Preface

Chapter 1  Overview

VNX System ................................................................. 12
  Main components ...................................................... 12
Control Station .............................................................. 13
  Accessing the Control Station ............................... 13
  Accessing the command line interface .................. 13
Role-based access ......................................................... 14
Command set conventions .......................................... 15
  Synopsis ...................................................................... 15
  Command prefixes ..................................................... 16
General notes .............................................................. 17

Chapter 2  The cel and cs Commands

cel_fs ........................................................................ 20
cs_standby .................................................................. 25

Chapter 3  The fs Commands

fs_ckpt ....................................................................... 28
fs_dedupe .................................................................... 39
fs_dhsm ...................................................................... 58
fs_group ...................................................................... 87
fs_rdf ......................................................................... 91
fs_timefinder ............................................................ 98
<table>
<thead>
<tr>
<th>Chapter 4</th>
<th>The nas Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>nas_acl</td>
<td>110</td>
</tr>
<tr>
<td>nas_automountmap</td>
<td>113</td>
</tr>
<tr>
<td>nas_ca_certificate</td>
<td>115</td>
</tr>
<tr>
<td>nas_cel</td>
<td>118</td>
</tr>
<tr>
<td>nas_checkup</td>
<td>131</td>
</tr>
<tr>
<td>nas_ckpt_schedule</td>
<td>136</td>
</tr>
<tr>
<td>nas_connecthome</td>
<td>143</td>
</tr>
<tr>
<td>nas_config</td>
<td>152</td>
</tr>
<tr>
<td>nas_copy</td>
<td>156</td>
</tr>
<tr>
<td>nas_cs</td>
<td>160</td>
</tr>
<tr>
<td>nas_devicegroup</td>
<td>166</td>
</tr>
<tr>
<td>nas_disk</td>
<td>170</td>
</tr>
<tr>
<td>nas_diskmark</td>
<td>178</td>
</tr>
<tr>
<td>nas_emailuser</td>
<td>181</td>
</tr>
<tr>
<td>nas_environment</td>
<td>184</td>
</tr>
<tr>
<td>nas_event</td>
<td>192</td>
</tr>
<tr>
<td>nas_fs</td>
<td>199</td>
</tr>
<tr>
<td>nas_fsck</td>
<td>228</td>
</tr>
<tr>
<td>nas_halt</td>
<td>232</td>
</tr>
<tr>
<td>nas_inventory</td>
<td>234</td>
</tr>
<tr>
<td>nas_license</td>
<td>238</td>
</tr>
<tr>
<td>nas_logviewer</td>
<td>240</td>
</tr>
<tr>
<td>nas_message</td>
<td>243</td>
</tr>
<tr>
<td>nas_mview</td>
<td>244</td>
</tr>
<tr>
<td>nas_pool</td>
<td>253</td>
</tr>
<tr>
<td>nas_quotas</td>
<td>267</td>
</tr>
<tr>
<td>nas_rdf</td>
<td>283</td>
</tr>
<tr>
<td>nas_replicate</td>
<td>298</td>
</tr>
<tr>
<td>nas_rp</td>
<td>321</td>
</tr>
<tr>
<td>nas_server</td>
<td>329</td>
</tr>
<tr>
<td>nas_slice</td>
<td>340</td>
</tr>
<tr>
<td>nas_stats</td>
<td>344</td>
</tr>
<tr>
<td>nas_storage</td>
<td>351</td>
</tr>
<tr>
<td>nas_task</td>
<td>367</td>
</tr>
<tr>
<td>nas_version</td>
<td>371</td>
</tr>
<tr>
<td>nas_volume</td>
<td>373</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 5</th>
<th>The server Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_archive</td>
<td>382</td>
</tr>
<tr>
<td>server_arp</td>
<td>396</td>
</tr>
<tr>
<td>server_certificate</td>
<td>398</td>
</tr>
</tbody>
</table>
## Contents

<table>
<thead>
<tr>
<th>Command Line Reference</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_cdms</td>
<td>404</td>
</tr>
<tr>
<td>server_cepp</td>
<td>410</td>
</tr>
<tr>
<td>server_checkup</td>
<td>413</td>
</tr>
<tr>
<td>server_cifs</td>
<td>430</td>
</tr>
<tr>
<td>server_cifssupport</td>
<td>455</td>
</tr>
<tr>
<td>server_cpu</td>
<td>469</td>
</tr>
<tr>
<td>server_date</td>
<td>471</td>
</tr>
<tr>
<td>server_dbms</td>
<td>476</td>
</tr>
<tr>
<td>server_devconfig</td>
<td>483</td>
</tr>
<tr>
<td>server_df</td>
<td>491</td>
</tr>
<tr>
<td>server_dns</td>
<td>493</td>
</tr>
<tr>
<td>server_export</td>
<td>495</td>
</tr>
<tr>
<td>server_file</td>
<td>508</td>
</tr>
<tr>
<td>server_ftp</td>
<td>509</td>
</tr>
<tr>
<td>server_http</td>
<td>525</td>
</tr>
<tr>
<td>server_ifconfig</td>
<td>531</td>
</tr>
<tr>
<td>server_ip</td>
<td>538</td>
</tr>
<tr>
<td>server_iscsi</td>
<td>543</td>
</tr>
<tr>
<td>server_kerberos</td>
<td>564</td>
</tr>
<tr>
<td>server_ldap</td>
<td>572</td>
</tr>
<tr>
<td>server_log</td>
<td>580</td>
</tr>
<tr>
<td>server_mount</td>
<td>587</td>
</tr>
<tr>
<td>server_mountpoint</td>
<td>595</td>
</tr>
<tr>
<td>server_mpfs</td>
<td>597</td>
</tr>
<tr>
<td>server_mt</td>
<td>600</td>
</tr>
<tr>
<td>server_name</td>
<td>602</td>
</tr>
<tr>
<td>server_netstat</td>
<td>603</td>
</tr>
<tr>
<td>server_nfs</td>
<td>607</td>
</tr>
<tr>
<td>server_nis</td>
<td>618</td>
</tr>
<tr>
<td>server_param</td>
<td>620</td>
</tr>
<tr>
<td>server_pax</td>
<td>628</td>
</tr>
<tr>
<td>server_ping</td>
<td>632</td>
</tr>
<tr>
<td>server_ping6</td>
<td>634</td>
</tr>
<tr>
<td>server_rip</td>
<td>636</td>
</tr>
<tr>
<td>server_route</td>
<td>637</td>
</tr>
<tr>
<td>server_security</td>
<td>640</td>
</tr>
<tr>
<td>server_setup</td>
<td>643</td>
</tr>
<tr>
<td>server_snmpd</td>
<td>646</td>
</tr>
<tr>
<td>server_ssh</td>
<td>650</td>
</tr>
<tr>
<td>server_standby</td>
<td>662</td>
</tr>
<tr>
<td>server_stats</td>
<td>665</td>
</tr>
<tr>
<td>server_sysconfig</td>
<td>694</td>
</tr>
<tr>
<td>server_sysstat</td>
<td>703</td>
</tr>
</tbody>
</table>
Contents

server_tftp ........................................................................................ 705
server_umount ................................................................................ 708
server_uptime ............................................................................... 710
server_user .................................................................................... 711
server_userrmapper ...................................................................... 714
server_version ............................................................................. 718
server_viruschk .......................................................................... 719
server_vtlu .................................................................................... 723

Appendix A Using Command Scripts for VNX

Scripting guidelines ....................................................................... 734

Appendix B GNU General Public License

GNU General Public License ............................................................ 742

Index
As part of an effort to improve and enhance the performance and capabilities of its product line, EMC periodically releases revisions of its hardware and software. Therefore, some functions described in this document may not be supported by all revisions of the software or hardware currently in use. For the most up-to-date information on product features, refer to your product release notes.

If a product does not function properly or does not function as described in this document, please contact your EMC representative.

 Audience  
This guide is part of the VNX documentation set, and is intended for use by administrators and users of the EMC VNX system.

 Related documentation  
Other VNX system publications are available on the EMC Online Support website. To search for technical documentation, go to http://Support.EMC.com. After logging in to the website, click the VNX Support by Product page to locate information for the specific feature required.

 Conventions used in this guide  
EMC uses the following conventions for notes, cautions, warnings, and danger notices.

 Note: A note presents information that is important, but not hazard-related.

 CAUTION  
A caution contains information essential to avoid data loss or damage to the system or equipment. The caution may apply to hardware or software.
WARNING
A warning contains information essential to avoid a hazard that can cause severe personal injury, death, or substantial property damage if you ignore the warning.

DANGER
A danger notice contains information essential to avoid a hazard that will cause severe personal injury, death, or substantial property damage if you ignore the message.

Typographical conventions
EMC uses the following type style conventions in this guide:

**Bold**
- User actions (what the user clicks, presses, or selects)
- Interface elements (button names, dialog box names)
- Names of keys, commands, programs, scripts, applications, utilities, processes, notifications, system calls, services, applications, and utilities in text

*italic*
- Book titles
- New terms in text
- Emphasis in text

*Courier*
- Prompts
- System output
- Filenames
- Pathnames
- URLs
- Syntax when shown in command line or other examples

*Courier, bold*
- User entry
- Options in command-line syntax

*Courier italic*
- Arguments in examples of command-line syntax
- Variables in examples of screen or file output
- Variables in pathnames

<>
Angle brackets for parameter values (variables) supplied by user.

[]
Square brackets for optional values.

|
Vertical bar symbol for alternate selections. The bar means or.

...
Ellipsis for nonessential information omitted from the example.
Where to get help

EMC support, product, and licensing information can be obtained as follows.

Product information — For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to the EMC Online Support website (registration required) at http://Support.EMC.com.

Technical support — For technical support and service requests, go to EMC Customer Service on the EMC Online Support website. After logging in, locate the applicable Support by Product page, and choose either Live Chat or Create a service request. To open a service request through EMC Online Support, you must have a valid support agreement. Contact your EMC Sales Representative for details about obtaining a valid support agreement or with any questions about your account.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send your opinion of this guide to:

techpubcomments@EMC.com
This chapter provides a brief description of the commands that can be used to configure and manage the VNX. Topics included are:

- VNX System ........................................................................................................... 12
- Control Station .................................................................................................... 13
- Role-based access ............................................................................................... 14
- Command set conventions .................................................................................. 15
- General notes ...................................................................................................... 17
Overview

VNX System

The EMC® VNX™ is a dedicated file server solution that is easily integrated into existing networks by using standard network interface protocols.

Main components

The VNX hardware platform is the VNX cabinet that works with a storage system to produce shared network-attached storage. The VNX cabinet:

- Contains up to 14 Data Movers available as dedicated file servers that provide access to file systems resident on the storage system to a requesting client on the network.
- Includes a Control Station that manages all Data Mover functions both locally and remotely.
Control Station

The Control Station provides utilities for managing, configuring, and monitoring of the Data Movers in the VNX.

As the system administrator, you may type commands through the Control Station to perform tasks that include the following:

- Managing and Configuring the database and Data Movers
- Monitoring statistics of the VNX cabinet components

Accessing the Control Station

You may use either local or remote access to the Control Station.

Local

Local access to the command line interface is available directly at the Control Station console.

Remote

Remote access to the command line interface by using a secure, encrypted login application allows the use of the VNX command set.

Accessing the command line interface

A description of how to gain local or remote access to the command line interface for the VNX follows.

Local

For local access to the command line interface, at the prompt, log in with your administrative username and password.

Remote

For remote access to the command line interface:

1. Use a secure, encrypted, remote login application. Type the IP address of the Control Station.
2. Log in with your administrative username and password.
Overview

Role-based access

The administrative user account you use to access the command line interface is associated with specific privileges, also referred to as roles. A role defines the privileges (operations) a user can perform on a particular VNX object. The ability to select a predefined role or define a custom role that gives a user certain privileges is supported for users who access VNX through the CLI, EMC Unisphere™, and the XML API.

The Security Configuration Guide for VNX provides detailed information about how role-based access is used to determine the commands a particular user can execute. You create and manage user accounts and roles in Unisphere by using Settings > User Management.
Command set conventions

This manual uses commonly known command set conventions for the VNX for File man pages. Each man page presents the command name at the top of the man page followed by a brief overview of what the command does. The synopsis contains the actual command usage. The description contains a more detailed breakdown of the features of the command, and the options describe what each switch or option does specifically.

The ‘See Also’ section refers to the technical modules that support the feature, in addition to any other commands that interact with the command. The examples are at the end of the command.

The naming convention for the Data Mover variable in the command line interface is <movername> (default = server_2 to server_15).

The commands are prefixed, then appear in alphabetical order.

Synopsis

The synopsis is usage of each command. The synopsis appears in courier typeface, with variables such as movername, filename, and device name enclosed by angle brackets, with the command name appearing in bold. The switches and other options also appear in bold and, in most cases, are prefixed by a minus sign:

server_umount {<movername> | ALL} [-perm | -temp] { -all | <fs_name> | <pathname> }

Required entries

A switch or variable enclosed with curly brackets, or not enclosed at all, indicates a required entry:

{<movername> | ALL}

Optional entries

A switch or variable enclosed with square brackets indicates an optional entry:

[ -perm | -temp ]

Formatting

The variable name enclosed by angle brackets indicates the name of a specified object:

{<movername> | ALL}

Options

An option is prefixed with a minus (-) sign: -perm

If the option is spelled out, for example, -perm, in the command syntax, you may use just the first letter: -p
Options and names are case-sensitive. If an uppercase letter is specified in the syntax, a lowercase letter is not accepted.

The vertical bar symbol ( | ) represents or, meaning an alternate selection:

\{-all|<fs_name>|<pathname>\}

**Command prefixes**

Commands are prefixed depending on what they are administering. For example, commands prefixed with:

- `cel_` execute to the remotely linked VNX system.
- `cs_` execute to the Control Station.
- `fs_` execute to the specified file system.
- `nas_` execute directly to the Control Station database.
- `server_` require a movername entry and execute directly to a Data Mover.
General notes

- If a command is interrupted by using Ctrl-C, then the following messages or traces on the console are expected:
  - nas_cmd: system execution failed.
  - nas_cmd: PANIC: caught signal #11 (Segmentation fault) -- Giving up.
- Use VNX CLI for file to add IPv6 addresses to the NFS export host list. Enclose the IPv6 address in { } or square brackets in the CLI. The IPv6 addresses added to the NFS export list by using the CLI are displayed as read-only fields in the Unisphere software.
This chapter lists the VNX Command Set provided for managing, configuring, and monitoring of Data Movers. The commands are prefixed with `cel` or `cs`, and appear alphabetically. The prefix assigned to a command indicates what the command is administering. The `cel` prefix administers to the remote VNX, and `cs` administers to the local Control Station. The command line syntax (Synopsis), a description of the options, and an example of usage are provided for each command.
**cel_fs**

Manages file systems residing on a remotely linked VNX.

**SYNOPSIS**

```
cel_fs <cel_name>
   -list
   -info [-size] {-all|{<fs_name>|id=<fs_id>}},...
   -Extract {-all|{<fs_name>|id=<fs_id>}},...
   -Import {{<fs_name>,...}|-file <filename>}
```

**DESCRIPTION**

From the remotely linked VNX, cel_fs imports and enables the definition of a remote file system known as the R2 onto the local system known as the R1. The <cel_name> is the default hostname of the remote Control Station active during installation, or when performing a `nas_rdf -init` to set up the Remote Data Facility (RDF) environment. To find the <cel_name>, go to `nas_cel` and execute the `-list` option. The local VNX, by default, has an ID of zero.

cel_fs displays a listing of the file systems residing on the remote VNX including their attributes.

This command is used in FarCopy and NearCopy environments.

**OPTIONS**

- **-list**
  
  Lists all of the file systems residing on the remote VNX as specified by its <cel_name>.

  **Note:** The ID of the object is an integer and is assigned automatically. The name of a file system may be truncated if it is too long for the display. To display the full name, use the `-info` option with a file system ID.

- **-info [-size] { -all|{<fs_name>|id=<fs_id>} },...**

  Displays the attributes and size of the file system residing on the <cel_name>. The <fs_name> can be a file system name or a group name and can be up to 16 characters.

- **-Extract {-all|{<fs_name>|id=<fs_id>} },...**

  Extracts all file system attributes that can be redirected for import into the VNX. The second <fs_name> specifies a file system where the attributes can be imported.

- **-Import {{<fs_name>,...}|-file <filename>}**

  Imports file system attributes obtained from the extraction of a file system definition either from the specified <fs_name> or <filename>.
The cel and cs Commands

SEE ALSO
Using SRDF/S with VNX for Disaster Recovery, nas_cel, nas_fs, nas_rdf, and fs_group.

EXAMPLE #1
To display a listing of the file systems residing on the remote VNX cs30, type:

```
$ cel_fs cs30 -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>inuse</th>
<th>type</th>
<th>acl</th>
<th>volume</th>
<th>name</th>
<th>server</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>66</td>
<td>root_fs_1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>68</td>
<td>root_fs_2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>70</td>
<td>root_fs_3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>72</td>
<td>root_fs_4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>74</td>
<td>root_fs_5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>76</td>
<td>root_fs_6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>78</td>
<td>root_fs_7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>80</td>
<td>root_fs_8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>82</td>
<td>root_fs_9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>84</td>
<td>root_fs_10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>86</td>
<td>root_fs_11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>88</td>
<td>root_fs_12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>90</td>
<td>root_fs_13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>92</td>
<td>root_fs_14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>94</td>
<td>root_fs_15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>96</td>
<td>root_fs_common_4,3,2,1</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>n</td>
<td>5</td>
<td>0</td>
<td>145</td>
<td>root_fs_ufslog_4,3,2,1</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>156</td>
<td>ufs1</td>
<td>1</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of a file system (assigned automatically).</td>
</tr>
<tr>
<td>inuse</td>
<td>Whether a file system has been registered into the mount table.</td>
</tr>
<tr>
<td>type</td>
<td>What type of file system. nas_fs provides information.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control value for a file system. nas_fs provides information.</td>
</tr>
<tr>
<td>volume</td>
<td>Name of the volume on which a file system resides.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to a file system.</td>
</tr>
<tr>
<td>server</td>
<td>Server associated with a file system.</td>
</tr>
</tbody>
</table>

EXAMPLE #2
To display the attributes and the size for ufs1 on cs30, type:

```
$ cel_fs cs30 -info -size ufs1
```

```
id       = 18
name     = ufs1
acl      = 0
in_use   = True
type     = uxfs
volume   = mtv1
pool     =
            rw_servers= server_2
            ro_servers=
```
The cel and cs Commands

```
wr_vdms   =
ro_vdms   =
size      = total = 22660 avail = 22659 used = 0 ( 0% ) (sizes in MB)
volume: total = 23012 (sizes in MB)
stor_devs =
002804000190-0034,002804000190-0035,002804000190-0036,002804000190-0037
disks     = d3,d4,d5,d6
  disk=d3  stor_dev=002804000190-0034  addr=c0t3l8-15-0  server=server_2
  disk=d4  stor_dev=002804000190-0035  addr=c0t3l9-15-0  server=server_2
  disk=d5  stor_dev=002804000190-0036  addr=c0t3l10-15-0  server=server_2
  disk=d6  stor_dev=002804000190-0037  addr=c0t3l11-15-0  server=server_2
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of a file system (assigned automatically).</td>
</tr>
<tr>
<td>name</td>
<td>Name of the imported file system.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control value for a file system. nas_fs provides information.</td>
</tr>
<tr>
<td>inuse</td>
<td>Whether a file system has been registered into the mount table.</td>
</tr>
<tr>
<td>type</td>
<td>What type of file system. nas_fs provides information.</td>
</tr>
<tr>
<td>volume</td>
<td>Name of the volume on which a file system resides.</td>
</tr>
<tr>
<td>pool</td>
<td>Storage pool for the file system.</td>
</tr>
<tr>
<td>rw_servers</td>
<td>Servers with read-write access.</td>
</tr>
<tr>
<td>ro_servers</td>
<td>Servers with read-only access.</td>
</tr>
<tr>
<td>rw_vdms</td>
<td>VDM servers with read-write access to a file system.</td>
</tr>
<tr>
<td>ro_vdms</td>
<td>VDM servers with read-only access to a file system.</td>
</tr>
<tr>
<td>size</td>
<td>Total size, available size, and amount used in MB.</td>
</tr>
<tr>
<td>volume</td>
<td>Volume size that is used.</td>
</tr>
<tr>
<td>stor_devs</td>
<td>ID of the storage system device and the serial number.</td>
</tr>
<tr>
<td>disks</td>
<td>The disks on which a file system resides. - dd indicates a remote disk.</td>
</tr>
</tbody>
</table>

**EXAMPLE #3**  To extract a file system definition for ufs1 from cs30, type:

```
$ cel_fs cs30 -Extract ufs1
```

```
0:bofs:18
18:ufs1:0:y:1:164:1:::0:::
164:mtv1:0:y:3:2:18:156:
156:stv1:0:y:2:1:164:64:3,4,5,6:
6:d6:0:y:4:1:156:6:
6:d6:0:y:5753:002804000190:1,2,3,4:6:0037:5:
5:d5:0:y:4:1:156:5:
5:d5:0:y:5753:002804000190:1,2,3,4:5:0036:5:
4:d4:0:y:4:1:156:4:
4:d4:0:y:5753:002804000190:1,2,3,4:4:0035:5:
3:d3:0:y:4:1:156:3:
3:d3:0:y:5753:002804000190:1,2,3,4:3:0034:5:
0:eof:18
```
The output is the extracted file system definition.

**EXAMPLE #4**
To extract a file system definition for *ufs1* from *cs30* into the extract-file, type:

```bash
$ cel_fs cs30 -Extract ufs1 > extract-file
```

**EXAMPLE #5**
To import a remote file system definition for *ufs1* from *cs30* onto the current VNX, type:

```bash
$ cel_fs cs30 -Import ufs2
```

```plaintext
id        = 18
name      = ufs2
acl       = 0
in_use    = False
type      = uxfs
volume    = mttv1
pool      =
rw_servers=
ro_servers=
rw_vdms   =
ro_vdms   =
backup_of = 1901 Fri Apr 23 15:25:59 EDT 2004
remainder = 0 MB (0%)
stor_devs = 002804000218-0038,002804000218-0039,002804000218-003A,002804000218-003B
disks     = dd7,dd8,dd9,dd10
```

**EXAMPLE #2** provides an output description of all values except those which are unique to the **-Import** option.

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>backup_of</td>
<td>System ID of the remote file system.</td>
</tr>
<tr>
<td>remainder</td>
<td>Number of MB copied from the file system on the remote VNX.</td>
</tr>
</tbody>
</table>

**EXAMPLE #6**
To import a remote file system definition from the file, extract-file, from *cs30* into the current VNX, type:

```bash
$ cel_fs cs30 -Import -file extract-file
```

```plaintext
id        = 19
name      = ufs1
acl       = 0
in_use    = False
type      = uxfs
volume    = mttv1
pool      =
rw_servers=
ro_servers=
rw_vdms   =
```
ro_vdms =
backup_of = 18@1 Fri Apr 23 15:26:25 EDT 2004
remainder = 0 MB (0%)
stor_devs =
002804000218-0034,002804000218-0035,002804000218-0036,002804000218-0037
disks = dd3,dd4,dd5,dd6

**EXAMPLE #2** and **EXAMPLE #5** provide a description of command output.
The `cs_standby` command initiates a takeover and failover of a Control Station on a VNX with dual Control Stations.

**SYNOPSIS**

```
cs_standby
{--takeover|--failover}
```

**DESCRIPTION**

The `cs_standby` command initiates a Control Station takeover and failover. When a Control Station is activated, the name of the primary Control Station is displayed.

`su` to `root` and execute this command from the `/nas/sbin` or `/nasmcd/sbin` directory.

__Note:__ EMC SRDF® is not supported on the secondary Control Station.

**OPTIONS**

- **-takeover**
  Executed from the standby Control Station, initiates a reboot of the primary Control Station, then changes the state of the standby to that of the primary. The original primary Control Station now becomes the standby Control Station. The `-takeover` option can only be executed from the standby Control Station.

__CAUTION__

When executing a takeover or failover, Data Movers performing functions such as RDF, EMC TimeFinder®/FS, file system extends, or quotas may be interrupted.

__CAUTION__

If a primary Control Station fails over to a standby Control Station, for remote replication, service continues to run but replication management capabilities are no longer available.

__Note:__ After executing a takeover or failover, a few minutes may be needed to stop Linux and other services active on the Control Station.
-failover
Executed from the primary Control Station, initiates a reboot of the primary Control Station, then activates the standby to take over the role of the primary Control Station. A -failover can only be executed from the primary Control Station.

To display the primary Control Station, type:

$ nas/sbin/getreason

**EXAMPLE #1**
To change the state of the standby Control Station to primary, cd to the /nasmcd/sbin directory of the standby Control Station, then type:

`.cs_standby -takeover`

Taking over as Primary Control Station............done

If the takeover command is executed on the primary Control Station, the following error message appears:

The -takeover option is only valid on a standby Control Station

**EXAMPLE #2**
To initiate a failover from the primary Control Station to the standby Control Station, cd to the /nas/sbin directory of the primary Control Station, then type:

`.cs_standby -failover`

The system will reboot, do you wish to continue [yes or no]: y

Failing over from Primary Control Station
This chapter lists the VNX Command Set provided for managing, configuring, and monitoring the specified file system. The commands are prefixed with `fs` and appear alphabetically. The command line syntax (Synopsis), a description of the options, and examples of usage are provided for each command.
The fs Commands

**fs_ckpt**

Manages checkpoints using the EMC SnapSure™ functionality.

**SYNOPSIS**

```
fs_ckpt {<fs_name>|id=<fs_id>}
-list [-all]
  | [-name <name>] -Create [-readonly {y|n}][<volume_name>][-option <options>]
  | [-name <name>] -Create [-readonly {y|n}][size=<integer>[T|G|M|%]]
    [pool=<pool>][storage=<system_name>][-option <options>]
  | -refresh [-option <options>]
  | [-name <name>] -Restore [-Force][-option <options>]
  | -modify [%full=<value>][maxsavsize=<integer>[T|G|M]]
```

**DESCRIPTION**

The fs_ckpt command creates a checkpoint of a Production File System (PFS), lists associated checkpoints, refreshes a checkpoint to the current time, and restores a PFS back to a specific point in time using a checkpoint. Checkpoints are deleted using nas_fs.

**What is a checkpoint file system?**

A PFS is made up of blocks. When a block within a PFS is modified, a copy containing the original contents of that block is saved to a metavolume called the SavVol. Subsequent changes made to the same block in the PFS are not copied into the SavVol. The original blocks from the PFS (in the SavVol) and the unchanged PFS blocks (that remain in the PFS) are read according to a bitmap and blockmap data tracking structure. These blocks combine to provide a complete point-in-time file system image which is called a checkpoint.

**OPTIONS**

- **list**
  Displays all of the associated checkpoints for the specified file system. The -all option displays system-generated Replication checkpoints in addition to checkpoints created by the user.

- **Create**
  Creates, mounts, and optionally assigns a name to the checkpoint of the PFS. The checkpoint must be unmounted prior to unmounting the PFS. Names assigned to a checkpoint cannot be all numeric. If a name is not chosen, one is assigned by default.

  ```
  [-name <name>] -Create [-readonly {y|n}]
  ```

  Specifies whether a checkpoint is read only or not. y (default) sets the checkpoint as read only; n sets the checkpoint as writeable.
[<volume_name>]
Specifies an unused metavolume for the checkpoint.

**Note:** A volume can be specified for only the first checkpoint of a PFS since all of the subsequent checkpoints share the same SavVol. The minimum size required for a SavVol is 64 MB. The volume size is 10 GB unless the PFS is less than 10 GB, then the volume is the same size as the file system.

[<option <options>>]
Specifies the following comma-separated options:

*%full=<value>*
Specifies a value as the percentage threshold permitted for the SavVol. When that value is reached, a warning is sent to the server_log and the syslog files. The Control Station acknowledges the event and automatically extends the checkpoint. The SavVol is automatically extended by 10 GB if its default %full value is reached. If the %full value is set to zero, the option is disabled.

*maxsavsize=<integer>[T|G|M]*
Limits the final size to which the SavVol can be automatically extended when the high watermark value specified in %full has been reached. Automatic extension of the SavVol stops when the size of the SavVol reaches the value specified in maxsavsize. The range for maxsavsize is 64 MB to 16 TB.

*automount=no*
Stops the checkpoint from being automatically mounted.

[<name>] -Create
Creates, mounts, and optionally assigns a name to the checkpoint of the PFS. The checkpoint must be unmounted prior to unmounting the PFS. Names assigned to a checkpoint cannot be all numeric. If a name is not chosen, one is assigned by default.

[readonly {y|n}]
Specifies whether a checkpoint is read only or not. y (default) sets the checkpoint as read only; n sets the checkpoint as writeable.

[size=<integer>[T|G|M|%]]
Specifies a size for the checkpoint file system. Type an integer between 1 and 1024, specify T for terabytes, G for gigabytes (default) or M for megabytes, or type an integer representing the percentage of a file system’s size, followed by the percent sign.
The fs Commands

[pool=<pool>]
Specifies the storage pool to be used for the checkpoint. Storage pools can either be user-defined or system-defined. The nas_pool-list command displays a listing of available pool types.

[storage=<system_name>]
Specifies the attached storage system for the checkpoint SavVol to reside.

[-option <options>]
Specifies the following comma-separated options:

%full=<value>
Specifies a value as the percentage threshold permitted for the SavVol. When that value is reached, a warning is sent to the server_log and the syslog files. The Control Station acknowledges the event and automatically extends the checkpoint. The SavVol is automatically extended by 10 GB if its default %full value is reached. If the %full value is set to zero, the option is disabled. The default for <value> is 90 and it can be within the range of 10 to 99.

automount=no
Stops the checkpoint from being automatically mounted.

-refresh
Initiates an immediate update of a checkpoint, thereby allowing the SavVol space to be reused. Refreshing a checkpoint does not add to the number of checkpoints of the PFS.

[-option <options>] %full=<value>
Specifies a value as the percentage threshold permitted for the metavolume. When that value is reached, a warning is sent to the server_log and the syslog files. The Control Station acknowledges the event and automatically extends the checkpoint. The SavVol is automatically extended by 10 GB if its default %full value is reached. If the %full value is set to zero, the option is disabled. The default for <value> is 90.

-modify
Modifies one or all of the following options:

Note: The -modify action works only on the PFS and not on the checkpoint.

[%full=<value>]
Modifies the value of the percentage threshold permitted for the metavolume.
The fs Commands

[**maxsavsize=<integer>[T|G|M]**]
Modifies the final size to which the SavVol can be automatically extended, when the size specified in %full is reached.

[-**name <name>**] -**Restore**
Restores the PFS from the specified checkpoint and optionally assigns a name to the automatically created checkpoint. If a name is not chosen, one is assigned by default.

**Note:** As part of the restore, a new checkpoint is automatically created to capture the latest point-in-time image of the PFS. This is for protection in the event that the restored image is discarded.

[-**Force**]
The -Force option must be used when restoring a production file system with enabled.

**CAUTION**
Forcing a restore of a production file system with enabled from a checkpoint will delete or overwrite files that were written after this checkpoint was created or refreshed.

[-**option <options>**]
Specifies the following comma-separated options:

%**full**=<value>
Specifies a value as the percentage threshold permitted for the SavVol. When that value is reached, a warning is sent to the server_log and the syslog files. The Control Station acknowledges the event and automatically extends the checkpoint. The SavVol is automatically extended by 10 GB if its default %full value is reached. If the %full value is set to zero, the option is disabled. The <value> can be an integer between 10 and 75 (default).

automount=no
Stops the checkpoint from being automatically mounted.

**SEE ALSO** Using VNX Snapsure, nas_fs, and nas_pool.

**STORAGE SYSTEM OUTPUT**
The number associated with the storage device is dependent on the attached storage system. VNX for block displays a prefix of APM before a set of integers, for example, APM00033900124-0019. For example, EMC Symmetrix® storage systems display as 002804000190-003C. The outputs displayed in the examples use a VNX for block.
**EXAMPLE #1**  To display the checkpoint for the file system fs4, type:

```bash
$ fs_ckpt fs4 -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>ckpt_name</th>
<th>creation_time</th>
<th>inuse</th>
<th>fullmark</th>
<th>total_savvol_used</th>
<th>ckpt_usage_on_savvol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1406</td>
<td>fs4_ckpt1</td>
<td>05/26/2008-16:22:19-EDT</td>
<td>y</td>
<td>90%</td>
<td>51%</td>
<td>0%</td>
</tr>
</tbody>
</table>

**EXAMPLE #2**  To display all checkpoints including internal checkpoints for the file system fs4, type:

```bash
$ fs_ckpt fs4 -list -all
```

<table>
<thead>
<tr>
<th>id</th>
<th>ckpt_name</th>
<th>creation_time</th>
<th>inuse</th>
<th>fullmark</th>
<th>total_savvol_used</th>
<th>ckpt_usage_on_savvol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1401</td>
<td>root_rep_ckpt_1398_21625_1</td>
<td>05/26/2008-16:11:10-EDT</td>
<td>y</td>
<td>90%</td>
<td>51%</td>
<td>0%</td>
</tr>
<tr>
<td>1402</td>
<td>root_rep_ckpt_1398_21625_2</td>
<td>05/26/2008-16:11:22-EDT</td>
<td>y</td>
<td>90%</td>
<td>51%</td>
<td>0%</td>
</tr>
<tr>
<td>1406</td>
<td>fs4_ckpt1</td>
<td>05/26/2008-16:22:19-EDT</td>
<td>y</td>
<td>90%</td>
<td>51%</td>
<td>0%</td>
</tr>
</tbody>
</table>

**EXAMPLE #3**  To create a checkpoint of ufs1, on the volume, ssmtv1, type:

```bash
$ fs_ckpt ufs1 -Create ssmtv1
```

```bash
operation in progress (not interruptible)...id = 22
name = ufs1
acl = 0
in_use = True
type = uxfs
worm = off
volume = mtv1
pool =
  rw_servers= server_2
  ro_servers=
  rw_vdms =
  ro_vdms =
  ckpts = ufs1_ckpt1
  stor_devs = APM00043807043-0010,APM00043807043-0014
  disks = d7,d9
    disk=d7 stor_dev=APM00043807043-0010 addr=c0t1l0 server=server_2
    disk=d7 stor_dev=APM00043807043-0010 addr=c16t1l0 server=server_2
    disk=d9 stor_dev=APM00043807043-0014 addr=c0t1l4 server=server_2
    disk=d9 stor_dev=APM00043807043-0014 addr=c16t1l4 server=server_2

id = 24
name = ufs1_ckpt1
acl = 0
in_use = True
type = ckpt
worm = off
volume = vp132
pool =
  member_of =
  rw_servers=
  ro_servers= server_2
  rw_vdms =
```
The fs Commands

The output is as follows:

ro_vdms =
ckpt_of= ufs1 Wed Oct 13 18:01:04 EDT 2004
used = 0%
full(mark)= 90%
stor_devs = APM00043807043-0011,APM00043807043-0017
disks = d12,d15
disk=d12 stor_dev=APM00043807043-0011 addr=c16t1l1 server=server_2
disk=d12 stor_dev=APM00043807043-0011 addr=c0t1l1 server=server_2
disk=d15 stor_dev=APM00043807043-0017 addr=c16t1l7 server=server_2
disk=d15 stor_dev=APM00043807043-0017 addr=c0t1l7 server=server_2

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Automatically assigned ID of a file system or the checkpoint.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to the file system or the checkpoint.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control value for a file system. nas_acl provides information.</td>
</tr>
<tr>
<td>in_use</td>
<td>If a file system is registered into the mount table of a Data Mover.</td>
</tr>
<tr>
<td>type</td>
<td>Type of file system. -list provides a description of the types.</td>
</tr>
<tr>
<td>worm</td>
<td>Whether the feature is enabled.</td>
</tr>
<tr>
<td>volume</td>
<td>Volume on which a file system resides.</td>
</tr>
<tr>
<td>pool</td>
<td>Storage pool for the file system.</td>
</tr>
<tr>
<td>member_of</td>
<td>Group to which the file system belongs.</td>
</tr>
<tr>
<td>rw_servers</td>
<td>Servers with read-write access to a file system.</td>
</tr>
<tr>
<td>ro_servers</td>
<td>Servers with read-only access to a file system.</td>
</tr>
<tr>
<td>rw_vdms</td>
<td>VDM servers with read-write access to a file system.</td>
</tr>
<tr>
<td>ro_vdms</td>
<td>VDM servers with read-only access to a file system.</td>
</tr>
<tr>
<td>ckpts</td>
<td>Associated checkpoints for the file system.</td>
</tr>
<tr>
<td>checkpt_of</td>
<td>Name of the PFS related to the existing checkpoints.</td>
</tr>
<tr>
<td>used</td>
<td>Percentage of SavVol space used by the checkpoints of the PFS.</td>
</tr>
<tr>
<td>full(mark)</td>
<td>SavVol usage point which, when reached, sends a warning message to the system log, and auto-extends the SavVol as system space permits.</td>
</tr>
<tr>
<td>stor_devs</td>
<td>Storage system devices associated with a file system.</td>
</tr>
<tr>
<td>disks</td>
<td>Disks on which the metavolume resides.</td>
</tr>
</tbody>
</table>

**EXAMPLE #4**  To create a checkpoint of ufs1 named ufs1_ckpt2 with a size of 2 GB by using the clar_r5_performance pool, with the specified storage system, with the %full set to 95, type:

```
$ fs_ckpt ufs1 -name ufs1_ckpt2 -Create size=2G pool=clar_r5_performance storage=APM00043807043 -option %full=95
```

operation in progress (not interruptible)...id = 27
name = ufs1
acl = 0
in_use = True
type = uxfs
The fs Commands

```plaintext
worm = off
volume = mtv1
pool =
rw_servers= server_2
ro_servers=
rw_vdms =
ro_vdms =
ckpts = ufs1_ckpt1, ufs1_ckpt2
stor_devs = APM00043807043-0010, APM00043807043-0014
disks = d7,d9
disk=d7 stor_dev=APM00043807043-0010 addr=c0t1l0 server=server_2
disk=d7 stor_dev=APM00043807043-0010 addr=c16t1l0 server=server_2
disk=d9 stor_dev=APM00043807043-0014 addr=c0t1l4 server=server_2
disk=d9 stor_dev=APM00043807043-0014 addr=c16t1l4 server=server_2

id = 30
name = ufs1_ckpt2
acl = 0
in_use = True
type = ckpt
worm = off
volume = vp145
pool =
member_of =
rw_servers =
ro_servers = server_2
rw_vdms =
ro_vdms =
checkpt_of = ufs1 Wed Nov 10 14:00:20 EST 2004
used = 0%
full(mark) = 95%
stor_devs = APM00043807043-0011, APM00043807043-0017
disks = d12,d15
disk=d12 stor_dev=APM00043807043-0011 addr=c16t1l1 server=server_2
disk=d12 stor_dev=APM00043807043-0011 addr=c0t1l1 server=server_2
disk=d15 stor_dev=APM00043807043-0017 addr=c16t1l7 server=server_2
disk=d15 stor_dev=APM00043807043-0017 addr=c0t1l7 server=server_2
```

**EXAMPLE #3** provides a description of command output.

**EXAMPLE #5** To create a writeable checkpoint of baseline checkpoint `ufs1_ckpt1`, type:

```bash
$ fs_ckpt ufs1_ckpt1 -Create -readonly n
```

**operation in progress (not interruptible)...**

```plaintext
id = 45
name = ufs1_ckpt1
acl = 0
in_use = False
type = ckpt
worm = off
volume = vp145
pool = clar_r5_performance
```
**The fs Commands**

```
member_of =
rw_servers=
ro_servers=
rw_vdms =
ro_vdms =
checkpt_of = ufs1 Tue Nov  6 14:56:43 EST 2007
ckpts = ufs1_ckpt1_writeable1
used = 38%
full(mark)= 90%
stor_devs = APM00042000814-0029,APM00042000814-0024,APM00042000814-0021,APM00042000814-001C
disks = d34,d17,d30,d13

id = 46
name = ufs1_ckpt1_writeable1
acl = 0
in_use = True
type = wckpt
worm = off
volume = vpl45
pool = clar_r5_performance
member_of =
rw_servers= server_2
ro_servers=
rw_vdms =
ro_vdms =
checkpt_of = ufs1
baseline_ckpt = ufs1_ckpt1 Tue Nov  6 14:56:43 EST 2007
used = 38%
full(mark)= 90%
stor_devs = APM00042000814-0029,APM00042000814-0024,APM00042000814-0021,APM00042000814-001C
disks = d34,d17,d30,d13

disk=d34   stor_dev=APM00042000814-0029 addr=c16t219 server=server_2
disk=d34   stor_dev=APM00042000814-0029 addr=c32t219 server=server_2
disk=d34   stor_dev=APM00042000814-0029 addr=c0t219 server=server_2
disk=d34   stor_dev=APM00042000814-0029 addr=c48t219 server=server_2
disk=d17   stor_dev=APM00042000814-0024 addr=c0t214 server=server_2
disk=d17   stor_dev=APM00042000814-0024 addr=c48t214 server=server_2
disk=d17   stor_dev=APM00042000814-0024 addr=c16t214 server=server_2
disk=d17   stor_dev=APM00042000814-0024 addr=c32t214 server=server_2
disk=d30   stor_dev=APM00042000814-0021 addr=c16t211 server=server_2
disk=d30   stor_dev=APM00042000814-0021 addr=c32t211 server=server_2
disk=d30   stor_dev=APM00042000814-0021 addr=c0t211 server=server_2
disk=d30   stor_dev=APM00042000814-0021 addr=c48t211 server=server_2
disk=d13   stor_dev=APM00042000814-001C addr=c0t1112 server=server_2
disk=d13   stor_dev=APM00042000814-001C addr=c48t1112 server=server_2
disk=d13   stor_dev=APM00042000814-001C addr=c16t1112 server=server_2
disk=d13   stor_dev=APM00042000814-001C addr=c32t1112 server=server_2
```
The *fs* Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline_ckpt</td>
<td>Name of the read-only checkpoint from which the writeable checkpoint is created.</td>
</tr>
</tbody>
</table>

**EXAMPLE #3** provides a description of command output.

**EXAMPLE #6** To list checkpoints for *ufs1*, type:

```
$ fs_ckpt ufs1 -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>ckpt_name</th>
<th>creation_time</th>
<th>inuse</th>
<th>full(mark)</th>
<th>used</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>ufs1_ckpt1</td>
<td>11/04/2004-14:54:06-EST</td>
<td>n</td>
<td>95%</td>
<td>0%</td>
</tr>
<tr>
<td>30</td>
<td>ufs1_ckpt2</td>
<td>11/10/2004-14:00:20-EST</td>
<td>y</td>
<td>95%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Automatically assigned ID of a file system or checkpoint.</td>
</tr>
<tr>
<td>ckpt_name</td>
<td>Name assigned to the checkpoint.</td>
</tr>
<tr>
<td>creation_time</td>
<td>Date and time the checkpoint was created.</td>
</tr>
<tr>
<td>inuse</td>
<td>If a checkpoint is registered into the mount table of a Data Mover.</td>
</tr>
<tr>
<td>full(mark)</td>
<td>SavVol-usage point which, when reached, sends a warning message to the system log, and auto-extends the SavVol as system space permits.</td>
</tr>
<tr>
<td>used</td>
<td>Percentage of SavVol space used by checkpoints of the PFS.</td>
</tr>
</tbody>
</table>

**EXAMPLE #7** To refresh *ufs1_ckpt2* by using the `%full` at 85, type:

```
$ fs_ckpt ufs1_ckpt2 -refresh -option %full=85
```

```
operation in progress (not interruptible)...id        = 30
name                  = ufs1_ckpt2
acl                   = 0
in_use                = True
type                  = ckpt
worm                  = off
volume                = vp145
pool                  =
member_of             =
rw_servers            =
ro_servers            = server_2
rw_vdms               =
ro_vdms               =
checkpoint_of         = ufs1 Wed Nov 10 14:02:59 EST 2004
used                  = 0%
full(mark)= 85%
stor_devs = APM00043807043-0011,APM00043807043-0017
disks                = d12,d15
disk=d12  stor_dev=APM00043807043-0011 addr=c16t111 server=server_2
disk=d12  stor_dev=APM00043807043-0011 addr=c0t111   server=server_2
disk=d15  stor_dev=APM00043807043-0017 addr=c16t117 server=server_2
disk=d15  stor_dev=APM00043807043-0017 addr=c0t117   server=server_2
```
EXAMPLE #3 provides a description of command output.

EXAMPLE #8 Using root command, to restore ufs1_ckpt2 and capture the latest point-in-time image of the PFS on ufs1_ckpt3, type:

$ /nas/sbin/rootfs_ckpt ufs1_ckpt2 -name ufs1_ckpt3

-Restore

operation in progress (not interruptible)...id = 30
name = ufs1_ckpt2
acl = 0
in_use = True
type =ckpt
worm = off
volume = vp145
pool =
member_of =
rw_servers=
ro_servers= server_2
rw_vdms =
ro_vdms =
checkpoint_of = ufs1 Wed Nov 10 14:02:59 EST 2004
used = 0%
full (mark)= 90%
stor_devs = APM000043807043-0011, APM000043807043-0017
disks = d12, d15
  disk=d12  stor_dev=APM000043807043-0011 addr=c16t111  server=server_2
  disk=d12  stor_dev=APM000043807043-0011 addr=c0t111  server=server_2
  disk=d15  stor_dev=APM000043807043-0017 addr=c16t117  server=server_2
  disk=d15  stor_dev=APM000043807043-0017 addr=c0t117  server=server_2

EXAMPLE #3 provides a description of command output.

EXAMPLE #9 To modify the %full value of the SavVol associated with the file system ufs1 and set it to 95, type:

$ fs_ckpt ufs1 -modify %full=95

operation in progress (not interruptible)...id = 33
name = ufs1
acl = 0
in_use = True
type =uxfs
worm = off
volume = vp145
pool =
rw_servers= server_2
ro_servers= server_2
rw_vdms =
ro_vdms =
auto_ext = no, virtual_provision=no
ckpts = wipckpt
stor_devs = APM00062400708-0014, APM00062400708-0016
disks = d26, d27
The fs Commands

EXAMPLE #10 To modify the maxsavsize value of the SavVol associated with the file system ufs1 and set it to 65 GB, type:

```
$ fs_ckpt ufs1 -modify maxsavsize=65G
```

```
operation in progress (not interruptible)...id        = 33
name      = ufs1
acl       = 0
in_use    = True
type      = uxfs
worm      = off
volume    = vp145
pool      =
  rw_servers= server_2
  ro_servers=
  rw_vdms   =
  ro_vdms   =
auto_ext  = no,virtual_provision=no
ckpts     = wipckpt
stor_devs = APM00062400708-0014,APM00062400708-0016
disks     = d26,d27
```

DIAGNOSTICS The fs_ckpt command returns one of the following return codes:

- **0** — Command completed successfully
- **1** — Usage error
- **2** — Invalid object error
- **3** — Unable to acquire lock
- **4** — Permission error
- **5** — Communication error
- **6** — Transaction error
- **7** — Dart error
- **8** — Backend error
The fs Commands

fs_dedupe

Manages file system deduplication state.

SYNOPSIS

fs_dedupe {
   -list
   | -info { -all | <fs_name>|id=<fs_id>} [-state
   | -modify { <fs_name>|id=<fs_id>} [-state
   | -minimum_scan_interval <days>][ -minimum_size <KB>]
   | -maximum_size <MB>][ -access_time <days>][ -modification_time <days>]
   | -case_sensitive {yes|no}][ -pathname_exclude_list <path_list>]
   | -file_ext_exclude_list <ext_list>][ -duplicate_detection_method
   | {sha1|byte|off}][ -savvol_threshold <percent>][ -backup_data_threshold
   | <percent>][ -cifs_compression_enabled {yes|no}]
   | -clear { <fs_name>|id=<fs_id>} [-minimum_scan_interval][-minimum_size]
   | [-maximum_size][-access_time][-modification_time][-case_sensitive]
   | [-pathname_exclude_list][-file_ext_exclude_list]
   | [-duplicate_detection_method][-savvol_threshold]
   | [-backup_data_threshold][-cifs_compression_enabled]
   | -default {
   |   -info {<mover_name>| -all}
   |   | -set {<mover_name>| -all}[-minimum_scan_interval <days>]
   |   | [-minimum_size <KB>][-maximum_size <MB>][-access_time
   |     <days>][-modification_time <days>][-case_sensitive
   |     {yes|no}][-file_ext_exclude_list <ext_list>]
   |   | [-duplicate_detection_method {sha1|byte|off}][-savvol_threshold
   |     <percent>][-cpu_usage_low_watermark <percent>]
   |   | [-cpu_usage_high_watermark <percent>][-backup_data_threshold
   |     <percent>][-cifs_compression_enabled {yes|no}]
   |   | -clear {<mover_name>| -all}
   |   | [-minimum_scan_interval][-minimum_size][-maximum_size][-access_time]
   |   | [-modification_time][-case_sensitive][-file_ext_exclude_list]
   |   | [-duplicate_detection_method][-savvol_threshold]
   |   | [-cpu_usage_low_watermark][-cpu_usage_high_watermark]
   |   | [-backup_data_threshold <percent>][-cifs_compression_enabled]
   }
}

DESCRIPTION

fs_dedupe allows the VNX administrator to enable, suspend, and undo all deduplication processing on a file system or a Data Mover. The Data Mover settings are the global settings that can be used for both the Data Mover and the file system. If a user sets a value for a specific file system, then that value overrides the Data Mover global value. If a user clears a value set for a specific file system, then that value is reset to the Data Mover global value.
The fs Commands

OPTIONS

- **list**
  Lists all deduplication-enabled file systems on the VNX.

- **info {-all|<fs_name>|id=<fs_id>}**
  Lists the existing file systems and provides information on the state of deduplication processing.

  - **-all**
    Lists all file systems and provides detailed information on the state of deduplication processing.

  - **<fs_name>**
    Lists the file system information for the specified file system name.

  - **id=<fs_id>**
    Lists the file system information for the specified identifier.

The file system state and status information displayed includes:

- If the state is **off** and the status is **not reduplicating**:
  - ID
  - Name
  - Deduplication state

- If the state is **off** and the status is **reduplicating**:
  - ID
  - Name
  - Deduplication state
  - Progress information (the percentage of files scanned)

- If the state of the file system is **on** or **suspended**, and the status is **Idle** or **Scanning**:
  - ID
  - Name
  - Reduplication state
  - Status
  - The percentage of files scanned
  - Last system scan time
  - Number of files scanned
  - Number of files deduplicated
  - The percentage of files deduplicated
  - File system capacity
  - Logical data size
- The fs Commands

- Percentage of file system usage
- Space saved (in MB and percent)

modify \(<fs\_name>|id=<fs\_id>\) \[-state \{\text{off|suspended|on}\}\]

Modifies the deduplication state of the file system for each specified file system identifier or file system name. The state can be set to \text{off}, \text{on}, or \text{suspended).

- \text{-minimum\_scan\_interval} \(<\text{days}>\)

Defines the minimum number of days between completing one scan of a file system and before scanning the same file system again. The values range from 1 to 365 and the default value is 7 days.

- \text{-minimum\_size} \(<\text{KB}>\)

Defines the file size in KB that limits deduplication. File sizes equal to this value or smaller will not be deduplicated. Setting this value to zero disables it. This value should not be set lower than 24 KB. The values range from 0 to 1000 and the default value is 24 KB.

- \text{-maximum\_size} \(<\text{MB}>\)

Defines the file size in MB of the largest file to be processed for deduplication. Files larger than this size in MB will not be deduplicated. Setting this value too high may affect system write performance as the first write operation reduplicates the file in its entirety. Setting this value to zero disables it. The values range from 0 to 8388608 and the default value is 8388608 MB.

- \text{-access\_time} \(<\text{days}>\)

Defines the minimum required file age in days based on read access time. Files that have been read within the specified number of days will not be deduplicated. This setting does not apply to files with an FLR locked state. Setting this value to zero disables it. The values range from 0 to 365 and the default value is 15 days.

- \text{-modification\_time} \(<\text{days}>\)

Defines the minimum required file age in days based on modification time. Files updated within the specified number of days will not be deduplicated. Setting this value to zero disables it. The values range from 0 to 365 and the default value is 15 days.

- \text{-case\_sensitive} \{\text{yes|no}\}

Defines whether case-sensitive (for NPS environments) or case-insensitive (for CIFS environments) string comparisons will
be used during scans. By default, case-insensitive comparisons will be done to be consistent for CIFS environments. The default value is zero (false).

\[-pathname\_exclude\_list\ <path\_list>\]
This is a file system setting only (no global setting). It is empty by default.

Defines a semicolon-delimited list of relative pathnames, in UTF-8 format, to be excluded from deduplication. Any directory below a specified pathname will be excluded from deduplication. You can specify a maximum of 10 pathnames and each one can be up to 1024 bytes. The default value is ‘’ (empty).

\[-file\_ext\_exclude\_list\ <ext\_list>\]
Specifies a colon-delimited list of filename extensions to be excluded from deduplication. Each extension must include the leading dot. The default value is ‘’ (empty).

\[-duplicate\_detection\_method\ \{sha1\|byte\|off\}\]
0 (off) — This means that duplicate data detection is disabled. With this setting, every deduplicated file is considered unique and the only space savings made are accomplished with compression.

1 (sha1) — The SHA-1 hash is used to detect duplicate data. It is faster than a byte comparison. This is the default method.

2 (byte) — This will use a byte-by-byte comparison to detect duplicate data. This adds considerable overhead especially for large files.

\[-savvol\_threshold\ <percent>\]
Represents the percentage of the configured save volume (SavVol) auto extension threshold that can be used during deduplication. When the specified amount of SavVol is used, deduplication stops on this file system. By default, this value is 90 percent and the SavVol auto extension is also 90 percent; this option will apply when the SavVol is 81 percent full (90 * 90). Setting this value to zero disables it. The values range from 0 to 100.
The fs Commands

[-backup_data_threshold <percent>]
Indicates the full percentage that a deduplicated file has to be below in order to trigger space-reduced backups for NDMP. For example, when set to 90, any deduplicated file whose physical size (compressed file plus changed blocks) is greater than 90 percent of the logical size of the file will have the entire file data backed up without attempting to back it up in a space-reduced format. Setting this value to zero disables it. The values range from 0 to 200 and the default value is 90 percent.

[-cifs_compression_enabled {yes|no}]
This option controls whether CIFS compression is allowed. When the default is yes, enable CIFS compression is allowed. When set to yes and the deduplication state of the file system is either on or suspended, then CIFS compression is enabled. If the deduplication state is either off or in the process of being turned off, then CIFS compression is not allowed, regardless of whether this option is set to yes.

-clear {<fs_name>|id=<fs_id>}
Sets the file system setting back to the Data Mover setting, which is the default setting.

[-minimum_scan_interval]
Defines the minimum number of days between completing one scan of a file system and before scanning the same file system again. The values range from 1 to 365 and the default value is 7 days.

[-minimum_size]
Defines the file size in KB that limits deduplication. File sizes equal to this value or smaller will not be deduplicated. File sizes greater than this value will be candidates for deduplication. Setting this value to zero disables it. This value should not be set lower than 24 KB. The values range from 0 to 1000 and the default value is 24 KB.

[-maximum_size]
Specifies the file size in MB of the largest file to be processed for deduplication. Files larger than this size in MB will not be deduplicated. Setting this value too high may affect system write performance as the first write operation reduplicates the file in its entirety. Setting this value to zero disables it. The values range from 0 to 8388608 and the default value is 8388608 MB.
[-access_time]
Defines the minimum required file age in days based on read access time. Files that have been read within the specified number of days will not be deduplicated. This setting does not apply to files with an FLR locked state. Setting this value to zero disables it. The values range from 0 to 365 and the default value is 15 days.

[-modification_time]
Defines the minimum required file age in days based on modification time. Files updated within the specified number of days will not be deduplicated. Setting this value to zero disables it. The values range from 0 to 365 and the default value is 15 days.

[-case_sensitive]
Defines whether case-sensitive (for NPS environments) or case-insensitive (for CIFS environments) string comparisons will be used during scans. By default, case-insensitive comparisons will be done to be consistent for CIFS environments. The default value is zero (false).

[-pathname_exclude_list]
This is a file system setting only (no global setting). It is empty by default.

Specifies a semicolon-delimited list of relative pathnames, in UTF-8 format, to be excluded from deduplication. Any directory below a specified pathname will be excluded from deduplication. You can specify a maximum of 10 pathnames and each one can be up to 1024 bytes. The default value is ’ ’ (empty).

[-file_ext_exclude_list]
Specifies a colon-delimited list of filename extensions to be excluded from deduplication. Each extension must include the leading dot. The default value is ’ ’ (empty).

[-duplicate_detection_method {sha1|byte|off}]
0 (off) — This means that duplicate data detection is disabled. With this setting, every deduplicated file is considered unique and the only space savings made are accomplished with compression.

1 (sha1) — The SHA-1 hash is used to detect duplicate data. It is faster than a byte comparison. This is the default method.

2 (byte) — This will use a byte-by-byte comparison to detect duplicate data. This adds considerable overhead especially for large files.
The fs Commands

[-savvol_threshold]
Represents the percentage of the configured save volume (SavVol) auto extension threshold that can be used during deduplication. After the specified amount of SavVol is used, deduplication stops on this file system. By default, this value is 90 percent and the SavVol auto extension is also 90 percent; this option will apply when the SavVol is 81 percent full (90 * 90). Setting this value to zero disables it. The values range from 0 to 100.

[-backup_data_threshold]
Indicates the full percentage that a deduplicated file has to be below in order to trigger space-reduced backups for NDMP. For example, when set to 90, any deduplicated file whose physical size (compressed file plus changed blocks) is greater than 90 percent of the logical size of the file will have the entire file data backed up without attempting to back it up in a space-reduced format. Setting this value to zero disables it. The values range from 0 to 200 and the default value is 90 percent.

[-cifs_compression_enabled]
This option controls whether CIFS compression is allowed. When the default is yes, enable CIFS compression. When set to yes and the deduplication state of the file system is either on or suspended, then CIFS compression is allowed. If the deduplication state is either off or in the process of being turned off, then CIFS compression is not allowed, regardless of whether this option is set to yes.

| -default { -info {<mover_name>|-all} | -set {<mover_name>|-all} }
Manages the Data Mover settings. The -set option determines the Data Mover settings.

[-minimum_scan_interval <days>]
Defines the minimum number of days between completing one scan of a file system and before scanning the same file system again. The values range from 1 to 365 and the default value is 7 days.

[-minimum_size <KB>]
Defines the file size in KB that limits deduplication. File sizes equal to this value or smaller will not be deduplicated. File sizes greater than this value will be candidates for deduplication.
Setting this value to zero disables it. This value should not be set lower than 24 KB. The values range from 0 to 1000 and the default value is 24 KB.

[-maximum_size <MB>]
Defines the file size in MB of the largest file to be processed for deduplication. Files larger than this size in MB will not be deduplicated. Setting this value too high may affect system write performance as the first write operation reduplicates the file in its entirety. Setting this value to zero disables it. The values range from 0 to 8388608 and the default value is 8388608 MB.

[-access_time <days>]
Specifies the minimum required file age in days based on read access time. Files that have been read within the specified number of days will not be deduplicated. This setting does not apply to files with an FLR locked state. Setting this value to zero disables it. The values range from 0 to 365 and the default value is 15 days.

[-modification_time <days>]
Specifies the minimum required file age in days based on modification time. Files updated within the specified number of days will not be deduplicated. Setting this value to zero disables it. The values range from 0 to 365 and the default value is 15 days.

[-case_sensitive {yes|no}]
Defines whether case-sensitive (for NPS environments) or case-insensitive (for CIFS environments) string comparisons will be used during scans. By default, case-insensitive comparisons will be done to be consistent for CIFS environments. The default value is zero (false).

[-file_ext_exclude_list <ext_list>]
Specifies a colon-delimited list of filename extensions to be excluded from deduplication. Each extension must include the leading dot. The default value is ‘’ (empty).

[-duplicate_detection_method {sha1|byte|off}]
0 (off) — This means that duplicate data detection is disabled. With this setting, every deduplicated file is considered unique and the only space savings made are accomplished with compression.

1 (sha1) — The SHA-1 hash is used to detect duplicate data. It is faster than a byte comparison. This is the default method.
2 (byte) — This will use a byte-by-byte comparison to detect duplicate data. This adds considerable overhead especially for large files.

[-savvol_threshold <percent>]
Represents the percentage of the configured save volume (SavVol) auto extension threshold that can be used during deduplication. After the specified amount of SavVol is used, deduplication stops on this file system. By default, this value is 90 percent and the SavVol auto extension is also 90 percent; this option will apply when the SavVol is 81 percent full (90 * 90). Setting this value to zero disables it. The values range from 0 to 100.

[-cpu_usage_low_watermark <percent>]
Defines the average percent of CPU usage that can be used during the deduplication process at which full throttle mode is re-entered. The values range from 0 to 100 and the default value is 40 percent. This is a global setting only.

[-cpu_usage_high_watermark <percent>]
Defines the average percent of CPU usage that can be used during the deduplication process which should trigger a slow throttle mode. The system starts in full throttle mode. The values range from 0 to 100 and the default value is 75 percent. This is a global setting only.

[-backup_data_threshold <percent>]
Defines the full percentage that a deduplicated file has to be below in order to trigger space-reduced backups for NDMP. For example, when set to 90, any deduplicated file whose physical size (compressed file plus changed blocks) is greater than 90 percent of the logical size of the file will have the entire file data backed up without attempting to back it up in a space-reduced format. Setting this value to zero disables it. The values range from 0 to 200 and the default value is 90 percent.

[-cifs_compression_enabled {yes|no}]
This option controls whether CIFS compression is allowed. When the default is yes, enable CIFS compression. When set to yes and the deduplication state of the file system is either on or suspended, then CIFS compression is allowed. If the deduplication state is either off or in the process of being turned off, then CIFS compression is not allowed, regardless of whether this option is set to yes.
The **-clear** option sets the global setting back to the default value.

<table>
<thead>
<tr>
<th><strong>-minimum_scan_interval</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines the minimum number of days between completing one scan of a file system and before scanning the same file system again. The values range from 1 to 365 and the default value is 7 days.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>-minimum_size</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines the file size in KB that limits deduplication. File sizes equal to this value or smaller will not be deduplicated. File sizes greater than this value will be candidates for deduplication. Setting this value to zero disables it. This value should not be set lower than 24 KB. The values range from 0 to 1000 and the default value is 24 KB.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>-maximum_size</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines the file size in MB of the largest file to be processed for deduplication. Files larger than this size in MB will not be deduplicated. Setting this value too high may affect system write performance as the first write operation reduplicates the file in its entirety. Setting this value to zero disables it. The values range from 0 to 8388608 and the default value is 8388608 MB.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>-access_time</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines the minimum required file age in days based on read access time. Files that have been read within the specified number of days will not be deduplicated. This setting does not apply to files with an FLR locked state. Setting this value to zero disables it. The values range from 0 to 365 and the default value is 15 days.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>-modification_time</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines the minimum required file age in days based on modification time. Files updated within the specified number of days will not be deduplicated. Setting this value to zero disables it. The values range from 0 to 365 and the default value is 15 days.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>-case_sensitive</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines whether case-sensitive (for NPS environments) or case-insensitive (for CIFS environments) string comparisons will be used during scans. By default, case-insensitive comparisons will be done to be consistent for CIFS environments. The default value is zero (false).</td>
</tr>
</tbody>
</table>
The fs Commands

[-file_ext_exclude_list]
Specifies a colon-delimited list of filename extensions to be excluded from deduplication. Each extension must include the leading dot. The default value is '' (empty).

[-duplicate_detection_method]
0 (off) — This means that duplicate data detection is disabled. With this setting, every deduplicated file is considered unique and the only space savings made are accomplished with compression.
1 (sha1) — The SHA-1 hash is used to detect duplicate data. It is faster than a byte comparison. This is the default method.
2 (byte) — This will use a byte-by-byte comparison to detect duplicate data. This adds considerable overhead especially for large files.

[-savvol_threshold]
Represents the percentage of the configured save volume (SavVol) auto extension threshold that can be used during deduplication. After the specified amount of SavVol is used, deduplication stops on this file system. By default, this value is 90 percent and the SavVol auto extension is also 90 percent; this option will apply when the SavVol is 81 percent full (90 * 90). Setting this value to zero disables it. The values range from 0 to 100.

[-cpu_usage_low_watermark]
Specifies the average percent of CPU usage that can be used during the deduplication process at which full throttle mode is re-entered. The values range from 0 to 100 and the default value is 25 percent. This is a global setting only.

[-cpu_usage_high_watermark]
Specifies the average percent of CPU usage that can be used during the deduplication process which should trigger a slow throttle mode. The system starts in full throttle mode. The values range from 0 to 100 and the default value is 75 percent. This is a global setting only.

[-backup_data_threshold <percent>]
Specifies the full percentage that a deduplicated file has to be below in order to trigger space-reduced backups for NDMP. For example, when set to 90, any deduplicated file whose physical size (compressed file plus changed blocks) is greater than 90 percent of the logical size of the file will have the entire file data
backed up without attempting to back it up in a space-reduced format. Setting this value to zero disables it. The values range from 0 to 200 and the default value is 90 percent.

[-cifs_compression_enabled]
This option controls whether CIFS compression is allowed. When the default is yes, enable CIFS compression. When set to yes and the deduplication state of the file system is either on or suspended, then CIFS compression is allowed. If the deduplication state is either off or in the process of being turned off, then CIFS compression is not allowed, regardless of whether this option is set to yes.

SEE ALSO:  nas_fs

EXAMPLE #1  To list the file systems and their deduplication states, type:

$ fs_dedupe -list

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>state</th>
<th>status</th>
<th>time_of_last_scan</th>
<th>original_data_size</th>
<th>usage</th>
<th>space_saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>141</td>
<td>ranap1_replica</td>
<td>Suspended</td>
<td>Idle</td>
<td>Wed Nov 12 09:04:45 EST 2008</td>
<td>5 MB</td>
<td>0%</td>
<td>0 MB (0%)</td>
</tr>
<tr>
<td>104</td>
<td>ds850gb_replica</td>
<td>On</td>
<td>Idle</td>
<td>Fri Nov 21 10:31:15 EST 2008</td>
<td>875459 MB</td>
<td>84%</td>
<td>341590 MB (39%)</td>
</tr>
<tr>
<td>495</td>
<td>cworm</td>
<td>On</td>
<td>Idle</td>
<td>Thu Nov 20 09:14:09 EST 2008</td>
<td>3 MB</td>
<td>0%</td>
<td>0 MB (0%)</td>
</tr>
<tr>
<td>33</td>
<td>chrisfsl</td>
<td>On</td>
<td>Idle</td>
<td>Sat Nov 22 10:04:33 EST 2008</td>
<td>1100 MB</td>
<td>18%</td>
<td>424 MB (38%)</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>File system identifier.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the file system.</td>
</tr>
<tr>
<td>state</td>
<td>Deduplication state of the file system. The file data is transferred to the storage which performs the deduplication and compression on the data. The states are:</td>
</tr>
<tr>
<td></td>
<td>• On — Deduplication on the file system is enabled.</td>
</tr>
<tr>
<td></td>
<td>• Suspended — Deduplication on the file system is suspended. Deduplication does not perform any new space reduction but existing space-reduced files remain the same.</td>
</tr>
<tr>
<td></td>
<td>• Off — Deduplication on the file system is disabled. Deduplication does not perform any new space reduction and the data is now reduplicated, which is the process used to restore a file that was deduplicated to its original condition.</td>
</tr>
</tbody>
</table>
The fs Commands

EXAMPLE #2

To list the file systems and provide detailed reports on the state of the deduplication processing, type:

```
$ fs_dedupe -info -all
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Current state of the deduplication-enabled file system. The progress statuses are:</td>
</tr>
<tr>
<td></td>
<td>• Idle — Deduplication process is currently idle.</td>
</tr>
<tr>
<td></td>
<td>• Scanning — File system is being scanned for deduplication. It displays the percentage of scanned files in the file system.</td>
</tr>
<tr>
<td></td>
<td>• Reduplicating — File system files are being reduplicated from the deduplicated files. It displays the percentage of reduplicated files.</td>
</tr>
<tr>
<td>time_of_last_scan</td>
<td>Time when the file system was last scanned.</td>
</tr>
<tr>
<td>original_data_size</td>
<td>Original size of the file system before deduplication.</td>
</tr>
<tr>
<td>usage</td>
<td>The current space usage of the file system.</td>
</tr>
<tr>
<td>space_saved</td>
<td>The file system space saved after deduplication.</td>
</tr>
</tbody>
</table>

**Value Definition**

- **status**: Current state of the deduplication-enabled file system. The progress statuses are:
  - Idle — Deduplication process is currently idle.
  - Scanning — File system is being scanned for deduplication. It displays the percentage of scanned files in the file system.
  - Reduplicating — File system files are being reduplicated from the deduplicated files. It displays the percentage of reduplicated files.

  - **time_of_last_scan**: Time when the file system was last scanned.
  - **original_data_size**: Original size of the file system before deduplication.
  - **usage**: The current space usage of the file system.
  - **space_saved**: The file system space saved after deduplication.

**EXAMPLE #2**

To list the file systems and provide detailed reports on the state of the deduplication processing, type:

```
$ fs_dedupe -info -all
```

```
Id = 53
Name = svr2fs1
Deduplication = Off
File system parameters:
  Case Sensitive = no
  Duplicate Detection Method = sha1
  Access Time = 15
  Modification Time = 15
  Minimum Size = 24 KB
  Maximum Size = 8388608 MB
  File Extension Exclude List =
  Minimum Scan Interval = 7
  Savevol Threshold = 90
  Backup Data Threshold = 90
  Cifs Compression Enabled = yes
  Pathname Exclude List =

Id = 2040
Name = server_2_fsltest2
Deduplication = Suspended
As of the last file system scan (Mon Aug 17 11:33:38 EDT 2009):
  Files scanned = 4
  Files deduped = 3 (75% of total files)
  File system capacity = 2016 MB
  Original data size = 6 MB (0% of current file system capacity)
  Space saved = 0 MB (0% of original data size)
File system parameters:
  Case Sensitive = no
  Duplicate Detection Method = sha1
  Access Time = 15
  Modification Time = 15
  Minimum Size = 24 KB
```
The fs Commands

Maximum Size = 8388608 MB
File Extension Exclude List =
Minimum Scan Interval = 7
Savevol Threshold = 90
Backup Data Threshold = 90
Cifs Compression Enabled = yes
Pathname Exclude List =

Id = 506
Name = demofs
Deduplication = Off

File system parameters:
   Case Sensitive = no
   Duplicate Detection Method = sha1
   Access Time = 15
   Modification Time = 15
   Minimum Size = 24 KB
   Maximum Size = 8388608 MB
File Extension Exclude List =
Minimum Scan Interval = 7
Savevol Threshold = 90
Backup Data Threshold = 90
Cifs Compression Enabled = yes
Pathname Exclude List =

Id = 2113
Name = testrdef
Deduplication = Suspended

As of the last file system scan (Thu Aug 13 14:22:31 EDT 2009):
   Files scanned = 1
   Files deduped = 0 (0% of total files)
   File system capacity = 1008 MB
   Original data size = 0 MB (0% of current file system capacity)
   Space saved = 0 MB (0% of original data size)

File system parameters:
   Case Sensitive = no
   Duplicate Detection Method = sha1
   Access Time = 15
   Modification Time = 15
   Minimum Size = 24 KB
   Maximum Size = 8388608 MB
File Extension Exclude List =
Minimum Scan Interval = 7
Savevol Threshold = 90
Backup Data Threshold = 90
Cifs Compression Enabled = yes
Pathname Exclude List =

Id = 2093
Name = kfs_ckpt1
Deduplication = Off

File system parameters:
The fs Commands

Case Sensitive = no
Duplicate Detection Method = sha1
Access Time = 15
Modification Time = 15
Minimum Size = 24 KB
Maximum Size = 8388608 MB
File Extension Exclude List =
Minimum Scan Interval = 7
Savevol Threshold = 90
Backup Data Threshold = 90
Cifs Compression Enabled = yes
Pathname Exclude List =

Id = 2095
Name = ranap-test3
Deduplication = On
Status = Idle

As of the last file system scan (Tue Aug 11 17:37:58 EDT 2009):
Files scanned = 30
Files deduped = 2 (7% of total files)
File system capacity = 5041 MB
Original data size = 1109 MB (22% of current file system capacity)
Space saved = 0 MB (0% of original data size)

File system parameters:
Case Sensitive = no
Duplicate Detection Method = sha1
Access Time = 15
Modification Time = 15
Minimum Size = 24 KB
Maximum Size = 8388608 MB
File Extension Exclude List =
Minimum Scan Interval = 7
Savevol Threshold = 90
Backup Data Threshold = 90
Cifs Compression Enabled = yes
Pathname Exclude List =

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deduplication</td>
<td>Current deduplication state of the file system.</td>
</tr>
<tr>
<td>Status</td>
<td>Progress status of the files being scanned.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the file system.</td>
</tr>
<tr>
<td>Id</td>
<td>File system identifier.</td>
</tr>
<tr>
<td>Files scanned</td>
<td>Number of files scanned.</td>
</tr>
<tr>
<td>Files deduped</td>
<td>Number of files in the file system that has been deduplicated.</td>
</tr>
<tr>
<td>Original data size</td>
<td>Proportion of space in use with respect to the file system capacity.</td>
</tr>
<tr>
<td>File system capacity</td>
<td>Current space usage of the file system.</td>
</tr>
<tr>
<td>Space saved</td>
<td>Proportion of space saved with respect to the original data size.</td>
</tr>
</tbody>
</table>
The fs Commands

<table>
<thead>
<tr>
<th>Case Sensitive</th>
<th>Method of string comparison: case-sensitive or case-insensitive.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate Detection Method</td>
<td>Method of duplication detection: 0, sha-1, or byte-by-byte.</td>
</tr>
<tr>
<td>Access Time</td>
<td>Minimum required file age in days based on read access time.</td>
</tr>
<tr>
<td>Modification Time</td>
<td>Minimum required file age in days based on modification time.</td>
</tr>
<tr>
<td>Minimum Size</td>
<td>Minimum file size to be processed for deduplication.</td>
</tr>
<tr>
<td>Maximum Size</td>
<td>Maximum file size to be processed for deduplication.</td>
</tr>
<tr>
<td>File Extension Exclude List</td>
<td>Lists filename extensions to be excluded from the deduplication.</td>
</tr>
<tr>
<td>Minimum Scan Interval</td>
<td>Minimum number of days between completing one scan of a file system and before scanning the same file system again.</td>
</tr>
<tr>
<td>SavVol Threshold</td>
<td>Percentage of SavVol space that can be used during deduplication.</td>
</tr>
<tr>
<td>Backup Data Threshold</td>
<td>Percentage below which a deduplicated file has to be in order to trigger space-reduced NDMP backups.</td>
</tr>
<tr>
<td>Cifs Compression Enabled</td>
<td>Controls whether CIFS permission is enabled.</td>
</tr>
<tr>
<td>Pathname Exclude List</td>
<td>Lists relative pathnames to be excluded from the deduplication.</td>
</tr>
</tbody>
</table>

**Note:** If reduplication fails, then the state transitions to the suspended state and a CCMD message will be sent to the server’s event log. If reduplication succeeds, then it remains in the off state.

**EXAMPLE #3**

To list the file systems for a given file system name, type:

```
$ fs_dedupe -info server3_fs3
```

<table>
<thead>
<tr>
<th>Id</th>
<th>98</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>server3_fs3</td>
</tr>
<tr>
<td>Deduplication</td>
<td>On</td>
</tr>
<tr>
<td>Status</td>
<td>Idle</td>
</tr>
</tbody>
</table>

As of the last file system scan on Tue Sep 23 13:28:01 EDT 2008:

- Files deduped: 30 (100%)
- File system capacity: 413590 MB
- Original data size: 117 MB (0% of current file system capacity)
- Space saved: 106 MB (90% of original data size)

File system parameters:

- Case Sensitive: yes
- Duplicate Detection Method: sha1
- Access Time: 30
- Modification Time: 30
- Minimum Size: 20
- Maximum Size: 200
- File Extension Exclude List: .jpg:.db:.pst
- Minimum Scan Interval: 1
- SavVol Threshold: 90
- Backup Data Threshold: 90
- Pathname Exclude List: root;etc

**EXAMPLE #2** provides a description of command output.
EXAMPLE #4  To list the file systems for a given file system name with the deduplication properties set to off, type:

```
$ fs_dedupe -info server3_fs3
```

<table>
<thead>
<tr>
<th>Id</th>
<th>98</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>server3_fs3</td>
</tr>
<tr>
<td>Deduplication</td>
<td>Off</td>
</tr>
</tbody>
</table>

File system parameters:
- Case Sensitive: yes
- Duplicate Detection Method: sha1
- Access Time: 30
- Modification Time: 30
- Minimum Size: 20
- Maximum Size: 200
- File Extension Exclude List: .jpg:.db:.pst
- Minimum Scan Interval: 1
- SavVol Threshold: 90
- Backup Data Threshold: 90
- Pathname Exclude List: root;etc

EXAMPLE #2 provides a description of command output.

EXAMPLE #5  To list the file systems for a given file system name with the deduplication properties set to off, type:

```
$ fs_dedupe -info server3_fs3
```

<table>
<thead>
<tr>
<th>Id</th>
<th>98</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>server3_fs3</td>
</tr>
<tr>
<td>Deduplication</td>
<td>Off</td>
</tr>
</tbody>
</table>

Status: re-duplicating - 20% complete

File system parameters:
- Case Sensitive: yes
- Duplicate Detection Method: sha1
- Access Time: 30
- Modification Time: 30
- Minimum Size: 20
- Maximum Size: 200
- File Extension Exclude List: .jpg:.db:.pst
- Minimum Scan Interval: 1
- SavVol Threshold: 90
- Backup Data Threshold: 90
- Cifs Compression Enabled: yes
- Pathname Exclude List: root;etc

EXAMPLE #2 provides a description of command output.

EXAMPLE #6  To list the duplication properties of a given Data Mover, type:

```
$ fs_dedupe -default -info server_2
```

Server parameters:
The fs Commands

Case Sensitive = yes
Duplicate Detection Method = sha1
Access Time = 30
Modification Time = 30
Minimum Size = 20
Maximum Size = 200
File Extension Exclude List = .jpg:.db:.pst
Minimum Scan Interval = 1
SavVol Threshold = 90
Backup Data Threshold = 90
CPU % Usage Low Water Mark = 25
CPU % Usage High Water Mark = 90
Cifs Compression Enabled = yes

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deduplication</td>
<td>Current deduplication state of the file system.</td>
</tr>
<tr>
<td>Status</td>
<td>Progress status of the files being scanned.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the file system.</td>
</tr>
<tr>
<td>Id</td>
<td>File system identifier.</td>
</tr>
<tr>
<td>Files scanned</td>
<td>Number of files scanned.</td>
</tr>
<tr>
<td>Files deduped</td>
<td>Number of files in the file system that has been deduplicated.</td>
</tr>
<tr>
<td>Original data size</td>
<td>Proportion of space in use with respect to the file system capacity.</td>
</tr>
<tr>
<td>File system capacity</td>
<td>Current space usage of the file system.</td>
</tr>
<tr>
<td>Space saved</td>
<td>Proportion of space saved with respect to the original data size.</td>
</tr>
<tr>
<td>Case Sensitive</td>
<td>Method of string comparison: case-sensitive or case-insensitive.</td>
</tr>
<tr>
<td>Duplicate Detection Method</td>
<td>Method of duplication detection: 0, sha-1, or byte-by-byte.</td>
</tr>
<tr>
<td>Access Time</td>
<td>Minimum required file age in days based on read access time.</td>
</tr>
<tr>
<td>Modification Time</td>
<td>Minimum required file age in days based on modification time.</td>
</tr>
<tr>
<td>Minimum Size</td>
<td>Minimum file size to be processed for deduplication.</td>
</tr>
<tr>
<td>Maximum Size</td>
<td>Maximum file size to be processed for deduplication.</td>
</tr>
<tr>
<td>File Extension Exclude List</td>
<td>Lists filename extensions to be excluded from the deduplication.</td>
</tr>
<tr>
<td>Minimum Scan Interval</td>
<td>Minimum number of days between completing one scan of a file system and before scanning the same file system again.</td>
</tr>
<tr>
<td>SavVol Threshold</td>
<td>Percentage of SavVol space that can be used during deduplication.</td>
</tr>
<tr>
<td>Backup Data Threshold</td>
<td>Percentage below which a deduplicated file has to be in order to trigger space-reduced NDMP backups.</td>
</tr>
<tr>
<td>CPU % Usage Low Water Mark</td>
<td>Average percentage of CPU usage which should trigger full throttle mode.</td>
</tr>
<tr>
<td>CPU % Usage High Water Mark</td>
<td>Average percentage of CPU usage which should trigger slow throttle mode.</td>
</tr>
<tr>
<td>Pathname Exclude List</td>
<td>Lists relative pathnames to be excluded from the deduplication.</td>
</tr>
</tbody>
</table>
EXAMPLE #7  To modify the file system, type:

$  fs_dedupe -modify testrdefs -state on

Done

EXAMPLE #8  To modify the file system settings to the user-specified values, type:

$  fs_dedupe -modify testrdefs -maximum_size 100
  -file_extension_exclude_list .jpg:.db:.pst

Done

EXAMPLE #9  To modify specific Data Mover settings, type:

$  fs_dedupe -default -set server_2 -maximum_size 100
  -minimum_size 20 -duplicate_detection_method sha1

Done

EXAMPLE #10  To reset the file system settings to the default settings (which are the Data Mover settings), type:

$  fs_dedupe -clear testrdefs -maximum_size -minimum_size
  -duplicate_detection_method

Done

EXAMPLE #11  To reset specific Data Mover settings to the default settings, type:

$  fs_dedupe -default -clear server_2 -maximum_size
  -minimum_size -duplicate_detection_method

Done

EXAMPLE #12  To reset all options for a specific Data Mover to the default settings, type:

$  fs_dedupe -default -clear server_2

Done

EXAMPLE #13  To reset all options on all Data Movers to the default settings, type:

$  fs_dedupe -default -clear -all

Done
The fs Commands

**fs_dhsm**

Manages the VNX FileMover file system connections.

**SYNOPSIS**

```
fs_dhsm
- list
| - info [<fs_name>|id=<fs_id>]
| - modify {<fs_name>|id=<fs_id>}[ -state enabled ]
| | [-popup_timeout <sec>][ -backup { offline | passthrough } ]
| | [-log { on | off }][ -max_log_size <mb>][ -offline_attr { on | off } ]
| | [ -read_policy_overide { none | full | passthrough | partial } ]
| - modify {<fs_name>|id=<fs_id>}[ -state disabled ]
| - connection {<fs_name>|id=<fs_id>}
| | [-state { enabled | disabled | recallonly } ]
| | [ -read_policy_overide { none | full | passthrough | partial } ]
| - create -type { nfsv3 | nfsv2 } -secondary <nfs_server>:<path>
| | [ -read_policy_overide { full | passthrough | partial | none } ]
| | [ -useRootCred { true | false } ] [ -proto { UDP | TCP } ] [ -nfsPort <port> ]
| | [ -mntPort <port> ] [ -mntVer { 3 | 2 | 1 } ] [ -localPort <port> ]
| - create -type cifs -admin [<fqdn>\]<admin_name>
| | [ -secondary \ [<fqdn>\]<share>[\<path>]
| | [ -local_server <host_name> ] [ -wins <address> ] [ -password <password> ]
| | [ -read_policy_overide { full | passthrough | partial | none } ]
| - create -type http -secondary http://<host><url_path>
| | [ -read_policy_overide { full | passthrough | partial | none } ]
| | [ -httpPort <port> ] [ -localPort <port> ]
| | [ -user <username> ] [ -password <password> ]
| | [ -timeout <seconds> ] [ -cgi { y | n } ]
| - create -type https -secondary https://<host><url_path>
| | [ -read_policy_overide { full | passthrough | partial | none } ]
| | [ -httpsPort <port> ] [ -localPort <port> ]
| | [ -user <username> ] [ -password <password> ]
| | [ -timeout <seconds> ] [ -cgi { y | n } ]
| - delete { -all | <cid> |,... } [ -recall_policy { check | no | yes } ]
| - modify { -all | <cid> |,... } [ -state { enabled | disabled | recallonly } ]
| | [ -read_policy_overide { full | passthrough | partial | none } ]
| | [ [-nfs_server <address> ] [ -localPort <port> ]
| | [ [-proto { TCP | UDP } ] [ -useRootCred { true | false } ] ]
| | [ [-cifs_server <fqdn>][ -local_server <host_name> ]
| | [ -password <password> ] [ -admin [<fqdn>\]<admin_name> ]
| | [ -wins <address> ]
| | [ [-http_server <host>][ -httpPort <port>][ -httpsPort <port> ]
| | [ -localPort <port>][ -user <username> ]
| | [ -password <password>][ -timeout <seconds> ]
```
The fs Commands

DESCRIPTION

The fs_dhsm command modifies the properties on file systems enabled for VNX FileMover. The fs_dhsm command creates, deletes, and modifies NFS, CIFS, and HTTP connections to remote hosts, lists VNX FileMover file systems, and provides information on the connections.

OPTIONS

-list
Lists all file systems enabled with the VNX FileMover.

-info [<fs_name>|id=<fs_id>]
Displays information for the specified VNX FileMover file systems.

-modify {<fs_name>|id=<fs_id>}
Sets VNX FileMover parameters for the specified file system.

Note: When specifying the -modify option on a disabled file system, the state is automatically changed to enabled. When specifying the -state disabled option, it is not possible to specify any other parameter to modify.

[ -state enabled ]
Enables VNX FileMover operations on the specified file system. The file system must be enabled to accept other options.

[ -state disabled ]
Disables VNX FileMover operations on the specified file system. New FileMover attributes cannot be specified as part of a disable command, nor can be specified for a file system that is in the disabled state. The attributes persist. If the file system is enabled after a disable command, then the attributes prior to the disable command take effect.

[ -popup_timeout <sec> ]
Specifies the Windows popup timeout value in seconds. If a CIFS I/O request cannot be processed within the specified time, then a popup notification of the delay is sent to the CIFS client. The default for <sec> is 0 (zero) which disables Windows popups.

Note: It may take up to 10 seconds before the popup is displayed.

[ -backup {offline|passthrough} ]
Specifies the nature of CIFS network backups. The offline option backs up the stub file only. The passthrough (default) option backs up all of the file data by using passthrough read.
The fs Commands

[-log {on|off}]
Enables or disables VNX FileMover logging. The default log filename is dhsm.log; it resides in the /etc directory on the FileMover-enabled file system.

[-max_log_size <mb>]
Specifies the maximum size of the log file. The current log file, in addition to four old log files, is saved. The minimum log file size is 10 MB.

[-offline_attr {on|off}]
Specifies whether the Data Mover should set the CIFS offline file attributes on the stub files. The default is on.

CAUTION
It is recommended that you do not disable the CIFS offline attributes.

[-read_policy_override {none|full|passthrough|partial}]
Specifies the migration method option used by the VNX, in the connection level or file system level, to override the migration method specified in the stub file. none (default) specifies no override, full recalls the whole file to the VNX on a read request before the data is returned, passthrough retrieves data without recalling the data to the VNX, and partial recalls only the blocks required to satisfy the client read request.

Note: The full migration may take several minutes or hours if the file is very large.

-connection {<fs_name>|id=<fs_id>} -list
Lists all connections for the specified file system.

-connection {<fs_name>|id=<fs_id>} -info [<cid>]
Displays details on all connections for the specified file system. If the <cid> is specified, only information for that connection is displayed.

Note: A connection ID is automatically created when a connection is established. The connection ID is displayed using the -list and is referred to as the <cid> in other commands.

NFS CONNECTIONS
-connection {<fs_name>|id=<fs_id>} -create -type {nfsv3|nfsv2} -secondary <nfs_server>:/<path>
The fs Commands

Creating a connection using the NFS protocol between the specified file system and the secondary file system. The secondary file system stores migrated data. The -type option specifies the NFS version that the Data Mover should use when connecting to the secondary server.

**Note:** VNX FileMover does not currently support NFSv4 protocol.

The -secondary option specifies the location of the remote file system.

**Note:** Although an IP address can be specified for an <nfs_server>, EMC strongly suggests using the hostname of the server, which allows you to take advantage of Domain Name System (DNS) failover capability.

```
[-read_policy_override {full | passthrough | partial | none}]
```

Specifies the migration method for data recall in response to client read requests. full migrates the whole file before it returns the requested blocks. passthrough leaves the stub file, but retrieves the requested data from the secondary file system. partial migrates only the blocks required to satisfy the client read request. none (default) defaults to the read method option specified in the stub file.

**Note:** The full migration may take minutes or hours if the file is very large.

```
[-useRootCred {true | false}]
```

Specifies the user credentials that the Data Mover uses when requesting data from the secondary VNX. When set to true, the Data Mover requests data as the root user (UID 0). When set to false (default), the Data Mover requests data as the owner of the file as specified in the stub file.

**Note:** If the -useRootCred option is set to true, the secondary storage NFS server must grant the Data Mover root privilege for NFS traffic.

```
[-proto {TCP | UDP}]
```

Specifies the protocol for the Data Movers to use for communication to the secondary <nfs_server>. TCP is the default.
The `fs` Commands

```bash
[-nfsPort <port>]
```
Specifies an NFS port on the secondary `<nfs_server>`. A default port is discovered automatically.

```bash
[-mntPort <port>]
```
Specifies a mount port on the secondary `<nfs_server>`. A default mount port is discovered automatically.

**Note:** The `-nfsPort` and the `-mntPort` options are used for secondary servers which do not have the Portmapper running. The admin starts the `nfsd` and `mountd` daemons on specific ports to avoid hackers.

```bash
[-mntVer {1 | 2 | 3}]
```
Specifies the mount version for the NFS connection. If the `-type` is `nfsv3`, then the `-mntVer` must be 3. If the `-type` is `nfsv2`, then 1 or 2 can be specified. The default for `nfsv2` is 2.

```bash
[-localPort <port>]
```
Overrides the default port that the Data Mover uses during connection to be compatible with firewalls. The default for UDP is 1020. By default, TCP uses a random port over 1024 to make the connection.

```bash
-connection {<fs_name>| id=<fs_id>} -modify { -all | <cid>[, <cid>...]} 
```
Changes parameters on an existing NFS VNX FileMover connection. Either all connections can be removed or just the specified `<cid>` connection can be removed.

```bash
[-state { enabled | disabled | recallonly } ]
```
Sets the state of VNX FileMover operations on the specified file system. `enabled` (default) allows both the creation of stub files and data migration through reads and writes. If the state is `disabled`, neither stub files nor data migration is possible. Data currently on the VNX can be read and written to in the disabled state.

If the state is `recallonly`, the policy engine is not allowed to create stub files, but the user is still able to trigger data migration using a read or write request from the secondary file system to the VNX.

```bash
[-read_policy_override { full | passthrough | partial | none } ]
```
Specifies the migration method option used by the VNX, in the connection level or file system level, to override the migration method specified in the stub file. `none` (default) specifies no
override, **full** recalls the whole file to the VNX on read request before the data is returned, **passthrough** retrieves data without recalling the data to the VNX, and **partial** recalls only the blocks required to satisfy the client read request.

---

**Note:** The full migration may take minutes or hours if the file is very large.

---

**[ -nfs_server <address> ]**

Specifies the name or IP address of the secondary NFS server.

---

**Note:** Although an IP address can be specified for the <nfs_server>, EMC strongly suggests using the hostname of the server, which allows use of the DNS failover capability.

---

**[ -localPort <port> ]**

Specifies a port to override the default port used by the Data Mover during connection for compatibility with firewalls.

---

**[ -proto {TCP|UDP} ]**

Specifies the protocol for the Data Mover to use for NFS communications to the secondary <nfs_server>. TCP is the default.

---

**[ -useRootCred {true|false} ]**

Specifies the user credentials that the Data Mover uses when requesting data from the secondary VNX. When set to true, the Data Mover requests data as the root user (UID 0). When set to false (default), the Data Mover requests data as the owner of the file as specified in the stub file.

---

**Note:** If the -useRootCred option is set to true, the secondary storage NFS server must grant the Data Mover root privilege for NFS traffic.

---

**-connection <fs_name> -delete { -all | <cid> [, <cid> ... ] }**

Removes an existing NFS connection between the file system and the secondary file system. Either all connections can be removed or just the specified <cid> connection can be removed.

---

**[ -recall_policy { check | no | yes } ]**

Specifies the recall policy for any migrated file during the -delete. check (default) scans the file system for stub files that depend on the connection and fails on the first one. no deletes the connection
without checking for stub files that depend on the connection, and yes migrates the files back to the VNX before the connection is removed. If no is specified and stub files exist, an I/O error appears when the file is read because the connection no longer exists.

**CIFS CONNECTIONS**

`-connection {<fs_name>|id=<fs_id>} -create -type cifs`

Creates a connection using the CIFS protocol between the specified file system and a secondary file system. A connection ID is automatically created when a connection is established. The connection ID is seen using the `-list` and is referred to as the `<cid>` in other commands.

`-admin [fqdn]\<admin_name>`

Specifies the `<admin_name>` used to make the CIFS connection. If an optional `<fqdn>` is specified, it must be a fully qualified domain name. The `<fqdn>\<admin_name>` entry must be enclosed within quotes as shown in EXAMPLE #2. If the `<fqdn>` is not specified, the `-local_server` domain is used.

`-secondary \\<fqdn>\<share>[\<path>]`

Specifies the CIFS server, the share, and path for the secondary server for connection. The `<fqdn>\<share>[\<path>]` entry must be enclosed within quotes. The domain must be fully qualified; an IP address will not work.

`-local_server <host_name>`

Specifies the NetBIOS name or computer name of the local CIFS server on the Data Mover.

`[-wins <address>]`

Specifies a WINS server to resolve names in a Windows domain.

`[-password <password>]`

Allows the user to specify the admin password. The password is not recorded in the command log. If the `-password` option is given but no password is specified, the user is prompted interactively.

**CAUTION**

When specifying the password with this option, be aware it is unmasked, and visible to other users. The command may also be read from the log of the shell.
[-read_policy.override {full | passthrough | partial | none}]

Specifies the migration method for data recall in response to client read requests. **full** migrates the whole file before it returns the requested blocks. **passthrough** leaves the stub file, but retrieves the requested data from the secondary file system. **partial** migrates only the blocks required to satisfy the client read request. **none** (default) defaults to the read method option specified in the stub file.

**Note:** The full migration may take several minutes or hours if the file is very large.

-connection {<fs_name>|id=<fs_id>}  -modify { -all | <cid>, <cid>...]

Changes parameters on an existing NFS VNX FileMover connection.

[-state {enabled | disabled | recallonly}]

Sets the state of VNX FileMover operations on the specified file system. **enabled** (default) allows both the creation of stub files and data migration through reads and writes. If the state is **disabled**, neither stub files nor data migration is possible. Data currently on the VNX can be read and written to in the disabled state.

If the state is **recallonly**, the policy engine is not allowed to create stub files, but the user is still able to trigger data migration using a read or write request from the secondary file system to the VNX.

[-read_policy.override {full | passthrough | partial | none}]

Specifies the migration method option used by the VNX, in the connection level or file system level, to override the migration method specified in the stub file. **none** (default) specifies no override, **full** recalls the whole file to the VNX on read request before the data is returned, **passthrough** retrieves data without recalling the data to the VNX, and **partial** recalls only the blocks required to satisfy the client read request.

**Note:** The full migration may take minutes or hours if the file is very large.
The fs Commands

[-cifs_server <fqdn>]  
Specifies the fully qualified domain name of the secondary CIFS server.

[-local_server <host_name>]  
Specifies the NetBIOS name or computer name of the local CIFS server on the Data Mover.

[-password <password>]  
Allows the user to specify the admin password. The password is not recorded in the command log. If the -password option is given but no password is specified, the user is prompted interactively.

When specifying the password with this option, be aware it is unmasked, and visible to other users. The command may also be read from the log of the shell.

[-admin [ <fqdn>\]<admin_name>]  
Specifies the <admin_name> used to make the CIFS connection. If an optional <fqdn> is specified, it must be a fully qualified domain name. If the <fqdn> is not specified, the -local_server domain is used.

[-wins <address>]}  
Specifies a WINS server to resolve names in a Windows domain.

-connection <fs_name> -delete {-all|<cid> [,<cid>...]}  
Removes an existing CIFS connection between the file system and the secondary file system.

[-recall_policy {check|no|yes}]  
Specifies the recall policy for any migrated file during the -delete option. check (default) scans the file system for stub files that depend on the connection and fails on the first one. no deletes the connection without checking for stub files that depend on the connection, and yes migrates the files back to the VNX before the connection is removed. If no is specified and stub files exist, an I/O error appears when the file is read because the connection no longer exists.

HTTP CONNECTIONS  
-connection {<fs_name>|id=<fs_id>} -create -type http -secondary http://<host><url_path>  
Creates a connection using the HTTP protocol between the specified primary file system and a secondary file system. There are two types
of HTTP connections: CGI and non-CGI. For CGI connections, the value of the `-secondary` option specifies the hostname of the server running the secondary storage HTTP server and the location of the CGI application that provides access to a storage system. For non-CGI connections, the value for the `-secondary` option specifies the hostname and, optionally, a portion of the hierarchical namespace published by the web server.

**Note:** Although an IP address can be specified for a `<host>`, EMC strongly suggests using the hostname of the server, which allows the DNS failover capability.

```
[ -read_policy_override { full | passthrough | partial | none } ]
```

Specifies the migration method option used by the VNX, in the connection level or file system level, to override the migration method specified in the stub file. `none` (default) specifies no override, `full` recalls the whole file to the VNX on read request before the data is returned, `passthrough` retrieves data without recalling the data to the VNX, and `partial` recalls only the blocks required to satisfy the client read request.

**Note:** The full migration may take several minutes or hours if the file is very large.

```
-httpPort <port>
```

Specifies the remote port number that the Data Mover delivers the HTTP request to. If not specified, the Data Mover issues HTTP requests to port 80 on the secondary storage HTTP server.

```
-localPort <port>
```

Specifies the local port number the Data Mover uses to issue HTTP requests to the web server active on the secondary storage. The `<port>` specified should be an integer no less than 1024. If not specified, the Data Mover selects a port to issue the HTTP requests.

**Note:** The two end points of an HTTP connection are specified by the file system name and the value specified for the `-secondary` option. If multiple connections are created by using identical end points with different attributes such as `-cgi`, `-user`, `-password`, `-localPort`, `-httpPort`, the connection will fail.
The fs Commands

[-user <username>]
Defines the username the HTTP client uses if digest authentication is required by the secondary storage HTTP server.

[-password <password>]
Allows the user to specify the admin password. The password is not recorded in the command log. If the -password option is given but no password is specified, the user is prompted interactively.

Use the -password option when digest authentication is required by the secondary storage HTTP server.

[-timeout <seconds>]
Specifies the timeout value in seconds. By default, the VNX HTTP client waits 30 seconds for a reply from the HTTP server and then retries the operation once.

[-cgi {y|n}]
Specifies the HTTP connection type: CGI or non-CGI. By default, FileMover assumes that the web server is using CGI connections to access migrated file data by using a CGI application. For non-CGI connections, set the -cgi option to n; FileMover then assumes the web server has direct access to migrated file content on secondary storage.

-connection {<fs_name>|id=<fs_id>} -modify {-all|<cid>[,<cid>...]}]
Changes parameters on an existing NFS VNX FileMover connection.

[-state {enabled|disabled|recallonly}]
Sets the state of VNX FileMover operations on the specified file system. enabled (default) allows both the creation of stub files and data migration through reads and writes. If the state is disabled, neither stub files nor data migration is possible. Data currently on the VNX can be read and written to in the disabled state.

If the state is recallonly, the policy engine is not allowed to create stub files, but the user is still able to trigger data migration by using a read or write request from the secondary file system to the VNX.

[-read_policy_override {full|passthrough|partial|none}]
Specifies the migration method option used by the VNX, in the connection level or file system level, to override the migration
method specified in the stub file. **none** (default) specifies no override, **full** recalls the whole file to the VNX on read request before the data is returned, **passthrough** retrieves data without recalling the data to the VNX, and **partial** recalls only the blocks required to satisfy the client read request.

---

**Note:** The full migration may take minutes or hours if the file is very large.

---

**[-http_server <host>]**

Specifies the hostname of the secondary storage HTTP server.

**-httpPort <port>**

Specifies the remote port number that the Data Mover delivers the HTTP request to. If not specified, the Data Mover issues HTTP requests to port 80 on the secondary storage HTTP server.

**-localPort <port>**

Specifies the local port number the Data Mover uses to issue HTTP requests to the web server active on the secondary storage. The <port> specified should be an integer no less than 1024. If not specified, the Data Mover selects a port to issue the HTTP requests.

---

**Note:** If you attempt to create multiple HTTP connections by using identical end points with different attributes such as **-cgi**, **-user**, **-password**, **-localPort**, **-httpPort**, the connection will fail.

---

**[-user <username>]**

An optional attribute used to define the username the HTTP client uses if digest authentication is required by the secondary storage HTTP server.

**[-password <password>]**

Allows the user to specify the admin password. The password is not recorded in the command log. If the **-password** option is given but no password is specified, the user is prompted interactively.

**[-timeout <sec>]**

Specifies the timeout value in seconds. By default, VNX’s HTTP client waits 30 seconds for a reply from the HTTP server and then retries the operation once before commencing the failover operation.
The fs Commands

-connection <fs_name> -delete {-all |<cid> [,<cid>... ]}
Removes an existing HTTP connection between the file system and the secondary file system. Either all connections can be removed or just the specified <cid> connection can be removed.

[-recall_policy {check|no|yes}]
Specifies the recall policy for any migrated file during the -delete option. The check (default) argument scans the file system for stub files that depend on the connection and fails on the first one. no deletes the connection without checking for stub files that depend on the connection, and yes migrates the files back to the VNX before the connection is removed. If no is specified and stub files exist, an I/O error appears when the file is read because the connection no longer exists.

HTTPS CONNECTIONS
-connection {<fs_name>|id=<fs_id>} -create -type https -secondary https://<host><url_path>
Creates a connection by using the HTTPS protocol between the specified primary file system and a secondary file system. There are two types of HTTPS connections: CGI and non-CGI. For CGI connections, the value of the -secondary option specifies the hostname of the server running the secondary storage HTTPS server and the location of the CGI application that provides access to a storage system. For non-CGI connections, the value for the -secondary option specifies the hostname and, optionally, a portion of the hierarchical namespace published by the web server.

Note: Although an IP address can be specified for a <host>, EMC strongly suggests using the hostname of the server, which allows the DNS failover capability.

[-read_policy_override {full|passthrough|partial |none}]
Specifies the migration method option used by the VNX, in the connection level or file system level, to override the migration method specified in the stub file. none (default) specifies no override, full recalls the whole file to the VNX on read request before the data is returned, passthrough retrieves data without recalling the data to the VNX, and partial recalls only the blocks required to satisfy the client read request.
Note: The full migration may take several minutes or hours if the file is very large.

[-httpsPort <port> ]
Specifies the remote port number that the Data Mover delivers the HTTPS request to. If not specified, the Data Mover issues HTTPS requests to port 443 on the secondary storage HTTPS server.

[-localPort <port> ]
Specifies the local port number the Data Mover uses to issue HTTPS requests to the web server active on the secondary storage. The <port> specified should be an integer no less than 1024. If not specified, the Data Mover selects a port to issue the HTTPS requests.

Note: The two end points of an HTTPS connection are specified by the file system name and the value specified for the -secondary option. If multiple connections are created by using identical end points with different attributes such as -cgi, -user, -password, -localPort, -httpsPort, the connection will fail.

[-user <username>]
Defines the username the HTTPS client uses if digest authentication is required by the secondary storage HTTPS server.

[-password <password>]
Allows the user to specify the admin password. The password is not recorded in the command log. If the -password option is given but no password is specified, the user is prompted interactively.

Use the -password option when digest authentication is required by the secondary storage HTTPS server.

[-timeout <seconds>]
Specifies the timeout value in seconds. By default, the VNX HTTPS client waits 30 seconds for a reply from the HTTPS server and then retries the operation once.

[-cgi {y|n}]
Specifies the HTTPS connection type: CGI or non-CGI. By default, FileMover assumes that the web server is using CGI connections to access migrated file data by using a CGI application. For
non-CGI connections, set the -cgi option to n; FileMover then assumes the web server has direct access to migrated file content on secondary storage.

```
-connection {<fs_name>|id=<fs_id>} -modify {--all|<cid>[,<cid>...]}
```

Changes parameters on an existing NFS VNX FileMover connection.

```
[-state {enabled|disabled|recallonly}]
```

Sets the state of VNX FileMover operations on the specified file system. `enabled` (default) allows both the creation of stub files and data migration through reads and writes. If the state is `disabled`, neither stub files nor data migration is possible. Data currently on the VNX can be read and written to in the disabled state.

If the state is `recallonly`, the policy engine is not allowed to create stub files, but the user is still able to trigger data migration by using a read or write request from the secondary file system to the VNX.

```
[-read_policy_override {full|passthrough|partial|none}]
```

Specifies the migration method option used by the VNX, in the connection level or file system level, to override the migration method specified in the stub file. `none` (default) specifies no override, `full` recalls the whole file to the VNX on read request before the data is returned, `passthrough` retrieves data without recalling the data to the VNX, and `partial` recalls only the blocks required to satisfy the client read request.

**Note:** The full migration may take minutes or hours if the file is very large.

```
[-http_server <host>]
```

Specifies the hostname of the secondary storage HTTPS server.

```
-httpsPort <port>
```

Specifies the remote port number that the Data Mover delivers the HTTPS request to. If not specified, the Data Mover issues HTTPS requests to port 443 on the secondary storage HTTPS server.
The fs Commands

Note: Although the -http_server option is used to modify the name of the secondary storage HTTPS server, files that can be converted into a stub by using an HTTPS connection can be brought back online using only HTTPS and not using NFS, CIFS, or even HTTP.

-localPort <port>
Specifies the local port number the Data Mover uses to issue HTTPS requests to the web server active on the secondary storage. The <port> specified should be an integer no less than 1024. If not specified, the Data Mover selects a port to issue the HTTPS requests.

Note: If you attempt to create multiple HTTPS connections by using identical end points with different attributes such as -cgi, -user, -password, -localPort, -httpsPort, the connection will fail.

-connection <fs_name> -delete {all|<cid> [,<cid>...]}{check|no|yes}
Removes an existing HTTPS connection between the file system and the secondary file system. Either all connections can be removed or just the specified <cid> connection can be removed.

Specifies the recall policy for any migrated file during the -delete. check (default) scans the file system for stub files that depend on the connection and fails on the first one. no deletes the connection without checking for stub files that depend on the connection,
and **yes** migrates the files back to the VNX before the connection is removed. If **no** is specified and stub files exist, an I/O error appears when the file is read because the connection no longer exists.

**SEE ALSO** *Using VNX FileMover, server_cifs, server_http, and server_nfs.*

**EXAMPLE #1** To enable VNX FileMover on a file system, type:

```
$ fs_dhsm -modify ufs1 -state enabled
```

```
ufs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = none
log file = on
max log size = 10MB
```

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>Whether VNX FileMover is enabled or disabled on the file system.</td>
</tr>
<tr>
<td>offline attr</td>
<td>Whether CIFS clients should be notified that a file is migrated.</td>
</tr>
<tr>
<td>popup timeout</td>
<td>Timeout value in seconds, before Windows popup notification is sent to the CIFS client.</td>
</tr>
<tr>
<td>backup</td>
<td>Nature of CIFS network backups.</td>
</tr>
<tr>
<td>read policy override</td>
<td>Migration method option used to override the read method specified in the stub file.</td>
</tr>
<tr>
<td>log file</td>
<td>Whether FileMover logging is enabled or disabled.</td>
</tr>
<tr>
<td>max log size</td>
<td>Maximum size of the log file.</td>
</tr>
</tbody>
</table>

**EXAMPLE #2** To create a CIFS connection for **ufs1** to the secondary file system `\\winserver2.nasdocs.emc.com\dhsm1` with a specified administrative account `nasdocs.emc.com\Administrator` and local server `dm102-cge0`:

```
$ fs_dhsm -connection ufs1 -create -type cifs -admin 'nasdocs.emc.com\Administrator' -secondary '\\winserver2.nasdocs.emc.com\dhsm1' -local_server dm102-cge0
```

Enter Password:********

```
ufs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = none
log file = on
```
max log size = 10MB
cid = 0
type = CIFS
secondary = \winserver2.nasdocs.emc.com\dhsm1\nstate = enabled
read policy override = none
write policy = full
local_server = DM102-CGE0.NASDOCS.EMC.COM
admin = nasdocs.emc.com\Administrator
wins =

Done

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>Whether VNX FileMover is enabled or disabled on the file system.</td>
</tr>
<tr>
<td>offline attr</td>
<td>Whether CIFS clients should be notified that a file is migrated.</td>
</tr>
<tr>
<td>popup timeout</td>
<td>Timeout value, in seconds, before a popup notification is sent to CIFS client.</td>
</tr>
<tr>
<td>backup</td>
<td>Nature of CIFS network backups.</td>
</tr>
<tr>
<td>read policy override</td>
<td>Migration method option used to override the read method specified in the stub file.</td>
</tr>
<tr>
<td>log file</td>
<td>Whether FileMover logging is enabled or disabled.</td>
</tr>
<tr>
<td>max log size</td>
<td>Maximum size of the log file.</td>
</tr>
<tr>
<td>cid</td>
<td>Connection ID.</td>
</tr>
<tr>
<td>type</td>
<td>Type of file system. The -list option provides information for a description of the types.</td>
</tr>
<tr>
<td>secondary</td>
<td>Hostname or IP address of the remote file system.</td>
</tr>
<tr>
<td>state</td>
<td>Specifies whether VNX FileMover is enabled or disabled on the file system.</td>
</tr>
<tr>
<td>read policy override</td>
<td>Migration method option used to override the read method specified in the stub file.</td>
</tr>
<tr>
<td>write policy</td>
<td>Write policy option used to recall data from secondary storage.</td>
</tr>
<tr>
<td>local_server</td>
<td>Name of the local CIFS server used to authenticate the CIFS connection.</td>
</tr>
</tbody>
</table>

EXAMPLE #3

To create a CIFS connection for ufs1 to the secondary file system \winserver2.nasdocs.emc.com\dhsm2 with a specified administrative account nasdocs.emc.com\Administrator, local server dm102-cge0, WINS server, and with the migration method set to full, type:

```
$ fs_dhsm -connection ufs1 -create -type cifs -admin 'nasdocs.emc.com\Administrator' -secondary '\winserver2.nasdocs.emc.com\dhsm1' -local_server dm102-cge0 -wins 172.24.102.25 -read_policy_override full
```

Enter Password:********

ufs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
EXAMPLE #2 provides a description of command output.

EXAMPLE #4 To display connection information for ufs1, type:

```bash
$ fs_dhsm -connection ufs1 -info 1
```

ufs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = none
log file = on
max log size = 10MB

EXAMPLE #2 provides a description of command output.

EXAMPLE #5 To modify the read_policy_override setting for connection 0 for ufs1, type:

```bash
$ fs_dhsm -connection ufs1 -modify 0
-read_policy_override passthrough
```

ufs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = none
log file = on
max log size = 10MB
cid = 0
type = CIFS
secondary = \winserver2.nasdocs.emc.com\dhsm1\
state = enabled
read policy override = pass
write policy = full
local_server = DM102-CGE0.NASDOCS.EMC.COM
admin = nasdocs.emc.com\Administrator
wins = 172.24.102.25
EXAMPLE #2 provides a description of command output.

EXAMPLE #6 To modify the VNX FileMover connection for ufs1, type:

$ fs_dhsm -connection ufs1 -modify 0 -nfs_server 172.24.102.115 -proto TCP

ufs1:
state = enabled
offline attr = on
popup timeout = 10
backup = offline
read policy override = full
log file = on
max log size = 25MB
cid = 0
type = NFSV3
secondary = 172.24.102.115:/export/dhsm1
state = enabled
read policy override = full
write policy = full
options = useRootCred=true proto=TCP
cid = 1
type = CIFS
secondary = \winserver2.nasdocs.emc.com\dhsm1\
state = enabled
read policy override = none
write policy = full
local_server = DM102-CGE0.NASDOCS.EMC.COM
admin = nasdocs.emc.com\Administrator
wins = 172.24.102.25
cid = 2
type = HTTP
secondary = http://172.24.102.115/export/dhsm1
state = enabled
read policy override = none
write policy = full
user =
options = cgi=n

Done

EXAMPLE #7 To create the NFSv3 connection for ufs1 to the secondary file system 172.24.102.115:/export/dhsm1 with the migration method set to full, the -useRootCred set to true and the protocol set to UDP, type:

$ fs_dhsm -connection ufs1 -create -type nfsv3 -secondary
172.24.102.115:/export/dhsm1 -read_policy_override full
-useRootCred true
/proto UDP

ufs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = none
log file = on
max log size = 10MB
cid = 0
type = CIFS
secondary = \winserver2.nasdocs.emc.com\dhsm1\
state = enabled
read policy override = pass
write policy = full
local_server = DM102-CGE0.NASDOCS.EMC.COM
admin = nasdocs.emc.com\Administrator
wins = 172.24.102.25
cid = 1
type = NFSV3
secondary = 172.24.102.115:/export/dhsm1
state = enabled
read policy override = full
write policy = full
options = useRootCred=true proto=UDP

Done

EXAMPLE #2 provides a description of command output.

EXAMPLE #8 To modify the VNX FileMover connection for ufs1, type:

```
$ fs_dhsm -connection ufs1 -modify 1 -proto TCP
```

ufs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = none
log file = on
max log size = 10MB
cid = 0
type = CIFS
secondary = \winserver2.nasdocs.emc.com\dhsm1\
state = enabled
read policy override = pass
write policy = full
local_server = DM102-CGE0.NASDOCS.EMC.COM
admin = nasdocs.emc.com\Administrator
wins = 172.24.102.25
EXAMPLE #2 provides a description of command output.

**EXAMPLE #9** To display VNX FileMover connection information for *ufs1*, type:

```
$ fs_dhsm -info ufs1
```

```
ufs1:
state                = enabled
offline attr         = on
popup timeout        = 0
backup               = passthrough
read policy override = none
log file             = on
max log size         = 10MB
```

**EXAMPLE #1** provides a description of command output.

EXAMPLE #10 To list VNX FileMover connections, type:

```
$ fs_dhsm -connection ufs1 -list
```

```
id   name   cid
29    ufs1   0
29    ufs1   1
29    ufs1   2
```

**EXAMPLE #11** To modify the VNX FileMover connection for *ufs1*, type:

```
$ fs_dhsm -modify ufs1 -popup_timeout 10 -backup offline
-log on -max_log_size 25 -offline_attr on
-read_policy_override full
```

```
ufs1:
state                = enabled
offline attr         = on
popup timeout        = 10
backup               = offline
read policy override = full
log file             = on
max log size         = 25MB
cid                 = 0
type                 = CIFS
secondary            = \winserver2.nasdocs.emc.com\dhsm1\nstate               = enabled
read policy override = pass
write policy         = full
```
EXAMPLE #2 provides a description of command output.

EXAMPLE #12

To modify the state of the VNX FileMover connection 0 for ufs1, type:

```bash
$ fs_dhsm -connection ufs1 -modify 0 -state disabled
```

ufs1:
state = enabled
offline attr = on
popup timeout = 10
backup = offline
read policy override = full
log file = on
max log size = 25MB
cid = 0
type = CIFS
secondary = \winserver2.nasdocs.emc.com\dhsm1\nstate = disabled
read policy override = pass
write policy = full
local_server = DM102-CGE0.NASDOCS.EMC.COM
admin = nasdocs.emc.com\Administrator
wins = 172.24.102.25
cid = 1
type = NFSV3
secondary = 172.24.102.115:/export/dhsm1
state = enabled
read policy override = full
write policy = full
options = useRootCred=true proto=TCP

Done

EXAMPLE #2 provides a description of command output.

EXAMPLE #13

To modify the state of the VNX FileMover connection 1 for ufs1, type:

```bash
$ fs_dhsm -connection ufs1 -modify 1 -state recallonly
```

ufs1:
state = enabled
offline attr  = on
popup timeout = 10
backup       = offline
read policy override = full
log file     = on
max log size = 25MB
cid          = 0
type          = CIFS
secondary     = \winserver2.nasdocs.emc.com\dhsm1\nstate         = enabled
read policy override = pass
write policy   = full
local_server  = DM102-CGE0.NASDOCS.EMC.COM
admin         = nasdocs.emc.com\Administrator
wins          = 172.24.102.25
cid          = 1
type          = NFSV3
secondary     = 172.24.102.115:/export/dhsm1
state         = recallonly
read policy override = full
write policy   = full
options       = useRootCred=true proto=TCP

Done

EXAMPLE #2 provides a description of command output.

EXAMPLE #14 To delete the VNX FileMover connections 1 and 2 for ufs1, and specify the recall policy for any migrated files during the delete, type:

```
$ fs_dhsm -connection ufs1 -delete 0,1 -recall_policy no
```

ufs1:
state = enabled
offline attr = on
popup timeout = 10
backup = offline
read policy override = full
log file = on
max log size = 25MB

Done

EXAMPLE #2 provides a description of command output.

EXAMPLE #15 To change the state of the VNX FileMover connection for ufs1 to disabled, type:

```
$ fs_dhsm -modify ufs1 -state disabled
```

ufs1:
state = enabled
offline attr = on
popup timeout = 10
EXAMPLE #1 provides a description of command output.

EXAMPLE #16
To create an HTTP connection for ufs1 to the secondary file system /export/dhsm1 on the web server http://172.24.102.115 which has direct access to the storage, type:

```
$ fs_dhsm -connection ufs1 -create -type http -secondary http://172.24.102.115/export/dhsm1 -cgi n
```

ufs1:
state                = enabled
offline attr         = on
popup timeout        = 10
backup               = offline
read policy override = full
log file             = on
max log size         = 25MB
cid                 = 2
type                 = HTTP
secondary            = http://172.24.102.115/export/dhsm1
state                = enabled
read policy override = none
write policy         = full
user                 =
options              = cgi=n

Done

EXAMPLE #2 provides a description of command output.

EXAMPLE #17
To create an HTTP connection for ufs1 to the secondary file system using CGI connections to access migrated file data using a CGI application, type:

```
$ fs_dhsm -connection ufs1 -create -type http -secondary http://www.nasdocs.emc.com/cgi-bin/access.sh
```

ufs1:
state                = enabled
offline attr         = on
popup timeout        = 0
backup               = passthrough
read policy override = none
log file             = on
max log size         = 10MB
cid                 = 0
type                 = HTTP
secondary = http://www.nasdocs.emc.com/cgi-bin/access.sh
state = enabled
read policy override = none
write policy = full
user =
options =
Done

EXAMPLE #2 provides a description of command output.

EXAMPLE #18 To create an HTTPS connection for server2_fs1 on the web server https://int16543 with read_policy_override set to full, type:

$ fs_dhsm -connection server2_fs1 -create -type https -secondary https://int16543 -read_policy_override full -cgi n

server2_fs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = passthrough
log file = on
max log size = 10MB
cid = 0
type = HTTPS
secondary = https://int16543
state = enabled
read policy override = full
write policy = full
user =
options =
Done

EXAMPLE #2 provides a description of command output.

EXAMPLE #19 To create an HTTPS connection for ufs1 to the secondary file system using CGI connections to access migrated file data using a CGI application, type:

$ fs_dhsm -connection ufs1 -create -type https -secondary https://www.nasdocs.emc.com/cgi-bin/access.sh

ufs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = none
log file = on
max log size = 10MB
cid = 0
type = HTTPS
EXAMPLE #2 provides a description of command output.

EXAMPLE #20
To create an HTTPS connection on **httpsPort 443** for **server2_ufs1** on the web server **https://int16543** with read_policy_override set to **passthrough**, type:

```
$ fs_dhsm -connection server2_fs1 -create -type https -secondary https://int16543 -read_policy_override passthrough -httpsPort 443 -cgi n
```

```
server2_fs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = passthrough
log file = on
max log size = 10MB
cid = 1
type = HTTPS
secondary = https://int16543
state = enabled
read policy override = pass
write policy = full
user =
options =
```

Done

EXAMPLE #2 provides a description of command output.

EXAMPLE #21
To create an HTTPS connection on **localPort 80** for **server2_ufs1** on the web server **https://int16543** with read_policy_override set to **passthrough**, type:

```
$ fs_dhsm -connection server2_fs1 -create -type https -secondary https://int16543 -read_policy_override passthrough -localPort 80 -cgi n
```

```
server2_fs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = passthrough
log file = on
max log size = 10MB
```
EXAMPLE #2 provides a description of command output.

**EXAMPLE #22**
To create an HTTPS connection on httpsPort 443 for server2_ufs1 on the web server https://int16543 with a specified user dhsm_user, type:

```bash
$ fs_dhsm -connection server2_fs1 -create -type https -secondary https://int16543 -read_policy_override full -httpsPort 443 -user dhsm_user -password dhsm_user -cgi
```

```
server2_fs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = passthrough
log file = on
max log size = 10MB
cid = 1
type = HTTPS
secondary = https://int16543
state = enabled
read policy override = full
write policy = full
user = dhsm_user
options =
```

Done

EXAMPLE #2 provides a description of command output.

**EXAMPLE #23**
To modify the read_policy_override setting for connection 1 from server2_fs1, type:

```bash
$ fs_dhsm -connection server2_fs1 -modify 1 -read_policy_override passthrough
```

```
server2_fs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = passthrough
log file = on
```
The fs Commands

max log size = 10MB
cid = 1
type = HTTPS
secondary = https://int16543
state = enabled
read policy override = pass
write policy = full
user = dhsm_user
options =

Done

EXAMPLE #2 provides a description of command output.

EXAMPLE #24 To delete the VNX FileMover connection 0 for ufs1, type:

$ fs_dhsm -connection 0 ufs1 -delete 0

ufs1:
state = enabled
offline attr = on
popup timeout = 0
backup = passthrough
read policy override = none
log file = on
max log size = 10MB

Done

EXAMPLE #1 provides a description of command output.
**fs_group**

Creates a file system group from the specified file systems or a single file system.

**SYNOPSIS**

```
fs_group
  -list
  -delete <fs_group_name>
  -info {<fs_group_name>|id=<fs_group_id>}
  [-name <name>] -create {<fs_name>,...}
  -xtend <fs_group_name> {<fs_name>,...}
  -shrink <fs_group_name> {<fs_name>,...}
```

**DESCRIPTION**

The *fs_group* command combines file systems to be acted upon simultaneously as a single group for TimeFinder/FS.

**OPTIONS**

- **-list**
  Displays a listing of all file system groups.

  __Note:__ The ID of the object is an integer and is assigned automatically. The name of a file system may be truncated if it is too long for the display. To display the full name, use the -info option with a file system ID.

- **-delete <fs_group_name>**
  Deletes the file system group configuration. Individual file systems are not deleted.

- **-info {<fs_group_name>|id=<fs_group_id>}**
  Displays information about a file system group, either by name or group ID.

- **[-name <name>] -create {<fs_name>,...}**
  Creates a file system group from the specified file systems. If a name is not specified, one is assigned by default.

- **-xtend <fs_group_name> {<fs_name>,...}**
  Adds the specified file systems or group to a file system group.

- **-shrink <fs_group_name> {<fs_name>,...}**
  Removes the specified file systems or group from a file system group. Individual file systems are not deleted.

**SEE ALSO**

*Managing Volumes and File Systems for VNX Manually* and *Using TimeFinder/FS, NearCopy, and FarCopy on VNX for File, fs_timefinder*, and *nas_fs*. 
The number associated with the storage device is dependent on the attached storage system. VNX for block displays a prefix of APM before a set of integers, for example, APM00033900124-0019. Symmetrix storage storage systems appear as 002804000190-003C.

**EXAMPLE #1**

To create a file system group named, `ufsg1`, and add `ufs1`, type:

```
$ fs_group -name ufsg1 -create ufs1
```

| id    | = 22  |
| name  | = ufsg1 |
| acl   | = 0 |
| in_use| = False |
| type  | = group |
| fs_set| = ufs1 |
| pool  | = |
| stor_devs | = 000187940268-0006,000187940268-0007,000187940268-0008,000187940268-0009 |
| disks | = d3,d4,d5,d6 |

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the group that is automatically assigned.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to the group.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control value for the group.</td>
</tr>
<tr>
<td>in_use</td>
<td>Whether a file system is used by a group.</td>
</tr>
<tr>
<td>type</td>
<td>Type of file system.</td>
</tr>
<tr>
<td>fs_set</td>
<td>File systems that are part of the group.</td>
</tr>
<tr>
<td>pool</td>
<td>Storage pool given to the file system group.</td>
</tr>
<tr>
<td>stor_devs</td>
<td>Storage system devices associated with the group.</td>
</tr>
<tr>
<td>disks</td>
<td>Disks on which the metavolume resides.</td>
</tr>
</tbody>
</table>

**EXAMPLE #2**

To list all file system groups, type:

```
$ fs_group -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>acl</th>
<th>in_use</th>
<th>type</th>
<th>member_of</th>
<th>fs_set</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>ufsg1</td>
<td>0</td>
<td>n</td>
<td>100</td>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>member_of</td>
<td>Groups which the file system group belong to.</td>
</tr>
</tbody>
</table>
EXAMPLE #3  To display information for the file system group, ufsg1, type:

```
$ fs_group -info ufsg1
```

id = 22
name = ufsg1
acl = 0
in_use = False
type = group
fs_set = ufs1
pool =
stor_devs =
000187940268-0006,000187940268-0007,000187940268-0008,000187940268-0009
disks = d3,d4,d5,d6

EXAMPLE #1 provides a description of command output.

EXAMPLE #4  To add file system, ufs2, to the file system group, ufsg1, type:

```
$ fs_group -xtend ufsg1 ufs2
```

id = 22
name = ufsg1
acl = 0
in_use = False
type = group
fs_set = ufs1,ufsg1
pool =
stor_devs =
000187940268-0006,000187940268-0007,000187940268-0008,000187940268-0009,00018794
0268-000A,000187940268-000B,000187940268-000C,000187940268-000D
disks = d3,d4,d5,d6,d7,d8,d9,d10

EXAMPLE #1 provides a description of command output.

EXAMPLE #5  To remove file system, ufs2, from the file system group, ufsg1, type:

```
$ fs_group -shrink ufsg1 ufs2
```

id = 22
name = ufsg1
acl = 0
in_use = False
type = group
fs_set = ufs1
pool =
stor_devs =
000187940268-0006,000187940268-0007,000187940268-0008,000187940268-0009

disks = d3,d4,d5,d6

EXAMPLE #1 provides a description of command output.
**EXAMPLE #6**  To delete file system group, *ufsg1*, type:

```
$ fs_group -delete ufsg1
```

<table>
<thead>
<tr>
<th>id</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>ufsg1</td>
</tr>
<tr>
<td>acl</td>
<td>0</td>
</tr>
<tr>
<td>in_use</td>
<td>False</td>
</tr>
<tr>
<td>type</td>
<td>group</td>
</tr>
<tr>
<td>fs_set</td>
<td></td>
</tr>
<tr>
<td>stor_devs</td>
<td></td>
</tr>
<tr>
<td>disks</td>
<td></td>
</tr>
</tbody>
</table>

**EXAMPLE #1** provides a description of command output.
fs_rdf

Manages the Remote Data Facility (RDF) functionality for a file system residing on RDF drives.

SYNOPSIS

```
fs_rdf {<fs_name>|id=<fs_id>} 
   -Mirror {on|off|refresh}
   | -Restore [-Force]
   | -info
```

Note: RDF is supported only on a VNX attached to a Symmetrix.

DESCRIPTION

The `fs_rdf` command turns mirroring on and off for an RDF file system and displays information about RDF relationships.

OPTIONS

- **-Mirror {on|off|refresh}**
  The `on` option resumes the link between the RDF drives of a file system thereby enabling mirroring for the RDF file system. The `off` option halts mirroring between the file systems, and the `refresh` option does an immediate mirror `on` then `off` which refreshes the file system image.

- **-Restore [-Force]**
  Restores a file system from the R2 side (remote) when remote TimeFinder/FS FarCopy is used. The `-Restore` can only be executed on the R1 side. The `-Force` option must be used when restoring a file system with enabled.

- **-info**
  Displays information about RDF relationships.

SEE ALSO


EXAMPLE #1

To turn on mirroring for **ufs1_snap1** from the R1 Control Station, type:

```
$ fs_rdf ufs1_snap1 -Mirror on
```

```
id        = 20
name      = ufs1_snap1
acl       = 0
in_use    = False
type      = uxfs
volume    = v168
```
The fs Commands

```plaintext
pool =
rw_servers=
ro_servers=
rw_vdms =
ro_vdms =
backup_of = ufs1 Fri Apr 23 16:29:23 EDT 2004
stor_devs =
002804000190-0052,002804000190-0053,002804000190-0054,002804000190-0055
disks = rootd33,rootd34,rootd35,rootd36

RDF Information:
remote_symid = 002804000218
remote_sym_devname =
ra_group_number = 2
dev_rdf_type = R1
dev_ra_status = READY
dev_link_status = READY
rdf_mode = SYNCHRONOUS
rdf_pair_state = SYNCINPROG
rdf_domino = DISABLED
adaptive_copy = DISABLED
adaptive_copy_skew = 65535
num_r1_invalid_tracks = 0
num_r2_invalid_tracks = 736440
dev_rdf_state = READY
remote_dev_rdf_state = WRITE_DISABLED
rdf_status = 0
link_domino = DISABLED
prevent_auto_link_recovery = DISABLED
link_config =
suspend_state = NA
consistency_state = DISABLED
adaptive_copy_wp_state = NA
prevent_ra_online_upon_pwron = ENABLED
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of a file system that is assigned automatically.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to a file system.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control value for a file system.</td>
</tr>
<tr>
<td>in_use</td>
<td>Whether a file system is registered into the mount table.</td>
</tr>
<tr>
<td>type</td>
<td>Type of file system. <a href="#">nas_fs</a> provides a description of the types.</td>
</tr>
<tr>
<td>volume</td>
<td>Volume on which a file system resides.</td>
</tr>
<tr>
<td>pool</td>
<td>Storage pool for the file system.</td>
</tr>
<tr>
<td>rw_servers</td>
<td>Servers with read-write access to a file system.</td>
</tr>
<tr>
<td>ro_servers</td>
<td>Servers with read-only access to a file system.</td>
</tr>
<tr>
<td>rw_vdms</td>
<td>VDM servers with read-write access to a file system.</td>
</tr>
<tr>
<td>ro_vdms</td>
<td>VDM servers with read-only access to a file system.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>backup_of</td>
<td>The remote RDF file system.</td>
</tr>
<tr>
<td>stor_devs</td>
<td>The storage system devices associated with a file system.</td>
</tr>
<tr>
<td>disks</td>
<td>The disks on which the metavolume resides.</td>
</tr>
<tr>
<td>remote_symid</td>
<td>The serial number of the storage system containing the target volume.</td>
</tr>
<tr>
<td>remote_sym_devname</td>
<td>The storage system device name of the remote device in an RDF pair.</td>
</tr>
<tr>
<td>ra_group_number</td>
<td>The RA group number (1-n).</td>
</tr>
<tr>
<td>dev_rdf_type</td>
<td>The type of RDF device. Possible values are: R1 and R2.</td>
</tr>
<tr>
<td>dev_ra_status</td>
<td>RA status. Possible values are: READY, NOT READY, WRITE_DISABLED, STATUS NA,</td>
</tr>
<tr>
<td></td>
<td>STATUS_MIXED.</td>
</tr>
<tr>
<td>dev_link_status</td>
<td>Link status. Possible values are: READY, NOT READY, WRITE_DISABLED, NA, MIXED.</td>
</tr>
<tr>
<td>rdf_mode</td>
<td>The RDF mode. Possible values are: SYNCHRONOUS, SEMI_SYNCHRONOUS, ADAPTIVE COPY, MIXED.</td>
</tr>
<tr>
<td>rdf_pair_state</td>
<td>The composite state of the RDF pair. Possible values: INVALID, SYNCPROG, SYNCHRONIZED, SPLIT, SUSPENDED, FAILED_OVER, PARTITIONED, R1_UPDINPROG, MIXED.</td>
</tr>
<tr>
<td>rdf_domino</td>
<td>The RDF device domino. Possible values are: ENABLED, DISABLED, MIXED.</td>
</tr>
<tr>
<td>adaptive_copy</td>
<td>Possible values are: DISABLED, WP_MODE, DISK_MODE, MIXED.</td>
</tr>
<tr>
<td>adaptive_copy_skew</td>
<td>Number of invalid tracks when in Adaptive copy mode.</td>
</tr>
<tr>
<td>num_r1_invalid_tracks</td>
<td>Number of invalid tracks on the source (R1) device.</td>
</tr>
<tr>
<td>num_r2_invalid_tracks</td>
<td>Number of invalid tracks on the target (R2) device.</td>
</tr>
<tr>
<td>dev_rdf_state</td>
<td>Specifies the composite RDF state of the RDF device. Possible values are: READY, NOT READY, WRITE_DISABLED, NA, MIXED.</td>
</tr>
<tr>
<td>remote_dev_rdf_state</td>
<td>Specifies the composite RDF state of the remote RDF device. Possible values are: READY, NOT READY, WRITE_DISABLED, NA, MIXED.</td>
</tr>
<tr>
<td>rdf_status</td>
<td>Specifies the RDF status of the device. Possible values are: READY, NOT READY, WRITE_DISABLED, NA, MIXED.</td>
</tr>
<tr>
<td>link_domino</td>
<td>RDF link domino. Possible values are: ENABLED, DISABLED.</td>
</tr>
<tr>
<td>prevent_auto_link_recovery</td>
<td>When enabled, prevents the automatic resumption of data copy across the RDF links as soon as the links have recovered. Possible values are: ENABLED, DISABLED.</td>
</tr>
<tr>
<td>link_config</td>
<td>Possible values are: CONFIG_ESCON, CONFIG_T3.</td>
</tr>
<tr>
<td>suspend_state</td>
<td>Specifies the status of R1 devices in a consistency group. Possible states are: NA, OFFLINE, OFFLINE_PEND, ONLINE_MIXED.</td>
</tr>
<tr>
<td>consistency_state</td>
<td>Specifies state of an R1 device related to consistency groups. Possible states are: ENABLED, DISABLED.</td>
</tr>
<tr>
<td>adaptive_copy_wp_state</td>
<td>Specifies state of the adaptive copy mode. Possible states are: NA, OFFLINE, OFFLINE_PEND, ONLINE_MIXED.</td>
</tr>
<tr>
<td>prevent_ra_online_upon_pwrnn</td>
<td>Specifies the state of the RA director coming online after power on. Possible states are: ENABLED, DISABLED.</td>
</tr>
</tbody>
</table>
EXAMPLE #2  To display RDF-related information for ufs1_snap1 from the R2 Control Station, type:

```bash
$ fs_rdf ufs1_snap1 -info
```

```
id        = 20
name      = ufs1_snap1
acl       = 0
in_use    = False
type      = uxfs
volume    = v168
pool      =
rw_servers=
ro_servers=
rw_vdms   =
ro_vdms   =
backup_of = ufs1 Fri Apr 23 16:29:23 EDT 2004
stor_devs = 002804000190-0052,002804000190-0053,002804000190-0054,002804000190-0055
disks     = rootd33,rootd34,rootd35,rootd36

RDF Information:
remote_symid                 = 002804000218
remote_sym_devname           =
ra_group_number              = 2
dev_rdf_type                 = R1
dev_ra_status                = READY
dev_link_status              = READY
rdf_mode                     = SYNCHRONOUS
rdf_pair_state               = SYNCINPROG
rdf_domino                   = DISABLED
adaptive_copy                = DISABLED
adaptive_copy_skew           = 65535
num_r1_invalid_tracks        = 0
num_r2_invalid_tracks        = 696030
dev_rdf_state                = READY
remote_dev_rdf_state         = WRITE_DISABLED
rdf_status                   = 0
link_domino                  = DISABLED
prevent_auto_link_recovery   = DISABLED
link_config                  =
suspend_state                = NA
consistency_state            = DISABLED
adaptive_copy_wp_state       = NA
prevent_ra_online_upon_pwron = ENABLED
```

EXAMPLE #1 provides a description of command output.
EXAMPLE #3  To turn the mirroring off for ufs1_snap1 on the R1 Control Station, type:

```
$ fs_rdf ufs1_snap1 -Mirror off
```

```text
remainder(MB) = 20548..17200..13110..8992..4870..746 0
id = 20
name = ufs1_snap1
acl = 0
in_use = False
type = uxfs
volume = v168
pool =
rw_servers=
ro_servers=
rw_vdms =
ro_vdms =
backup_of = ufs1 Fri Apr 23 16:29:23 EDT 2004
stor_devs = 002804000190-0052,002804000190-0053,002804000190-0054,002804000190-0055
disks = rootd33,rootd34,rootd35,rootd36

RDF Information:
remote_symid = 002804000218
remote_sym_devname =
ra_group_number = 2
dev_rdf_type = R1
dev_ra_status = READY
dev_link_status = NOT_READY
rdf_mode = SYNCHRONOUS
rdf_pair_state = SUSPENDED
rdf_domino = DISABLED
adaptive_copy = DISABLED
adaptive_copy_skew = 65535
num_r1_invalid_tracks = 0
num_r2_invalid_tracks = 0
dev_rdf_state = READY
remote_dev_rdf_state = WRITE_DISABLED
rdf_status = 0
link_domino = DISABLED
prevent_auto_link_recovery = DISABLED
link_config =
suspend_state = OFFLINE
consistency_state = DISABLED
adaptive_copy_wp_state = NA
prevent_ra_online_upon_pwron = ENABLED
```

EXAMPLE #1 provides a description of command output.
EXAMPLE #4 To perform a mirror refresh for ufs1_snap1 on the R1 Control Station, type:

```
$ fs_rdf ufs1_snap1 -Miror refresh
```

remainder(MB) = 1 0

id        = 20
name      = ufs1_snap1
acl       = 0
in_use    = False
type      = uXfs
volume    = v168
pool      =

```
RDF Information:
remote_symid = 002804000218
remote_sym_devname =
ra_group_number = 2
dev_rdf_type = R1
dev_ra_status = READY
dev_link_status = NOT_READY
rdf_mode = SYNCHRONOUS
rdf_pair_state = SUSPENDED
rdf_domino = DISABLED
adaptive_copy = DISABLED
adaptive_copy_skew = 65535
num_r1_invalid_tracks = 0
num_r2_invalid_tracks = 0
dev_rdf_state = READY
remote_dev_rdf_state = WRITE_DISABLED
rdf_status = 0
link_domino = DISABLED
prevent_auto_link_recovery = DISABLED
link_config =
suspend_state = OFFLINE
consistency_state = DISABLED
adaptive_copy_wp_state = NA
prevent_ra_online_upon_pwron = ENABLED
```

EXAMPLE #1 provides a description of command output.
EXAMPLE #5 To restore the file system ufs1_snap1 from the R1 Control Station, type:

```
$ /nas/sbin/rootfs_rdf ufs1_snap1 -Restore
```
The `fs` Commands

**fs_timefinder**

Manages the TimeFinder/FS functionality for the specified file system or file system group.

**SYNOPSIS**

```
fs_timefinder {<fs_name>|id=<fs_id>}
    -Mirror {on|off|refresh [-Force]}
    | [-name <name>] -Snapshot [volume <volume_name>][option <options>]
    | -Restore [-Force][option <options>]

Note: TimeFinder/FS is supported only on a VNX attached to a Symmetrix.
```

**DESCRIPTION**

The `fs_timefinder` command creates a copy of a file system or file system group that can be placed into a mirrored mode with its original file system. The Symmetrix must already have Business Continuance Volumes (BCV) configured to the same size as the volumes on the VNX. After the copy of the file system has been made, it can be mounted on any Data Mover.

**OPTIONS**

```
-Mirror {on|off|refresh [-Force]}
```

- `on`: Places the unmounted file system copy, created by using the `-Snapshot` option, into mirrored mode with its original file system. The file system copy is frozen and remains unavailable to users until mirrored mode is turned `off`.

- `refresh`: Initiates an immediate `-Mirror on then off` for the unmounted file system copy, thereby refreshing the file system copy.

The file system copy should not be mounted read-write when placed into mirrored mode or when refreshed. If the file system copy is mounted read-write, the `-Force` option can be used to force a refresh if the metavolume is an STD type. The `-Force` option requires root command and must be executed by using `/nas/sbin/rootfs_timefinder`.

**CAUTION**

Performing a mirror refresh may be time consuming, relative to the amount of data that has changed in the file system.

```
-name <name> -Snapshot
```

Creates a copy of a file system and assigns an optional name to the file system copy. If a name is not specified, one is assigned by default.
If no options are provided, a name and metavolume are automatically assigned. Use `nas_fs` to delete the copy of the file system.

**CAUTION**
Creating a copy by using -Snapshot may be time consuming, relative to the size of a file system.

```
[-volume <volume_name>]
Assigns a metavolume to a file system copy. The metavolume must be created by using the `nas_volume -Clone` command prior to executing this option. The metavolume must be a BCV type and have the same characteristics as the metavolume of the original file system.

[-option <options>]
Specifies the following comma-separated options:

mirror=on
Leaves the file system copy in mirrored mode.

disktype=<disktype>
For systems with both local and R1BCVs, specifies the type of volume to use when creating a snapshot. In a TimeFinder/FS FarCopy configuration, use `disktype=R1BCV` for creating a snapshot of the PFS on the local VNX for file. For creating a snapshot of an imported FarCopy snapshot on the remote VNX for file, use `disktype=STD`. This option is supported for only RAID group based disk volumes and cannot be combined with the "mapped_pool=" option.

By default, the system uses the first available R1BCV or BCV, or R1STD or STD device.

Use the `disktype=` option to designate which to use if there are R1 devices in your configuration.

pool=<mapped_pool>
Specifies the mapped pool to use when creating a snapshot from that pool. This option is supported only for mapped pool disk volumes and cannot be combined with the `disktype=` option.

A mapped pool is a VNX for file storage pool that is dynamically generated when diskmark is run. It is a one-to-one mapping with either a VNX for block storage pool or a Symmetrix Storage Group.
Note: If the pool= option is used when creating a snapshot, the disk volume will be selected from only this pool. If the pool does not have enough disk volumes to create a snapshot for the source file system, the fs_timefinder command reports an error.

-Restore
Restores a file system to its original location by using the unmounted file system copy created with the -Snapshot option. The original file system must not have any associated SnapSure checkpoints.

CAUTION
Restoring a file system may be time consuming, relative to the amount of data that has changed in the file system.

[-Force]
Forces a restore of a file system copy that is mounted on the metavolume as read-only, or if the volume is an STD type.

[-option <options>]
Specifies the following comma-separated options:

mirror=on
Places the file system copy in mirrored mode.

SEE ALSO Using TimeFinder/FS, NearCopy, and FarCopy on VNX for File, fs_ckpt, fs_group, and nas_fs.

EXAMPLE #1 To create a TimeFinder/FS copy of the PFS, type:

$ fs_timefinder ufs1 -Snapshot

operation in progress (not interruptible)...
remainder (MB) = 43688...37205...31142...24933...18649...12608...7115...4991...4129...3281...2457...1653...815...0
operation in progress (not interruptible)...id = 18
name = ufs1
acl = 0
in_use = True
type = uxfs
worm = off
volume = mtv1
pool =

rw_servers= server_2
ro_servers=

rw_vdms =
ro_vdms =
backups = ufs1_snap1
stor_devs =
00187940268-0006, 00187940268-0007, 00187940268-0008, 00187940268-0009
disks = d3, d4, d5, d6
disk=d3  stor_dev=00187940268-0006  addr=c0t110-48-0  server=server_2
disk=d3  stor_dev=00187940268-0006  addr=c16t110-33-0  server=server_2
disk=d4  stor_dev=00187940268-0007  addr=c0t111-48-0  server=server_2
disk=d4  stor_dev=00187940268-0007  addr=c16t111-33-0  server=server_2
disk=d5  stor_dev=00187940268-0008  addr=c0t112-48-0  server=server_2
disk=d5  stor_dev=00187940268-0008  addr=c16t112-33-0  server=server_2
disk=d6  stor_dev=00187940268-0009  addr=c0t113-48-0  server=server_2
disk=d6  stor_dev=00187940268-0009  addr=c16t113-33-0  server=server_2

id = 19
name = ufs1_snap1
acl = 0
in_use = False
type = uxfs
worm = off
volume = v456
pool =
rw_servers=
ro_servers=
rw_vdms =
ro_vdms =
backup_of = ufs1 Thu Oct 28 14:13:30 EDT 2004
stor_devs =
00187940268-0180, 00187940268-0181, 00187940268-0182, 00187940268-0183
disks = rootd378, rootd379, rootd380, rootd381

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name assigned to the file system.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control value for a file system. nas_acl provides information.</td>
</tr>
<tr>
<td>in_use</td>
<td>If a file system is registered into the mount table of a Data Mover.</td>
</tr>
<tr>
<td>type</td>
<td>Type of file system. -list provides a description of the types.</td>
</tr>
<tr>
<td>worm</td>
<td>Whether is enabled.</td>
</tr>
<tr>
<td>volume</td>
<td>Volume on which the file system resides.</td>
</tr>
<tr>
<td>pool</td>
<td>Storage pool for the file system.</td>
</tr>
<tr>
<td>rw_servers</td>
<td>Servers with read-write access to a file system.</td>
</tr>
<tr>
<td>ro_servers</td>
<td>Servers with read-only access to a file system.</td>
</tr>
<tr>
<td>rw_vdms</td>
<td>VDM servers with read-write access to a file system.</td>
</tr>
<tr>
<td>ro_vdms</td>
<td>VDM servers with read-only access to a file system.</td>
</tr>
<tr>
<td>backups</td>
<td>Name of associated backups.</td>
</tr>
</tbody>
</table>
EXAMPLE #2  To create a TimeFinder/FS copy of the PFS, ufs1, and leave a file system copy in mirrored mode, type:

```
$ fs_timefinder ufs1 -Snapshot -option mirror=on
```

operation in progress (not interruptible)...id        = 18
name      = ufs1
acl       = 0
in_use    = True
type      = uxfs
worm      = off
volume    = mtv1
pool      =
rw_servers= server_2
ro_servers=
rw_vdms   =
ro_vdms   =
backups   = ufs1_snap1
stor_devs =
000187940268-0006,000187940268-0007,000187940268-0008,000187940268-0009
disks     = d3,d4,d5,d6
disk=d3    stor_dev=000187940268-0006 addr=c0t110-48-0 server=server_2
disk=d3    stor_dev=000187940268-0006 addr=c16t110-33-0 server=server_2
disk=d4    stor_dev=000187940268-0007 addr=c0t111-48-0 server=server_2
disk=d4    stor_dev=000187940268-0007 addr=c16t111-33-0 server=server_2
disk=d5    stor_dev=000187940268-0008 addr=c0t112-48-0 server=server_2
disk=d5    stor_dev=000187940268-0008 addr=c16t112-33-0 server=server_2
disk=d6    stor_dev=000187940268-0009 addr=c0t113-48-0 server=server_2
disk=d6    stor_dev=000187940268-0009 addr=c16t113-33-0 server=server_2

id        = 19
name      = ufs1_snap1
acl       = 0
in_use    = False
type      = mirrorfs
worm      = off
volume    = v456
pool      =
rw_servers=
ro_servers=
rw_vdms   =
ro_vdms   =
backup_of = ufs1 Thu Oct 28 14:19:03 EDT 2004
remainder = 0 MB (0%)
stor_devs =
000187940268-0180,000187940268-0181,000187940268-0182,000187940268-0183
disks     = rootd378,rootd379,rootd380,rootd381

<table>
<thead>
<tr>
<th>backup_of</th>
<th>File system that the file system copy is made from.</th>
</tr>
</thead>
<tbody>
<tr>
<td>stor_devs</td>
<td>Storage system devices associated with a file system. The storage device output is the result of the Symmetrix hardware storage system.</td>
</tr>
<tr>
<td>disks</td>
<td>Disks on which the metavolume resides.</td>
</tr>
</tbody>
</table>
EXAMPLE #1 provides a description of command output.

EXAMPLE #3 To turn mirroring off for a file system copy, ufs1_snap1, type:

```
$ fs_timefinder ufs1_snap1 -Mirror off
```

operation in progress (not interruptible)...
remainder (MB) = 0
operation in progress (not interruptible)... id = 18
name = ufs1
acl = 0
in_use = True
type = uxfs
worm = off
volume = mtv1
pool =
rw_servers = server_2
ro_servers =
rw_vdms =
ro_vdms =
backups = ufs1_snap1
stor_devs =
000187940268-0006, 000187940268-0007, 000187940268-0008, 000187940268-0009
disks = d3, d4, d5, d6
  disk=d3  stor_dev=000187940268-0006  addr=c0t1l10-48-0  server=server_2
disk=d3  stor_dev=000187940268-0006  addr=c16t1l10-33-0  server=server_2
disk=d4  stor_dev=000187940268-0007  addr=c0t1l11-48-0  server=server_2
disk=d4  stor_dev=000187940268-0007  addr=c16t1l11-33-0  server=server_2
disk=d5  stor_dev=000187940268-0008  addr=c0t1l12-48-0  server=server_2
disk=d5  stor_dev=000187940268-0008  addr=c16t1l12-33-0  server=server_2
disk=d6  stor_dev=000187940268-0009  addr=c0t1l13-48-0  server=server_2
disk=d6  stor_dev=000187940268-0009  addr=c16t1l13-33-0  server=server_2
id = 19
name = ufs1_snap1
acl = 0
in_use = False
type = uxfs
worm = off
volume = v456
pool =
rw_servers =
ro_servers =
rw_vdms =
ro_vdms =
backup_of = ufs1 Thu Oct 28 14:21:50 EDT 2004
stor_devs =
000187940268-0180, 000187940268-0181, 000187940268-0182, 000187940268-0183
disks = rootd378, rootd379, rootd380, rootd381

EXAMPLE #1 provides a description of command output.
EXAMPLE #4  To turn mirroring on for a file system copy, ufs1_snap1, type:

    $ fs_timefinder ufs1_snap1 -Mirror on

operation in progress (not interruptible)...id        = 18
name      = ufs1
acl       = 0
in_use    = True
type      = uxfs
worm      = off
volume    = mtv1
pool      =
  rw_servers= server_2
  ro_servers= 
  rw_vdms   =
  ro_vdms   =
backups   = ufs1_snap1
stor_devs =
  000187940268-0006,000187940268-0007,000187940268-0008,000187940268-0009
disks     = d3,d4,d5,d6
  disk=d3    stor_dev=000187940268-0006    addr=c0t110-48-0    server=server_2
  disk=d3    stor_dev=000187940268-0006    addr=c16t110-33-0    server=server_2
  disk=d4    stor_dev=000187940268-0007    addr=c0t111-48-0    server=server_2
  disk=d4    stor_dev=000187940268-0007    addr=c16t111-33-0    server=server_2
  disk=d5    stor_dev=000187940268-0008    addr=c0t112-48-0    server=server_2
  disk=d5    stor_dev=000187940268-0008    addr=c16t112-33-0    server=server_2
  disk=d6    stor_dev=000187940268-0009    addr=c0t113-48-0    server=server_2
  disk=d6    stor_dev=000187940268-0009    addr=c16t113-33-0    server=server_2

id        = 19
name      = ufs1_snap1
acl       = 0
in_use    = False
type      = mirrorfs
worm      = off
volume    = v456
pool      =
  rw_servers=
  ro_servers= 
  rw_vdms   =
  ro_vdms   =
backup_of = ufs1 Thu Oct 28 14:21:50 EDT 2004
remainder = 0 MB (0%)
stor_devs =
  000187940268-0180,000187940268-0181,000187940268-0182,000187940268-0183
disks     = rootd378,rootd379,rootd380,rootd381

EXAMPLE #1 provides a description of command output.

EXAMPLE #5  To perform a mirror refresh on ufs1_snap1, type:

    $ fs_timefinder ufs1_snap1 -Mirror refresh

operation in progress (not interruptible)...

EXAMPLE #1 provides a description of command output.

EXAMPLE #6 To restore the file system copy, ufs1_snap1, to its original location, type:

```bash
$ /nas/sbin/rootfs_timefinder ufs1_snap1 -Restore -Force
```

remainder(MB) = 4991..4129..3281..2457..1653..815..0
operation in progress (not interruptible)...id = 18
name = ufs1
acl = 0
in_use = True
type = uxfs
worm = off
volume = mtv1
pool = r
rw_servers= server_2
ro_servers=
rw_vdms =
ro_vdms =
backups = ufs1_snap1
stor_devs =
00187940268-0006,00187940268-0007,00187940268-0008,00187940268-0009
disks = d3,d4,d5,d6
disk=d3  stor_dev=00187940268-0006  addr=0110-48-0  server=server_2
disk=d3  stor_dev=00187940268-0006  addr=0111-33-0  server=server_2
disk=d4  stor_dev=00187940268-0007  addr=00187940268-0007  server=server_2
disk=d4  stor_dev=00187940268-0007  addr=0111-33-0  server=server_2
disk=d5  stor_dev=00187940268-0008  addr=0112-48-0  server=server_2
disk=d5  stor_dev=00187940268-0008  addr=0112-33-0  server=server_2
disk=d6  stor_dev=00187940268-0009  addr=0113-48-0  server=server_2
disk=d6  stor_dev=00187940268-0009  addr=0113-33-0  server=server_2
id = 19
name = ufs1_snap1
acl = 0
in_use = False
type = uxfs
worm = off
volume = v456
pool = r
rw_servers=
ro_servers=
rw_vdms =
ro_vdms =
backup_of = ufs1 Thu Oct 28 14:25:21 EDT 2004
stor_devs =
00187940268-0180,00187940268-0181,00187940268-0182,00187940268-0183
disks = rootd378,rootd379,rootd380,rootd381

operation in progress (not interruptible)...id = 19
name = ufs1_snap1
acl  = 0
in_use = False
type = uxfs
worm  = off
volume = v456
pool  =
    rw_servers=
    ro_servers=
    rw_vdms  =
    ro_vdms  =
backup_of = ufs1 Thu Oct 28 14:25:21 EDT 2004
stor_devs =
    000187940268-0180,000187940268-0181,000187940268-0182,000187940268-0183
disks  = rootd378,rootd379,rootd380,rootd381
id     = 18
name   = ufs1
acl    = 0
in_use = True
type   = uxfs
worm   = off
volume = mtv1
pool   =
    rw_servers= server_2
    ro_servers=
    rw_vdms  =
    ro_vdms  =
backups  = ufs1_snap1
stor_devs =
    000187940268-0006,000187940268-0007,000187940268-0008,000187940268-0009
disks   = d3,d4,d5,d6
disk=d3 stor_dev=000187940268-0006 addr=c0t1l0-48-0 server=server_2
disk=d3 stor_dev=000187940268-0006 addr=c16t1l0-33-0 server=server_2
disk=d4 stor_dev=000187940268-0007 addr=c0t1l1-48-0 server=server_2
disk=d4 stor_dev=000187940268-0007 addr=c16t1l1-33-0 server=server_2
disk=d5 stor_dev=000187940268-0008 addr=c0t1l2-48-0 server=server_2
disk=d5 stor_dev=000187940268-0008 addr=c16t1l2-33-0 server=server_2
disk=d6 stor_dev=000187940268-0009 addr=c0t1l3-48-0 server=server_2
disk=d6 stor_dev=000187940268-0009 addr=c16t1l3-33-0 server=server_2

EXAMPLE #7 To create a snapshot for a mapped pool, type:

$ fs_timefinder ufs1 -name ufs1_snap1 -Snapshot -option
    pool=bcv_sg

operation in progress (not interruptible)...remainder(MB) = ..14184..0
operation in progress (not interruptible)...id     = 87
name   = ufs1
acl    = 0
in_use = False
The fs Commands

```plaintext
type = uufs
worm = off
volume = mtv1
pool =
rw_servers=
ro_servers=
rw_vdms =
ro_vdms =
backups = ufs1_snap1
auto_ext = no, thin=no
deduplication = unavailable
stor_devs = 000194900546-0037
disks = d11

id = 88
name = ufs1_snap1
acl = 0
in_use = False
type = uufs
worm = off
volume = v456
pool = bcv_sg
member_of = root_avm_fs_group_49
rw_servers=
ro_servers=
rw_vdms =
ro_vdms =
backup_of = ufs1 Fri Oct 1 12:03:10 EDT 2010
auto_ext = no, thin=no
deduplication = unavailable
thin_storage = False
tiering_policy = thickfp2
mirrored = False
stor_devs = 000194900546-003C
disks = rootd16
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_ext</td>
<td>Indicates whether auto-extension and thin provisioning are enabled.</td>
</tr>
</tbody>
</table>
| deduplication    | Deduplication state of the file system. The file data is transferred to the storage which performs the deduplication and compression on the data. The states are:  
                    • On — Deduplication on the file system is enabled.  
                    • Suspended — Deduplication on the file system is suspended. Deduplication does not perform any new space reduction but the existing files that were reduced in space remain the same.  
                    • Off — Deduplication on the file system is disabled. Deduplication does not perform any new space reduction and the data is now reduplicated. |
<table>
<thead>
<tr>
<th>thin_storage</th>
<th>Indicates whether the block storage system uses thin provisioning. Values are: True, False, Mixed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tiering_policy</td>
<td>Indicates the tiering policy in effect. If the initial tier and the tiering policy are the same, the values are: Auto-Tier, Highest Available Tier, Lowest Available Tier. If the initial tier and the tiering policy are not the same, the values are: Auto-Tier/No Data Movement, Highest Available Tier/No Data Movement, Lowest Available Tier/No Data Movement.</td>
</tr>
<tr>
<td>mirrored</td>
<td>Indicates whether the disk is mirrored.</td>
</tr>
</tbody>
</table>
This chapter lists the VNX Command Set provided for managing, configuring, and monitoring of NAS database. The commands are prefixed with `nas` and appear alphabetically. The command line syntax (Synopsis), a description of the options, and an example of usage are provided for each command.
The nas Commands

**nas_acl**

Manages the access control level table.

**SYNOPSIS**

```
  nas_acl
    -list
  | -info {-user|-group|-owner} <numerical_id>
  | -delete {-user|-group} <numerical_id>
  | [-name <name>] -create {-user|-group} <numerical_id> level=<acl_level>
  | -modify {-user|-group} <numerical_id>
        {{num_id=<numerical_id>}[,level=<acl_level>]
```

**DESCRIPTION**

`nas_acl` creates, lists, and displays information for access control level entries within the table, and deletes the specified group or entries. The access control level table is created and recognized in the NAS database and contains assigned levels for users and groups. A user must be defined in the `/etc/passwd` file prior to being assigned an entry in the table. Creating an access control level entry defines the access level allowed for the user or group once a value has been established for an object.

Note: root privileges are required to create, modify, or delete the access control level table. The root user is permitted access to all objects.

**OPTIONS**

- **-list**
  Lists the access control level table.

- **-info {-user|-group|-owner} <numerical_id>**
  Displays information for the user, group, or index entry of the owner as specified by the `<numerical_id>`.

- **-delete {-user|-group} <numerical_id>**
  Deletes the entry for the specified user or group from the access control level table.

- **-create {-user|-group} <numerical_id> level=<acl_level>**
  Creates an access control level entry for the specified user or group. The `<numerical_id>` can be a user ID (UID) or group ID (GID).

  **Note:** Before executing this command, the user or group must exist in the Control Station in the `/etc/passwd` file or the `/etc/group` file.
The <acl_level> is a single-digit (between numbers 2 and 9) input representing available access control levels. Levels 2, 3, and 4 which are established by default are:

2 — admin — Is the most privileged level and includes privileges allowed from the operator and observer levels.
3 — operator — Includes privileges from the observer level.
4 — observer — The least privileged.

Levels 5—9 are available for configuration.

[-name <name>]
The name is case-sensitive and indicates a name by which the entry is referred.

Once a value has been set, the level assigned the user or group is checked in the ACL table and the level of access to the object is determined.

-modify {-user|-group} <numerical_id> {{num_id=<numerical_id>}} [level=<acl_level>] } Modifies the <numerical_id> and level for an access control level entry.

SEE ALSO Controlling Access to System Objects on VNX, nas_fs, nas_volume, nas_rp, and nas_storage.

EXAMPLE #1 Before creating access control level entries, su to root. To create entries in the access control level table, type:

```sh
# nas_acl -name user1 -create -user 211 level=3 done
# nas_acl -name user2 -create -user 212 level=2 done
# nas_acl -name user3 -create -user 213 level=4 done
# nas_acl -name user4 -create -user 214 level=2 done
# nas_acl -name user5 -create -user 215 level=3 done
# nas_acl -name user6 -create -user 216 level=4 done
```
EXAMPLE #2  To display the access control level table, type:

```
$ nas_acl -list
```

<table>
<thead>
<tr>
<th>index</th>
<th>type</th>
<th>level</th>
<th>num_id</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>user</td>
<td>admin</td>
<td>201</td>
<td>nasadmin</td>
</tr>
<tr>
<td>2</td>
<td>user</td>
<td>operator</td>
<td>211</td>
<td>user1</td>
</tr>
<tr>
<td>3</td>
<td>user</td>
<td>admin</td>
<td>212</td>
<td>user2</td>
</tr>
<tr>
<td>4</td>
<td>user</td>
<td>observer</td>
<td>213</td>
<td>user3</td>
</tr>
<tr>
<td>5</td>
<td>user</td>
<td>admin</td>
<td>214</td>
<td>user4</td>
</tr>
<tr>
<td>6</td>
<td>user</td>
<td>operator</td>
<td>215</td>
<td>user5</td>
</tr>
<tr>
<td>7</td>
<td>user</td>
<td>observer</td>
<td>216</td>
<td>user6</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>Access control level table index entry number.</td>
</tr>
<tr>
<td>type</td>
<td>User or group for the entry.</td>
</tr>
<tr>
<td>level</td>
<td>Level of access permitted.</td>
</tr>
<tr>
<td>num_id</td>
<td>Numerical ID for identifying the entry.</td>
</tr>
<tr>
<td>name</td>
<td>Name given to the entry.</td>
</tr>
</tbody>
</table>

EXAMPLE #3  To display information for an access control level entry, type:

```
$ nas_acl -info -user 211
```

```
id        = 2
name      = user1
level     = operator
user_id   = 211
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Index entry.</td>
</tr>
<tr>
<td>name</td>
<td>Name given for the entry.</td>
</tr>
<tr>
<td>level</td>
<td>Level of access permitted.</td>
</tr>
<tr>
<td>user_id</td>
<td>Also known as the num_id.</td>
</tr>
</tbody>
</table>

EXAMPLE #4  To modify an access control level entry, type:

```
# nas_acl -modify -user 211 level=7
done
```

EXAMPLE #5  To delete an access control level entry, type:

```
# nas_acl -delete -user 211
done
```
nas_automountmap

Manages the automount map file.

SYNOPSIS

nas_automountmap
   -list_conflict <infile> [-out <outfile>]
   | -create [-in <infile>] [-out <outfile>]

DESCRIPTION

nas_automountmap creates and displays an automount map containing all permanently exported file systems used by the automount daemon.

OPTIONS

- list_conflict <infile>
   Prints a list of the mount points that are used more than once.
   [-out <outfile>]
   Prints a conflicting list and saves it to an <outfile>.

- create
   Creates an automount map and prints it to the screen only.
   [-in <infile>] [-out <outfile>]
   Merges an automount map with an existing map <infile> and outputs it to an <outfile>.
   [-out <outfile>]
   Creates an automount map and outputs it to an <outfile>.

EXAMPLE #1

To create an automount map, type:

$ nas_automountmap -create
ufs1 -rw,intr,nosuid 127.0.0.1,10.172.128.47,128.221.253.2,128.221.252.2:/ufs1
ufs2 -rw,intr,nosuid 127.0.0.1,10.172.128.47,128.221.253.2,128.221.252.2:/ufs2

EXAMPLE #2

To create an automount map and save it to a file, type:

$ nas_automountmap -create -out automountmap
$ more automountmap
ufs1 -rw,intr,nosuid 127.0.0.1,10.172.128.47,128.221.253.2,128.221.252.2:/ufs1
ufs2 -rw,intr,nosuid 127.0.0.1,10.172.128.47,128.221.253.2,128.221.252.2:/ufs2

EXAMPLE #3

To print a conflicting list, type:

$ nas_automountmap -list_conflict automountmap
Conflicting lists:
ufs1   -rw,intr,suid 172.16.21.202:/ufs1
ufs1_172.16.21.203   -rw,intr,suid 172.16.21.203:/ufs1
EXAMPLE #4  To merge an automount map file with an existing map file, type:

$ nas_automountmap -create -in automountmap -out automountmap1
nas_ca_certificate

Manages the Control Station as a Certificate Authority (CA) for VNX’s Public Key Infrastructure (PKI).

SYNOPSIS

```
nas_ca_certificate
  -display
  -generate
```

DESCRIPTION

**nas_ca_certificate** generates a public/private key set and a CA certificate for the Control Station. When the Control Station is serving as a CA, it must have a private key with which to sign the certificates it generates for the Data Mover. The Control Station CA certificate contains the corresponding public key, which is used by clients to verify the signature on a certificate received from the Data Mover.

**nas_ca_certificate** also displays the text of the CA certificate so you can copy it and distribute it to network clients. In order for a network client to validate a certificate sent by a Data Mover that has been signed by the Control Station, the client needs the Control Station CA certificate (specifically the public key from the CA certificate) to verify the signature of the Data Mover’s certificate.

The initial Control Station public/private key set and CA certificate are generated automatically during a VNX software 5.6 install or upgrade. A new Control Station public/private key set and CA certificate is not required unless the CA key set is compromised or the CA certificate expires. The Control Station CA certificate is valid for 5 years.

You must be **root** to execute the **-generate** option from the /nas/sbin directory.

Once a Control Station CA certificate is generated, you must perform several additional tasks to ensure that the new certificate is integrated into VNX’s PKI framework. The *Security Configuration Guide for File* and the Unisphere online help for the PKI interface explain these tasks.

OPTIONS

**-display**

Displays the Control Station CA certificate. The certificate text is displayed on the terminal screen. Alternatively, you can redirect it to a file.
-generate
Generates a new CA public/private key set and certificate for the Control Station. This certificate is valid for 5 years from the date it is generated.

SEE ALSO
server_certificate.

EXAMPLE #1
To generate a new Control Station CA certificate, type:

```
# /nas/sbin/nas_ca_certificate -generate
```

New keys and certificate were successfully generated.

EXAMPLE #2
To display the Control Station’s CA certificate, type:

```
# /nas/sbin/nas_ca_certificate -display
```

Note: Clients need only the certificate text enclosed by BEGIN CERTIFICATE and END CERTIFICATE although most clients can handle the entire output.

Certificate:
Data:
Version: 3 (0x2)
Serial Number: 3 (0x3)
Signature Algorithm: sha1WithRSAEncryption
Issuer: O=Celerra Certificate Authority, CN=eng173100
Validity
 Not Before: Mar 23 21:07:40 2007 GMT
 Not After : Mar 21 21:07:40 2012 GMT
Subject: O=Celerra Certificate Authority, CN=eng173100
Subject Public Key Info:
 Public Key Algorithm: rsaEncryption
 RSA Public Key: (2048 bit)
 Modulus (2048 bit):
 b7:e6:f1:7f:21:f0:71:2d:c4:8a:8f:20:d1:ab:5a:
 22:29:00:11:e0:a1:12:4b:02:79:fb:0f:fc:54:90:
 00:15:04:77:47:03:ec:c5:7a:a2:bf:32:0e:4d:d8:
 35:b1:e9:a8:a8:2a:72:8a:b8:c4:bf:04:70:12:
b3:31
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Subject Key Identifier:
X509v3 Authority Key Identifier:
  DirName:/O=Celerra Certificate Authority/CN=eng173100
  serial:00
X509v3 Basic Constraints:
  CA:TRUE
X509v3 Subject Alternative Name:
  DNS:eng173100
Signature Algorithm: sha1WithRSAEncryption
  96:bf:fe:6f
-----BEGIN CERTIFICATE-----
MIIDoDCCAoigAwIBAgIBAzANBgkqhkiG9w0BAQUFADA8MSYwJAYDVQQKEExlDZWxl
ecnJhIENlcnRpZmljYXRlIEF1dGhvcml0eTÜmBAAGA1UEAxMJZW5nMTczMTAwMB4X
DTA3MDMyMzIxMDM0FOzDEyMTEyMDMyMTIzMjQ2VzZXJy
YSBDZXJ0aWZpY2F0ZS0bQ2VydGlmaWNhdGUgQ2VydGlmaWNhdGUgQXV0aG9yaXR5
MRIwEAYDVQQHEI9Db21wb250LnN0YXJ0aWZpY2F0ZS5hbmdlbC8=
-----END CERTIFICATE-----
**nas_cel**

Performs management of remotely linked VNX or a linked pair of Data Movers.

**SYNOPSIS**

```
nas_cel
- list
  - delete {<cel_name>|id=<cel_id>} [-Force]
  - info {<cel_name>|id=<cel_id>}
  - update {<cel_name>|id=<cel_id>}
  - modify {<cel_name>|id=<cel_id>}
    {[-passphrase <passphrase>] [-name <new_name>] [-ip <ipaddr>]}
  - create <cel_name> -ip <ipaddr> -passphrase <passphrase>
  - interconnect <interconnect_options>
```

**DESCRIPTION**

`nas_cel` manages the linking of the remote VNX to the local VNX. `nas_cel` also creates the trusted relationship between source and destination VNX Control Stations in configurations such as VNX Replicator.

For VNX Replicator only, `nas_cel -interconnect` also builds the connection (interconnect) between a pair of Data Movers.

Linked VNX systems are acknowledged:

- Automatically during the installation
- When executing the `nas_cel -create`
- When performing a `nas_rdf -init` to set up the SRDF relationship between two VNX.

**OPTIONS**

- **-list**
  Lists all VNX linked to the current VNX. The hostname of the Control Station active during installation appears as the `<cel_name>`.

  **Note:** The ID of the object is an integer and is assigned automatically. The name of the VNX might be truncated if it is too long for the display. To view the full name, use the `-info` option with the VNX ID.

- **-delete <cel_name>|id=<cel_id> [-Force]**
  Deletes the relationship of the remote VNX, and removes its entry from the NAS database on the local VNX.

  The `-Force` option applies to SRDF and EMC MirrorView™/S configurations only. If the VNX to be deleted is part of an SRDF or
The nas Commands

MirrorView/S configuration, **-delete** must be specified with the **-Force** option; otherwise, an error is generated. You cannot use **-Force** if the specified VNX is also being used by VNX Replicator, file system copy (for example, with **nas_copy**), or TimeFinder/FS NearCopy or FarCopy. If the deletion is necessary, clean up these configurations before performing the forced deletion.

**-info** {<cel_name>|id=<cel_id>}
Displays information for the remote VNX. To view the <cel_id> of configured VNX, use **-list**.

**-update** {<cel_name>|id=<cel_id>}
Updates the local VNX entry with the local Control Station’s hostname and IP address configuration. It also updates the local Data Mover-to-Data Mover authentication setup.

For the remote VNX, updates all Data Movers that were down or experiencing errors during the **-create** or **-modify** and restores them to service by using the configuration required for Data Mover authentication.

**Note:** Data Mover authentication is used in iSCSI replication as the mechanism enabling two Data Movers (local or remote) to authenticate themselves and perform the requested operations. The **-update** option communicates with each Data Mover and either updates the configuration, or creates the configuration if it is being done for the first time.

**-modify** {<cel_name>|id=<cel_id>}
{[-passphrase <passphrase>][-name <new_name>][-ip <ipaddr>]
Changes the current passphrase, name, or IP address of the remote VNX to the new passphrase, name, or IP address in the local VNX database and modifies the remote Data Mover authentication setup by communicating with each Data Mover in the cabinet. The passphrase must have 6 to 15 characters.

**-create** <cel_name> -ip <ipaddr> -passphrase <passphrase>
Builds the trusted relationship between one VNX and another VNX in a configuration such as VNX Replicator, SRDF, and MirrorView/S.

The **-create** must be executed twice to ensure communication from both sides, first on the source VNX (to identify the destination VNX) and then on the destination VNX (to identify the source VNX). You must assign a name when you create the relationship (for example, a name that identifies the remote VNX in a local entry). The IP address
The nas Commands

specified represents the appropriate remote VNX’s primary Control Station (in slot 0); the passphrase specified is used to manage the remote VNX. The passphrase must have 6 to 15 characters and be the same between the source and destination VNXs to enable communication.

INTERCONNECT OPTIONS

Type `nas_cel -interconnect` to display interconnect options:

```
-interconnect
  { -create <name>
      -source_server <movername>
      -destination_system {<cel_name>|id=<cel_id>}
      -destination_server <movername>
      -source_interfaces {<name_service_interface_name>|ip=<ipaddr>}
      [...] {<name_service_interface_name>|ip=<ipaddr>},...
      -destination_interfaces {<name_service_interface_name>|ip=<ipaddr>}
      [...] {<name_service_interface_name>|ip=<ipaddr>},...
      [-bandwidth <bandwidthSched>]
    } -modify {<name>|id=<interConnectId>}
    { [-source_interfaces {<name_service_interface_name>|ip=<ipaddr>},...
        [-destination_interfaces
        {<name_service_interface_name>|ip=<ipaddr>},...
        [-bandwidth <bandwidthSched>]
        [-name <newName>]
      ]
    } -pause {<name>|id=<interConnectId>}
    -resume {<name>|id=<interConnectId>}
    -delete {<name>|id=<interConnectId>}
    -info {<name>|id=<interConnectId>|-all}
    -list [-destination_system {<cel_name>|id=<cel_id>}]   
    -validate {<name>|id=<interconnectId>})
```

An interconnect supports VNX Replicator sessions by defining the communications path between a given Data Mover pair located on the same cabinet or different cabinets. The interconnect configures a list of local (source) and peer (destination) interfaces (using IP addresses and interface names), and a bandwidth schedule for all replication sessions using the interconnect. Only one interconnect can be established for a given Data Mover pair in any direction.

Note: You must delete all user-defined interconnects configured for a Data Mover before you can rename it. After you rename the Data Mover, you must re-create the source and peer interconnects with the new Data Mover name and then restart any associated replication sessions.

To fully establish an interconnect, `nas_cel -interconnect` must be issued twice, once from each side (the local side and its peer side). Both sides of the interconnect must exist before VNX Replicator
sessions for local or remote replication can use the interconnect. Only the local side of an interconnect on which the source replication object resides is specified when creating the replication session. Loopback interconnects are created and named automatically and can be viewed using `nas_cel -interconnect -list`. You cannot create, modify, or delete loopback interconnects.

**-create <name>**
Assigns a name, up to 255 characters, to the appropriate side of the interconnect. The name must be unique for each Data Mover. Make the name meaningful, identifying servers and, for remote replication, VNX names or sites.

Remote replication naming example:
```
s2CelA_s3CelB or NYs3_LAs4 (local side)
s3CelB_s2CelA or LAs4_NYs3 (peer side)
```

Local replication naming example:
```
s2_s3 (source side on local system)
s3_s2 (peer side on the same system)
```

**-source_server <moverName>**
Specifies the name of an available local Data Mover to use for the local side of the interconnect.

**-destination_system {<cel_name>|id=<cel_id>}**
Specifies the name or ID of the VNX where the peer Data Mover resides.

**-destination_server <movername>**
Specifies the name of an available Data Mover, on the same or different system, to use for the peer side of the interconnect.

**-source_interfaces {<name_service_interface_name>|ip=<ipaddr>} [,{<name_service_interface_interface_name>|ip=<ipaddr>},...]**
Configures a list of interfaces available for the local side of the interconnect. You can define the list by using IP addresses (IPv4 or IPv6) or name service interface names or a combination of both, but how you specify an interface determines how it must be specified by the replication session later (by name service interface name or IP address).
If you define an interface by using an IP address, make sure that the source interface list uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

For each network protocol type (IPv4/IPv6) specified in the source interface list, at least one interface from the same type must be specified in the destination interfaces list and vice versa. For example, if the source interface list includes one or more IPv6 addresses, the destination interface list must also include at least one IPv6 address.

The name service interface name is a fully qualified name given to a network interface that must resolve to a single IP address (for example, using a DNS server).

**Note:** To prevent potential errors during interface selection (especially after a failover/switchover), it is highly recommended that you specify the same local and peer interface lists when configuring each side of the interconnect.

```
-destination_interfaces
{name_service_interface_name}| ip=<ipaddr>
[,{name_service_interface_name}| ip=<ipaddr>],...
```

Configures a list of interfaces available on the peer side of the interconnect. You can define the list by using IP addresses (IPv4 or IPv6) or name service interface names or a combination of both, but how you specify each interface determines how it is specified by the replication session.

If you define an interface using an IP address, make sure that the source interface list uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

For each network protocol type (IPv4/IPv6) specified in the destination interface list, at least one interface from the same type must be specified in the source interfaces list and vice versa. For example, if the source interface list includes one or more IPv6 addresses, the destination interface list must also include at least one IPv6 address. The name service interface name is a fully qualified name given to a network interface that must resolve to a single IP address (for example, using a DNS server).
The nas Commands

[-bandwidth <bandwidthSched>]
Specifies a schedule to control the interconnect bandwidth used on specific days, or times instead of using all available bandwidth at all times for the interconnect (the default).

Note: The bandwidth schedule executes based on Data Mover time, not Control Station time.

The schedule applies to all VNX Replicator sessions using the interconnect. Specify a schedule with one or more comma-separated entries, most specific to least specific, as follows:

[{Su|Mo|Tu|We|Th|Fr|Sa}][HH:00-HH:00]/[Kbps], [ <next_entry>],[...]

Example:
MoTuWeThFr07:00-18:00/2000,/8000 means use a limit of 2000 Kb/s from 7 A.M. to 6 P.M. Monday through Friday; otherwise, use a bandwidth limit of 8000 Kb/s.

-interconnect
-modify {<name>|id=<interConnectId>}
Modifies one or more of the following characteristics of an existing interconnect, as specified by the name or ID for the appropriate side of the interconnect.

Note: You cannot modify the peer side of an interconnect configured on a remote system; you must modify it from that system. Also, you cannot modify an interface in use by a replication session.

[-source_interfaces {<name_service_interface_name>| ip= <ipAddr>}, ... ]
Modifies the list of interfaces (name service interface name or IP addresses or both) available for use on the local side of an interconnect. The new list of interfaces completely replaces the previous list.

Note: To avoid problems with interface selection, any changes made to the interface lists should be reflected on both sides of an interconnect.

[-destination_interfaces {<name_service_interface_name>| ip= <ipAddr>}, ... ]
Modifies the list of interfaces (name service interface name or IP
addresses or both) available for use on the peer side of an interconnect. The new list of interfaces completely replaces the previous list.

[-bandwidth <bandwidth>]
Modifies the existing bandwidth schedule for the specified interconnect, or creates a schedule if none existed previously. The schedule allocates the interconnect bandwidth for specific days or times or both instead of using all available bandwidth at all times for the interconnect (the default). The schedule applies to all replication sessions using the interconnect. Specify a schedule with one or more comma-separated entries, most specific to least specific, as follows:

[[Su|Mo|Tu|We|Th|Fr|Sa]}[HH:00-HH:00] [/Kbps]
, [ <nextEntry>], [...]

Example:
MoTuWeThFr07:00-18:00/2000,/8000 means use a limit of 2000 Kb/s from 7 A.M. to 6 P.M. Monday through Friday; otherwise, use a bandwidth limit of 8000 Kb/s.

[-name <newName>]
Changes the name of the specified interconnect to a new name.

-interconnect -pause {<name> | id=<interConnectId>}
Halts data transmission over the existing Data Mover interconnect until you resume transmission over the interconnect or delete the interconnect. This affects all replication sessions using the specified interconnect.

-interconnect -resume {<name> | id=<interConnectId>}
Resumes data transmission over the Data Mover interconnect, making the interconnect available for use by replication sessions.

-interconnect -delete {<name> | id=<interConnectId>}
Deletes the Data Mover interconnect, thereby making the interconnect unavailable for use by any replication sessions. You cannot delete an interconnect if it is in use by a replication session. You can delete a paused interconnect.

-interconnect -info {<name> | id=<interConnectId> | -all}
Displays information about the specified interconnect or about all interconnects known to the local system.

-interconnect -list [-destination_system <cel_name> | id=<cel_id>]}
By default, lists the interconnects available on the local VNX. Specifying the name or ID of a remote VNX also lists the interconnects available on that VNX.

```
-interconnect -validate {<name>|id=<interconnectId>}
```

Verifies the interconnect, verifying that authentication is configured properly by opening the connection between the Data Mover pair. Validation is done for loopback, local, and remote configuration.

**SEE ALSO**  Using VNX Replicator, nas_copy, nas_replicate, and nas_task.

**EXAMPLE #1**

To create an entry for the remote VNX, type:

```
$ nas_cel -create cs110 -ip 172.24.102.240 -passphrase nasdocs
```

```
operation in progress (not interruptible)...
```

```
id         = 3
name       = cs110
owner      = 0
device     =
channel    =
net_path   = 172.24.102.240
VNX_id = APM000438070430000
passphrase = nasdocs
```

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the remote VNX on the local VNX.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned in the local view to the remote VNX.</td>
</tr>
<tr>
<td>owner</td>
<td>ACL ID assigned automatically.</td>
</tr>
<tr>
<td>device</td>
<td>R2 device mounted by the local Control Station to read the database of the remote Control Station in the SRDF environment. This value is unique to the Symmetrix storage system.</td>
</tr>
<tr>
<td>channel</td>
<td>Pair of devices used in the rdf channel. One is used for writing messages to the remote (wdev), the other to read messages from them. This value is unique to the Symmetrix storage system.</td>
</tr>
<tr>
<td>net_path</td>
<td>IP address of the remote VNX.</td>
</tr>
<tr>
<td>VNX_id</td>
<td>Unique VNX ID number.</td>
</tr>
<tr>
<td>passphrase</td>
<td>Used for authentication with a remote VNX.</td>
</tr>
</tbody>
</table>

**EXAMPLE #2**

For the VNX for block, to list all remote VNXs, type:

```
$ nas_cel -list
```

```
id    name          owner mount_dev  channel    net_path        CMU
0     cs100         0                           172.24.102.236  APM000420008180000
3     cs110         0                           172.24.102.240  APM000438070430000
```

For the VNX with a Symmetrix storage system, to list all remote VNXs, type:
### The nas Commands

#### $ nas_cel -list

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>owner</th>
<th>mount_dev</th>
<th>channel</th>
<th>net_path</th>
<th>CMU</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>cs30</td>
<td>0</td>
<td></td>
<td></td>
<td>172.24.172.152</td>
<td>0028040001900006</td>
</tr>
<tr>
<td>1</td>
<td>cs40</td>
<td>500</td>
<td>/dev/sdj1</td>
<td>/dev/sdg</td>
<td>172.24.172.151</td>
<td>0028040002180000</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the remote VNX on the local VNX.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned in the local view to the remote VNX.</td>
</tr>
<tr>
<td>owner</td>
<td>ACL ID assigned automatically.</td>
</tr>
<tr>
<td>mount_dev</td>
<td>Mounted database from the remote VNX in the SRDF environment. This value is unique to the Symmetrix storage system.</td>
</tr>
<tr>
<td>channel</td>
<td>RDF channel from where information is read and written. This value is unique to the Symmetrix storage system.</td>
</tr>
<tr>
<td>net_path</td>
<td>IP address of the remote VNX.</td>
</tr>
<tr>
<td>CMU</td>
<td>VNX Management Unit (unique VNX ID number).</td>
</tr>
</tbody>
</table>

#### EXAMPLE #3

To display information for the remote VNX, cs110, type:

```
$ nas_cel -info cs110
```

```
id = 3
name = cs110
owner = 0
device =
channel =
net_path = 172.24.102.240
VNX_id = APM000438070430000
passphrase = nasdocs
```

**EXAMPLE #1** provides information for a description of command outputs.

#### EXAMPLE #4

To update the Control Station entry for cs110, type:

```
$ nas_cel -update cs110
```

```
operation in progress (not interruptible)...
id = 3
name = cs110
owner = 0
device =
channel =
net_path = 172.24.102.240
VNX_id = APM000438070430000
passphrase = nasdocs
```

**EXAMPLE #1** provides information for a description of command outputs.
EXAMPLE #5 To modify the passphrase and name for the remote Control Station cs110, type:

```
$ nas_cel -modify cs110 -passphrase nasdocs_replication
     -name cs110_target
```

operation in progress (not interruptible)...

id         = 3
name       = cs110_target
owner      = 0
device     =
channel    =
net_path   = 172.24.102.240
VNX_id      = APM000438070430000
passphrase = nasdocs_replication

EXAMPLE #1 provides information for a description of command outputs.

EXAMPLE #6 To delete the Control Station entry of the remote VNX, cs110_target, type:

```
$ nas_cel -delete cs110_target
```

operation in progress (not interruptible)...

id         = 3
name       = cs110_target
owner      = 0
device     =
channel    =
net_path   = 172.24.102.240
VNX_id      = APM000438070430000
passphrase = nasdocs_replication

EXAMPLE #1 provides information for a description of command outputs.

EXAMPLE #7 To create an interconnect NYs3_LAs2 between Data Mover server_3 and remote Data Mover server_2, and use a bandwidth limit of 2000 Kb/s from 7 A.M. to 6 P.M. Monday through Friday; otherwise, use a bandwidth limit of 8000 Kb/s, type:

```
$ nas_cel -interconnect -create NYs3_LAs2 -source_server server_3
     -destination_system cs110 -destination_server server_2
     -source_interfaces ip=10.6.3.190 -destination_interfaces ip=10.6.3.173
     -bandwidth MoTuWeThFr07:00-18:00/2000,/8000
```

operation in progress (not interruptible)...

id                                 = 30003
name                               = NYs3_LAs2
source_server                      = server_3
The nas Commands

source_interfaces                  = 10.6.3.190
destination_system                 = cs110
destination_server                 = server_2
destination_interfaces             = 10.6.3.173
bandwidth schedule                 = MoTuWeThFr07:00-18:00/2000,/8000
crc enabled                        = yes
number of configured replications  = 0
number of replications in transfer = 0
status                             = The interconnect is OK.

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the interconnect.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the interconnect.</td>
</tr>
<tr>
<td>source_server</td>
<td>Name of an available local Data Mover to use for the local side of the interconnect.</td>
</tr>
<tr>
<td>source_interfaces</td>
<td>IP addresses available for the local side of the interconnect (at least one, or a name service interface name).</td>
</tr>
<tr>
<td>destination_system</td>
<td>Control Station names of the VNX systems available for use in a remote replication session. Local System is the default.</td>
</tr>
<tr>
<td>destination_server</td>
<td>Name of an available peer Data Mover to use for the peer side of the interconnect.</td>
</tr>
<tr>
<td>destination_interfaces</td>
<td>IP addresses available for the peer side of the interconnect (at least one, or a name service interface name). For loopback interconnects, the interface is fixed at 127.0.0.1.</td>
</tr>
<tr>
<td>bandwidth schedule</td>
<td>Bandwidth schedule with one or more comma-separated entries, most specific to least specific.</td>
</tr>
<tr>
<td>crc enabled</td>
<td>Indicates that the Cyclic Redundancy Check (CRC) method is in use for verifying the integrity of data sent over the interconnect. CRC is automatically enabled and cannot be disabled.</td>
</tr>
<tr>
<td>number of configured replications</td>
<td>Number of replication sessions currently configured.</td>
</tr>
<tr>
<td>number of replications in transfer</td>
<td>Number of replications are currently in transfer.</td>
</tr>
<tr>
<td>status</td>
<td>Status of the interconnect.</td>
</tr>
</tbody>
</table>

EXAMPLE #8  To modify the bandwidth schedule of the interconnect NYs3_LAs2, type:

```bash
$ nas_cel -interconnect -modify NYs3_LAs2 -bandwidth MoWeFr07:00-18:00/2000,TuTh07:00-18:00/4000,/8000
```

operation in progress (not interruptible)...

id                   = 30003
name                 = NYs3_LAs2
source_server        = server_3
source_interfaces    = 10.6.3.190
destination_system   = cs110
destination_server   = server_2
destination_interfaces = 10.6.3.173
EXAMPLE #7 provides a description of the command outputs.

EXAMPLE #9
To list available interconnects, type:

```
$ nas_cel -interconnect -list
```

id     name               source_server   destination_system   destination_server
20001  loopback           server_2        cs100                server_2
30001  loopback           server_3        cs100                server_3
30003  NYs3_LAs2          server_3        cs110                server_2

EXAMPLE #10
To pause the interconnect with id=30003, type:

```
$ nas_cel -interconnect -pause id=30003
done
```

EXAMPLE #11
To resume the interconnect NYs3_LAs2, type:

```
$ nas_cel -interconnect -resume NYs3_LAs2
done
```

EXAMPLE #12
To validate the interconnect NYs3_LAs2, type:

```
$ nas_cel -interconnect -validate NYs3_LAs2
NYs3_LAs2: validating 9 interface pairs: please wait...ok
```

EXAMPLE #13
To display the detailed information about the interconnect NYs3_LAs2, type:

```
$ nas_cel -interconnect -info NYs3_LAs2
```

id                                 = 30003
name                               = NYs3_LAs2
source_server                      = server_3
source_interfaces                  = 10.6.3.190
destination_system                 = cs110
destination_server                 = server_2
destination_interfaces             = 10.6.3.173
bandwidth schedule                 = MoWeFr07:00-18:00/2000,TuTh07:00-18:00/4000,/8000
crc enabled                        = yes
number of configured replications  = 0
number of replications in transfer = 0
status                             = The interconnect is OK.

EXAMPLE #7 provides a description of the command outputs.
EXAMPLE #14  To delete interconnect NYs3_LAs2, type:

```
$ nas_cel -interconnect -delete NYs3_LAs2
```

operation in progress (not interruptible)...

id = 30003
name = NYs3_LAs2
source_server = server_3
source_interfaces = 10.6.3.190
destination_system = cs110
destination_server = server_2
destination_interfaces = 10.6.3.173
bandwidth schedule = MoWeFr07:00-18:00/2000,TuTh07:00-18:00/4000,/8000
crc enabled = no
number of configured replications = 0
number of replications in transfer = 0
status = The interconnect is OK.

EXAMPLE #7 provides a description of the command outputs.
**nas_checkup**

Provides a system health checkup for the VNX.

**SYNOPSIS**

```bash
nas_checkup [-version | -help | -rerun]
```

**DESCRIPTION**

`nas_checkup` runs scheduled and unscheduled health checks on the VNX and reports problems that are found, and the actions needed to fix the problem and acts as a system health monitor.

The scheduled run time for the `nas_checkup` command is every 2 weeks by default. If a warning or error is discovered during this time, an alert is posted on the Unisphere.

Set up email notification for warnings or errors in the Unisphere Notifications page, or modify and load the sample `nas_checkup` event configuration file.

If a problem is discovered that requires EMC Service Personnel assistance, `nas_checkup` will notify EMC.

**OPTIONS**

- **No arguments**
  Runs a series of system health checks on the VNX and reports the problems that are found and the actions needed to fix the problem.

---

**Note:** No email, callhome, or Unisphere alert is posted when the health check is run unscheduled.

---

- **-version**
  Displays the version of health check that is run on the VNX.

- **-help**
  Provides help.

- **-rerun**
  Reruns the checks that produce error messages in the previous health checkup. It does not rerun the checks that produce warning or information messages. If there are no checks that produce error messages, then the -rerun switch generates a message that there is nothing to rerun.
The nas Commands

CHECKS

Nas_checkup runs a subset of the available checks based on the configuration of your system. The complete list of available checks are:

Control Station Checks:
Check if minimum free space exists
Check if minimum free space exists ns
Check if enough free space exists
Check if enough free space exists ns
Check if NAS Storage API is installed correctly
Check if NAS Storage APIs match
Check if NBS clients are started
Check if NBS configuration exists
Check if NBS devices are accessible
Check if NBS service is started
Check if standby is up
Check if Symapi data is present
Check if Symapi is synced with Storage System
Check integrity of NASDB
Check if primary is active
Check all callhome files delivered
Check if NAS partitions are mounted

Data Mover Checks:
Check boot files
Check if hardware is supported
Check if primary is active
Check if root filesystem has enough free space
Check if using standard DART image
Check MAC address
Check network connectivity
Check status

Storage System Checks:
Check disk emulation type
Check disk high availability access
Check disks read cache enabled
Check disks and storage processors write cache enabled
Check if access logix is enabled
Check if FLARE is committed
Check if FLARE is supported
Check if microcode is supported
Check no disks or storage processors are failed over
Check that no disks or storage processors are faulted
Check that no hot spares are in use
Check that no hot spares are rebuilding
Check control lun size
Check if storage processors are read cache enabled
FILES

The files associated with system health checkups are:

/nas/log/nas_checkup-run.<timestamp>.log Contains information about the checks that were run, problems found, and actions needed to fix the problem.

/nas/log/nas_checkup.<timestamp>.log Produced when a scheduled nas_checkup is run and contains the same information as the nas_checkup-run.<timestamp>.log.

/nas/log/syslog Provides a sample nas_checkup event configuration file. This is the file to be modified to add email addresses and load the file.

SEE ALSO

Configuring Events and Notifications on VNX for File.

EXAMPLE #1

To run a health check on the VNX, type:

$ nas_checkup

Check Version:  5.6.23.1
Check Command: /nas/bin/nas_checkup
Check Log : /nas/log/checkup-run.070611-064115.log

-------------------------------------Checks-------------------------------------
Control Station: Checking if file system usage is under limit.............. Pass
Control Station: Checking if file systems have enough space to upgrade..... Pass
Control Station: Checking if NAS Storage API is installed correctly....... Pass
Control Station: Checking if NBS clients are started....................... Pass
Control Station: Checking if NBS configuration exists...................... Pass
Control Station: Checking if NBS devices are accessible.................... Pass
Control Station: Checking if NBS service is started......................... Pass
Control Station: Checking if standby is up................................. N/A
Control Station: Checking if Symapi data is present........................ Pass
Control Station: Checking if Symapi is synced with Storage System........ Pass
Control Station: Checking integrity of NASDB............................... Pass
Control Station: Checking all callhome files delivered..................... Pass
Control Station: Checking resolv conf...................................... Pass
Control Station: Checking if NAS partitions are mounted................... Pass
Control Station: Checking ipmi connection.................................. Pass
Control Station: Checking nas site eventlog configuration.................. Pass
Control Station: Checking nas sys mcd configuration........................ Pass
Control Station: Checking nas sys eventlog configuration................... Pass
Control Station: Checking logical volume status............................ Pass
Control Station: Checking ups is available.................................. Fail
Data Movers : Checking boot files............................................ Pass
Data Movers : Checking if primary is active................................ Pass
Data Movers : Checking if root filesystem has enough free space......... Pass
Data Movers : Checking if using standard DART image........................ Pass
Data Movers : Checking network connectivity................................ Pass
Data Movers : Checking status.................................................. Pass
Data Movers : Checking dart release compatibility.......................... Pass
Data Movers: Checking dart version compatibility........................ Pass
Data Movers: Checking server name........................................ Pass
Data Movers: Checking unique id.......................................... Pass
Data Movers: Checking CIFS file server configuration.................... N/A
Data Movers: Checking domain controller connectivity and configuration. N/A
Data Movers: Checking DNS connectivity and configuration................ N/A
Data Movers: Checking connectivity to WINS servers........................ N/A
Data Movers: Checking connectivity to NTP servers........................ N/A
Data Movers: Checking connectivity to NIS servers........................ Pass
Data Movers: Checking virus checker server configuration................. N/A
Data Movers: Checking if workpart is OK................................. Pass
Data Movers: Checking if free full dump is available...................... Fail
Data Movers: Checking if each primary data mover has standby.......... Pass
Storage System: Checking disk emulation type................................ Pass
Storage System: Checking disk high availability access................... Pass
Storage System: Checking disks read cache enabled........................ Pass
Storage System: Checking disks and storage processors write cache enabled. Pass
Storage System: Checking if access logix is enabled....................... Pass
Storage System: Checking if FLARE is committed......................... Pass
Storage System: Checking if FLARE is supported.......................... Pass
Storage System: Checking if microcode is supported....................... Pass
Storage System: Checking no disks or storage processors are failed over... Pass
Storage System: Checking that no disks or storage processors are faulted.. Pass
Storage System: Checking that no hot spares are in use................ Pas
Storage System: Checking that no hot spares are rebuilding.............. Pass
Storage System: Checking minimum control lun size...................... Pass
Storage System: Checking maximum control lun size...................... Fail
Storage System: Checking system lun configuration........................ Pass
Storage System: Checking if storage processors are read cache enabled..... Pass
Storage System: Checking if auto assign are disabled for all luns........ Pass
Storage System: Checking if auto trespass are disabled for all luns..... Pass
Storage System: Checking backend connectivity............................ Pass

One or more warnings are shown below. It is recommended that you follow the instructions below to correct the problem then try again.

-----------------------------------------------------------Information---------------------------------------------------
Control Station: Check ups is available
Symptom: The following UPS emcnasUPS_i0 emcnasUPS_i1 is(are) not available

Data Movers: Check if each primary data mover has standby
Symptom: The following primary Data Movers server_2, server_3 does not have a standby Data Mover configured. It is recommended that each primary Data Mover have a standby configured for it with automatic failover policy for high availability.

Storage System: Check maximum control lun size
Symptom:
* The size of control LUN 5 is 32 GB. It is larger than the recommended size of 14 GB. The additional space will be reserved by the system.

---

Warnings

Data Movers: Check if free full dump is available
Symptom: Cannot get workpart structure. Command failed.
* Command: /nas/sbin/workpart -r
* Command output: open: Permission denied

* Command exit code: 2
Action: Contact EMC Customer Service and refer to EMC Knowledgebase emc146016. Include this log with your support request.

EXAMPLE #2
To display help for nas_checkup, type:

```bash
$ nas_checkup -help
```

Check Version: 5.6.23.1
Check Command: /nas/bin/nas_checkup

```
usage: nas_checkup
    [ -help | -version ]
```

EXAMPLE #3
To display the version of nas_checkup utility, type:

```bash
$ nas_checkup -version
```

Check Version: 5.6.23.1
Check Command: /nas/bin/nas_checkup

DIAGNOSTICS

nas_checkup returns one of the following exit statuses:

- 0 — No problems found
- 1 — nas_checkup posted information
- 2 — nas_checkup discovered a warning
- 3 — nas_checkup discovered an error
- 255 — Any other error

Examples of errors that could cause a 255 exit status include, but are not limited to:

- If nas_checkup is run when another instance of nas_checkup is running
- If nas_checkup is run by someone other than root or the administrator group (generally nasadmin)
- If nas_checkup is run on the standby Control Station
nas_ckpt_schedule

Manages SnapSure checkpoint scheduling for the VNX.

SYNOPSIS

nas_ckpt_schedule
- list
  | - info {-all|<name>|id=<id>}
  | - create <name>
    - filesystem {<name>|id=<id>} [-description <description>]
    - recurrence 
      once [ -start_on <YYYY-MM-DD> ] - runtimes <HH:MM>
      [- ckpt_name <ckpt_name>]
      | daily [ -every <number_of_days> ]
      [ -start_on <YYYY-MM-DD> ] [-end_on <YYYY-MM-DD>]
      [- runtimes <HH:MM> [, . . . ]
      [ -keep <number_of_ckpts> | - ckpt_names <ckpt_name> [, . . . ]]
    | weekly [ -every <number_of_weeks> ]
      - days_of_week {Mon|Tue|Wed|Thu|Fri|Sat|Sun} [, . . . ]
      [ -start_on <YYYY-MM-DD> ] [-end_on <YYYY-MM-DD>]
      [- runtimes <HH:MM> [, . . . ]
      [ -keep <number_of_ckpts> | - ckpt_names <ckpt_name> [, . . . ]]
    | monthly [ -every <number_of_months> ] - days_of_month <1-31> [, . . . ]
      [ -start_on <YYYY-MM-DD> ] [-end_on <YYYY-MM-DD>]
      [- runtimes <HH:MM> [, . . . ]
      [ -keep <number_of_ckpts> | - ckpt_names <ckpt_name> [, . . . ]]
    [{ - cvfsname_prefix <prefix> | - time_based_cvfsname }]
  | - modify {<name>|id=<id>}
    [ - name <new_name>]
    [{ - cvfsname_prefix <prefix> | - time_based_cvfsname }]
    [ - description <description>]
    [ - recurrence { daily | weekly | monthly }]
    [ - every { number_of_days | number_of_weeks | number_of_months }]
    [ - days_of_week { Mon | Tue | Wed | Thu | Fri | Sat | Sun } [, . . . ] ]
    [ - days_of_month < 1-31 > [, . . . ] [ - start_on < YYYY-MM-DD > ]
    [ - end_on < YYYY-MM-DD > ] [ - runtimes <HH:MM> [, . . . ]]
  | - delete {<name>|id=<id>}
  | - pause {<name>|id=<id>}
  | - resume {<name>|id=<id>}

DESCRIPTION

nas_ckpt_schedule creates and lists the schedules for the SnapSure checkpoints. Schedules can be run once, daily, weekly, or monthly and can be modified, paused, resumed, and deleted.

OPTIONS

-list
Lists all checkpoint schedules on the system, the name of the schedule, the next run date, the state, and the description.
-info {-all | <name> | id=<id>}
Lists detailed information for all schedules or for the specified schedule.

(create <name> -filesystem {<name> | id=<id>})
[-description <description>] -recurrence {
Creates a checkpoint schedule for the file system that is specified by
<name> or <id>. The schedule name in -create <name> must be
unique. The -description option provides a label for the schedule.
The -recurrence option specifies if the checkpoint operation occurs
once, daily, weekly, or monthly.

Note: It is recommended that a time interval of at least 15 minutes in between
the creation of two checkpoints on the same production file system. Using
VNX SnapSure provides information on checkpoint scheduling.

once [ -start_on <YYYY-MM-DD> ] -runtime <HH:MM>
[ -ckpt_name <ckpt_name> ]
If once is specified, the hours and minutes for the snapshot to be
run must be specified. A start date and name may be optionally
assigned to the checkpoint.

For a one-time checkpoint schedule, only one runtime can be
provided. For one-time schedules, the option -ckpt_name can
specify a name for the single checkpoint; if omitted, the default
naming is used ( <schedule_name>_<fs_name>_<num> ) where
<num> is a four digit integer beginning with 0001.

daily [ -every <number_of_days> ] [-start_on
<YYYY-MM-DD>] [-end_on <YYYY-MM-DD>]
-runtimes <HH:MM>[,...]
[ -keep <number_of_ckpts> ] -ckpt_names
<ckpt_name>[,...]
If daily is specified, the checkpoint is taken every day unless
-every is specified indicating the number of days between runs.
The -start_on option indicates the day when the checkpoints will
start and -end_on indicates the day when they end.

The -runtimes option specifies one or more times to take a
checkpoint on each scheduled day. The -keep option specifies the
maximum number of checkpoints to be kept at any one time
(using default checkpoint naming). <number_of_ckpts> should
be equal to the number of checkpoint names specified for a
schedule. The -ckpt_name option assigns one or more specific
names to each checkpoint as it is taken.
The nas Commands


If weekly is specified, the checkpoint is taken every week unless the -every option is specified indicating the number of weeks between runs. The -days_of_week option specifies one or more days during the week on which to run the schedule. The -start_on option indicates the day when the checkpoints will start and -end_on indicates the day when they end.

The -runtimes option specifies one or more times to take a checkpoint on each scheduled day. The -keep option specifies the maximum number of checkpoints to be kept at any one time (using default checkpoint naming). The -ckpt_name option assigns one or more specific names to each checkpoint as it is taken.


If monthly is specified, the checkpoint is taken every month unless the -every is specified indicating the number of months between runs. The -days_of_month option specifies one or more days during the month on which to run the schedule. <days> is specified as an integer 1 through 31. The -start_on option indicates the day when the checkpoints will start and -end_on indicates the day when they end.

The -runtimes option specifies one or more times to take a checkpoint on each scheduled day. The -keep option specifies either the maximum number of checkpoints to be kept at any one time (using default checkpoint naming) or using the -ckpt_name option, one or more specific names to assign each checkpoint as it is taken.

Note: The schedule that is set takes effect immediately unless -start_on is specified. Daily, weekly, and monthly schedules run indefinitely unless -end_on is included.
The `cvfsname_prefix` option specifies the customized prefix of a CVFS name. This prefix along with the `cvfsname_delimiter` and the `cvfs_starting_index` make up the CVFS name. The `time_based_cvfsname` option specifies the CVFS name based on the creation time of the CVFS. It is the default method for generating CVFS names and will be used if the prefix is not specified.

**Note:** The prefix must be a PFS-wide unique string and can contain up to 20 ASCII characters. The prefix must not include intervening spaces, colons (:), or slashes (/).

```bash
-modify {<name>|id=<id>} [-name <new_name>] [-cvfsname_prefix <prefix>| -time_based_cvfsname] [-description <description>] [-recurrence {daily|weekly|monthly}] [-every <number_of_days>|<number_of_weeks>|<number_of_months>] [-days_of_week {Mon|Tue|Wed|Thu|Fri|Sat|Sun}[,...]] [-days_of_month <1-31>[,...]] [-start_on <YYYY-MM-DD>] [-end_on <YYYY-MM-DD>] [-runtimes <HH:MM>[,...]]
```

Modifies the scheduled checkpoint entry as specified.

```bash
-delete {<name>|id=<id>}
```

Deletes the specified checkpoint schedule. This operation does not delete any checkpoints.

```bash
-pause {<name>|id=<id>}
```

Pauses the specified checkpoint schedule, including checkpoint creations.

```bash
-resume {<name>|id=<id>}
```

Resumes a paused checkpoint schedule.

**SEE ALSO** *Using VNX SnapSure.*
EXAMPLE #1
To create a checkpoint schedule that creates a checkpoint of the file system ufs1 daily at 8 A.M. and 8 P.M. starting on 11/13/06 with the last run on 12/13/07, and keep 7 checkpoints, type:

```bash
$ nas_ckpt_schedule -create ufs1_ckpt_sched1 -filesystem ufs1 -description "Daily Checkpoint schedule for ufs1" -recurrence daily -every 1 -start_on 2006-11-13 -end_on 2007-12-13 -runtimes 8:00,20:00 -keep 7 -cvfsname_prefix daily
```

This command returns no output.

EXAMPLE #2
To create a checkpoint schedule that creates a checkpoint of the file system ufs1 weekly on Mondays at 6 P.M., starting on 11/13/06 with the last run on 12/13/07, and name new checkpoints ufs1_ckpt_mon1, ufs1_ckpt_mon2, ufs1_ckpt_mon3, ufs1_ckpt_mon4, type:

```bash
$ nas_ckpt_schedule -create ufs1_ckpt_sched2 -filesystem ufs1 -description "Weekly Checkpoint schedule for ufs1" -recurrence weekly -every 1 -days_of_week Mon -start_on 2006-11-13 -end_on 2007-12-13 -runtimes 18:00 -ckpt_names ufs1_ckpt_mon1,ufs1_ckpt_mon2,ufs1_ckpt_mon3,ufs1_ckpt_mon4 -cvfsname_prefix weekly
```

This command returns no output.

EXAMPLE #3
To create a checkpoint schedule that creates a checkpoint of the file system ufs1 every other 15th of the month at 7 P.M., and keep 12 checkpoints, type:

```bash
$ nas_ckpt_schedule -create ufs1_ckpt_sched3 -filesystem ufs1 -description "Monthly Checkpoint schedule for ufs1" -recurrence monthly -every 2 -days_of_month 15 -runtimes 19:00 -keep 12 -cvfsname_prefix monthly
```

This command returns no output.

EXAMPLE #4
To create a checkpoint schedule that creates a checkpoint of the file system ufs1 once at 3:09 P.M., type:

```bash
$ nas_ckpt_schedule -create ufs1_ckpt_sched4 -filesystem ufs1 -description "One-time Checkpoint Schedule for ufs1" -recurrence once -runtimes 15:09
```

Command Line Interface Reference for File 7.0
This command returns no output.

**EXAMPLE #5** To list all checkpoint schedules, type:

```
$ nas_ckpt_schedule -list
```

```
id = 6
name = ufs1_ckpt_sched2
description = Weekly Checkpoint schedule for ufs1
state = Pending
next run = Mon Nov 13 18:00:00 EST 2006
```

```
id = 80
name = ufs1_ckpt_sched4
description = One-time Checkpoint Schedule for ufs1
state = Pending
next run = Tue Nov 14 15:09:00 EST 2006
```

```
id = 5
name = ufs1_ckpt_sched1
description = Daily Checkpoint schedule for ufs1
state = Pending
next run = Mon Nov 13 20:00:00 EST 2006
```

```
id = 7
name = ufs1_ckpt_sched3
description = Monthly Checkpoint schedule for ufs1
state = Pending
next run = Wed Nov 15 19:00:00 EST 2006
```

**EXAMPLE #6** To modify the recurrence of the checkpoint schedule `ufs1_ckpt_sched3` to run every 10th of the month, type:

```
$ nas_ckpt_schedule -modify ufs1_ckpt_sched3 -recurrence monthly -every 1 -days_of_month 10
```

This command returns no output.

**EXAMPLE #7** To get detailed information about checkpoint schedule, type:

```
$ nas_ckpt_schedule -info ufs1_ckpt_sched3
```

```
id = 7
name = ufs1_ckpt_sched3
description = Monthly Checkpoint schedule for ufs1
CVFS name prefix = monthly
tasks = Checkpoint ckpt.ufs1ckpt sched3.001 on filesystem id=25, Checkpoint ckpt.ufs1ckpt sched3.002 on filesystem id=25, Checkpoint ckpt.ufs1ckpt sched3.003 on filesystem id=25, Checkpoint ckpt.ufs1ckpt sched3.004 on filesystem id=25, Checkpoint ckpt.ufs1ckpt sched3.005 on filesystem id=25, Checkpoint ckpt.ufs1ckpt sched3.006 on filesystem id=25, Checkpoint ckpt.ufs1ckpt sched3.007 on filesystem id=25, Checkpoint
```
The nas Commands

ckpt.ufs1_ckpt_sched3_008 on filesystem id=25, Checkpoint
ckpt.ufs1_ckpt_sched3_009 on filesystem id=25, Checkpoint
ckpt.ufs1_ckpt_sched3_010 on filesystem id=25, Checkpoint
ckpt.ufs1_ckpt_sched3_011 on filesystem id=25, Checkpoint
ckpt.ufs1_ckpt_sched3_012 on filesystem id=25
next run = Sun Dec 10 19:00:00 EST 2006
state = Pending
recurrence = every 1 months
start on = Mon Nov 13 16:47:51 EST 2006
end on =
at which times = 19:00
on which days of week =
on which days of month = 10

**EXAMPLE #8**  To pause a checkpoint schedule, type:

```
$ nas_ckpt_schedule -pause ufs1_ckpt_sched1
```
This command returns no output.

**EXAMPLE #9**  To resume a checkpoint schedule, type:

```
$ nas_ckpt_schedule -resume ufs1_ckpt_sched1
```
This command returns no output.

**EXAMPLE #10**  To delete a checkpoint schedule, type:

```
$ nas_ckpt_schedule -delete ufs1_ckpt_sched2
```
This command returns no output.
nas_connecthome

Configures email, FTP, and modem transport mechanisms for transporting Callhome event files to user-configured destinations.

SYNOPSIS

nas_connecthome
 -info
 | -test { -email_1 | -email_2 | -ftp_1 | -ftp_2 | -modem_1 | -modem_2 }
 | -modify [-modem_priority { Disabled | 1 | 2 | 3 }]
 | [-modem_number <phone_number>]
 | [-modem_number_2 <phone_number>]
 | [-ftp_priority { Disabled | 1 | 2 | 3 }]
 | [-ftp_server { <hostname> | <ip_addr> }]
 | [-ftp_port <port>]
 | [-ftp_user <username>]
 | [-ftp_passwd <passwd>]
 | [-ftp_folder <path>]
 | [-ftp_ipprotocol { IPv4 | IPv6 }]
 | [-ftp_mode { active | passive }]
 | [-ftp_server_2 { <hostname> | <ip_addr> }]
 | [-ftp_port_2 <port>]
 | [-ftp_user_2 <username>]
 | [-ftp_passwd_2 <passwd>]
 | [-ftp_folder_2 <path>]
 | [-ftp_ipprotocol_2 { IPv4 | IPv6 }]
 | [-ftp_mode_2 { active | passive }]
 | [-email_priority { Disabled | 1 | 2 | 3 }]
 | [-email_from <email_addr>]
 | [-email_to { <email_addr>, <email_addr> }]
 | [-email_subject <email_subject>]
 | [-email_server { <hostname> | <ip_addr> }]
 | [-email_ipprotocol { IPv4 | IPv6 }]
 | [-email_server_2 { <hostname> | <ip_addr> }]
 | [-email_ipprotocol_2 { IPv4 | IPv6 }]
 | [-https_priority { Disabled | 1 | 2 | 3 }]
 | [-https_url <url>]
 | [-https_ipprotocol { IPv4 | IPv6 }]
 | [-dial_in_number <phone_number>]
 | [-serial_number <serial_number>]
 | [-site_id <site_id>]
 | [-encryption_enabled { yes | no }]
 | [-dial_in_enabled { yes | no }]
 | [-service_info]
 | -help

DESCRIPTION

nas_connecthome pauses and resumes the ConnectHome service, displays and configures parameters for email, FTP, and modem, which are mechanisms used for transmitting event files.
nas_connecthome enables a user to configure primary and optional secondary destinations for each transport mechanism.

nas_connecthome also tests connectivity to the destination configured for a transport mechanism.

This command must be executed from /nas/sbin/.

**OPTIONS**

- **-info**
  Displays the enabled and disabled configuration parameters for all three transport mechanisms.

- **-test** `-email_1` `-email_2` `-ftp_1` `-ftp_2` `-modem_1` `-modem_2`
  Tests connectivity to the destination configured and enabled for the specified transport mechanism.

- **-modify**
  Modifies the following configuration parameters for any or all three transport mechanisms:

    `[-modem_priority {Disabled|1|2|3}]`
  Enables modem as a Primary, Secondary, or Tertiary transport mechanism. Specifying Disabled removes modem as a transport mechanism.

    `[-modem_number <phone_number>]`
  Sets or modifies the primary phone number of the modem.

    **Note:** Specifying "" (empty double quotes) disables the use of the existing phone number.

    `[-modem_number_2 <phone_number>]`
  Sets or modifies the secondary phone number of the modem.

    **Note:** Specifying "" (empty double quotes) disables the use of the existing phone number for this transport mechanism.

    `[-ftp_priority {Disabled|1|2|3}]`
  Enables FTP as a Primary, Secondary, or Tertiary transport mechanism. Specifying Disabled removes FTP as a transport mechanism.

    `[-ftp_server {<hostname>|<ip_addr>}]`
  Sets or modifies the hostname or IP address of the primary FTP server and corresponding port. The allowable input is IPv4 address, IPv6 address, or domain name.
The nas Commands

[-ftp_port <port>]
Sets or modifies the port of the primary FTP server and corresponding port. The valid input is an integer between 1 and 65535. If an empty string "" is provided for this option, the port number is reset to the default value 21.

[-ftp_user <username>]
Sets or modifies the username of the login account on the primary FTP server.

Note: Specifying "" (empty double quotes) reverts to the default value of onalert.

[-ftp_passwd [<passwd>]]
Sets or modifies the password of the login account on the primary FTP server.

Note: Specifying "" (empty double quotes) reverts to the default value of EMCCCONNECT.

[-ftp_folder <path>]
Sets or modifies the path to the folder on the primary FTP server where the event files have to be deposited.

Note: Specifying "" (empty double quotes) reverts to the default value of incoming.

[-ftp_ipprotocol {IPV4 | IPV6}]
Sets or modifies the transfer mode of the primary FTP transport mechanism. If an IPv4 address is provided to FTP server, the corresponding IP protocol is changed to IPv4 automatically. If an IPv6 address is used, the IP protocol is changed to IPv6. When hostname is specified, no IP protocol change is made.

[-ftp_mode {active | passive}]
Sets or modifies the transfer mode of the primary FTP transport mechanism.

Note: Specifying "" (empty double quotes) reverts to the default value of active.
[-ftp_server_2 <hostname><ip_addr>]]
Sets or modifies the hostname or IP address of the secondary FTP server and corresponding port. The allowable input is IPv4 address, IPv6 address, or domain name.

[-ftp_port_2 <port>]
Sets or modifies the port of the secondary FTP server and corresponding port. The valid input is an integer between 1 and 65535. If an empty string "" is provided for this option, the port number is reset to the default value of 21.

[-ftp_user_2 <username>]
Sets or modifies the username of the login account on the secondary FTP server.

**Note:** Specifying "" (empty double quotes) reverts to the default value of onalert.

[-ftp_passwd_2 [passwd]]
Sets or modifies the password of the login account on the secondary FTP server.

**Note:** Specifying "" (empty double quotes) reverts to the default value of EMCCONNECT.

[-ftp_folder_2 <path>]
Sets or modifies the path of the folder on the secondary FTP server where the event files have to be deposited.

**Note:** Specifying "" (empty double quotes) reverts to the default value of incoming.

[-ftp_ipprotocol_2 {IPv4 | IPv6}]
Sets or modifies the transfer mode of the secondary FTP transport mechanism.

[-ftp_mode_2 { active | passive}]
Sets or modifies the transfer mode of the secondary FTP transport mechanism.

**Note:** Specifying "" (empty double quotes) reverts to the default value of active.
The nas Commands

[-email_priority {Disabled|1|2|3 }]
Enables email as a Primary, Secondary, or Tertiary transport mechanism. Specifying Disabled removes email as a transport mechanism.

[-email_from <email_addr>]
Sets or modifies the sender's email address. The maximum number of characters that can be specified is 63.

Note: Specifying "" (empty double quotes) reverts to the default value of connectemc@emc.com.

[-email_to <email_addr> [, <email_addr> ]]
Sets or modifies the destination email addresses that receive the event files. Multiple email addresses can be specified with a comma separating each address. The maximum number of characters that can be specified is 255.

Note: Specifying "" (empty double quotes) reverts to the default value of emailalert@emc.com.

[-email_subject <email_subject>]
Sets or modifies the subject of the email message.

Note: Specifying "" (empty double quotes) reverts to the default value of CallHome Alert.

[-email_server { <hostname> | <ip_addr> }]
Sets or modifies the primary email server that accepts and routes email messages.

Note: Specifying "" (empty double quotes) disables the use of the existing email server for this transport mechanism.

[-email_ipprotocol {IPv4 | IPv6 }]
Sets or modifies the secondary email server that accepts and routes email messages.

[-email_server_2 { <hostname> | <ip_addr> }]
Sets or modifies the secondary email server that accepts and routes email messages.
Note: Specifying "" (empty double quotes) disables the use of the existing email server for this transport mechanism.

[-email_ipprotocol_2 {IPv4 | IPv6}]  
Sets or modifies the secondary email server that accepts and routes email messages.

[-https_priority {Disabled | 1 | 2 | 3}]  
Enables https as a Primary, Secondary, or Tertiary transport mechanism. Specifying Disabled removes https as a transport mechanism.

[-https_url]  
The url of the monitoring station.

[-https_ipprotocol {IPv4 | IPv6}]  
Sets or modifies the transfer mode of the secondary HTTPS transport mechanism.

[-dial_in_number <phone_number>]  
Sets or modifies the dial-in phone number of the modem.

Note: Specifying "" (empty double quotes) does not disable the number or restore its default value. The empty string is stored as is.

[-serial_number <serial_number>]  
Sets or modifies the VNX serial number, if it was not automatically detected.

Note: Specifying "" (empty double quotes) does not disable the number or restore its default value. The empty string is stored as is.

[-site_id <site_id>]  
Sets or modifies the site ID.

Note: Specifying "" (empty double quotes) does not disable the number or restore its default value. The empty string is stored as is.

[-encryption_enabled {yes | no}]  
Enables or disables the encryption of the CallHome payload during transmission.

Note: Specifying "" (empty double quotes) reverts to the default value of yes.
The nas Commands

[-dial_in_enabled {yes|no}]
Enables or disables dial-in login sessions.

Note: Specifying "" (empty double quotes) reverts to the default value of yes.

SEE ALSO Configuring Events and Notifications on VNX for File.
EXAMPLE #1  To display configuration information, type:

    # /nas/sbin/nas_connecthome -info ConnectHome

    Configuration:
    Encryption Enabled         = yes
    Dial In :
        Enabled                 = yes
        Modem phone number      = 9123123123
        Site ID                 = MY SITE
        Serial number           = APM00054703223
    Email :
        Priority               = 1
        Sender Address         = admin@yourcompany.com
        Recipient Address(es)  = emailalert@emc.com
        Subject                = CallHome Alert
        Primary :
            Email Server       = backup.mailhub.company.com
        Secondary :
            Email Server       =
    FTP :
        Priority               = 2
        Primary :
            FTP Server          = 1.2.3.4
            FTP Port            = 22
            FTP User Name       = onalert
            FTP Password        = **********
            FTP Remote Folder   = incoming
            FTP Transfer Mode   = active
        Secondary :
            FTP Server          = 1.2.4.4
            FTP Port            = 22
            FTP User Name       = onalert
            FTP Password        = **********
            FTP Remote Folder   = incoming
            FTP Transfer Mode   = active
    Modem :
        Priority               = Disabled
        Primary :
            Phone Number       =
            BT Tymnet          = no
        Secondary :
            Phone Number       =
            BT Tymnet          = no

EXAMPLE #2  To test the primary email server, type:

    # /nas/sbin/nas_connecthome -test -email_1
RSC API Version: 2.0.27-bl18
Copyright (C) EMC Corporation 2003-2007, all rights reserved.
------------------------------------------
Reading configuration file: ConnectEMC.ini.
Run Service begin...
Test succeeded for Primary Email.
The nas Commands

---

**nas_config**

Manages a variety of configuration settings on the Control Station, some of which are security based.

**SYNOPSIS**

```
nas_config
    -IPalias {-list | -create [-name <device_name>] <numeric_id> | -delete [-name <device_name>] <numeric_id>}
    | -ssl
    | -sessiontimeout [<number_in_minutes> | off]
    | -password -default
```

**DESCRIPTION**

`nas_config -IPalias` configures different IP addresses to point to the same network device allowing use of IP aliasing to manage the Control Station. This enables communication with the primary Control Station using a single IP address regardless of whether the primary Control Station is running in slot 0 or slot 1.

`nas_config -ssl` generates an X.509 digital certificate on the Control Station. Unisphere uses the Secure Socket Layer (SSL) protocol to create a secure connection between a user’s Web browser and the Control Station’s Apache Web server. When a VNX system is initially installed, a generic certificate is generated. After configuring the Control Station’s network configuration (hostname, DNS domain name, or IP address) and before using the Unisphere, a new certificate should be generated.

`nas_config -sessiontimeout` sets a system-wide value that automatically times out a Control Station shell session after a specified period of inactivity.

`nas_config -password` supports a password quality policy by requiring that passwords chosen by VNX users adhere to certain rules.

You must be `root` to execute this command from the `/nas/sbin` directory.
OPTIONS

-IPalias -list
Lists IP aliases configured on the VNX.

-IPalias -create [-name <device_name>] <numeric_id>
Creates an IP alias for the Control Station.

<device_name> is the name for a specified device:
◆ If you specify a device name that device must have an IP address.
◆ If you do not specify a device name, the system uses the external network interface.

<numeric_id> is a user-defined number, and can be an integer between 0 and 255. The system allows up to 256 aliases for any device.

-delete [-name <device_name>] <numeric_id>
Deletes an IP alias for the Control Station.

-ssl
Installs a SSL certificate on the Control Station and restarts the HTTP server.

-sessiontimeout [<number_in_minutes>|off]
Displays the current session timeout value in minutes.
<number_in_minutes> sets the number of minutes a Control Station shell session can be inactive before it is timed out. Possible values are 5 to 240 minutes. The default value is 60 minutes. Session timeout is enabled by default. To disable session timeout, type off or 0 to indicate zero minutes.

The -sessiontimeout option enables the native timeout properties of the underlying shells on the Control Station. The relevant shell man page provides a description of how the mechanism works.

-password
Prompts for specific password policy definitions. The current value for each policy definition is shown in brackets.

[-min <6..15>] defines the minimum length of the new password. The default length is eight characters. The length has to be a value between 6 and 15 characters.

[-retries <max_allowed>] defines the number of attempts a user can make to define an acceptable new password before the command fails. The default value is three attempts.
The nas Commands

[-newchars <min_num>] defines the minimum number of characters that must be in the new password that were not included in the old password. The default value is three characters.

[-digits <min_num>] defines the minimum number of digits that must be included in the new password. The default value is one digit.

[-spechars <min_num>] defines the minimum number of special characters (such as ! @ # $ % & ^ and *) that must be included in the new password. The default value is 0.

[-lcase <min_num>] defines the minimum number of lowercase characters that must be included in the new password. The default value is 0.

[-ucase <min_num>] defines the minimum number of uppercase characters that must be included in the new password. The default value is 0.

-password -default
Resets the password policy definitions to their default values.


EXAMPLE #1 To create an IP alias for the Control Station, type:

    # /nas/sbin/nas_config -IPalias -create 0
Do you want slot_0 IP address <1.2.3.4> as your alias [yes or no]: no
Please enter an IP address to use as an alias: 1.2.3.6
Do you want slot_0 IP address <1.2.3.4> as your alias [yes or no]: yes
Please enter a new IP address for slot_0: 1.2.3.6

EXAMPLE #2 To view the IP alias that you created, type:

    # /nas/sbin/nas_config -IPalias -list
alias IPaddress state
eth2:0 1.2.3.6 UP

EXAMPLE #3 To delete an IP alias, type:

    # /nas/sbin/nas_config -IPalias -delete 0
All current sessions using alias eth2:0 will terminate
Do you want to continue [yes or no]: yes
done
EXAMPLE #4  To generate and install a certificate for the Apache Web server on the Control Station, type:

```
# /nas/sbin/nas_config -ssl
```

Installing a new SSL certificate requires restarting the Apache web server. Do you want to proceed? [y/n]: y

New SSL certificate has been generated and installed successfully.

EXAMPLE #5  To change the session timeout value from the default value of 60 minutes to 100 minutes, type:

```
# /nas/sbin/nas_config -sessiontimeout 100
```

done

EXAMPLE #6  To disable session timeout, type:

```
# /nas/sbin/nas_config -sessiontimeout 0
```

done

or

```
# /nas/sbin/nas_config -sessiontimeout off
```

done

EXAMPLE #7  To set specific password policy definitions, type:

```
# /nas/sbin/nas_config -password
```

Minimum length for a new password (Between 6 and 15): [8]  
Number of attempts to allow before failing: [3]  
Number of new characters (not in the old password): [3]  
Number of digits that must be in the new password: [1]  
Number of special characters that must be in a new password: [0]  
Number of lower case characters that must be in password: [0]  
Number of upper case characters that must be in password: [0]

EXAMPLE #8  To set the minimum length of a new password to 10 characters, type:

```
# /nas/sbin/nas_config -password -min 10
```

EXAMPLE #9  To reset the current password policy definitions to their default values, type:

```
# /nas/sbin/nas_config -password -default
```
nas_copy

Creates a replication session for a one-time copy of a file system. This command is available with VNX Replicator only.

SYNOPSIS

nas_copy
   -name <sessionName>
   -source
      { -fs { <name> | id=<fsId> } | -ckpt { <ckptName> | id=<ckptId> } }
   -destination
      { -fs { id=<dstFsId> | <existing_dstFsName> } |
        -pool { id=<dstStoragePoolId> | <dstStoragePool> } }
      [ -from_base { <ckpt_name> | id=<ckptId> } ]
   -interconnect { <name> | id=<interConnectId> }
   [ -source_interface { <nameServiceInterfaceName> | ip=<ipaddr> } ]
   [ -destination_interface { <nameServiceInterfaceName> | ip=<ipaddr> } ]
   [ -overwrite_destination ]
   [ -refresh ]
   [ -full_copy ]
   [ -background ]

DESCRIPTION

nas_copy from the Control Station on the source side, performs a one-time copy of a source read-only file system or a checkpoint file system.

Note: Depending on the size of the data in the source, this command may take some time to complete. Once a copy session begins, you can monitor it or interrupt it if necessary using the nas_task command. You can list all replication sessions, including copy sessions, using the nas_replicate -list command.

OPTIONS

-name <sessionName>  -source -fs
{ <name> | id=<fsId> } | -ckpt { <ckptName> | id=<ckptId> }
   -destination -fs { <existing_dstFsName> | id=<dstFsId> |
      -pool <dstStoragePool> | id=<dstStoragePoolId> } [ -from_base { <ckpt_name> |
      id=<ckptId> } ] -interconnect { <name> | id= <interConnectId> }

Creates a VNX Replicator session that performs a one-time copy of a source read-only file system or a checkpoint file system.

The session name assigned must be unique for the Data Mover pair as defined by the interconnect. The naming convention <source_fs_or_ckpt_name>_replica<#> is used if a read-only file system or checkpoint at the destination already has the same name as
the source. An integer between 1 and 4 is assigned according to how many replicas of that file system or checkpoint already exist.

The **-source** specifies the name or ID of an existing read-only file system or checkpoint file system as the source for this copy session. This is to be used as a common base for the initial transfer. The checkpoint is identified by checkpoint name or checkpoint file system ID. This option is intended to accommodate upgrade situations to VNX Replicator.

The **-destination** specifies either an existing destination file system or the storage needed to create the destination file system automatically as part of the copy operation. An existing destination file system must be read-only and the same size as the source. Specifying a storage pool or ID creates the destination file system automatically, as read-only, using the same name and size as the source file system.

\[-from\_base \{ckpt\_name>\mid id=<ckptId>\}\]  
Specifies an existing source file system checkpoint to be used as a common base for the initial data transfer. The checkpoint is identified by the checkpoint name or ID.

The **-interconnect** specifies the local (source) side of an established Data Mover interconnect to use for this copy session. Use the **nas_cel -interconnect -list** command on the source VNX to list the interconnects available to VNX Replicator sessions.

\[-source\_interface \{<nameServiceInterfaceName> \mid ip= <ipAddr>\}\]  
Instructs the copy session to use a specific local interface defined for the interconnect on the source VNX instead of selecting the local interface supporting the lowest number of sessions (the default). If this local interface was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address. The **source\_interfaces** field of the output from the **nas_cel -interconnect -info** command shows how the source interface was defined. This option does not apply to a loopback interconnect, which always uses 127.0.0.1.

\[-destination\_interface \{<nameServiceInterfaceName> \mid ip=<ipaddr>\}\]  
Instructs the copy session to use a specific interface defined for the interconnect on the destination VNX instead of selecting the peer interface supporting the lowest number of sessions (the default). If this peer interface was defined for the interconnect
using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address. The destination_interfaces field of the output from the nas_cel -interconnect -info command shows how the peer interface was defined. This option does not apply to a loopback interconnect, which always uses 127.0.0.1.

[-overwrite_destination]
For an existing destination, discards any changes made to the destination object and restores it from the established common base (differential copy). If this option is not specified and an existing destination object contains different content than the established common base, an error is returned.

[-refresh {<name>|id=<session_id>}
Updates a destination checkpoint that has the same name as the copied checkpoint. This option does not refresh the source object; it refreshes only the destination for a existing checkpoint. If you specify this option and no checkpoint exists with the same name, the command returns an error.

[-full_copy]
For an existing destination object, if a common base checkpoint exists, performs a full copy of the source checkpoint to the destination, instead of a differential copy. If this option is not specified and an existing destination object has different content than the established common base, an error is returned.

[-background]
Executes the command in asynchronous mode. Use the nas_task command to check the status of the command.

SEE ALSO
nas_cel, nas_replicate, nas_task.

EXAMPLE #1
To create a one-time copy of a checkpoint file system with session name ufs1_replica1 with the source checkpoint ufs_ckpt1 and destination pool clar_r5_performance on the interconnect NYs3_LAs2, source interface 10.6.3.190, and destination interface 10.6.3.173, type:

$ nas_copy -name ufs1_replica1 -source -ckpt ufs1_ckpt1 -destination -pool clar_r5_performance -interconnect NYs3_LAs2 -source_interface 10.6.3.190 -destination_interface 10.6.3.173

OK
EXAMPLE #2  To create a one-time copy of a read-only file system for the session `ufs1_replica1` with source file system `ufs1` and overwrite an existing destination file system `ufs1` on the interconnect `NYs3_LAs2`, source interface `10.6.3.190`, and destination interface `10.6.3.173`, type:

```
$ nas_copy -name ufs1_replica1 -source -fs ufs1 -destination -fs ufs1 -interconnect NYs3_LAs2 -source_interface 10.6.3.190 -destination_interface 10.6.3.173 -overwrite_destination
```

OK

EXAMPLE #3  To initiate a differential copy of `ufs_ckpt2` to the `ufs1_destination` file system using `ufs1_ckpt1` as the common base, using the `-from_base` option, type:

```
$ nas_copy -name ufs1_replica1 -source -ckpt -ufs1_ckpt2 -destination -fs ufs1_destination -from_base ufs1_ckpt1 -interconnect NYs3_LAs2
```

OK

CAUTION

Using the `-from_base` option overrides any common base that may exist. Make sure that the specified checkpoint represents the correct state of the destination file system.

EXAMPLE #4  To refresh the destination of the replication session `ufs1_replica1` for the source checkpoint `ufs1_ckpt1` and destination file system `ufs1` on the interconnect `NYs3_LAs2`, type:

```
$ nas_copy -name ufs1_replica1 -source -ckpt ufs1_ckpt1 -destination -fs ufs1 -interconnect NYs3_LAs2 -refresh
```

OK

EXAMPLE #5  To perform a full copy of the source checkpoint to the destination for the replication session `ufs1_replica1` with the source file system `ufs1` and destination file system `ufs1` on the interconnect `NYs3_LAs2`, type:

```
$ nas_copy -name ufs1_replica1 -source -fs ufs1 -destination -fs ufs1 -interconnect NYs3_LAs2 -overwrite_destination -full_copy -background
```

Info 26843676673: In Progress: Operation is still running. Check task id 4177 on the Task Status screen for results.
The nas Commands

nas_cs

Manages the configuration properties of the Control Station.

SYNOPSIS

nas_cs
   -info [-timezones]
| -set [-hostname <hostname>]
   [-ip4address <ipv4_address>]
   [-ip4netmask <ipv4_netmask>]
   [-ip4gateway <ipv4_gateway>]
   [-ip6address <ipv6_address[/prefix_length]>
   [-ip6gateway <ipv6_gateway>]
   [-dns_domain <dns_domain_name>]
   [-search_domains <domain_name>[,...]]
   [-dns_servers <ip_addr>[,...]]
   [-session_idle_timeout <minutes>]
   [-session_monitor_timeout <days>]
   [-time <yyyyymmddhhmm [ss]>]
   [-timezone <time_zone_str>]
   [-ntp_servers <ip_addr>[,...]]
| -clear [-ip4gateway]
   [-ip6address]
   [-ip6gateway]
   [-dns]
   [-search_domains]
   [-session_monitor_timeout]
   [-session_idle_timeout]
   [-ntp_servers]
| -reboot

DESCRIPTION

nas_cs sets, clears, and lists the Control Station configuration. nas_cs can be used to reboot the Control Station.

OPTIONS

-info [-timezones]
Displays the Control Station configuration. The -timezones option displays all supported time zones on the Control Station.

-set [-hostname <hostname>]
Sets the user configurable paratmeters of the Control Station configuration. Sets the hostname of the primary Control Station. To specify a hostname, the maximum number of characters is 64, excluding white spaces and dot characters.

   [-ip4address <ipv4_address>]
Sets the IPv4 network address of the primary Control Station. The IPv4 address must be a valid address.
The nas Commands

[-ip4netmask <ipv4_netmask>]
Sets the subnet mask for a valid IPv4 network address on the primary Control Station.

[-ip4gateway <ipv4_gateway>]
Sets the IPv4 network address of the gateway machine for the primary Control Station on the network. The IPv4 address must be a valid address.

[-ip6address <ipv6_addr[/prefix_length]>]
Sets the IPv6 network address of the primary Control Station. The IPv6 address must be a valid address. The /prefix_length option sets the integer value, between 8 and 128, for the prefix length of the IPv6 address of the primary Control Station.

[-ip6gateway <ipv6_gateway>]
Sets the IPv6 network address of the gateway machine for the primary Control Station on the network. The IPv6 address must be a valid address.

[-dns_domain <dns_domain_name>]
Sets the Domain Name System of which the primary Control Station is a member. It can accept valid domain names.

[-search_domains <domain_name>[,...]]
Sets the order in which DNS domains are searched to resolve a comma separated list of valid domain names.

[-dns_servers <ip_addr>[,...]]
Sets the IP addresses of the DNS servers of the domain. It is a comma separated list of valid IPv4 or IPv6 addresses and can have a maximum of three DNS addresses.

[-session_idle_timeout <minutes>]
Sets the timeout period in minutes for an inactive administrator session to become invalid.

[-session_monitor_timeout <days>]
Sets the limit for the number of days until when a valid login is allowed to run queries on the primary Control Station. Any active management session requires a login on the primary Control Station.

[-time <yyyymmdhhmm [ss]>]
Sets the current system date and time in the format <yyyymmdhhmm [ss]>.
[**timezone** <time_zone_str>]
Sets a valid time zone value on the primary Control Station.

[**ntp_servers** <ip_addr>[,...]]
Sets the IP addresses of the NTP server used by the primary Control Station. It is a comma separated list of valid IPv4 or IPv6 addresses and can have a maximum of four NTP addresses.

-**clear** [-ip4gateway]
Clears the user configurable parameters of the Control Station configuration. Clears the IPv4 network address of the gateway machine for the primary Control Station on the network.

[-ip6address]
Clears the IPv6 network address of the primary Control Station.

[-ip6gateway]
Clears the IPv6 network address of the gateway machine for the primary Control Station on the network.

[-dns]
Clears the Domain Name System of which the primary Control Station is a member.

[-search_domains]
Clears the order in which DNS domains are searched to resolve a list of valid domain names.

[-session_idle_timeout]
Clears the timeout period in minutes for an inactive administrator session to become invalid.

[-session_monitor_timeout]
Clears the limit for the number of days until when a valid login is allowed to run queries on the primary Control Station. Any active management session requires a login on the primary Control Station.

[-ntp_servers]
Clears the IP addresses of the NTP server used by the primary Control Station.

-**reboot**
Reboots the primary Control Station.
EXAMPLE #1  To display the configuration properties of the primary Control Station, type:

$ nas_cs -info

Example #2  To set the hostname, IPv4 network address, subnet mask, and IPv4 gateway for the primary Control Station, type:

$ nas_cs -set -hostname eng24416 -ip4address 172.24.250.26

OK

Example #3  To set the IPv6 address and the IPv6 gateway for the primary Control Station, type:


OK

Example #4  To set the DNS domain, search domains, and DNS servers for the primary Control Station, type:


OK
**EXAMPLE #5**  To set the session monitor timeout and session idle timeout for the primary Control Station, type:

\$ nas_cs -set -session_monitor_timeout 2
- session_idle_timeout 30

OK

**EXAMPLE #6**  To set the date, time, timezone, and NTP servers for the primary Control Station, type:

\$ nas_cs -set -time 200811070205 -timezone America/New_York -ntp_server 128.221.252.0

OK

**EXAMPLE #7**  To clear the IPv4 gateway for the primary Control Station, type:

\$ nas_cs -clear -ip4gateway

OK

**EXAMPLE #8**  To clear the IPv6 network address and IPv6 gateway for the primary Control Station, type:

\$ nas_cs -clear -ip6address -ip6gateway

OK

**EXAMPLE #9**  To clear the DNS domain and DNS server configuration for the primary Control Station, type:

\$ nas_cs -clear -dns

OK

**EXAMPLE #10**  To clear the domain search configuration for the primary Control Station, type:

\$ nas_cs -clear -search_domains

OK

**EXAMPLE #11**  To clear the NTP server configuration for the primary Control Station, type:

\$ nas_cs -clear -ntp_servers

OK

**EXAMPLE #12**  To reboot the primary Control Station, type:

\$ nas_cs -reboot

OK
nas_devicegroup

Manages an established MirrorView/Synchronous (MirrorView/S) consistency group, also known as a device group.

SYNOPSIS

nas_devicegroup
   -list
   -info {<name> | id=<id> | -all} [-sync [yes | no]]
   -acl <acl_value> {<name> | id=<id>}
   -suspend {<name> | id=<id>}
   -resume {<name> | id=<id>}

DESCRIPTION

nas_devicegroup lists the device group information for a MirrorView/S configuration, gets detailed information about a consistency group, specifies an access control level value for the group, suspends MirrorView/S operations, or resumes operations of the device group.

A MirrorView/S with a VNX configuration involves source and destination VNXs attached to old versions of storage systems. MirrorView/S performs synchronous mirroring of source storage logical units (LUNs) representing production images, where the mirrored LUNs are part of a MirrorView/S consistency group.

On the source VNX, you must be root to issue the -acl, -suspend, and -resume options.

nas_devicegroup must be run from a Control Station in slot 0; it will report an error if run from a Control Station in slot 1.

OPTIONS

-list
Displays a list of available configured MirrorView/S device groups.

-info {<name> | id=<id> | -all} [-sync [yes | no]]
Displays detailed information about the MirrorView/S configuration for a specific device group or for all groups.

   [-sync [yes | no]]
   The -sync option first synchronizes the Control Station’s view with the VNX for block before displaying configuration information. The default is yes.

-acl <acl_value> {<name> | id=<id>}
Sets an access control level value that defines the owner of the storage system, and the level of access allowed for users and groups defined
in the access control level table. The `nas_acl` command provides more information).

**CAUTION**

The access control level value for the group should not be changed from the default setting. A change in access control level value can prevent MirrorView/S from functioning properly.

```bash
-suspend {<name>|id=<id>}
```

Temporarily halts mirroring from the source to the destination, thereby suspending the link. Changes can still be made to the source LUNs, but are not applied to the destination LUNs until operations are resumed.

```bash
-resume {<name>|id=<id>}
```

Resumes device group operations and restarts mirroring, synchronizing the destination LUNs with the source LUNs.

**SEE ALSO**

`Using MirrorView/Synchronous with VNX for Disaster Recovery`, `nas_acl`, and `nas_logviewer`.

**STORAGE SYSTEM OUTPUT**

The number associated with the storage device is dependent on the attached storage system of the system; for MirrorView/S, some VNX for block display a prefix of APM before a set of integers, for example, APM0003900124-0019. The VNX for block supports the following system-defined AVM storage pools for MirrorView/S only: `cm_r1`, `cm_r5_performance`, `cm_r5_economy`, `cmata_archive`, and `cmata_r3`.

**EXAMPLE #1**

To list the configured MirrorView/S device groups that are available, type:

```
$ nas_devicegroup -list
```

```
ID name                owner   storage ID      acl type
2   mviewgroup            500     APM00053001549  0   MVIEW
```

**EXAMPLE #2**

To display detailed information for a MirrorView/S device group, type:

```
$ nas_devicegroup -info mviewgroup
```

```
Sync with CLARiiON backend ...... done
name                   = mviewgroup
description            =
uid                    = 50:6:1:60:B0:60:27:20:0:0:0:0:0:0:0:0
state                  = Synchronized
role                   = Primary
condition              = Active
recovery policy        = Automatic
```
number of mirrors = 16
mode = SYNC
owner = 500
mirrored disks =
local clarid = APM00053001549
remote clarid = APM00053001552
mirror direction = local -> remote

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sync with CLARiiON storage system</td>
<td>Indicates that a sync with the VNX for block was performed to retrieve the most recent information. This does not appear if you specify -info -sync no.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the device group.</td>
</tr>
<tr>
<td>description</td>
<td>Brief description of device group.</td>
</tr>
<tr>
<td>uid</td>
<td>UID assigned, based on the system.</td>
</tr>
<tr>
<td>state</td>
<td>State of the device group (for example, Consistent, Synchronized, Out-of-Sync, Synchronizing, Scrambled, Empty, Incomplete, or Local Only).</td>
</tr>
<tr>
<td>role</td>
<td>Whether the current system is the Primary (source) or Secondary (destination).</td>
</tr>
<tr>
<td>condition</td>
<td>Whether the group is functioning (Active), Inactive, Admin Fractured (suspended), Waiting on Sync, System Fractured (which indicates link down), or Unknown.</td>
</tr>
<tr>
<td>recovery policy</td>
<td>Type of recovery policy (Automatic is the default and recommended value for group during storage system configuration; if Manual is set, use -resume after a link down failure).</td>
</tr>
<tr>
<td>number of mirrors</td>
<td>Number of mirrors in group.</td>
</tr>
<tr>
<td>mode</td>
<td>MirrorView mode (always SYNC in this release).</td>
</tr>
<tr>
<td>owner</td>
<td>User whom the object is assigned to, and is indicated by the index number in the access control level table. nas_acl provides information.</td>
</tr>
<tr>
<td>mirrored disks</td>
<td>Comma-separated list of disks that are mirrored.</td>
</tr>
<tr>
<td>local clarid</td>
<td>APM number of local VNX for block storage array.</td>
</tr>
<tr>
<td>remote clarid</td>
<td>APM number of remote VNX for block storage array.</td>
</tr>
<tr>
<td>mirror direction</td>
<td>On primary system, local to remote (on primary system); on destination system, local from remote.</td>
</tr>
</tbody>
</table>

**EXAMPLE #3**

To display detailed information about a MirrorView/S device group without synchronizing the Control Station’s view with the VNX for block, type:

```
$ nas_devicegroup -info id=2 -sync no
```

name = mviewgroup
description =
uid = 50:6:1:60:B0:60:27:20:0:0:0:0:0:0:0:0:0
state = Consistent
role = Primary
condition = Active
recovery policy = Automatic
number of mirrors = 16
The `nas Commands`

```
mode                  = SYNC
owner                 = 500
mirrored disks        =
local clarid          = APM00053001549
remote clarid         = APM00053001552
mirror direction      = local -> remote
```

**EXAMPLE #4**  To halt operation of the specified device group, as **root** user, type:

```
# nas_devicegroup -suspend mviewgroup
```

Sync with CLARiiON backend ...... done
STARTING an MV 'SUSPEND' operation.
  Device group: mviewgroup ............ done
The MV 'SUSPEND' operation SUCCEEDED.
done

**EXAMPLE #5**  To resume operations of the specified device group, as **root** user, type:

```
# nas_devicegroup -resume mviewgroup
```

Sync with CLARiiON backend ...... done
STARTING an MV 'RESUME' operation.
  Device group: mviewgroup ............ done
The MV 'RESUME' operation SUCCEEDED.
done
The nas Commands

**nas_disk**

Manages the disk table.

**SYNOPSIS**

nas_disk
   -list
   | -delete <disk_name> ![[-perm] | [-unbind]]
   | -info {<disk_name>|id=<disk_id>}
   | -rename <old_name> <new_name>

**DESCRIPTION**

nas_disk displays a list of known disks and renames, deletes, or displays information for the specified disk.

**OPTIONS**

- **-list**
  Lists the disk table.

*Note: The ID of the object is an integer and is assigned automatically. The name of the disk might be truncated if it is too long for the display. To display the full name, use the -info option with the disk ID.*

- **-delete <disk_name> ![[-perm] | [-unbind]]**
  Deletes an entry from the disk table. In a VNX, restores the VNX for block LUN name to its default value.

  Unless -perm is specified, the disk is still identified as a VNX disk and can be discovered and marked again using server_devconfig. The -perm option removes the entry from the disk table and deletes the diskmark. The disk is then available to be deployed for use by another platform. The -unbind option removes the LUN from the VNX Storage group (if EMC Access Logix™ is enabled). The -unbind option permanently destroys the LUN and its contents. If this is the last LUN using a RAID group, then the RAID group will be deleted.

- **-info {<disk_name>|id=<disk_id>}**
  Displays information for a specific <disk_name> or <disk_id> such as size, type, and ACL.

- **-rename <old_name> <new_name>**
  Renames a disk to <new_name>.

*Note: If a VNX for block LUN uses the default name, renames it in the format VNX_<VNX-hostname>_<lun-id>_VNX-dvol-name.*
STORAGE SYSTEM OUTPUT

The number associated with the storage device is dependent on the attached storage system. VNX for block display a prefix of alphabetic characters before a set of integers, for example, FCNTR074200038-0019. Symmetrix storage systems display as a set of integers, for example, 002804000190-003C.

EXAMPLE #1

To list the disk table for a VNX for block, type:

```
$ nas_disk -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>inuse</th>
<th>sizeMB</th>
<th>storageID-devID</th>
<th>type</th>
<th>name</th>
<th>servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>y</td>
<td>11263</td>
<td>FCNTR074200038-0000</td>
<td>CLSTD</td>
<td>root_disk</td>
<td>1,2</td>
</tr>
<tr>
<td>2</td>
<td>y</td>
<td>11263</td>
<td>FCNTR074200038-0001</td>
<td>CLSTD</td>
<td>root_ldisk</td>
<td>1,2</td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td>2047</td>
<td>FCNTR074200038-0002</td>
<td>CLSTD</td>
<td>d3</td>
<td>1,2</td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td>2047</td>
<td>FCNTR074200038-0003</td>
<td>CLSTD</td>
<td>d4</td>
<td>1,2</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>2047</td>
<td>FCNTR074200038-0004</td>
<td>CLSTD</td>
<td>d5</td>
<td>1,2</td>
</tr>
<tr>
<td>6</td>
<td>y</td>
<td>32767</td>
<td>FCNTR074200038-0005</td>
<td>CLSTD</td>
<td>d6</td>
<td>1,2</td>
</tr>
<tr>
<td>7</td>
<td>n</td>
<td>1023</td>
<td>FCNTR074200038-0010</td>
<td>CLSTD</td>
<td>d7</td>
<td>1,2</td>
</tr>
<tr>
<td>8</td>
<td>n</td>
<td>1023</td>
<td>FCNTR074200038-0012</td>
<td>CLSTD</td>
<td>d8</td>
<td>1,2</td>
</tr>
<tr>
<td>9</td>
<td>n</td>
<td>1023</td>
<td>FCNTR074200038-0011</td>
<td>CLSTD</td>
<td>d9</td>
<td>1,2</td>
</tr>
<tr>
<td>10</td>
<td>n</td>
<td>1023</td>
<td>FCNTR074200038-0013</td>
<td>CLSTD</td>
<td>d10</td>
<td>1,2</td>
</tr>
<tr>
<td>11</td>
<td>n</td>
<td>10239</td>
<td>FCNTR074200038-0014</td>
<td>MIXED</td>
<td>d11</td>
<td>1,2</td>
</tr>
<tr>
<td>12</td>
<td>n</td>
<td>10239</td>
<td>FCNTR074200038-0015</td>
<td>MIXED</td>
<td>d12</td>
<td>1,2</td>
</tr>
<tr>
<td>13</td>
<td>n</td>
<td>10239</td>
<td>FCNTR074200038-0015</td>
<td>MIXED</td>
<td>d13</td>
<td>1,2</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the disk (assigned automatically).</td>
</tr>
<tr>
<td>inuse</td>
<td>Used by any type of volume or file system.</td>
</tr>
<tr>
<td>sizeMB</td>
<td>Total size of disk.</td>
</tr>
<tr>
<td>storageID-devID</td>
<td>ID of the storage system and device associated with the disk.</td>
</tr>
<tr>
<td>type</td>
<td>Type of disk contingent on the storage system attached; CLSTD, CLATA, CMSTD, CLFED, CMEFD, CMATA, MIXED (indicates tiers used in the pool contain multiple disk types), Performance, Capacity, Extreme_performance, Mirrored_mixed, Mirrored_performance, Mirrored_capacity, and Mirrored_extreme_performance are VNX disk types and STD, BCV, R1BCV, R2BCV, R1STD, R2STD, ATA, R1ATA, R2ATA, BCVA, R1BCA, R2BCA, EF0, BCVMIXED, R1MIXED, R2MIXED, R1BCVMIXED, and R2BCVMIXED are Symmetrix disk types.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the disk; 'dd' in a disk name indicates a remote disk.</td>
</tr>
<tr>
<td>servers</td>
<td>Servers that have access to this disk.</td>
</tr>
</tbody>
</table>

EXAMPLE #2

To list the disk table for the system with a Symmetrix storage system, type:

```
$ nas_disk -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>inuse</th>
<th>sizeMB</th>
<th>storageID-devID</th>
<th>type</th>
<th>name</th>
<th>servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>y</td>
<td>11263</td>
<td>FCNTR074200038-0000</td>
<td>CLSTD</td>
<td>root_disk</td>
<td>1,2</td>
</tr>
<tr>
<td>2</td>
<td>y</td>
<td>11263</td>
<td>FCNTR074200038-0001</td>
<td>CLSTD</td>
<td>root_ldisk</td>
<td>1,2</td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td>2047</td>
<td>FCNTR074200038-0002</td>
<td>CLSTD</td>
<td>d3</td>
<td>1,2</td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td>2047</td>
<td>FCNTR074200038-0003</td>
<td>CLSTD</td>
<td>d4</td>
<td>1,2</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>2047</td>
<td>FCNTR074200038-0004</td>
<td>CLSTD</td>
<td>d5</td>
<td>1,2</td>
</tr>
<tr>
<td>6</td>
<td>y</td>
<td>32767</td>
<td>FCNTR074200038-0005</td>
<td>CLSTD</td>
<td>d6</td>
<td>1,2</td>
</tr>
<tr>
<td>7</td>
<td>n</td>
<td>1023</td>
<td>FCNTR074200038-0010</td>
<td>CLSTD</td>
<td>d7</td>
<td>1,2</td>
</tr>
<tr>
<td>8</td>
<td>n</td>
<td>1023</td>
<td>FCNTR074200038-0012</td>
<td>CLSTD</td>
<td>d8</td>
<td>1,2</td>
</tr>
<tr>
<td>9</td>
<td>n</td>
<td>1023</td>
<td>FCNTR074200038-0011</td>
<td>CLSTD</td>
<td>d9</td>
<td>1,2</td>
</tr>
<tr>
<td>10</td>
<td>n</td>
<td>1023</td>
<td>FCNTR074200038-0013</td>
<td>CLSTD</td>
<td>d10</td>
<td>1,2</td>
</tr>
<tr>
<td>11</td>
<td>n</td>
<td>10239</td>
<td>FCNTR074200038-0014</td>
<td>MIXED</td>
<td>d11</td>
<td>1,2</td>
</tr>
<tr>
<td>12</td>
<td>n</td>
<td>10239</td>
<td>FCNTR074200038-0015</td>
<td>MIXED</td>
<td>d12</td>
<td>1,2</td>
</tr>
<tr>
<td>13</td>
<td>n</td>
<td>10239</td>
<td>FCNTR074200038-0015</td>
<td>MIXED</td>
<td>d13</td>
<td>1,2</td>
</tr>
</tbody>
</table>
The nas Commands

1 y 11507 000190100530-00FB STD root_disk 1,2,3,4,5,6,7,8
2 y 11507 000190100530-00FC STD root_ldisk 1,2,3,4,5,6,7,8
3 y 2076 000190100530-00FD STD d3 1,2,3,4,5,6,7,8
4 y 2076 000190100530-00FE STD d4 1,2,3,4,5,6,7,8
5 y 2076 000190100530-00FF STD d5 1,2,3,4,5,6,7,8
6 y 65536 000190100530-04D4 STD d6 1,2,3,4,5,6,7,8
7 n 28560 000190100530-0102 STD d7 1,2,3,4,5,6,7,8
8 n 28560 000190100530-0103 STD d8 1,2,3,4,5,6,7,8
9 n 28560 000190100530-0104 STD d9 1,2,3,4,5,6,7,8
10 n 28560 000190100530-0105 STD d10 1,2,3,4,5,6,7,8
11 n 28560 000190100530-0106 STD d11 1,2,3,4,5,6,7,8
12 n 28560 000190100530-0107 STD d12 1,2,3,4,5,6,7,8
13 n 28560 000190100530-0108 STD d13 1,2,3,4,5,6,7,8
14 n 28560 000190100530-0109 STD d14 1,2,3,4,5,6,7,8
15 n 28560 000190100530-010A STD d15 1,2,3,4,5,6,7,8
16 n 28560 000190100530-010B STD d16 1,2,3,4,5,6,7,8
17 n 28560 000190100530-010C STD d17 1,2,3,4,5,6,7,8
18 n 28560 000190100530-010D STD d18 1,2,3,4,5,6,7,8
19 n 28560 000190100530-010E STD d19 1,2,3,4,5,6,7,8
20 n 28560 000190100530-010F STD d20 1,2,3,4,5,6,7,8
21 n 28560 000190100530-0110 STD d21 1,2,3,4,5,6,7,8
22 n 28560 000190100530-0111 STD d22 1,2,3,4,5,6,7,8
23 n 28560 000190100530-0112 STD d23 1,2,3,4,5,6,7,8
24 n 28560 000190100530-0113 STD d24 1,2,3,4,5,6,7,8
[...]
155 n 28560 000190100530-0196 STD d155 1,2,3,4,5,6,7,8
156 n 28560 000190100530-0197 STD d156 1,2,3,4,5,6,7,8
157 n 28560 000190100530-0198 BCV rootd157 1,2,3,4,5,6,7,8
158 n 28560 000190100530-0199 BCV rootd158 1,2,3,4,5,6,7,8
159 n 28560 000190100530-019A BCV rootd159 1,2,3,4,5,6,7,8
160 n 28560 000190100530-019B BCV rootd160 1,2,3,4,5,6,7,8
161 n 28560 000190100530-019C BCV rootd161 1,2,3,4,5,6,7,8
162 n 28560 000190100530-019D BCV rootd162 1,2,3,4,5,6,7,8
163 n 28560 000190100530-019E BCV rootd163 1,2,3,4,5,6,7,8
164 n 28560 000190100530-019F BCV rootd164 1,2,3,4,5,6,7,8
165 n 28560 000190100530-01A0 BCV rootd165 1,2,3,4,5,6,7,8
166 n 28560 000190100530-01A1 BCV rootd166 1,2,3,4,5,6,7,8
167 n 28560 000190100530-01A2 BCV rootd167 1,2,3,4,5,6,7,8
168 n 28560 000190100530-01A3 BCV rootd168 1,2,3,4,5,6,7,8
169 n 28560 000190100530-01A4 BCV rootd169 1,2,3,4,5,6,7,8
170 n 28560 000190100530-01A5 BCV rootd170 1,2,3,4,5,6,7,8
171 n 28560 000190100530-01A6 BCV rootd171 1,2,3,4,5,6,7,8
172 n 28560 000190100530-01A7 BCV rootd172 1,2,3,4,5,6,7,8
173 n 28560 000190100530-01A8 BCV rootd173 1,2,3,4,5,6,7,8
174 n 28560 000190100530-01A9 BCV rootd174 1,2,3,4,5,6,7,8
175 n 28560 000190100530-01AA BCV rootd175 1,2,3,4,5,6,7,8
176 n 28560 000190100530-01AB BCV rootd176 1,2,3,4,5,6,7,8
177 n 28560 000190100530-01AC BCV rootd177 1,2,3,4,5,6,7,8
178 n 28560 000190100530-01AD BCV rootd178 1,2,3,4,5,6,7,8
179 n 28560 000190100530-01AE BCV rootd179 1,2,3,4,5,6,7,8
180 n 28560 000190100530-01AF BCV rootd180 1,2,3,4,5,6,7,8
181 n 28560 000190100530-01B0 BCV rootd181 1,2,3,4,5,6,7,8
The nas Commands

Example #1 provides a description of command outputs.

**Example #3**

To view information for disk d7 for a system with a VNX for block, type:

```
$ nas_disk -info d7
```

```
id        = 7
name      = d7
acl       = 0
in_use    = True
pool      = TP1
size (MB) = 273709
type      = Mixed
protection= RAID5(4+1)
stor_id   = FCNTR074200038
stor_dev  = 0012
volume_name = d7
storage_profiles = TP1
thin      = True
tiering_policy = Auto-tier
compressed= False
mirrored  = False
servers   = server_2,server_3,server_4,server_5
  server = server_2          addr=c0t1l2
  server = server_2          addr=c32t1l2
```
server = server_2          addr=c16t1l2
server = server_2          addr=c48t1l2
server = server_3          addr=c0t1l2
server = server_3          addr=c32t1l2
server = server_3          addr=c16t1l2
server = server_3          addr=c48t1l2
server = server_4          addr=c0t1l2
server = server_4          addr=c32t1l2
server = server_4          addr=c16t1l2
server = server_4          addr=c48t1l2
server = server_5          addr=c0t1l2
server = server_5          addr=c32t1l2
server = server_5          addr=c16t1l2
server = server_5          addr=c48t1l2

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the disk (assigned automatically).</td>
</tr>
<tr>
<td>name</td>
<td>Name of the disk.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value of the disk.</td>
</tr>
<tr>
<td>in_use</td>
<td>Used by any type of volume or file system.</td>
</tr>
<tr>
<td>pool</td>
<td>Name of the storage pool in use.</td>
</tr>
<tr>
<td>size (MB)</td>
<td>Total size of the disk.</td>
</tr>
<tr>
<td>type</td>
<td>Type of disk contingent on the storage system attached; VNX for block disk types are CLSTD, CLATA, CMSTD, CLEFD, CLSAS, CMEFD, CMATA, MIXED (indicates tiers used in the pool contain multiple disk types), Performance, Capacity, Extreme_performance, Mirrored_MIXED, Mirrored_performance, Mirrored_capacity, and Mirrored_extreme_performance.</td>
</tr>
<tr>
<td>protection</td>
<td>The type of disk protection that has been assigned.</td>
</tr>
<tr>
<td>stor_id</td>
<td>ID of the storage system associated with the disk.</td>
</tr>
<tr>
<td>stor_dev</td>
<td>ID of the device associated with the disk.</td>
</tr>
<tr>
<td>volume_name</td>
<td>Name of the volume residing on the disk.</td>
</tr>
<tr>
<td>storage_profiles</td>
<td>The storage profiles to which the disk belongs.</td>
</tr>
<tr>
<td>thin</td>
<td>Indicates whether the block storage system uses thin provisioning. Values are: True, False.</td>
</tr>
<tr>
<td>tiering_policy</td>
<td>Indicates the tiering policy in effect. If the initial tier and the tiering policy are the same, the values are: Auto-Tier, Highest Available Tier, Lowest Available Tier. If the initial tier and the tiering policy are not the same, the values are: Auto-Tier/No Data Movement, Highest Available Tier/No Data Movement, Lowest Available Tier/No Data Movement.</td>
</tr>
<tr>
<td>compressed</td>
<td>For VNX for block, indicates whether data is compressed. Values are: True, False, Mixed (indicates some of the LUNs, but not all, are compressed).</td>
</tr>
<tr>
<td>mirrored</td>
<td>Indicates whether the disk is mirrored.</td>
</tr>
<tr>
<td>servers</td>
<td>Lists the servers that have access to this disk.</td>
</tr>
<tr>
<td>addr</td>
<td>Path to storage system (SCSI address).</td>
</tr>
</tbody>
</table>

**EXAMPLE #4**  To view information for disk **d205** for the system with a Symmetrix storage system, type:
The `nas_disk` Command

```
$ nas_disk -info d205
```

id        = 205
name      = d205
acl       = 0
in_use    = True
pool      = SG0
size (MB) = 28560
type      = Mixed
protection= RAID1
symm_id   = 000190100530
symm_dev  = 0539
volume_name = d205
storage_profiles = SG0_000192601245
thin      = True
tiering_enabled = True
server_2,server_3,server_4,server_5,server_6,server_7,server_8,server_9
  server = server_2          addr=c0t14l0      FA=03A    FAport=0
  server = server_2          addr=c16t14l0     FA=04A    FAport=0
  server = server_3          addr=c0t14l0      FA=03A    FAport=0
  server = server_3          addr=c16t14l0     FA=04A    FAport=0
  server = server_4          addr=c0t14l0      FA=03A    FAport=0
  server = server_4          addr=c16t14l0     FA=04A    FAport=0
  server = server_5          addr=c0t14l0      FA=03A    FAport=0
  server = server_5          addr=c16t14l0     FA=04A    FAport=0
  server = server_6          addr=c0t14l0      FA=03A    FAport=0
  server = server_6          addr=c16t14l0     FA=04A    FAport=0
  server = server_7          addr=c0t14l0      FA=03A    FAport=0
  server = server_7          addr=c16t14l0     FA=04A    FAport=0
  server = server_8          addr=c0t14l0      FA=03A    FAport=0
  server = server_8          addr=c16t14l0     FA=04A    FAport=0
  server = server_9          addr=c0t14l0      FA=03A    FAport=0
  server = server_9          addr=c16t14l0     FA=04A    FAport=0

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the disk (assigned automatically).</td>
</tr>
<tr>
<td>name</td>
<td>Name of the disk.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value of the disk.</td>
</tr>
<tr>
<td>in_use</td>
<td>Used by any type of volume or file system.</td>
</tr>
<tr>
<td>pool</td>
<td>Name of the storage pool in use.</td>
</tr>
<tr>
<td>size (MB)</td>
<td>Total size of disk.</td>
</tr>
<tr>
<td>type</td>
<td>Type of disk contingent on the storage system attached; Symmetrix disk types are STD, BCV, R1BCV, R2BCV, R1STD, R2STD, ATA, R1ATA, R2ATA, BCVA, R1BCA, R2BCA, EFD, BCVMIXED, R1MIXED, R2MIXED, R1BCVMIXED, and R2BCVMIXED. If multiple disk volumes are used, the type is Mixed.</td>
</tr>
<tr>
<td>protection</td>
<td>The type of disk protection that has been assigned.</td>
</tr>
<tr>
<td>symm_id</td>
<td>ID of the Symmetrix storage system associated with the disk.</td>
</tr>
</tbody>
</table>
### The nas Commands

<table>
<thead>
<tr>
<th>symm_dev</th>
<th>ID of the Symmetrix device associated with the disk.</th>
</tr>
</thead>
<tbody>
<tr>
<td>volume_name</td>
<td>Name of the volume residing on the disk.</td>
</tr>
<tr>
<td>storage_profiles</td>
<td>The storage profiles to which the disk belongs.</td>
</tr>
<tr>
<td>thin</td>
<td>Indicates whether the storage system uses thin provisioning. Values are: True, False, Mixed.</td>
</tr>
<tr>
<td>tiering_enabled</td>
<td>Indicates whether a tiering policy is being used.</td>
</tr>
<tr>
<td>mirrored</td>
<td>Indicates whether the disk is mirrored.</td>
</tr>
<tr>
<td>servers</td>
<td>Lists the servers that have access to this disk.</td>
</tr>
<tr>
<td>addr</td>
<td>Path to storage system (SCSI address).</td>
</tr>
</tbody>
</table>

**EXAMPLE #5**

To rename a disk in the system with a VNX for block, type:

```
$ nas_disk -rename d7 disk7
```

```text
id        = 7
name      = disk7
acl       = 0
in_use    = True
size (MB) = 273709
type      = CLSTD
protection= RAID5(4+1)
stor_id    = FCNTR074200038
stor_dev  = 0012
volume_name = disk7
storage_profiles = clar_r5_performance
virtually_provisioned = False
mirrored  = False
servers   = server_2,server_3,server_4,server_5
  server = server_2          addr=c0t1l2
  server = server_2          addr=c32t1l2
  server = server_2          addr=c16t1l2
  server = server_2          addr=c48t1l2
  server = server_3          addr=c0t1l2
  server = server_3          addr=c32t1l2
  server = server_3          addr=c16t1l2
  server = server_3          addr=c48t1l2
  server = server_4          addr=c0t1l2
  server = server_4          addr=c32t1l2
  server = server_4          addr=c16t1l2
  server = server_4          addr=c48t1l2
  server = server_5          addr=c0t1l2
  server = server_5          addr=c32t1l2
  server = server_5          addr=c16t1l2
  server = server_5          addr=c48t1l2
```

**EXAMPLE #3** provides a description of command outputs.

**EXAMPLE #6**

To delete a disk entry from the disk table for the system with a VNX for block, type:
$ nas_disk -delete d24

id = 24
name = d24
acl = 0
in_use = False
size (MB) = 456202
type = CLATA
protection= RAID5(6+1)
stor_id = FCNTR074200038
stor_dev = 0023
storage_profiles = clarata_archive
virtually_provisioned = False
mirrored = False
servers = server_2,server_3,server_4,server_5

EXAMPLE #3 provides a description of command outputs.
nas_diskmark

Queries the storage system and lists the SCSI devices.

SYNOPSIS

nas_diskmark
  -mark {-all | <movername>} [-discovery {y|n}] [-monitor {y|n}]
  [-Force {y|n}]
  | -list {-all | <movername>}

DESCRIPTION

nas_diskmark queries the available storage system device and tape
device configuration; saves the device configuration into the Data
Movers database, and lists SCSI devices.

CAUTION

It is recommended that all Data Movers have the same device
configuration. When adding devices to the device table for a single
Data Mover only, certain actions such as standby failover are not
successful unless the standby Data Mover has the same disk device
configuration as the primary Data Mover.

The -all option executes the command for all Data Movers.

OPTIONS

-mark {-all | <movername>}
Queries SCSI devices and saves them into the device table database
on the Data Mover.

Modifies VNX for block LUN names to the
VNX_<VNX-hostname>_<lun-id>_<VNX-dvol-name> format, if the
LUNs use the default Unisphere name.

CAUTION

The time taken to complete this command may be lengthy,
dependent on the number and type of attached devices.

-discovery {y|n}
Enables or disables the storage discovery operation.

CAUTION

Disabling the -discovery option should be done only under the
direction of an EMC Customer Service Engineer.
The nas Commands

[-monitor {y|n}]
Displays the progress of the query and discovery operations.

[-Force {y|n}]
 Overrides the health check failures and changes the storage configuration.

CAUTION

Use the -Force option only when directed to do so, as high
availability can be lost when changing storage configuration.

-list {-all <movername>}
Lists the SCSI devices for the specified Data Mover or all Data Movers.

**EXAMPLE #1**
To query SCSI devices on server_2 and display the progress of the query operation, type:

```
$ nas_diskmark -mark server_2 -monitor y
```

Discovering storage (may take several minutes)

server_2:
chain 0 ..........
chain 16 ........
chain 32 ........
chain 48 ........
chain 96 ........
chain 112 ........

Verifying disk reachability
Verifying file system reachability
Verifying local domain
Verifying disk health
Verifying gate keepers
Verifying device group
done

**EXAMPLE #2**
To list the SCSI devices for server_2, type:

```
$ nas_diskmark -list server_2
```

server_2: chain 0 :
chain= 0, scsi-0
stor_id= HK190807090011  VNX_id= HK1908070900110032
tid/lun= 0/0 type= disk sz= 11263 val= 1 info= DGC RAID 5 0324320000000032NI
tid/lun= 0/1 type= disk sz= 11263 val= 2 info= DGC RAID 5 03243300010033NI
tid/lun= 0/2 type= disk sz= 2047 val= 3 info= DGC RAID 5 03243400020034NI
tid/lun= 0/3 type= disk sz= 2047 val= 4 info= DGC RAID 5 03243500030035NI
tid/lun= 0/4 type= disk sz= 2047 val= 5 info= DGC RAID 5 03243600040036NI
tid/lun= 0/5 type= disk sz= 32767 val= 6 info= DGC RAID 5 03243700050037NI
The *nas* Commands

```
tid/lun= 1/0 type= disk sz= 274811 val= 7 info= DGC RAID 5 0324440010044NI
tid/lun= 1/1 type= disk sz= 274811 val= -5 info= DGC RAID 5 03244500110045NI
tid/lun= 1/2 type= disk sz= 274811 val= 8 info= DGC RAID 5 03244600120046NI
tid/lun= 1/3 type= disk sz= 274811 val= -5 info= DGC RAID 5 03244700130047NI
tid/lun= 1/4 type= disk sz= 274811 val= 9 info= DGC RAID 5 03245600140056NI
tid/lun= 1/5 type= disk sz= 274811 val= -5 info= DGC RAID 5 03245700150057NI
tid/lun= 1/6 type= disk sz= 274811 val= 10 info= DGC RAID 5 03245800160058NI
...```


```
tid/lun= 1/10 type= disk sz= 274811 val= 97 info= DGC RAID 5 03245C001A005CNI
...```

```
server_2 : chain 1 :
  no drives on chain
server_2 : chain 2 :
  no drives on chain
...```

Note: This is a partial listing due to the length of the output.
The nas Commands

nas_emailuser

Manages email notifications for serious system events.

SYNOPSIS

nas_emailuser
  -info
  | -test
  | -modify
    [ -enabled {yes|no} ]
    [ -to <email_addr> [,....] ]
    [ -cc <email_addr> [,....] ]
    [ -email_server <email_server> ]
    [ -subject_prefix <email_subject> ]
    [ -from <email_addr> ]
  | -init

DESCRIPTION

nas_emailuser enables, configures, and tests email notifications for serious system events.

OPTIONS

-info
  Displays the configuration for email notifications.

-test
  Generates a test event that sends a test email notification to the email addresses configured in -to and -cc. The recipient email address must be configured prior to testing email notification.

  Note: After the -test option is run, all the configured recipients must be asked to confirm whether they received the test email with the correct system identification information.

-modify
  Modifies one or more of the following configuration parameters:
    [ -enabled {yes|no} ]
    Enables email notification if yes is specified. The recipient email address must be configured prior to enabling email notification. Disables email notification if no is specified.

    [ -to <email_addr> [,....] ]
    Configures one or more recipient email addresses. The email addresses are comma-separated, enclosed in single-quotes, and follow the mailbox@fully_qualified_domain_name format. For example, 'storage_admin@yourcompany.com', 'backup_admin@yourcompany.com'.
Refer the following email address format guidelines for configuring email addresses. An email address can contain:

- A maximum of 63 characters; the field can contain a maximum of 255 characters:
  - ASCII characters: a through z, A through Z, 0 through 9, ! # $ % & * + - / = ? ` ^ _ { | } are allowed; a period, if it is not the first or last character in the mailbox
  - Alphanumeric strings
- Single quotes, if they are escaped in the format:
  - - your\'email@yourcompany.com
  - -'first''email@yourcompany.com,second''email@yourcompany.com'

[-cc <email_addr> [, ...]]
Configures a list of carbon-copy recipients. The email addresses are comma-separated, enclosed in single-quotes, and follow the mailbox@fully_qualified_domain_name format. For example, ‘storage_admin@yourcompany.com’. For the email address character set and format guidelines, refer the -to option.

[-email_server <email_server>]
Configures the email server that accepts and routes the email notifications. <email_server> specifies an IP address or the fully qualified domain name, which can have 1 to 63 characters. The IP addresses 0.0.0.0 and 255.255.255.255 are not allowed.

[-subject_prefix <email_subject>]
Specifies the email subject prefix. The subject prefix for the email notification can be from 1 to 63 characters long, is enclosed in quotes, and should contain printable ASCII characters. You can customize the subject prefix for specific needs like email filtering. The default subject is "VNX Notification."

[-from <email_addr>]
Configures the sender’s email address. If the sender’s email address is not specified, a default email address of the format root@<hostname> is configured. The email address follows the mailbox@fully_qualified_domain_name format. For example, ‘storage_admin@yourcompany.com’. For the email address character set and format guidelines, refer the -to option.
The nas Commands

-init
Initializes the default state; displays a status message if the feature has already been initialized. The -init option must be used only when directed.

SEE ALSO Configuring Events and Notifications on VNX for File.

EXAMPLE #1 To configure email notifications used by the Marketing Department using the email server, mail.server.yourcompany.com, with sender as administrator, recipients as storage admin and backup admin, while carbon-copying the network admin and IT support, type:

```
$ nas_emailuser -modify -to
  'storage_admin@yourcompany.com','backup_admin@yourcompany.com'
  -cc 'network_admin@yourcompany.com','it_support@yourcompany.com'
  -email_server mail.server.yourcompany.com
  -subject_prefix "VNX Notification - Marketing Department"
  -from 'administrator@yourcompany.com'
```

OK

EXAMPLE #2 To display information on email notifications, type:

```
$ nas_emailuser -info
```

```
Service Enabled      = Yes
Recipient Address(es) = storage_admin@yourcompany.com,backup_admin@yourcompany.com
Carbon copy Address(es) = network_admin@yourcompany.com, it_support@yourcompany.com
Email Server         = mail.server.yourcompany.com
Subject Prefix       = VNX Notification - Marketing Department
Sender Address       = administrator@yourcompany.com
```

EXAMPLE #3 To test email notifications, type:

```
$ nas_emailuser -test
```

OK

EXAMPLE #4 To disable email notification, type:

```
$ nas_emailuser -modify -enabled no
```

OK
The nas Commands

**nas_environment**

Reports the inlet air temperatures and input power to the user.

**SYNOPSIS**

`nas_environment -info`

```
{  
  -system [-present|-average]
  -array [-present|-average]
  -shelf {<shelf_id>|<-all>} [-intemp [f|c]|-power]
  [-present|-average]
  -battery [a|b] [-present|-average]
  -spe [-intemp [f|c]|-power] [-present|-average]
  -all
}
```

**DESCRIPTION**

`nas_environment -info` displays the inlet air temperatures of the Data Mover enclosures, disk array enclosures, the input power of the Data Mover enclosures, disk array enclosures, and standby power supply through the CLI and Unisphere GUI.

**OPTIONS**

**-system**
Displays the present or average input power information of the system, which includes file and block on VNX systems, and file only on gateway systems.

**-present**
Displays the current value, which is a sum of the present input power for all supported systems. The current value is computed as the 30 second average of the power consumption sampled every three seconds.

**-average**
Displays the average value. It requires an hour to calculate the correct value. N/A is displayed if there is less than one hour worth of data. The average value is computed as the 60 minute rolling average of the present power consumption values.

**-dme**
Displays the present or average inlet air temperature and input power information on a specified Data Mover enclosure. If a specific enclosure_id is not specified, all Data Mover enclosure information is displayed.
enclosure_id
Specifies a Data Mover enclosure id on which to display information.

-intemp [f|c]
Displays the inlet air temperature information. The f flag indicates Fahrenheit. The default value or c flag indicates Celsius.

-power
Displays the input power information.

-present
Displays the current value.

-average
Displays the average value. It requires an hour to calculate the correct value. N/A is displayed if there is less than one hour worth of data.

-array
Displays the present or average input power information on the array.

-present
Displays the current value.

-average
Displays the average value. It requires an hour to calculate the correct value. N/A is displayed if there is less than one hour worth of data.

-shelf
Allows to input a value for a selected enclosure. It displays the present and average inlet air temperature and input power information on a specified disk array enclosure. If a specific enclosure_id is not specified, all disk array enclosure information is displayed.

<shelf_id>
Specifies a disk array enclosure id on which to display information.

-power
Displays the input power information.

-present
Displays the current value.
The nas Commands

-**average**
Displays the average value. It requires an hour to calculate the correct value. N/A is displayed if there is less than one hour worth of data.

-**battery**
Displays the present and average input power information on a specified standby power supply. If no -a or -b is specified, then the information is displayed on both standby power supplies.

-**a**
Specifies a standby power supply A on which to display information.

-**b**
Specifies a standby power supply B on which to display information.

-**present**
Displays the current value.

-**average**
Displays the average value. It requires an hour to calculate the correct value. N/A is displayed if there is not one hour worth of data.

-**spe**
Displays the present and average input power information on a specified standby power supply.

-**intemp [f|c]**
Displays the inlet air temperature information. The f flag indicates Fahrenheit. The default value or c flag indicates Celsius.

-**power**
Displays the input power information.

-**present**
Displays the current value.

-**average**
Displays the average value. It requires an hour to calculate the correct value. N/A is displayed if there is less than one hour worth of data.

-**all**
Displays the following:

- System input power
◆ Data Mover enclosure inlet air temperatures and input power
◆ Array input power
◆ Disk array enclosure inlet air temperatures and input power
◆ Storage processor enclosure inlet air temperatures and input power
◆ Standby power supply input power

EXAMPLE #1  To view the present and average input power information for file and block on systems or file only on gateway system, type:

```
$ nas_environment -info -system
```

```
System = Celerra ns 600 APM 000237001650000
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
```

EXAMPLE #2  To view the average inlet air temperature on Data Mover enclosure 1 in degrees Farenheit, type:

```
$ nas_environment -info -dme 1 -intemp f -average
```

```
Component = DME 0 Data Mover 1
Temperature Status = OK
Rolling average (degrees) = 53F
Temperature Status = OK
Rolling average (degrees) = 53F
```

EXAMPLE #3  To view the average inlet air temperature on Data Mover enclosure 1 in degrees Celsius, type:

```
$ nas_environment -info -dme1 -intemp c -average
```

```
Data Mover Enclosure 1
Status: Valid
Inlet Air Temperature
Rolling average (degrees Celsius): 11.3
```

EXAMPLE #4  To view the present system information, type:

```
$ nas_environment -info -system -present
```

```
System = Celerra ns 600 APM 000237001650000
Power Status = OK
Present (watts) = 150
```

EXAMPLE #5  To view the information array (input power and inlet temperature), type:

```
$ nas_environment -info -array
```
EXAMPLE #6  To view the present and average inlet air temperature on all shelves, type:

```
$ nnas_environment -info -shelf -all
```

```
Component = Shelf 0/0 Shelf 0/0
Power Status = OK
Present (watts) = 150
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
Temperature Status = OK
Present (degrees) = 12C
Rolling average (degrees) = 11C

Component = Shelf 0/1 Shelf 0/1
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
Temperature Status = OK
Present (degrees) = 12C
Rolling average (degrees) = 11C

Component = Shelf 1/0 Shelf 1/0
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
Temperature Status = OK
Present (degrees) = 12C
Rolling average (degrees) = 11C

Component = Shelf 1/1 Shelf 1/1
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
Temperature Status = OK
Present (degrees) = 12C
Rolling average (degrees) = 11C
```

EXAMPLE #7  To view the present and average inlet air temperature on shelf 1 enclosure 1, type:

```
$ nas_environment -info -shelf 1/1 -average
```

```
Component = Shelf 1/1 Shelf 1/1
Power Status = OK
Rolling average (watts) = 150
Temperature Status = OK
Rolling average (degrees) = 11C
EXAMPLE #8  To view the present and average inlet air temperature on all spes, type:

$ nas_environment -info -spe

Component = SPE 0 SPE 0
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
Temperature Status = OK
Present (degrees) = 12C
Rolling average (degrees) = 11C

EXAMPLE #9  To view the present information for all batteries, type:

$ nas_environment -info -battery

Component = Shelf 0/0 SP A
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
Component = Shelf 0/0 SP B
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150

EXAMPLE #10  To view the present information for all batteries, type:

$ nas_environment -info -battery a -average

Component = Shelf 0/0 SP A
Power Status = OK
Rolling average (watts) = 150

EXAMPLE #11  To view all the components, type:

$ nas_environment -info -all

Component = Shelf 0/0 SP A
Power Status = OK
Rolling average (watts) = 150

EXAMPLE #12  To view the present and average inlet air temperature of the dme, shelf, spe components, type:

$ nas_environment -info -all

Component = Celerra ns600 APM000237001650000
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
Component = DME 0 Data Mover 0
Power Status = OK
Present (watts) = 200
Rolling average (watts) = 333
Temperature Status = OK
Present (degrees) = 12C
Rolling Average (degrees) = 11C
Component = DME 0 Data Mover 1
Power Status = Error 13690667102: Not Present
Present (watts) = N/A
Rolling average (watts) = N/A
Temperature Status = Error 13690667102: Unsupported
Present (watts) = N/A
Rolling average (watts) = N/A

Component = DME 0 Data Mover 2
Power Status = Error 13690667102: Uninitialized
Present (watts) = 150
Average (watts) = N/A
Temperature Status = Error 13690667102: Uninitialized
Present (degrees) = 12C
Average (degrees) = N/A

Component = DME 0 Data Mover 3
Power Status = Error 13690667102: Failed
Present (watts) = 150
Average (watts) = N/A
Temperature Status = Error 13690667102: Failed
Present (degrees) = 12C
Average (degrees) = N/A

Component = Shelf 0/0
Power Status = OK
Present (watts): 150
Rolling average (watts): 150
Present (watts): 150
Rolling average (watts): 150
Temperature Status = OK
Present (degrees) = 12C
Rolling average (degrees) = 11C
Component = Shelf 0/1
Power Status = Error 13690667102: Invalid
Present (watts) = N/A
Rolling average (watts) = N/A
Temperature Status = Error 13690667102: Invalid
Present (degrees) = N/A
Rolling average (degrees) = N/A

Component = Shelf 1/0
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
Temperature Status = OK
Present (degrees) = 12C
Rolling average (degrees) = 11C

Component = Shelf 1/1
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
Temperature Status = OK
Present (degrees) = 12C
Rolling average (degrees) = 11C

Component = CLARiiON CX600 APM00023700165
Power Status = OK
Present (watts) = 230
Rolling average (watts) = 245

Component = SPE 0 SPE 0
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150
Temperature Status = OK
Present (degrees) = 12C
Rolling average (degrees) = 11C

Component = Shelf 0/0 SP A
Power Status = OK
Present (watts) = 150
Power Status = OK
Present (watts) = 150
Rolling average (watts) = 150

Component = Shelf 0/0 SP B
Power Status = Error 13690667102: Not Present
Present (watts) = N/A
Rolling average (watts) = N/A
**nas_event**

Provides a user interface to system-wide events.

**SYNOPSIS**

```
 nas_event
     -Load {-info|<file_name>}
     | -Unload <file_name>
     | -list
     -action {-info|{trap|logfile|mail|callhome|exec|udprpc|tcprpc|terminate}
               [-component {<component>|<id>}] [-facility {<facility>
               | <id>}] [-severity {<severity>|<id>}]} [-id]}
     | -component {-info|{<component>|<id>}} [-facility {<facility>|<id>}] [-id]
     | -severity {-info|<severity>|<id>}} [-component {<component>|<id>}
               [-facility {<facility>|<id>}] [-id]}
     | -keyword <keyword> [-component {<component>|<id>}
               [-facility{<facility>|<id>]} [-severity {<severity>|<id>}]][-id]
```

**DESCRIPTION**

`nas_event` provides a user interface to system-wide events within the VNX. The VNX includes a default event configuration file that contains a mapping of facilities that generate events, and the associated actions triggered by those events.

To list the default configuration files, type:

```
$ nas_event -Load -info
```

Using a text editor, a new event configuration file can be created and loaded into the database to add an event, or change an action.

VNX facilities generate events that trigger specific actions. An event consists of:

- An assigned ID for the event and the facility issuing the event
- The high water mark for the severity of the event
- A description of the event
- The system-defined action to take when the event occurs

**CAUTION**

The callhome events in the system are carefully reviewed and configured to provide the right level of support. Do not add, delete, or modify lines that specify the callhome action in the event configuration files in the /nas/sys directory. User-defined event configuration files should not use the callhome action.
OPTIONS

-Load {-info|<file_name>}
Loads the event configuration file `<file_name>` into the system. The
-info option displays the currently loaded configuration files.

-Unload <file_name>
Unloads the configuration file `<file_name>` from the system.

CAUTION

The /nas/sys/nas_eventlog.cfg configuration file must not be deleted,
as this can lead to data loss. Unloading or modifying configuration
files that are loaded by default can affect logging, alerts,
notifications, and system operations.

-list
The -list option displays components and facilities that generate
events, and the actions that are taken as a result. You can search for an
event, facility, or action by using a keyword. Component, facility, and
severity can be specified by either the text name or ID. The output is
displayed with parameter names in the form $(paraname,
typeIndicator, fmtStr).

-action
{-info {trap|logfile|mail|callhome|exec|udprpc|tcprpc|
terminate}}
With the -info option, lists all the possible actions associated with
events. If one of the actions trap, logfile, mail, callhome, exec,
udprpc, tcprpc, or terminate is specified, lists the possible events
that trigger the specified action. These events are categorized by
component and facility:

[-component {<component>|<id>}] [-facility
{<facility> |<id>}]}
Lists the possible events in the specified component that
trigger the given action. If facility is specified, lists the events
in the specified component and facility that trigger the given
action.

[-severity {<severity>|<id>}]}
Lists the possible events with the specified severity that
trigger the given action.

[-id]
Lists the output with the MessageID number in addition to
BaseID, Severity, and Brief_Description.
**The nas Commands**

- **component** [-info | <component> | <id>] [-facility|<facility> | <id>]
  With the **-info** option, lists the ids and names of all the components. If the component is specified, lists the ids and names of all the facilities under that component. Specifying facility lists the events that can be generated by the specified facility and component.

  [-id]
  Lists the output with the message ID number in addition to BaseID and Brief_Description.

- **severity** {-info|<severity>|<id>}
  With the **-info** option, lists the severity levels. If severity is specified, lists the events with the specified severity level.

  [-component {<component>|<id>} [-facility|<facility> | <id>]]
  Lists the events filtered by the given severity and component. If facility is specified lists the events further filtered by the given facility.

  [-id]
  Lists the output with the MessageID number in addition to BaseID, Severity, and Brief_Description.

---

**Note:** To receive email notifications sent to multiple recipients, specify the email addresses within the quotes and separate them with a comma.

---

- **keyword** <keyword>
  Lists all events that match the specified keyword.

  [-component {<component>|<id>}]
  [-facility{<facility> | <id>}]
  Lists events filtered by the specified keyword and component. If facility is specified, lists the events further filtered by the given facility.

  [-severity {<severity>|<id>}]]
  Lists events filtered by the specified severity.

  [-id]
  Lists the output with the MessageID number in addition to BaseID, Severity, and Brief_Description.

---

**SEE ALSO**  Configuring Events and Notifications on VNX for File.
EXAMPLE #1  After using a text editor to create an event configuration file, to load the new configuration file into the NAS database, type:

```
$ nas_event -Load /nas/site/new_eventlog.cfg
```

EventLog : will load /nas/site/new_eventlog.cfg...done

EXAMPLE #2  To verify that the configuration file was loaded, type:

```
$ nas_event -Load -info
```

Loaded config. files:
1:  /nas/sys/nas_eventlog.cfg
2:  /nas/http/webui/etc/web_client_eventlog.cfg
3:  /nas/site/new_eventlog.cfg

EXAMPLE #3  To list actions, type:

```
$ nas_event -list -action -info
```

action
mail
terminate
trap
exec
callhome
logfile

EXAMPLE #4  To list the events that trigger the mail action, type:

```
$ nas_event -list -action mail
```

CS_PLATFORM(6)
|--> EventLog(130)

<table>
<thead>
<tr>
<th>BaseID</th>
<th>Severity</th>
<th>Brief_Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>EMERGENCY(0)</td>
<td>${text,8,%s}</td>
</tr>
<tr>
<td>51</td>
<td>ALERT(1)</td>
<td>${text,8,%s}</td>
</tr>
<tr>
<td>52</td>
<td>CRITICAL(2)</td>
<td>${text,8,%s}</td>
</tr>
</tbody>
</table>

EXAMPLE #5  To list the components, type:

```
$ nas_event -list -component -info
```

<table>
<thead>
<tr>
<th>Id</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DART</td>
</tr>
<tr>
<td>2</td>
<td>CS_CORE</td>
</tr>
<tr>
<td>5</td>
<td>XML_API</td>
</tr>
<tr>
<td>6</td>
<td>CS_PLATFORM</td>
</tr>
</tbody>
</table>

EXAMPLE #6  To list the facilities under the component DART, type:

```
$ nas_event -list -component DART
```

DART(1)
<table>
<thead>
<tr>
<th>Id</th>
<th>Facility</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>ADMIN</td>
</tr>
<tr>
<td>26</td>
<td>CAM</td>
</tr>
<tr>
<td>27</td>
<td>CFS</td>
</tr>
<tr>
<td>36</td>
<td>DRIVERS</td>
</tr>
</tbody>
</table>
EXAMPLE #7

To list the events generated by DART in the facility with the ID 146, type:

$ nas_event -list -component DART -facility 146

<table>
<thead>
<tr>
<th>BaseID</th>
<th>Severity</th>
<th>Brief_Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NOTICE(5)</td>
<td>CEPP server ${ipaddr,8,%s} of pool ${pool,8,%s} is ${status,8,%s}. Vendor ${vendor,8,%s}, ntStatus 0x${ntstatus,2,%x}.</td>
</tr>
<tr>
<td>2</td>
<td>ERROR(3)</td>
<td>Error on CEPP server ${ipaddr,8,%s} of pool ${pool,8,%s}: ${status,8,%s}. Vendor ${vendor,8,%s}, ntStatus 0x${ntstatus,2,%x}.</td>
</tr>
<tr>
<td>3</td>
<td>NOTICE(5)</td>
<td>The CEPP facility is started.</td>
</tr>
<tr>
<td>4</td>
<td>NOTICE(5)</td>
<td>The CEPP facility is stopped.</td>
</tr>
</tbody>
</table>
**EXAMPLE #8**  
To list events with severity 4 generated by component CS_CORE and facility DBMS, and to display the MessageID in the output, type:

```
$ nas_event -list -severity 4 -component CS_CORE -facility DBMS -id
```

<table>
<thead>
<tr>
<th>MessageID</th>
<th>BaseID</th>
<th>Brief_Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>86444212226</td>
<td>2</td>
<td>Db: Compact${compact_option,8,%s}: ${db_name,8,%s}: Failed: ${db_status,8,%s}.</td>
</tr>
<tr>
<td>86444212227</td>
<td>3</td>
<td>Db Env: ${db_env,8,%s}: Log Remove: Failed: ${db_status,8,%s}.</td>
</tr>
</tbody>
</table>

**EXAMPLE #9**  
To list events filtered by the keyword freeblocks, type:

```
$ nas_event -list -keyword freeblocks
```

<table>
<thead>
<tr>
<th>BaseID</th>
<th>Severity</th>
<th>Brief_Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>CRITICAL(2)</td>
<td>Only ${freeblocks,3,%llu} free blocks in the root file system (fsid ${fsid,2,%u}) of the VDM ${vdm,8,%s}.</td>
</tr>
<tr>
<td>3</td>
<td>ALERT(1)</td>
<td>The root file system (fsid ${fsid,2,%u}) of the VDM ${vdm,8,%s} is full. There are only ${freeblocks,3,%llu} free blocks.</td>
</tr>
</tbody>
</table>

**EXAMPLE #10**  
To list events with the keyword data generated in DART with the severity level 6, type:

```
$ nas_event -list -keyword data -component DART -severity 6
```

<table>
<thead>
<tr>
<th>BaseID</th>
<th>Severity</th>
<th>Brief_Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INFO(6)</td>
<td>The Usermapper database has been created.</td>
</tr>
<tr>
<td>4</td>
<td>INFO(6)</td>
<td>The Usermapper database has been destroyed.</td>
</tr>
<tr>
<td>8</td>
<td>INFO(6)</td>
<td>The migration of the Usermapper database to the VNX version 5.6 format has started.</td>
</tr>
<tr>
<td>9</td>
<td>INFO(6)</td>
<td>The Usermapper database has been successfully migrated.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BaseID</th>
<th>Severity</th>
<th>Brief_Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INFO(6)</td>
<td>The migration of the secmap database to the VNX version 5.6 format has started.</td>
</tr>
<tr>
<td>2</td>
<td>INFO(6)</td>
<td>The secmap database has been successfully migrated.</td>
</tr>
</tbody>
</table>

**EXAMPLE #11**  
To unload the event configuration file, type:

```
$ nas_event -Unload /nas/site/new_eventlog.cfg
```

EventLog : will unload /nas/site/new_eventlog.cfg... done
EXAMPLE #12  To receive email notifications that are sent to multiple recipients, add
the following line to your /nas/sys/eventlog.cfg file:

disposition severity=0-3, mail "nasadmin@nasdocs.emc.com,
helpdesk@nasdocs.emc.com"
The nas Commands

nas_fs

Manages local file systems for the VNX.

SYNOPSIS

nas_fs
- list [-all]
- delete <fs_name> [-option <options>] [-Force]
- info [-size] { -all | <fs_name>|id=<fs_id> } [-option <options>]
- rename <old_name> <new_name> [-Force]
- size <fs_name>
- acl <acl_value> <fs_name>
- translate <fs_name> -access_policy start
  - to { MIXED } - from { NT | NATIVE | UNIX | SECURE }
- translate <fs_name> -access_policy status
- xtend <fs_name> { <volume_name> | size=<integer>[T|G|M]% } [ pool=<pool> ]
  [ storage=<system_name> ] [-option <options>]
- modify <fs_name> -auto_extend { no | yes } [-thin { no | yes }]
  [-hwm <50-99>%][ -max_size <integer>[T|G|M] ]
- Type <type> <fs_name> -Force
  [- name <name> ][ -type <type> ] -create <volume_name>
    [ samesize=<fs_name>[:cel=<cel_name>]] [ worm={enterprise|compliance|off} ]
    [-option <options>]
  [- name <name> ][ -type <type> ] -create { size=<integer>[T|G|M]
    | samesize=<fs_name>[:cel=<cel_name>]
    pool=<pool> [ storage=<system_name> ] [ worm={enterprise|compliance|off} ]
    [-auto_extend { no | yes } | -thin { no | yes }]
    [-hwm <50-99>%][ -max_size <integer>[T|G|M] ]]
    [-option <options>]
- name <name> -type nmfs -create

DESCRIPTION

nas_fs creates, deletes, extends, and lists file systems. nas_fs displays the attributes of a file system, translates the access policy, enables, Automatic File System Extension and thin provisioning capabilities, and manages access control level values.

OPTIONS

- list [-all]
Displays a list of file systems and their attributes such as the name, ID, usage, type, access control level setting, the residing volume, and the server. The -all option displays all file systems including system-generated internal file systems. For example, Replicator internal checkpoints.
The ID is an integer and is assigned automatically, but not always sequentially, depending on ID availability. The name of a file system might be truncated if it is more than 19 characters. To display the full file system name, use the `-info` option with a file system ID.

File system types are:

- `1=uxfs` (default)
- `5=rawfs` (unformatted file system)
- `6=mirrorfs` (mirrored file system)
- `7=ckpt` (checkpoint)
- `8=mgfs` (migration file system)
- `100=group file system`
- `102=nmfs` (nested mount file system)

File system types `uxfs`, `mgfs`, `nmfs`, and `rawfs` are created by using `nas_fs`. Other file system types are created either automatically or with their specific commands.

**-delete <fs_name>**

Deletes the file system specified by file system name or ID. A file system cannot be deleted when it is mounted or part of a group.

```bash
[-option <options>]
```

Specifies the following comma-separated options:

- `volume`
  Deletes the file system’s underlying volume structure.

**Note:** If a checkpoint is created with a volume that has been specified by size, the underlying volume is deleted when the checkpoint is deleted. If a file system, using a storage pool is deleted, the underlying volume structure is also deleted.

```
[-Force]
```

Forces the deletion of a file system with SnapSure checkpoints known as the PFS, when a task scheduler such as an automated scheduler for SnapSure is running or is enabled.

**-info [-size] {<all>|<fs_name>|id=<fs_id>}**

Displays the attributes of a single file system, or all file systems, including the configuration of associated disks and replication sessions that are stopped or configured on the file system. If a file system is mounted, data is reported from the NAS database and the
Data Mover. If a file system is unmounted, data is reported from the NAS database only.

The **-size** option also displays the total size of the file system and the block count in megabytes.

```
[-option <options>]
```

Specifies the following comma-separated options:

- **mpd**
  Displays the current directory type and translation status for the specified Multi-Protocol Directory (MPD) file system.

- **-rename <old_name> <new_name>**
  Changes the file system name from <old_name> to <new_name>.
  
  [**-Force**]
  Forces the rename of the file system with SnapSure checkpoints known as the PFS.

- **-size <fs_name>**
  Displays the total size of the file system and the block count in megabytes. The total size of a file system relates to the mounted or unmounted status of a file system.

- **-acl <acl_value> <fs_name>**
  Sets an access control level value that defines the owner of a file system, and the level of access allowed for users and groups defined in the access control level table. The **nas_acl** command provides more information.

- **-translate <fs_name> -access_policy start -to {MIXED} -from {NT | NATIVE | UNIX | SECURE}**
  Synchronizes the UNIX and Windows permissions on the specified file system. Prior to executing the **-translate** option by using **server_mount**, mount the specified file system with the MIXED access-checking policy. The <fs_name> must be a **uxfs** file system type mounted as read/write.

  The policy specified in the **-from** option instructs the VNX about which operating system (UNIX or Windows) to derive permissions from, when migrating to the **MIXED** or **MIXED_COMPAT** access-checking policy (set with **server_mount**). For example, if you type UNIX in the **-from** option, all ACLs are regenerated from the UNIX mode bits. The policy typed in the **-from** option does not relate to the policy previously used by the file system object.
-translate <fs_name> -access_policy status
Prints the status of the access policy translation for the specified file system.

-xtend <fs_name> <volume_name>
Adds the specified volume to the mounted file system.

-xtend <fs_name> size=<integer>[T|G|M%]
Adds the volume as specified by its desired size to the file system or checkpoint. Type an integer within the range of 1 to 1024, then specify T for terabytes, G for gigabytes (default), M for megabytes, or type an integer representing the percentage of a file system’s size followed by the percent sign. The extended volume added to the file system by the system will have a size equal to or greater than the total size specified.

CAUTION
When executing this command, extends should be performed incrementally by using like volumes to reduce time consumption.

[pool=<pool>]
Applies the specified storage pool rule set to the volume that has been added to the mounted file system.

Note: The storage pool is a rule set that contains automatically created volumes and defines the type of disk volumes used and how they are aggregated.

[storage=<system_name>]
Specifies the storage system on which the checkpoint resides. If a storage system is not specified, the default storage system is the one on which the file system resides. If the file system spans multiple storage systems, the default is to use all the storage systems on which the file system resides. Use nas_storage -list to obtain attached storage system names.

[-option <options>]
Specifies the following comma-separated options:

slice={y|n}
Specifies whether the disk volumes used by the file system may be shared with other file systems that use a slice. The slice=y option allows the file system to share disk volumes with other file
The nas Commands

systems. The **slice=n** option gives the new file system exclusive access to the disk volumes it uses, and is relevant when using TimeFinder/FS.

When **symm_std**, **symm_std_rdf_src**, **symm_ata**, **symm_ata_rdf_src**, **symm_ata_rdf_tgt**, and **symm_std_rdf_tgt** pools are specified, the default is not to slice the volumes, which is overridden with **slice=y**. For **symm_efd**, the default is **slice=y**, because TimeFinder/FS is not supported with EFD disk types.

When **clar_r1**, **clar_r5_performance**, **clar_r5_economy**, **clar_r6**, **clarata_r3**, **clarata_r6**, **clarata_r10**, **clarata_archive**, **cm_r1**, **cm_r5_performance**, **cm_r5_economy**, **cm_r6**, **cmata_r3**, **cmata_archive**, **cmata_r6**, **cmata_r10**, **clarsas_archive**, **clarsas_r6**, **clarsas_r10**, **clarefd_r5**, **clarefd_r10**, **cmsas_archive**, **cmsas_r6**, **cmsas_r10**, and **cmefd_r5** pools are specified, the default for standard AVM pools is to slice the volumes (**slice=y**), which is overridden by using **slice=n**. The default for mapped pools is not to slice the volumes (**slice=n**). Use **nas_pool** to change the default slice option.

**src_only**
Extends only the source side for the PFS when replication is running. When extending the PFS while replication is running, by default, both the source and the destination sides are extended.

```
-modify <fs_name> -auto_extend {no|yes} [-thin {yes|no}] [-hwm <50-99>][[-max_size <integer> [T|G|M]]]
```

For an AVM file system, turns Automatic File System Extension and thin provisioning on or off, and sets a high water mark and maximum size for the file system. When file system extension is turned on, the file system is automatically extended up to the maximum size specified when the high water mark is reached. The default for **-auto_extend** is **no**.

Thin provisioning reports the maximum file system size to the CIFS, NFS, and FTP users, even if the actual size of the file system is smaller. If thin provisioning is disabled, the true file system size and maximum file system sizes are reported to the system administrator. Thin provisioning requires that a maximum file system size also be set. If a file system is created by using a virtual storage pool, the **-thin** option cannot be enabled because file-based thin provisioning and block-based thin provisioning cannot coexist on a file system.
Automatic File System Extension cannot be used for any file system that is part of an RDF configuration (for example, file systems on Data Movers configured with an RDF standby). Do not use the `nas_fs` command with the `-auto_extend` option for file systems associated with RDF configurations.

```
[-hwm <50-99>]
```

Specifies the size threshold that must be reached before the file system is automatically extended. Type an integer between the range of 50 and 99 to represent the percentage of file system usage. The default is 90.

```
[-max_size <integer> [T|G|M]]
```

Sets the maximum file system size to which a file system can be extended. Type an integer and specify T for terabytes, G for gigabytes (default), or M for megabytes. If the `-max_size` option is not specified, then it defaults to the maximum limit of the file system size which is 16 terabytes.

```
-Type <type> <fs_name> -Force
```

Changes the file system type from the one of `<fs_name>` to the new specified `<type>`.

---

**CAUTION**

The conversion from rawfs to uxfs will fail with "Error 3105: invalid filesystem specified" because a uxfs is not available on the rawfs. However, if the user initially creates a rawfs, and restores an NDMP volume backup on the rawfs, then the conversion from the rawfs to a uxfs will be successful.

---

**CREATING A FILE SYSTEM**

File systems can be created by using:

- A volume specified by name
- A volume specified by its size and desired storage pool
- An existing local or remote file system with the `samesize` option
- An existing local or remote file system with the `samesize` option and by using space from the available storage pool

```
[-name <name>][-type <type>] -create <volume_name>
```

Creates a file system on the specified volume and assigns an optional name to the file system. If a name is not specified, one is assigned automatically.

A file system name cannot:
The nas Commands

- Begin with a dash (-)
- Be comprised entirely of integers
- Be a single integer
- Contain the word root or contain a colon (:)

The **-type** option assigns the file system type to be **uxfs** (default), **mgfs**, or **rawfs**.

\[\text{samesize}=<\text{fs\_name}>[:\text{cel}=<\text{cel\_name}>]\]

Specifies that the new file system must be created with the same size as the specified local or remote file system. When using the **samesize** option by using the options described below, the **slice** must be set to **y**.

\[\text{worm}=(\text{enterprise}|\text{compliance}|\text{off})\]

Enables storage capability on a new file system. The **worm** option can only be specified when creating a new file system; existing file systems cannot be converted. After a file system is enabled, it is persistently marked as such until the time when it is deleted.

**Note:** The compliance file system cannot be deleted if it has protected files.

**CAUTION**

The enterprise version of this feature is intended for self-regulated archiving. The administrator is considered a trusted user and the capability does not protect the archived data from the administrator's actions. If the administrator attempts to delete the file system, the file system issues a warning message and prompts the administrator to confirm the operation. This version is not intended for high-end compliance applications such as pharmaceuticals, aerospace, or finance.

\[-\text{option} <\text{options}>\]

Specifies the following comma-separated options:

**nbpi**=<number>
The number of bytes per inode block. The default is 8192 bytes.

**mover**=<movername>
Assigns an optional Data Mover to build a file system. If no Data Mover is assigned, the system will automatically pick the first available Data Mover to build the file system.
The nas Commands

slice={y|n}
Specifies whether the disk volumes used by the new file system may be shared with other file systems by using a slice. The slice=y option allows the file system to share disk volumes with other file systems. The slice=n option ensures that the new file system has exclusive access to the disk volumes it uses, and is relevant when using TimeFinder/FS.

When symm_std, symm_std_rdf_src, symm_ata, symm_ata_rdf_src, symm_ata_rdf_tgt, and symm_std_rdf_tgt pools are specified, the default is not to slice the volumes. When slice=y is specified, it overrides the default. For symm_efd, the default is slice=y, because TimeFinder/FS is not supported with EFD disk types.

When clar_r1, clar_r5_performance, clar_r5_economy, clar_r6, clarata_r3, clarata_r6, clarata_r10, clarata_archive, cm_r1, cm_r5_performance, cm_r5_economy, cm_r6, cmata_r3, cmata_archive, cmata_r6, cmata_r10, clarsas_archive, clarsas_r6, clarsas_r10, Clarefd_r5, Clarefd_r10, Cmsas_archive, Cmsas_r6, Cmsas_r10, and Cmefd_r5 pools are specified, the default for standard AVM pools is to slice the volumes (slice=y), which is overridden by using slice=n. The default for mapped pools is not to slice the volumes (slice=n). Use nas_pool to change the default slice option.

id=<desired_id>
Specifies the ID to be assigned to the new file system. If a file system already exists with the specified ID, a warning is displayed indicating that the ID is not available, and the new file system is assigned the next available ID.

[-name <name>] [-type <type>] -create {size=<integer>[T|G|M] | samesize=<fs_name>[::cel=<cel_name>] } pool=<pool>
Creates a file system on the volume specified by its desired size and storage pool or by using the same size as a specified local or remote file system. Also assigns an optional name and file system type to a file system. If a name is not specified, one is assigned automatically. A file system name can be up to 240 characters, but cannot begin with a dash (-), be comprised entirely of integers or be a single integer, contain the word root or contain a colon (:). Available file system types are uxfs (default), mgfs, or rawfs.
When using the `samesize` option by using the options described below, the `slice=` should be set to `y`. The new file system is created with the same size as the specified file system.

The `pool` option specifies a rule set for the new file system that contains automatically created volumes and defines the type of disk volumes used and how they are aggregated. Storage pools are system defined (storage pool description provides more information) or user defined.

```
[worm=(enterprise|compliance|off)]
```

Enables the storage capability on the new file system. The capability can only be specified when creating a new file system; existing file systems cannot be converted. After a file system is enabled, it is persistently marked as such until the time when it is deleted.

**CAUTION**

The enterprise version of this feature is intended for self-regulated archiving. The administrator is considered a trusted user and feature does not protect the archived data from the administrator’s actions. If the administrator attempts to delete a file system, the file system issues a warning message and prompts the administrator to confirm the operation. This version of this feature is not intended for high-end compliance applications such as pharmaceuticals, aerospace, or finance.

```
[storage=<system_name>]
```

Specifies the storage system on which the file system resides. Use `nas_storage -list` to obtain a list of the available system names.

```
[-auto_extend {no|yes} [-thin {no|yes}]]
```

For an AVM file system, turns Automatic File System Extension and thin provisioning on or off, and sets a high water mark and maximum size for the file system. When Automatic File System Extension is turned on, the file system is automatically extended up to the maximum size specified when the high water mark is reached. The default for `-auto_extend` is `no`.

Thin provisioning reports the maximum file system size to the CIFS, NFS, and FTP users, even if the actual size of the file system is smaller. If disabled, the true file system size and maximum file system sizes are reported to the system administrator. Thin provisioning requires that a maximum file system size also be set.
If a file system is created using a virtual storage pool, the -thin option cannot be enabled because file-based thin provisioning and block-based thin provisioning cannot coexist on a file system.

**Note:** SRDF pools are not supported.

[-hwm <50-99>\%]
Specifies the size threshold that must be reached before the file system is automatically extended. Type an integer between the range of 50 and 99 to represent the percentage of file system usage. The default is 90.

[-max_size <integer> [T|G|M]]
Sets the maximum file system size to which a file system can be extended. Type an integer and specify T for terabytes, G for gigabytes (default), or M for megabytes. If the -max_size option is not specified, then it defaults to the maximum limit of the file system size which is 16 terabytes. Maximum size must be set to enable thin provisioning. The maximum size is what is presented to users as the file system size through thin provisioning.

[-option <options>]
Specifies the following comma-separated options:

nbpi=<number>
The number of bytes per inode block. The default is 8192 bytes.

mover=<movername>
Assigns an optional Data Mover on which to build a file system. If no Data Mover is assigned, the system will automatically pick the first available Data Mover to build the file system.

slice={y|n}
Specifies whether the disk volumes used by the new file system may be shared with other file systems by using a slice. The slice=y option allows the file system to share disk volumes with other file systems. The slice=n option ensures that the new file system has exclusive access to the disk volumes it uses, and is relevant when using TimeFinder/FS.

When symm_std, symm_std_rdf_src, symm_ata, symm_ata_rdf_src, symm_ata_rdf_tgt, and symm_std_rdf_tgt pools are specified, the default is not to slice the volumes, which is overridden with slice=y. For symm_efd, the default is slice=y, because TimeFinder/FS is not supported with EFD disk types.
When clar_r1, clar_r5_performance, clar_r5_economy, clar_r6, clarata_r3, clarata_r6, clarata_r10, clarata_archive, cm_r1, cm_r5_performance, cm_r5_economy, cm_r6, cmata_r3, cmata_archive, cmata_r6, cmata_r10, clarsas_archive, clarsas_r6, clarsas_r10, clarefd_r5, clarefd_r10, cmsas_archive, cmsas_r6, cmsas_r10, and cmefd_r5 pools are specified, the default for standard AVM pools is to slice the volumes (slice=y), which is overridden by using slice=n. The default for mapped pools is not to slice the volumes (slice=n). Use nas_pool to change the default slice option.

\[ -\text{name } <\text{name}> \] -type nmfs -create
Creates a nested mount file system (NMFS) that can be used to combine multiple uxfs file systems into a single virtual file system. The NMFS can then be mounted and exported as a single share or mount point.

**SEE ALSO**

**STORAGE SYSTEM OUTPUT**
The number associated with the storage device is dependent on the attached storage system. VNX for block displays a prefix of APM before a set of integers, for example, APM0003900124-0019. Symmetrix storage systems appear as 002804000190-003C. The outputs displayed in the examples use a VNX for block.

VNX for block supports the following system-defined storage pools: clar_r1, clar_r5_performance, clar_r5_economy, clar_r6, clarata_r3, clarata_r6, clarata_r10, clarata_archive, cm_r1, cm_r5_performance, cm_r5_economy, cm_r6, cmata_r3, cmata_archive, cmata_r6, cmata_r10, clarsas_archive, clarsas_r6, clarsas_r10, clarefd_r5, clarefd_r10, cmsas_archive, cmsas_r6, cmsas_r10, and cmefd_r5.

VNXs with a Symmetrix storage system support the following system-defined storage pools: symm_std_rdf_src, symm_std, symm_ata, symm_ata_rdf_src, symm_ata_rdf_tgt, symm_std_rdf_tgt, and symm_efd.

For user-defined storage pools, the difference in output is in the disk type. Disk types when using a Symmetrix are: STD, R1STD, R2STD, BCV, R1BCV, R2BCV, ATA, R1ATA, R2ATA, BCVA, R1BCA, R2BCA, and EFD.
Disk types when using VNX for block are: CLSTD, CLEFD, CLATA, MIXED (indicates that tiers used in the pool contain multiple disk types), Performance, Capacity, and Extreme_performance and for VNX for block involving mirrored disks are: CMEFD, CMSTD, CMATA, Mirrored_mixed, Mirrored_performance, Mirrored_capacity, and Mirrored_extreme_performance.

**EXAMPLE #1**

To create a file system named ufs1 on metavolume mtv1, type:

```
$ nas_fs -name ufs1 -create mtv1
```

```
id        = 37
name      = ufs1
acl       = 0
in_use    = False
type      = uxfs
worm      = compliance
worm_clock = Thu Mar  6 16:26:09 EST 2008
worm Max Retention Date = Fri April 18 12:30:40 EST 2008
volume    = mtv1
pool      =
rw_servers=
ro_servers=
rw_vdms   =
ro_vdms   =
auto_ext  = no,thin=no
deduplication = off
stor_devs = APM00042000818-0012,APM00042000818-0014
disks     = d7,d9
```

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Automatically assigned ID of a file system.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to a file system.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control value assigned to the file system.</td>
</tr>
<tr>
<td>in_use</td>
<td>If a file system is registered into the mount table of a Data Mover.</td>
</tr>
<tr>
<td>type</td>
<td>Type of file system. -list provides information for a description of the types.</td>
</tr>
<tr>
<td>worm</td>
<td>Write Once Read Many (WORM) state of file system. It states whether the state is disabled or set to either compliance or enterprise.</td>
</tr>
<tr>
<td>volume</td>
<td>Volume on which a file system resides.</td>
</tr>
<tr>
<td>pool</td>
<td>Storage pool for the file system.</td>
</tr>
<tr>
<td>rw_servers</td>
<td>Servers with read/write access to a file system.</td>
</tr>
<tr>
<td>ro_servers</td>
<td>Servers with read-only access to a file system.</td>
</tr>
<tr>
<td>rw_vdms</td>
<td>VDM servers with read/write access to a file system.</td>
</tr>
<tr>
<td>ro_vdms</td>
<td>VDM servers with read-only access to a file system.</td>
</tr>
<tr>
<td>worm_clock</td>
<td>Software clock maintained by the file system. The clock functions only when the file system is mounted read/write.</td>
</tr>
</tbody>
</table>
### The nas Commands

#### Note:
The deduplication state is **unavailable** when the file system is unmounted.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| `worm Max Retention Date` | Time when the protected files expire. The file system can be deleted only after this date. The special values returned are:  
  - 3 — The file system is FLR-E with protected files.  
  - 2 — The file system is scanning for max_retention period.  
  - 1 — The default value (no protected files created).  
  - 0 — Infinite retention period (if the server is up and running). |
| `auto_ext`     | Indicates whether auto-extension and thin provisioning are enabled. |
| `deduplication`| Deduplication state of the file system. The file data is transferred to the storage which performs the deduplication and compression on the data. The states are:  
  - On — Deduplication on the file system is enabled.  
  - Suspended — Deduplication on the file system is suspended. Deduplication does not perform any new space reduction but the existing files that were reduced in space remain the same.  
  - Off — Deduplication on the file system is disabled. Deduplication does not perform any new space reduction and the data is now reduplicated. |
| `stor_devs`    | Storage system devices associated with a file system. |
| `disks`        | Disks on which the metavolume resides. |

---

**EXAMPLE #2**

To display information about a file system by using the file system ID, 37, type:

```
$ nas_fs -info id=37
```

```bash
id = 37
name = ufs1
acl = 0
in_use = True
type = uxfs
worm = off
volume = mtv1
pool = marketing
member_of = root_avm_fs_group_3
rw_servers= server_2
ro_servers=
rw_vdms =
ro_vdms =
auto_ext = no,thin=no
deduplication = off
thin_storage = True
tiering_policy = Auto-tier
compressed = False
mirrored = False
stor_devs = APM00042000818-0012,APM00042000818-0014
disks = d7,d9
```
The nas Commands

Where

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>thin_storage</td>
<td>Indicates whether the block storage system uses thin provisioning. Values are: True, False, Mixed.</td>
</tr>
<tr>
<td>tiering_policy</td>
<td>Indicates the tiering policy in effect. If the initial tier and the tiering policy are the same, the values are: Auto-Tier, Highest Available Tier, Lowest Available Tier. If the initial tier and the tiering policy are not the same, the values are: Auto-Tier/No Data Movement, Highest Available Tier/No Data Movement, Lowest Available Tier/No Data Movement.</td>
</tr>
<tr>
<td>compressed</td>
<td>For VNX for block, indicates whether data is compressed. Values are: True, False, Mixed (indicates some of the LUNs, but not all, are compressed).</td>
</tr>
<tr>
<td>mirrored</td>
<td>Indicates whether the disk is mirrored.</td>
</tr>
</tbody>
</table>

**EXAMPLE #3**  
To display a list of file systems, type:

```bash
$ nas_fs -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>inuse</th>
<th>type</th>
<th>acl</th>
<th>volume</th>
<th>name</th>
<th>server</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>24</td>
<td>root_fs_1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>26</td>
<td>root_fs_2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>28</td>
<td>root_fs_3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>30</td>
<td>root_fs_4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>32</td>
<td>root_fs_5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>34</td>
<td>root_fs_6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>36</td>
<td>root_fs_7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>38</td>
<td>root_fs_8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>40</td>
<td>root_fs_9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>42</td>
<td>root_fs_10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>44</td>
<td>root_fs_11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>46</td>
<td>root_fs_12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>48</td>
<td>root_fs_13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>50</td>
<td>root_fs_14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>52</td>
<td>root_fs_15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>54</td>
<td>root_fs_common</td>
<td>2,1</td>
</tr>
<tr>
<td>17</td>
<td>n</td>
<td>5</td>
<td>0</td>
<td>87</td>
<td>root_fs_ufslog</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>n</td>
<td>5</td>
<td>0</td>
<td>90</td>
<td>root_panic_reserve</td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>315</td>
<td>v2src1</td>
<td>1</td>
</tr>
<tr>
<td>214</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>318</td>
<td>v2dst1</td>
<td></td>
</tr>
<tr>
<td>230</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>346</td>
<td>v2srclun1</td>
<td>1</td>
</tr>
<tr>
<td>231</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>349</td>
<td>v2dstlun1</td>
<td>2</td>
</tr>
<tr>
<td>342</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>560</td>
<td>root_fs_vdm_srcvdm1</td>
<td>1</td>
</tr>
<tr>
<td>343</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>563</td>
<td>root_fs_vdm_srcvdm2</td>
<td>1</td>
</tr>
<tr>
<td>987</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>1722</td>
<td>gtest</td>
<td>1</td>
</tr>
<tr>
<td>988</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>1725</td>
<td>src1</td>
<td>1</td>
</tr>
<tr>
<td>989</td>
<td>y</td>
<td>5</td>
<td>0</td>
<td>1728</td>
<td>dst1</td>
<td>1</td>
</tr>
<tr>
<td>1346</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2354</td>
<td>root_fs_vdm_srcvdm1</td>
<td>1</td>
</tr>
<tr>
<td>1350</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2367</td>
<td>fsl</td>
<td>v9</td>
</tr>
<tr>
<td>1354</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>2374</td>
<td>fsl_replica1</td>
<td></td>
</tr>
<tr>
<td>1361</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>2385</td>
<td>fsl_replica2</td>
<td></td>
</tr>
<tr>
<td>1365</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2392</td>
<td>fsl1365</td>
<td>1</td>
</tr>
<tr>
<td>1371</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2399</td>
<td>root_fs_vdm_v1</td>
<td>1</td>
</tr>
<tr>
<td>1372</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2401</td>
<td>f1</td>
<td>v40</td>
</tr>
<tr>
<td>1376</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2406</td>
<td>root_fs_vdm_v1_repl</td>
<td>2</td>
</tr>
</tbody>
</table>
The nas Commands

EXAMPLE #4 To list all the file systems including internal checkpoints, type:

$ nas_fs -list -all

id   inuse type acl   volume   name           server
1 n    1    0   24        root_fs_1
2 y    1    0   26        root_fs_2 1
3 y    1    0   28        root_fs_3 2
4 n    1    0   30        root_fs_4
5 n    1    0   32        root_fs_5
6 n    1    0   34        root_fs_6
7 n    1    0   36        root_fs_7
8 n    1    0   38        root_fs_8
9 n    1    0   40        root_fs_9
10 n   1    0   42        root_fs_10
11 n   1    0   44        root_fs_11
12 n   1    0   46        root_fs_12
13 n   1    0   48        root_fs_13
14 n   1    0   50        root_fs_14
15 n   1    0   52        root_fs_15
16 y   1    0   54        root_fs_common 2,1
17 n   5    0   87        root_fs.ufslog
18 n   5    0   90        root_panic_reserve
212 y  1    0   315       v2src1              1
213 y  101  0   0         root_avm_fs_group_3
214 n  1    0   318       v2dst1
230 y  1    0   346       v2srclun1           1
231 y  1    0   349       v2dstlun1           2
342 y  1    0   560       root_fs_vdm_srvcmdm1 1
343 y  1    0   563       root_fs_vdm_srvcmdm2 1
986 n  11   0   0         vpfs986
987 y  7    0   1722      gtest              1
988 y  1    0   1725      src1               1
989 y  5    0   1728      dst1               1
1343 n  11   0   0         vpfs1343
1344 y  7    0   2351      root_rep_ckpt_342_2_1
1345 y  7    0   2351      root_rep_ckpt_342_2_1
1346 y  1    0   2354      root_fs_vdm_srvcmdm1 1
1347 n  11   0   0         vpfs1347
1348 y  7    0   2358      root_rep_ckpt_1346_1
1349 y  7    0   2358      root_rep_ckpt_1346_1
1350 y  1    0   2367      fs1                 v9
1354 n  1    0   2374      fs1_replica1
### The nas Commands

<table>
<thead>
<tr>
<th>Number</th>
<th>Type</th>
<th>ID</th>
<th>Flags</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1358</td>
<td>n</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>vpfs1358</td>
</tr>
<tr>
<td>1359</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2383</td>
<td>root_rep_ckpt_1350_ v9</td>
</tr>
<tr>
<td>1360</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2383</td>
<td>root_rep_ckpt_1350_ v9</td>
</tr>
<tr>
<td>1361</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>2385</td>
<td>fs1_replica2</td>
</tr>
<tr>
<td>1362</td>
<td>n</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>vpfs1362</td>
</tr>
<tr>
<td>1363</td>
<td>n</td>
<td>7</td>
<td>0</td>
<td>2388</td>
<td>root_rep_ckpt_1361_</td>
</tr>
<tr>
<td>1364</td>
<td>n</td>
<td>7</td>
<td>0</td>
<td>2388</td>
<td>root_rep_ckpt_1361_</td>
</tr>
<tr>
<td>1365</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2392</td>
<td>fs1365</td>
</tr>
<tr>
<td>1366</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2383</td>
<td>root_rep_ckpt_1350_ v9</td>
</tr>
<tr>
<td>1367</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2383</td>
<td>root_rep_ckpt_1350_ v9</td>
</tr>
<tr>
<td>1368</td>
<td>n</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>vpfs1368</td>
</tr>
<tr>
<td>1369</td>
<td>n</td>
<td>7</td>
<td>0</td>
<td>2395</td>
<td>root_rep_ckpt_1354_</td>
</tr>
<tr>
<td>1370</td>
<td>n</td>
<td>7</td>
<td>0</td>
<td>2395</td>
<td>root_rep_ckpt_1354_</td>
</tr>
<tr>
<td>1371</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2399</td>
<td>root_fs_vdm_v1</td>
</tr>
<tr>
<td>1372</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2401</td>
<td>f1</td>
</tr>
<tr>
<td>1374</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2406</td>
<td>root_fs_vdm_v1_repl 2</td>
</tr>
<tr>
<td>1376</td>
<td>n</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>vpfs1376</td>
</tr>
<tr>
<td>1378</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2419</td>
<td>root_rep_ckpt_1383_ v41</td>
</tr>
<tr>
<td>1380</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2419</td>
<td>root_rep_ckpt_1383_ v41</td>
</tr>
<tr>
<td>1382</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2423</td>
<td>cworm</td>
</tr>
<tr>
<td>1384</td>
<td>n</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>vpfs1384</td>
</tr>
<tr>
<td>1386</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2432</td>
<td>root_rep_ckpt_1389_ 2</td>
</tr>
<tr>
<td>1387</td>
<td>n</td>
<td>3</td>
<td>0</td>
<td>2432</td>
<td>root_rep_ckpt_1389_ 2</td>
</tr>
<tr>
<td>1389</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>2425</td>
<td>cworm1</td>
</tr>
<tr>
<td>1391</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2427</td>
<td>fs2</td>
</tr>
<tr>
<td>1392</td>
<td>y</td>
<td>2</td>
<td>0</td>
<td>2429</td>
<td>fs3</td>
</tr>
<tr>
<td>1393</td>
<td>n</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>vpfs1393</td>
</tr>
<tr>
<td>1395</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2435</td>
<td>root_rep_ckpt_1390_ 2</td>
</tr>
<tr>
<td>1396</td>
<td>n</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>vpfs1396</td>
</tr>
<tr>
<td>1398</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2439</td>
<td>fs4</td>
</tr>
<tr>
<td>1399</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>2441</td>
<td>fs5</td>
</tr>
<tr>
<td>1400</td>
<td>n</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>vpfs1400</td>
</tr>
<tr>
<td>1401</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2444</td>
<td>root_rep_ckpt_1398_ 2</td>
</tr>
<tr>
<td>1402</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2444</td>
<td>root_rep_ckpt_1398_ 2</td>
</tr>
<tr>
<td>1403</td>
<td>n</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>vpfs1403</td>
</tr>
<tr>
<td>1404</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2447</td>
<td>root_rep_ckpt_1399_ 2</td>
</tr>
<tr>
<td>1405</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2447</td>
<td>root_rep_ckpt_1399_ 2</td>
</tr>
<tr>
<td>1406</td>
<td>y</td>
<td>7</td>
<td>0</td>
<td>2444</td>
<td>fs4_ckpt1</td>
</tr>
</tbody>
</table>

---

**Note:** NDMP and Replicator internal checkpoints can be identified by specific prefixes in the filename. Using VNX SnapSure provides more information for internal checkpoints naming formats.

---

**EXAMPLE #5** To create a uxfs file system named ufs1 on storage system APM00042000818, with a size of 100 GB, by using the
The nas Commands

clar_r5_performance pool and allowing the file system to share disk volumes with other file systems, type:

```bash
$ nas_fs -name ufs1 -type uxfs -create size=100G
pool=clar_r5_performance storage=APM00042000818 -option slice=y
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>member_of</td>
<td>File system group to which the file system belongs.</td>
</tr>
</tbody>
</table>

Where:

EXAMPLE #1 provides a description of command output. The clar_r5_performance storage pool is available with the VNX for block.

EXAMPLE #6 To create a rawfs file system named ufs3 with the same size as the file system ufs1 by using the clar_r5_performance pool and allowing the file system to share disk volumes with other file systems, type:

```bash
$ nas_fs -name ufs3 -type rawfs -create samesize=ufs1
pool=clar_r5_performance storage=APM00042000818 -option slice=y
```

id        = 38
name      = ufs1
acl       = 0
in_use    = False
type      = uxfs
worm      = off
volume    = v171
pool      = clar_r5_performance
member_of = root_avm_fs_group_3
rw_servers=          
ro_servers=          
rw_vdms    =
ro_vdms    =
auto_ext   = no,thin=no
deduplication = off
stor_devs  = APM00042000818-001F,APM00042000818-001D,APM00042000818-0019,APM00042000818-0016
disks     = d20,d18,d14,d11
The nas Commands

ro_servers=
rd_wdms =
ro_vdms =
auto_ext = no, thin=no
deduplication = off
stor_devs = APM00042000818-001F,APM00042000818-001D,APM00042000818-0019,APM00042000818-0016
disks = d20,d18,d14,d11

EXAMPLE #1 and EXAMPLE #3 provide for a description of command outputs. The clar_r5_performance storage pool is available with the VNX for block.

EXAMPLE #7
To create a ufs file system named ufs4, with a size of 100 GB, by using the clar_r5_performance pool, with enabled, 4096 bytes per inode, and server_3 for file system building, type:

```
$ nas_fs -name ufs4 -create size=100G pool=clar_r5_performance worm=enterprise -option nbpi=4096,mover=server_3
```

id = 40
name = ufs4
acl = 0
in_use = False
type = ufs
worm = enterprise
volume = v175
pool = clar_r5_performance
member_of = root_avm_fs_group_3
rw_servers=
ro_servers=
rd_wdms =
ro_vdms =
auto_ext = no, thin=no
deduplication = off
stor_devs = APM00042000818-001F,APM00042000818-001D,APM00042000818-0019,APM00042000818-0016
disks = d20,d18,d14,d11

To ensure retention of protected files, it can also be set to compliance by typing:

```
$ nas_fs -name ufs4 -create size=100G pool=clar_r5_performance worm=compliance -option nbpi=4096,mover=server_3
```

id = 40
name = ufs4
acl = 0
in_use = False
type = ufs
worm = compliance
volume    = v175  
pool      = clar_r5_performance  
member_of = root_avm_fs_group_3  
rw_servers=  
ro_servers=  
rw_vdms   =  
ro_vdms   =  
auto_ext  = no,thin=no  
deduplication = off  
stor_devs = APM00042000818-001F,APM00042000818-001D,APM00042000818-0019,APM00042000818-0016  
disks     = d20,d18,d14,d11

EXAMPLE #1 provides a description of command outputs. The clar_r5_performance storage pool is available with the VNX for block.

EXAMPLE #8 To create a uxfs file system named ufs1, with a size of 10 GB, by using the clar_r5_performance pool, and an ID of 8000 assigned to ufs1, type:

$ nas_fs -name ufs1 -type uxfs -create size=10G pool=clar_r5_performance -option slice=y,id=8000

id   = 8000  
name  = ufs1  
ac1   = 0  
in_use = False  
type  = uxfs  
worm  = off  
volume = v7193  
pool   = clar_r5_performance  
member_of = root_avm_fs_group_3  
rw_servers=  
ro_servers=  
rw_vdms   =  
ro_vdms   =  
auto_ext  = no,thin=no  
deduplication = off  
stor_devs = APM00043401231-0035  
disks     = d3

EXAMPLE #9 To create a uxfs file system named ufs2, with a size of 10 GB, by using the clar_r5_performance pool, and an ID of 8000 assigned to ufs1, type:

$ nas_fs -name ufs2 -type uxfs -create size=10G pool=clar_r5_performance -option slice=y,id=8000

id   = 12464  
name  = ufs2  
ac1   = 0  
in_use = False
The nas Commands

type = uxfs
worm = off
volume = v7195
pool = clar_r5_performance
member_of = root_avm_fs_group_3
rw_servers=
ro_servers=
rcdms =
ro_vdms =
auto_ext = no,thin=no
deduplication = off
stor_devs = APM00043401231-0034
disks = d5

Warning 17716815881: unavailable id : 8000

**Note:** This output is displayed if the desired ID is not available.

**EXAMPLE #10**

To view the size of *ufs1*, type:

```bash
$ nas_fs -size ufs1
```

```plaintext
total = 393 avail = 393 used = 1 ( 0% ) (sizes in MB) ( blockcount = 819200 )
volume: total = 400 (sizes in MB) ( blockcount = 819200 ) avail = 393 used = 7 ( 2% )
```

When a file system is mounted, the size info for the volume and a file system, as well as the number of blocks that are used are displayed.

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>Total size of the file system.</td>
</tr>
<tr>
<td>blockcount</td>
<td>Total number of blocks used.</td>
</tr>
</tbody>
</table>

**EXAMPLE #11**

To rename a file system from *ufs1* to *ufs2*, type:

```bash
$ nas_fs -rename ufs1 ufs2
```

```plaintext
id        = 38
name      = ufs2
acl       = 0
in_use    = True
type      = uxfs
worm      = off
volume    = v171
pool      = clar_r5_performance
member_of = root_avm_fs_group_3
rw_servers= server_2
ro_servers=
rcdms =
ro_vdms =
auto_ext = no,thin=no
deduplication = off
```
EXAMPLE #1 and EXAMPLE #3 provide a description of command outputs. The clar_r5_performance storage pool is available with the VNX for block.

EXAMPLE #12 To list all of the file systems in the NAS database, type:

```
$ nas_fs -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>inuse</th>
<th>type</th>
<th>acl</th>
<th>volume</th>
<th>name</th>
<th>server</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>10</td>
<td>root_fs_1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>root_fs_2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>14</td>
<td>root_fs_3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>16</td>
<td>root_fs_4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>18</td>
<td>root_fs_5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>root_fs_6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>22</td>
<td>root_fs_7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>24</td>
<td>root_fs_8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>26</td>
<td>root_fs_9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>28</td>
<td>root_fs_10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>30</td>
<td>root_fs_11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>32</td>
<td>root_fs_12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>34</td>
<td>root_fs_13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>36</td>
<td>root_fs_14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>38</td>
<td>root_fs_15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>40</td>
<td>root_fs_common</td>
<td>2,4,3,1</td>
</tr>
<tr>
<td>17</td>
<td>n</td>
<td>5</td>
<td>0</td>
<td>73</td>
<td>root_fs_ufslog</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>n</td>
<td>5</td>
<td>0</td>
<td>76</td>
<td>root_fs_d3</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>n</td>
<td>5</td>
<td>0</td>
<td>77</td>
<td>root_fs_d4</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>n</td>
<td>5</td>
<td>0</td>
<td>78</td>
<td>root_fs_d5</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>n</td>
<td>5</td>
<td>0</td>
<td>79</td>
<td>root_fs_d6</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>171</td>
<td>ufs2</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>175</td>
<td>ufs4</td>
<td></td>
</tr>
</tbody>
</table>

EXAMPLE #1 provides a description of command outputs.

EXAMPLE #13 To extend the file system, ufs1, with the volume, emtv2b, type:

```
$ nas_fs -xtend ufs1 emtv2b
```

id    = 38
name  = ufs1
acl   = 0
The nas Commands

<table>
<thead>
<tr>
<th>in_use</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>uxfs</td>
</tr>
<tr>
<td>worm</td>
<td>off</td>
</tr>
<tr>
<td>volume</td>
<td>vl71</td>
</tr>
<tr>
<td>pool</td>
<td>clar_r5_performance</td>
</tr>
<tr>
<td>member_of</td>
<td>root_avm_fs_group_3</td>
</tr>
<tr>
<td>rw_servers</td>
<td>server_2</td>
</tr>
<tr>
<td>ro_servers</td>
<td></td>
</tr>
<tr>
<td>rw_vdms</td>
<td></td>
</tr>
<tr>
<td>ro_vdms</td>
<td></td>
</tr>
<tr>
<td>auto_ext</td>
<td>no, thin=no</td>
</tr>
<tr>
<td>deduplication</td>
<td>off</td>
</tr>
<tr>
<td>stor_devs</td>
<td>APM00042000818-001F, APM00042000818-001D, APM00042000818-0019, APM00042000818-0016, APM00042000818-001C</td>
</tr>
<tr>
<td>disks</td>
<td>d20, d18, d14, d11, d17</td>
</tr>
<tr>
<td>disk</td>
<td>stor_dev</td>
</tr>
<tr>
<td>d20</td>
<td>APM00042000818-001F</td>
</tr>
<tr>
<td>addr</td>
<td>c0t1l15</td>
</tr>
<tr>
<td></td>
<td>c32t1l15</td>
</tr>
<tr>
<td>d18</td>
<td>APM00042000818-001D</td>
</tr>
<tr>
<td>addr</td>
<td>c0t1l13</td>
</tr>
<tr>
<td></td>
<td>c32t1l13</td>
</tr>
<tr>
<td>d14</td>
<td>APM00042000818-0019</td>
</tr>
<tr>
<td>addr</td>
<td>c0t1l19</td>
</tr>
<tr>
<td></td>
<td>c32t1l19</td>
</tr>
<tr>
<td>d11</td>
<td>APM00042000818-0016</td>
</tr>
<tr>
<td>addr</td>
<td>c0t1l16</td>
</tr>
<tr>
<td></td>
<td>c32t1l16</td>
</tr>
<tr>
<td>d17</td>
<td>APM00042000818-001C</td>
</tr>
<tr>
<td>addr</td>
<td>c0t1l12</td>
</tr>
<tr>
<td></td>
<td>c32t1l12</td>
</tr>
</tbody>
</table>

**EXAMPLE #1** provides a description of command outputs.

**EXAMPLE #14**  To extend the file system named *ufs2*, with the specified size of 100 GB, by using *clar_r5_performance* pool, type:

```
$ nas_fs -xtend ufs2 size=100G pool=clar_r5_performance
```

id        | 38         |
name      | ufs2       |
acl       | 0          |
in_use    | True       |
type      | uxfs       |
worm     | off        |
volume   | vl71       |
pool     | clar_r5_performance |
member_of | root_avm_fs_group_3 |
rw_servers| server_2  |
ro_servers|           |
rw_vdms  |            |
ro_vdms  |            |
auto_ext | no, thin=no|
deduplication | off   |
stor_devs | APM00042000818-001F, APM00042000818-001D, APM00042000818-0019, APM00042000818-0016, APM00042000818-001C |
disks    | d20, d18, d14, d11, d17 |
disk    | stor_dev |
disk    | d20       |
stor_dev | APM00042000818-001F  |
addr    | c0t1l15 | server=server_2 |
disk    | d20       |
stor_dev | APM00042000818-001F  |
addr    | c32t1l15 | server=server_2 |

EXAMPLE #1 provides a description of command outputs. The `clar_r5_performance` storage pool is available with the VNX for block.

EXAMPLE #15  To set the access control level for the file system `ufs1`, type:

$ nas_fs -acl 1432 ufs1

```
id        = 38
name      = ufs1
acl       = 1432, owner=nasadmin, ID=201
in_use    = True
type      = uXfs
worm      = off
volume    = v171
pool      = clar_r5_performance
member_of = root_avm_fs_group_3
rw_servers= server_2
ro_servers=
rw_vdms   =
ro_vdms   =
auto_ext  = no,thin=no
deduplication = off
stor_devs = APM00042000818-001F,APM00042000818-001D,APM00042000818-0019,APM00042000818-0016,APM00042000818-001C
disks     = d20,d18,d14,d11,d17
disk=d20stor_dev=APM00042000818-001F addr=c0t1l15 server=server_2
disk=d20stor_dev=APM00042000818-001F addr=c32t1l15 server=server_2
disk=d18stor_dev=APM00042000818-001D addr=c0t1l13 server=server_2
disk=d18stor_dev=APM00042000818-001D addr=c32t1l13 server=server_2
disk=d14stor_dev=APM00042000818-0019 addr=c0t1l13 server=server_2
disk=d14stor_dev=APM00042000818-0019 addr=c32t1l13 server=server_2
disk=d11stor_dev=APM00042000818-0016 addr=c0t1l16 server=server_2
disk=d11stor_dev=APM00042000818-0016 addr=c32t1l16 server=server_2
disk=d17stor_dev=APM00042000818-001C addr=c0t1l12 server=server_2
disk=d17stor_dev=APM00042000818-001C addr=c32t1l12 server=server_2
```

Note: The value 1432 specifies `nasadmin` as the owner and gives users with an access level of at least observer read access only, users with an access level of at least operator read/write access, and users with an access level of at least admin read/write/delete access.
EXAMPLE #1 provides a description of command outputs.

EXAMPLE #16 To convert a uxfs file system type named ufs1 to a rawfs, type:

$ nas_fs -Type rawfs ufs1 -Force

name      = ufs1
acl       = 1432, owner=nasadmin, ID=201
in_use    = True
type      = rawfs
worm      = off
volume    = v171
pool      = clar_r5_performance
member_of = root_avm_fs_group_3
rw_servers= server_2
ro_servers=
rw_vdms   =
ro_vdms   =
auto_ext  = no,thin=no
deduplication = off
stor_devs = APM00042000818-001F,APM00042000818-001D,APM00042000818-0019,APM00042000818-0016,APM00042000818-001C
disks     = d20,d18,d14,d17
disk=d20  stor_dev=APM00042000818-001F addr=c0t1l15      server=server_2
disk=d20  stor_dev=APM00042000818-001F addr=c32t1l15      server=server_2
disk=d18  stor_dev=APM00042000818-001D addr=c0t1l13      server=server_2
disk=d18  stor_dev=APM00042000818-001D addr=c32t1l13      server=server_2
disk=d14  stor_dev=APM00042000818-0019 addr=c0t1l19      server=server_2
disk=d14  stor_dev=APM00042000818-0019 addr=c32t1l19      server=server_2
disk=d11  stor_dev=APM00042000818-0016 addr=c0t1l16      server=server_2
disk=d11  stor_dev=APM00042000818-0016 addr=c32t1l16      server=server_2
disk=d17  stor_dev=APM00042000818-001C addr=c0t1l12      server=server_2
disk=d17  stor_dev=APM00042000818-001C addr=c32t1l12      server=server_2

EXAMPLE #1 provides a description of command outputs.

EXAMPLE #17 To start the conversion of the file system, ufs2, and to conform to the MIXED access policy mode, type:

$ nas_fs -translate ufs2 -access_policy start -to MIXED -from NT

id        = 38
name      = ufs2
acl       = 1432, owner=nasadmin, ID=201
in_use    = True
type      = uxfs
worm      = off
volume    = v171
pool      = clar_r5_performance
member_of = root_avm_fs_group_3
rw_servers= server_2
ro_servers=
rw_vdms   =
EXAMPLE #1 provides a description of command outputs. The clar_r5_performance storage pool is available with the VNX for block.

EXAMPLE #18 To display the status of access policy conversion for ufs2, type:

```
$ nas_fs -translate ufs2 -access_policy status
```

```
1130511626: CFS: 7: conversion in progress : 10/11 inodes done 90 % done
    percent_inode_scanned=90
1130511626: ADMIN: 4: Command succeeded: acl database=/ufs2 convertAccessPolicy
```

EXAMPLE #19 To create a nested mount file system, nmfs1, type:

```
$ nas_fs -name nmfs1 -type nmfs -create
```

EXAMPLE #1 provides a description of command outputs.
The *nas* Commands

**EXAMPLE #20** To delete *ufs1*, type:

```
$ nas_fs -delete ufs1
```

```
id        = 37
name      = ufs1
acl       = 0
in_use    = False
type      = uxfs
worm      = off
volume    = mtv1
rw_servers=
ro_servers=
rw_vdms   =
ro_vdms   =
auto_ext  = no,thin=no
deduplication = off
stor_devs = APM00042000818-0012,APM00042000818-0014
disks     = d7,d9
```

**EXAMPLE #1** provides a description of command outputs.

**EXAMPLE #21** To create a file system named *ufs3*, with a size of 20 GB, by using the *clar_r5_performance* pool, a maximum size of 100 GB and with auto-extend and thin provisioning enabled, type:

```
$ nas_fs -name ufs3 -create size=20G pool=clar_r5_performance -auto_extend yes -max_size 100G -thin yes
```

```
id        = 42
name      = ufs3
acl       = 0
in_use    = False
type      = uxfs
worm      = off
volume    = v180
pool      = clar_r5_performance
member_of = root_avm_fs_group_3
rw_servers=
ro_servers=
rw_vdms   =
ro_vdms   =
auto_ext  = hwm=90%,max_size=102400M,thin=yes
deduplication = off
thin_storage = True
tiering_policy = Auto-tier
compressed = False
mirrored  = False
stor_devs = APM00042000818-001F,APM00042000818-001D,APM00042000818-0019,APM00042000818-0016
000818-0016
disks     = d20,d18,d14,d11
```

**EXAMPLE #1** provides a description of command outputs.
EXAMPLE #22  To disable thin provisioning on ufs3, type:

```
$ nas_fs -modify ufs3 -thin no
```

id        = 42
name      = ufs3
acl       = 0
in_use    = False
type      = uxfs
worm      = off
volume    = v180
pool      = clar_r5_performance
member_of = root_avm_fs_group_3
rw_servers=
ro_servers=
rw_vdms   =
ro_vdms   =
auto_ext  = hwm=90%,max_size=102400M,thin=no
deduplication = off
stor_devs = APM00042000818-001F,APM00042000818-001D,APM00042000818-0019,APM00042000818-0016
disks     = d20,d18,d14,d11

EXAMPLE #1 provides a description of command outputs. The clar_r5_performance storage pool is available with the VNX for block.

EXAMPLE #23  To query the current directory type and translation status for MPD, type:

```
$ nas_fs -info ufs1 -option mpd
```

id        = 35
name      = ufs1
acl       = 0
in_use    = True
type      = uxfs
worm      = off
volume    = mtv1
pool      =
member_of = root_avm_fs_group_3
rw_servers= server_3
rw_vdms   =
ro_vdms   =
auto_ext  = no,thin=no
deduplication = off
thin_storage = True
tiering_policy = Auto-tier
compressed = False
mirrored  = False
stor_devs = 000190100559-0094,000190100559-0095,000190100559-0096,000190100559-0097
disks     = d59,d60,d61,d62
disk=d59  stor_dev=000190100559-0094  addr=c0t4l8-04-0  server=server_3
The state where both Default_directory_type=DIR3 and Needs_transalation=False assure that this file system’s directories are all in MPD format, and there are no directories of the obsolete single-protocol format.

Any other combination of states, for example,
The nas Commands

Needs_translation=True, indicates that there could be non-MPD directories in the file system which may not be compatible with a future release.
The nas Commands

nas_fsck

Manages fsck and aclchk utilities on specified file systems.

SYNOPSIS

**nas_fsck**
- **-list**
- **-info** {-all | <fs_name>|id=<fs_id>}
- **-start** {<fs_name>|id=<fs_id>} [-aclchkonly][-monitor][-mover <mover_name>][-Force]

DESCRIPTION

nas_fsck uses the fsck and aclchk utilities to perform a check for consistency and errors on the specified file system. nas_fsck also lists and displays the status of the fsck and aclchk utilities. File systems must be mounted read-write to use these utilities.

Depending on the size of the file system, the FSCK utility may use a significant portion of the system’s memory and may affect overall system performance. Hence, it should not be run on a server under heavy load as it is possible that the server may run out of resources. In most cases, the user will be notified if sufficient memory is not available to run a file system check. In these cases, one of the following can be done:

- Start the file system during off-peak hours.
- Reboot the server and let nas_fsck run on reboot.
- Run nas_fsck on a different server if the file system is unmounted.

OPTIONS

- **-list**
  Displays a list of all the file systems undergoing fsck or aclchk.

- **-info** {-all | <fs_name>|id=<fs_id>}
  Queries the Data Mover and displays information about the status of the fsck or aclchk utilities for the specified file system.

- **-start** {<fs_name>|id=<fs_id>}
  Starts the fsck and the aclchk utilities on the specified file system.
CAUTION

If file system check is started on a mounted file system, the file system will be unavailable for the duration of the check. NFS clients will display the message NFS server not responding and CIFS clients will lose connectivity with the server and will have to remap shares.

[-aclchkonly]
Initiates the aclchk utility only, which checks and corrects any errors in the ACL database and removes duplicate ACL information stored on the specified file system. The aclchkonly option can only be used on a file system that is not exported. The default is for both fsck and aclchk.

Note: The NDMP backup process must be stopped on the Data Mover before using the nas_fsck -aclchkonly command.

[-monitor]
Displays the status of fsck and aclchk until the command completes.

Note: For a mounted file system, a <movername> is not required since the fsck and aclchk utilities are run on the Data Mover where the file system is mounted.

[-Force]
Forces a fsck or aclchk to be run on a enabled file system.

SEE ALSO
Managing Volumes and File Systems for VNX Manually and nas_fs.

EXAMPLE #1
To start file system check on ufs1 and monitor the progress, type:

$ nas_fsck -start ufs1 -monitor

id = 27
name = ufs1
volume = mtv1
fsck_server = server_2
inode_check_percent = 10..20..30..40..60..70..80..100
directory_check_percent = 0..0..100
used_ACL_check_percent = 100
free_ACL_check_status = Done
cylinder_group_check_status = In Progress..Done
The nas Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Automatically assigned ID of a file system.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to the file system.</td>
</tr>
<tr>
<td>volume</td>
<td>Volume on which the file system resided.</td>
</tr>
<tr>
<td>fsck_server</td>
<td>Name of the Data Mover where the utility is being run.</td>
</tr>
<tr>
<td>inode_check_percent</td>
<td>Percentage of inodes in the file system checked and fixed.</td>
</tr>
<tr>
<td>directory_check_percent</td>
<td>Percentage of directories in the file system checked and fixed.</td>
</tr>
<tr>
<td>used_ACL_check_percent</td>
<td>Percentage of used ACLs that have been checked and fixed.</td>
</tr>
<tr>
<td>free_ACL_check_status</td>
<td>Status of the ACL check.</td>
</tr>
<tr>
<td>cylinder_group_check_status</td>
<td>Status of the cylinder group check.</td>
</tr>
</tbody>
</table>

EXAMPLE #2 To start ACL check on ufs1, type:

```
$ nas_fsck -start ufs1 -aclchkonly
```

ACLCHK: in progress for file system ufs1

EXAMPLE #3 To start a file system check on ufs2 using Data Mover server_5, type:

```
$ nas_fsck -start ufs2 -mover server_5
```

name = ufs2
id = 23
volume = v134
fsck_server = server_5
inode_check_percent = 40
directory_check_percent = 0
used_ACL_check_percent = 0
free_ACL_check_status = Not Started
cylinder_group_check_status = Not Started

EXAMPLE #1 provides a description of command outputs.

EXAMPLE #4 To list all current file system checks, type:

```
$ nas_fsck -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>type</th>
<th>state</th>
<th>volume</th>
<th>name</th>
<th>server</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>1</td>
<td>FSCK</td>
<td>134</td>
<td>ufs2</td>
<td>4</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>ACLCHK</td>
<td>144</td>
<td>ufs1</td>
<td>1</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Automatically assigned ID of a file system.</td>
</tr>
<tr>
<td>type</td>
<td>Type of file system.</td>
</tr>
<tr>
<td>state</td>
<td>Utility being run.</td>
</tr>
<tr>
<td>volume</td>
<td>Volume on which the file system resided.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to the file system.</td>
</tr>
<tr>
<td>server</td>
<td>Server on which fsck is being run.</td>
</tr>
</tbody>
</table>
**EXAMPLE #5** To display information about file system check for `ufs2` that is currently running, type:

```bash
$ nas_fsck -info ufs2
```

- name = ufs2
- id = 23
- volume = v134
- fsck_server = server_5
- inode_check_percent = 100
- directory_check_percent = 100
- used_ACL_check_percent = 100
- free_ACL_check_status = Done
- cylinder_group_check_status = In Progress

**EXAMPLE #6** To display information about all file system checks that are currently running, type:

```bash
$ nas_fsck -info -all
```

- name = ufs2
- id = 23
- volume = v134
- fsck_server = server_5
- inode_check_percent = 30
- directory_check_percent = 0
- used_ACL_check_percent = 0
- free_ACL_check_status = Not Started
- cylinder_group_check_status = Not Started

- name = ufs1
- id = 27
- volume = mtv1
- fsck_server = server_2
- inode_check_percent = 100
- directory_check_percent = 0
- used_ACL_check_percent = 0
- free_ACL_check_status = Not Started
- cylinder_group_check_status = Not Started

**EXAMPLE #1** provides a description of command outputs.
**The nas Commands**

---

### nas_halt

Performs a controlled halt of all Control Stations and Data Movers in the VNX.

**SYNOPSIS**

```
# nas_halt now
```

**DESCRIPTION**

`nas_halt` performs an orderly shutdown of the Control Stations and Data Movers in the VNX. `nas_halt` must be executed from the `/nas/sbin` directory.

**OPTIONS**

`now`

Performs an immediate halt for the VNX.

**SEE ALSO**

`VNX System Operations` and `server_cpu`.

**EXAMPLE #1**

To perform an immediate halt of the VNX, type:

```
# /nas/sbin/nas_halt now
```

**usage:**

```
usage: nas_halt now
```

Perform a controlled halt of the Control Stations and Data Movers

```
# /nas/sbin/nas_halt now
```

--------------------------------------------- WARNING! ---------------------------------------------

You are about to HALT this VNX including all of its Control Stations and Data Movers. DATA will be UNAVAILABLE when the system is halted. Note that this command does *not* halt the storage array.

ARE YOU SURE YOU WANT TO CONTINUE? [yes or no] : yes

Sending the halt signal to the Master Control Daemon...: Done

```
May  3 11:12:54 cs100 EMCServer: nas_mcd: Check and halt other CS...: Done
```

```
May  3 11:13:26 cs100 JSERVER: *** Java Server is exiting ***
```

```
May  3 11:13:31 cs100 ucd-snmp[11218]: Received TERM or STOP signal... shutting down...
```

```
May  3 11:13:31 cs100 snmpd: snmpd shutdown succeeded
```

```
May  3 11:13:32 cs100 setup_enclosure: Executing -dhcpd stop option
```

```
May  3 11:13:32 cs100 snmptrapd[11179]: Stopping snmptrapd
```

```
May  3 11:13:32 cs100 EV_AGENT[13721]: Signal TERM received
```

```
May  3 11:13:32 cs100 EV_AGENT[13721]: Agent is going down
```

```
May  3 11:13:40 cs100 DHCPDMON: Starting DHCPD on CS 0
```

```
May  3 11:13:41 cs100 setup_enclosure: Executing -dhcpd start option
```

```
May  3 11:13:41 cs100 dhcpd: Internet Software Consortium DHCP Server V3.0pl1
```

```
```

```
May  3 11:13:41 cs100 dhcpd: All rights reserved.
```

```
May  3 11:13:41 cs100 dhcpd: For info, please visit
```

```
http://www.isc.org/products/DHCP
```

---
May  3 11:13:41 cs100 dhcpd: Wrote 0 deleted host decls to leases file.
May  3 11:13:41 cs100 dhcpd: Wrote 0 new dynamic host decls to leases file.
May  3 11:13:41 cs100 dhcpd: Wrote 0 leases to leases file.
May  3 11:13:41 cs100 dhcpd: Listening on LPF/eth2/00:00:f0:9d:04:13/128.221.253.0/24
May  3 11:13:41 cs100 dhcpd: Sending on LPF/eth2/00:00:f0:9d:04:13/128.221.253.0/24
May  3 11:13:41 cs100 dhcpd: Listening on LPF/eth0/00:00:f0:9d:01:e5/128.221.252.0/24
May  3 11:13:41 cs100 dhcpd: Sending on LPF/eth0/00:00:f0:9d:01:e5/128.221.252.0/24
May  3 11:13:41 cs100 dhcpd: Sending on Socket/fallback/fallback-net
May  3 11:13:59 cs100 mcd_helper: : Failed to umount /nas (0)
May  3 11:13:59 cs100 EMCServer: nas_mcd: Failed to gracefully shutdown MCD and halt servers. Forcing halt and reboot...
May  3 11:13:59 cs100 EMCServer: nas_mcd: Halting all servers...
May  3 11:15:00 cs100 get_datamover_status: Data Mover server_5: COMMAND doesnt match.
The nas Commands

nas_inventory

Provides detailed information about hardware components in the system.

SYNOPSIS

nas_inventory

\{ 
  -list [-location]
  | { -info <location> | -all }
  -tree
\}

DESCRIPTION

nas_inventory displays detailed information about the hardware components that are configured on a system.

OPTIONS

-list
Displays a list of all hardware components and their associated name, type, status, and system ID.

  [-location]
  Displays the location string for each component in the output.
  The location string is a unique identifier for the component.

  Specifies the location string with enclosed double quotes (" ") and displays a list of detailed information for the specific component for which the string is the unique ID.

-info <location_string> | -all
Displays a list of all the properties for a component, including the component name, type, status, variant, associated storage system, serial number, part number, and history.

The -all option lists detailed information for all components in the system.

-tree
Displays a hierarchical tree of components, including the status of each component.
### EXAMPLE #1

To display a list of components on the system, type:

```
$ nas_inventory -list
```

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Status</th>
<th>System ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery A</td>
<td>Battery</td>
<td>OK</td>
<td>CLARiiON CX4-240 FCNTR083000055001A</td>
</tr>
<tr>
<td>VNX NS40G FCNTR083000055001A</td>
<td>VNX</td>
<td>Warning</td>
<td>VNX NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>CLEARiiON CX4-240</td>
<td>CLEARiiON</td>
<td>OK</td>
<td>CLEARiiON CX4-240 FCNTR083000055001A</td>
</tr>
<tr>
<td>DME 0 Data Mover 2</td>
<td>Data Mover</td>
<td>OK</td>
<td>VNX NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>Ethernet Module</td>
<td>Module</td>
<td>OK</td>
<td>VNX NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>DME 0 Data Mover 2</td>
<td>SFP</td>
<td>OK</td>
<td>VNX NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>SFP BE0</td>
<td>SFP</td>
<td>OK</td>
<td>VNX NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>SFP BE1</td>
<td>SFP</td>
<td>OK</td>
<td>VNX NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>SFP FE0</td>
<td>SFP</td>
<td>OK</td>
<td>VNX NS40G FCNTR083000055001A</td>
</tr>
</tbody>
</table>

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>Description of the component.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of component. Possible types are: battery, blower, VNX, Control Station, Data Mover, and disk.</td>
</tr>
<tr>
<td>Status</td>
<td>The current status of the component. Status is component type specific. There are several possible status values, each of which is associated with a particular component type.</td>
</tr>
<tr>
<td>System ID</td>
<td>The identifier for the VNX or the storage ID of the storage system containing the component.</td>
</tr>
</tbody>
</table>

### EXAMPLE #2

To display a list of components and component locations, type:

```
$ nas_inventory -list -location
```

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Status</th>
<th>System ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery A</td>
<td>Battery</td>
<td>OK</td>
<td>CLARiiON CX4-240 FCNTR083000055001A</td>
</tr>
<tr>
<td>system:NS40G:FCNTR083000055001A</td>
<td>clarionSystem:CX4-240:FCNTR083000055</td>
<td>sps::A</td>
<td></td>
</tr>
<tr>
<td>Celerra NS40G FCNTR083000055001A Celerra</td>
<td>Warning</td>
<td>Celerra NS40G FCNTR083000055001A</td>
<td></td>
</tr>
<tr>
<td>CLEARiiON CX4-240 FCNTR083000055001A CLRiiON</td>
<td>OK</td>
<td>CLEARiiON CX4-240 FCNTR083000055001A</td>
<td></td>
</tr>
<tr>
<td>system:NS40G:FCNTR083000055001A</td>
<td>clarionSystem:CX4-240:FCNTR083000055</td>
<td>sps::A</td>
<td></td>
</tr>
<tr>
<td>DME 0 Data Mover 2</td>
<td>Data Mover</td>
<td>OK</td>
<td>Celerra NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>system:NS40G:FCNTR083000055001A</td>
<td>enclosure:xpe:0</td>
<td>mover:NS40:2</td>
<td></td>
</tr>
<tr>
<td>DME 0 Data Mover 2 Ethernet Module</td>
<td>Module</td>
<td>OK</td>
<td>Celerra NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>system:NS40G:FCNTR083000055001A</td>
<td>module:ethernet:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DME 0 Data Mover 2 SFP BE0</td>
<td>SFP</td>
<td>OK</td>
<td>Celerra NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>system:NS40G:FCNTR083000055001A</td>
<td>enclosure:xpe:0</td>
<td>module:NS40:2</td>
<td>sfp::BE0</td>
</tr>
<tr>
<td>DME 0 Data Mover 2 SFP BE1</td>
<td>SFP</td>
<td>OK</td>
<td>Celerra NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>system:NS40G:FCNTR083000055001A</td>
<td>enclosure:xpe:0</td>
<td>module:NS40:2</td>
<td>sfp::BE1</td>
</tr>
<tr>
<td>DME 0 Data Mover 2 SFP FE0</td>
<td>SFP</td>
<td>OK</td>
<td>Celerra NS40G FCNTR083000055001A</td>
</tr>
<tr>
<td>system:NS40G:FCNTR083000055001A</td>
<td>enclosure:xpe:0</td>
<td>module:NS40:2</td>
<td>sfp::FE0</td>
</tr>
</tbody>
</table>
**EXAMPLE #3** To list information for a specific component, type:

```
$ nas_inventory -info "system:NS40G:FCNTR083000055001A|clariionSystem:CX4-240:FCNTR083000055|iomodule::B0"
```

Location = system:NS40G:FCNTR083000055001A|clariionSystem:CX4-240:
Component Name = IO Module B0
Type = IO Module
Status = OK
Variant = 4 PORT FIBRE IO MODULE
Storage System = CLARiiON CX4-240 FCNTR083000055
Serial Number = CF2YW082800426
Part Number = 103-054-100C
History = EMC_PART_NUMBER:103-054-100C
EMC_ARTWORK_REVISION:C01
EMC_ASSEMBLY_REVISION:C03
EMC_SERIAL_NUMBER:CF2YW082800426
VENDER_PART_NUMBER:N/A
VENDER_ARTWORK_NUMBER:N/A
VENDER_ASSEMBLY_NUMBER:N/A
VENDER_SERIAL_NUMBER:N/A
VENDOR_NAME:N/A
LOCATION_OF_MANUFACTURE:N/A
YEAR_OF_MANUFACTURE:N/A
MONTH_OF_MANUFACTURE:N/A
DAY_OF_MONTH_OF_MANUFACTURE:N/A
ASSEMBLY_NAME:4 PORT FIBRE IO MODULE

**Note:** The location string must be enclosed in double quotes.

**Where:**

<table>
<thead>
<tr>
<th><strong>Value</strong></th>
<th><strong>Definition</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>The unique identifier of the component and where the component is located in the component hierarchy.</td>
</tr>
<tr>
<td>Component</td>
<td>The description of the component.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of component. Possible types are: battery, blower, VNX for file, VNX for block, Control Station, Data Mover, and disk.</td>
</tr>
<tr>
<td>Status</td>
<td>The current condition of the component. Status is component type specific. There are several possible status values, each of which is associated with a particular component type.</td>
</tr>
<tr>
<td>Variant</td>
<td>The specific type of hardware.</td>
</tr>
<tr>
<td>Storage System</td>
<td>The model and serial number of the storage system.</td>
</tr>
<tr>
<td>Serial Number</td>
<td>The serial number of the hardware component.</td>
</tr>
<tr>
<td>Part Number</td>
<td>The part number of the hardware component.</td>
</tr>
<tr>
<td>History</td>
<td>If available, the history information of the component. Possible values are: part number, serial number, vendor, date of manufacture, and CPU information.</td>
</tr>
</tbody>
</table>
### EXAMPLE #4

To display components in a tree structure, type:

```
$ nas_inventory -tree
```

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Celerra NS40G FCNTR083000055001A</td>
<td>Celerra</td>
<td>Warning</td>
</tr>
<tr>
<td>CLARiiON CX4-240 FCNTR083000005</td>
<td>CLARiiON</td>
<td>OK</td>
</tr>
<tr>
<td>Battery A</td>
<td>Battery</td>
<td>OK</td>
</tr>
<tr>
<td>IO Module A0</td>
<td>IO Module</td>
<td>OK</td>
</tr>
<tr>
<td>IO Module A1</td>
<td>IO Module</td>
<td>OK</td>
</tr>
<tr>
<td>IO Module A2</td>
<td>IO Module</td>
<td>Empty</td>
</tr>
<tr>
<td>IO Module A3</td>
<td>IO Module</td>
<td>Empty</td>
</tr>
<tr>
<td>IO Module A4</td>
<td>IO Module</td>
<td>Empty</td>
</tr>
<tr>
<td>IO Module B0</td>
<td>IO Module</td>
<td>OK</td>
</tr>
<tr>
<td>IO Module B1</td>
<td>IO Module</td>
<td>OK</td>
</tr>
<tr>
<td>IO Module B2</td>
<td>IO Module</td>
<td>Empty</td>
</tr>
<tr>
<td>IO Module B3</td>
<td>IO Module</td>
<td>Empty</td>
</tr>
<tr>
<td>IO Module B4</td>
<td>IO Module</td>
<td>Empty</td>
</tr>
<tr>
<td>Power Supply A0</td>
<td>Power Supply</td>
<td>OK</td>
</tr>
<tr>
<td>Power Supply A1</td>
<td>Power Supply</td>
<td>OK</td>
</tr>
<tr>
<td>Power Supply B0</td>
<td>Power Supply</td>
<td>OK</td>
</tr>
<tr>
<td>Power Supply B1</td>
<td>Power Supply</td>
<td>OK</td>
</tr>
</tbody>
</table>
The nas Commands

nas_license

Enables software packages.

SYNOPSIS

nas_license
  -list
  -create <package_name>[=|<key_code>]
  -delete <package_name>
  -init

DESCRIPTION

nas_license enables software packages that are available for use with the system. The <key_code> is supplied by EMC.

All entries are case-sensitive.

OPTIONS

No arguments

Displays a usage message containing all available and valid software packages that can be installed.

-list

Displays the site_key as a string and any software packages for which a license has been installed. The site_key is a permanent license and cannot be deleted. The failover license is legacy and should only be used only when instructed to do so by EMC personnel.

Note: Licenses installed on the Control Station are read by the system. The site_key is a unique identifier which gets generated the first time nas_license is run. The site_key is also used to decode the key_code supplied by EMC personnel for special packages.

-create <package_name>[=|<key_code>]

Installs the license for the indicated <package_names>. Valid <package_names> are:

site key
nfs
cifs
failover
snapsure
advancedmanager
replicator
iscsi
replicatorV2
filelevelretention
The nas Commands

Note: These packages do not require key_code as they can be enabled from the GUI. Special packages are supplied along with the required Key_code by the EMC Customer Service Representative. The package failover requires key_code.

**-delete** `<package_name>`
Deletes the license for the specified `<package_name>`.

**-init**
Initializes the database and re-creates the license file by using the site_key that is already installed. The license file is located at /nas/site as nas_license. It contains license keys in an encrypted format. The `-init` option should be run only if the license file containing all the license information has been lost and the following error message is received:

```
license table is not initialized
```

Once the license file has been re-created, the rest of the entries, if present, should be re-added by using the `-create` option.

**EXAMPLE #1**
To install a license for the iscsi software package, type:

```
$ nas_license -create iscsi
done
```

**EXAMPLE #2**
To display all software packages with currently installed licenses, type:

```
$ nas_license -list

key                    status            value
site_key               online            42 de 6f d1
advancedmanager        online
nfs                    online
iscsi                  online
snapsure               online
replicator             online
replicatorV2           online
filelevelretention     online
```

**EXAMPLE #3**
To delete a license for specified software package, type:

```
$ nas_license -delete iscsi
done
```

**EXAMPLE #4**
To initialize the database and re-create the license file, type:

```
$ nas_license -init
done
```
The nas Commands

nas_logviewer

Displays the content of nas_eventlog generated log files.

SYNOPSIS

nas_logviewer <file_name>
[-f][-v][-t]

DESCRIPTION

nas_logviewer displays the event log and other logs created by nas_eventlog. The log files may be system generated, or created by the user. Information in the log file is read from oldest to newest.

OPTIONS

No arguments
Displays the contents of the specified logfile.

-f
Monitors the growth of the log by entering into an endless loop, pausing and reading the log as it is being generated. To exit, press Ctrl-C together.

[-v|-t]
Displays the log files in verbose or terse format.

SEE ALSO

Configuring Events and Notifications on VNX for File and server_log.

EXAMPLE #1

To view the contents of the sys_log file, type:

$ nas_logviewer /nas/log/sys_log|more

May 12 18:01:57 2007:CS_PLATFORM:NASDB:INFO:300:::nasdb_backup: NAS_DB checkpoint in progress
May 12 18:03:00 2007:CS_PLATFORM:NASDB:ERROR:202:::NAS database error detected
May 12 20:01:57 2007:CS_PLATFORM:NASDB:INFO:300:::nasdb_backup: NAS_DB checkpoint in progress
May 12 20:03:10 2007:
The nas Commands

Note: This is a partial listing due to the length of the outputs.

EXAMPLE #2  To display the contents of the log files in terse format, type:

```
$ nas_logviewer -t /nas/log/sys_log
```

logged time = May 12 18:01:57 2007
creation time = May 12 18:01:57 2007
slot id =
id = 96108871980
severity = INFO
component = CS_PLATFORM
facility = NASDB
baseid = 300
type = EVENT
brief discription = nasdb_backup: NAS_DB checkpoint in progress
full discription = The Celerra configuration database is being checkpoined.
recommended action = No action required.

logged time = May 12 18:02:59 2007
creation time = May 12 18:02:59 2007
slot id =
id = 96108871985
severity = INFO
component = CS_PLATFORM
facility = NASDB

EXAMPLE #3  To display the contents of the log files in verbose format, type:

```
$ nas_logviewer -v /nas/log/sys_log|more
```

logged time = May 12 18:01:57 2007
creation time = May 12 18:01:57 2007
slot id =
id = 96108871980
severity = INFO
component = CS_PLATFORM
facility = NASDB
baseid = 300
type = EVENT
brief discription = nasdb_backup: NAS_DB checkpoint in progress
full discription = The Celerra configuration database is being checkpoined.
recommended action = No action required.

logged time = May 12 18:02:59 2007
creation time = May 12 18:02:59 2007
slot id =
id = 96108871985
severity = INFO
component = CS_PLATFORM
facility = NASDB
baseid = 305
type = EVENT
brief description = nasdb_backup: NAS_DB Checkpoint done
full description = The NAS DB backup has completed a checkpoint of the current
reparation for performing a backup of NAS system data.
recommended action = No action required.

EXAMPLE #4 To monitor the growth of the current log, type:

$ nas_logviewer -f /nas/log/sys_log|more

May 12 18:01:57 2007:CS_PLATFORM:NASDB:INFO:300:::nasdb_backup: NAS_DB
checkpoint in progress
May 12 18:02:59 2007:CS_PLATFORM:NASDB:INFO:305:::nasdb_backup: NAS_DB
Checkpoint done
May 12 18:03:00 2007:CS_PLATFORM:NASDB:ERROR:202:::NAS database error
detected
Backup done

checkpoint in progress
Checkpoint done
detected
Backup done

checkpoint in progress
Checkpoint done
detected
Backup done

checkpoint in progress
Checkpoint done
detected
Backup done
nas_message

 Displays message description.

 SYNOPSIS
 nas_message
    -info <MessageId>

 DESCRIPTION
 nas_message provides detailed descriptions to a dedicated message.
 A brief description, full description, and recommended user action of
 the message are displayed.

 OPTIONS
    -info <MessageId>
 Displays detailed descriptions of the error message, including
 severity, component, facility, BaseID, and recommended user action.
 The message parameters are displayed in the form stateDesc,8,%s
 and not as parameter values. The <MessageId> must be a positive
 integer.

 SEE ALSO
 Celerra Network Server Error Messages Guide.

 EXAMPLE #1
 To display detailed descriptions for error message 13421838337, type:

 $ nas_message -info 13421838337

 MessageID = 13421838337
 BaseID    = 1
 Severity  = ERROR
 Component = CS_CORE
 Facility  = default
 Type      = STATUS
 Brief_Description  = Operation not permitted$(arg0,8,%s)
 Full_Description   = The operation is not permitted due to an ACL or ownership
 issue on the specified object.
 Recommended_Action = Check ownership or ACL of the object in question.If
 appropriate change the setting to resolve the conflict. Refer to the nas_acl and
 chmod man page.
The nas Commands

nas_mview

Performs MirrorView/Synchronous (MirrorView/S) operations on a system attached to an older version of VNX for block.

SYNOPSIS

nas_mview
  -info
  -init <cel_name>
  -activate
  -restore

DESCRIPTION

nas_mview retrieves MirrorView/S cabinet-level information, initializes the source and destination systems for MirrorView/S, activates a failover to a destination VNX for file, or restores the source site after a failover.

MirrorView/S is supported on a system attached to an older version of VNX for block array serving as the boot storage, not the secondary storage. nas_mview must be run from a Control Station in slot 0; it will report an error if run from a Control Station in slot 1.

nas_mview must be issued as root from the /nas/sbin directory. For the -init and -info options, log in with your administrative username and use the su root command to log in as root. For the -activate and -restore options, you must log in to the destination system using the remote administration account (for example, dradmin) and log in as root.

OPTIONS

- info
Displays disaster recovery information such as the MirrorView/S device group eligible, displays the MirrorView/S Data Mover configuration for the current system.

-init <cel_name>
Initializes the MirrorView/S relationship between the source and destination systems based on if the configuration is active/passive (unidirectional) or active/active’ (bidirectional).

Note: The apostrophe in active/active’ indicates that both sites have source LUNs mirrored at the other site.

The passphrase-protected relationship between the source and destination systems in the MirrorView/S configuration must be built prior to initialization using the nas_cel -create command:
On the destination Control Station in a MirrorView/S active/passive configuration, use the `-init` option to specify the name of the source system.

On the Control Station of each system in a MirrorView/S active/active’ configuration, use the `-init` option to specify the name of the remote system. The active/active configuration is a bidirectional configuration in which a VNX for file can serve both as source and destination for another system.

**-activate**

Executed from the destination system using the remote administration account, initiates a failover from the source to the destination system. The activation works as follows:

- If the source is available, the `-activate` option swaps the primary-secondary role for all mirrors in the MirrorView/S device group and makes the destination LUNs read/write. The standby Data Movers acquire the IP and MAC addresses, file systems, and export tables of their source counterparts.

- If the original source site is unavailable, the destination LUNs are promoted to the primary role, making them visible to the destination VNX for file. The original source LUNs cannot be converted to backup images; they stay visible to the source VNX for file, and the original destination site is activated with new source (primary) LUNs only. If the source cannot be shut down in a disaster scenario, any writes occurring after the forced activation will be lost during a restore.

**-restore**

Issued from the destination system using the remote administration account, restores a source system after a MirrorView/S failover, and fails back the device group to the source system.

The restore process begins by checking the state of the device group. If the device group state is Local Only (where each mirror has only the source LUN), the device group will be fully synchronized and rebuilt before the failback can occur. If the device group condition is fractured, an incremental synchronization is performed before the failback occurs. Source devices are then synchronized with the data on the original destination devices, I/O access is shut down, the original destination Data Movers are rebooted as remote standbys, and the mirrored devices are failed back. When the source side is restored, the source Data Movers and their services are restarted.
If the restore fails, the source Control Station is not reachable on the data network. To complete the restore, access the source, log in as root, and type `/nasmd/sbin/nas_mview -restore`.

**SEE ALSO**  
*Using MirrorView/Synchronous with VNX for Disaster Recovery*, *nas cel*, and *nas_checkup*.

**STORAGE SYSTEM OUTPUT**  
The number associated with the storage device reflects the attached storage system; for MirrorView/S, VNX for block displays a prefix of APM before a set of integers, for example, APM00033900124-0019.

The VNX for block supports the following system-defined AVM storage pools for MirrorView/S only: `cm_r1`, `cm_r5_performance`, `cm_r5_economy`, `cmata_archive`, `cmata_r3`, `cm_r6`, and `cmata_r6`.

**EXAMPLE #1**  
To initialize a destination VNX for file in an active/passive configuration to communicate with source site source_cs, from the destination Control Station, type:

```
# /nas/sbin/nas_mview -init source_cs
```

**Ce1lerra with MirrorView/Synchronous Disaster Recovery**

Initializing source_cs --> target_cs

Contacting source_cs for remote storage info

Local storage system: APM00053001549
Remote storage system: APM00053001552

Enter the Global CLARiiON account information
Username: emc
Password: *** Retype your response to validate
Password: ***

Discovering storage on source_cs (may take several minutes)
Setting security information for APM00053001549
Discovering storage APM00053001552 (may take several minutes)

Discovering storage (may take several minutes)

Contacting source_cs for remote storage info
Gathering server information...
Contacting source_cs for server capabilities...
Analyzing server information...

Source servers available to be configured for remote DR
-------------------------------------------------------------------------------
1. server_2:source_cs
2. server_3:source_cs [ local standby ]
3. Verify standby server configuration
The nas Commands

q. Quit initialization process
c. Continue initialization

Select a source_cs server: 1

Destination servers available to act as remote standby

1. server_2:target_cs [ unconfigured standby ]
2. server_3:target_cs [ unconfigured standby ]
b. Back

Select a target_cs server: 1

Source servers available to be configured for remote DR

1. server_2:source_cs [ remote standby is server_2:target_cs ]
2. server_3:source_cs [ local standby ]
v. Verify standby server configuration
q. Quit initialization process
c. Continue initialization

Select a source_cs server: 2

Destination servers available to act as remote standby

server_2:target_cs [ is remote standby for server_2:source_cs ]
2. server_3:target_cs [ unconfigured standby ]

b. Back

Select a target_cs server: 2

Source servers available to be configured for remote DR

1. server_2:source_cs [ remote standby is server_2:target_cs ]
2. server_3:source_cs [ remote standby is server_3:target_cs ]
v. Verify standby server configuration
q. Quit initialization process
c. Continue initialization

Select a source_cs server: c

Standby configuration validated OK

Enter user information for managing remote site source_cs
Username: dradmin
Password: *******
Retype your response to validate
Password: *******

Active/Active configuration
Initializing (source_cs-->target_cs)

Do you wish to continue? [yes or no] yes

Updating MirrorView configuration cache
Setting up server_3 on source_cs
Setting up server_2 on source_cs
Creating user account dradmin
Setting acl for server_3 on target_cs
Setting acl for server_2 on target_cs
Updating the Celerra domain information
Creating device group mviewgroup on source_cs

done

**EXAMPLE #2**
To get information about a source MirrorView configuration (for example, on new_york configured as active/passive), type:

```
# /nas/sbin/nas_mview -info
```

***** Device Group Configuration *****

```plaintext
name                  = mviewgroup
description           =
uid                   = 50:6:1:60:B0:60:26:BC:0:0:0:0:0:0:0:0
state                 = Consistent
role                  = Primary
condition             = Active
recovery policy       = Automatic
number of mirrors     = 16
mode                  = SYNC
owner                 = 0
mirrored disks        = root_disk,root_ldisk,d5,d8,d10,d11,d24,d25,d26,d27,d29,d30,d31,d32,d33,d39,
local clarid          = APM00053001552
remote clarid         = APM00053001549
mirror direction      = local -> remote
```

***** Servers configured with RDFstandby *****

```plaintext
id       = 1
name     = server_2
acl      = 1000, owner=nasadmin, ID=201
type     = nas
slot     = 2
member_of =
standby  = server_3, policy=auto
RDFstandby= slot=2
status :  
    defined = enabled
    actual = online, active

id       = 2
name     = server_3
acl      = 1000, owner=nasadmin, ID=201
type     = standby
slot     = 3
member_of =
standbyfor= server_2
RDFstandby= slot=3
status :  
```
defined = enabled
actual = online, ready

***** Servers configured as standby *****

No servers configured as standby

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device group configuration:</strong></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>Name of the consistency (device) group.</td>
</tr>
<tr>
<td>description</td>
<td>Brief description of device group.</td>
</tr>
<tr>
<td>uid</td>
<td>UID assigned, based on the system.</td>
</tr>
<tr>
<td>state</td>
<td>State of the device group (for example, Consistent, Synchronized, Out-of-Sync, Synchronizing, Scrambled, Empty, Incomplete, or Local Only).</td>
</tr>
<tr>
<td>role</td>
<td>Whether the current system is the Primary (source) or Secondary (destination) for this group.</td>
</tr>
<tr>
<td>condition</td>
<td>Whether the group is functioning (Active), Inactive, Admin Fractured (suspended), Waiting on Sync, System Fractured (which indicates link down), or Unknown.</td>
</tr>
<tr>
<td>recovery policy</td>
<td>Type of recovery policy (Automatic is the default and recommended value for group during storage system configuration; if Manual is set, you must use -resume after a link down failure).</td>
</tr>
<tr>
<td>number of mirrors</td>
<td>Number of mirrors in group.</td>
</tr>
<tr>
<td>mode</td>
<td>MirrorView mode (always SYNC in this release).</td>
</tr>
<tr>
<td>owner</td>
<td>ACL ID assigned (0 indicates no control). nas_acl provides information.</td>
</tr>
<tr>
<td>mirrored disks</td>
<td>Comma-separated list of disks that are mirrored.</td>
</tr>
<tr>
<td>local clarid</td>
<td>APM number of local VNX for block storage array.</td>
</tr>
<tr>
<td>remote clarid</td>
<td>APM number of remote VNX for block storage array.</td>
</tr>
<tr>
<td>mirror direction</td>
<td>On primary system, local to remote (on primary system); on destination system, local from remote.</td>
</tr>
</tbody>
</table>
EXAMPLE #3  To activate a failover, log in to destination Control Station using `dradmin` account, `su` to `root`, and type:

```
# /nas/sbin/nas_mview -activate
```

Sync with CLARiiON backend ...... done
Validating mirror group configuration ...... done
Is source site source_cs ready for complete shut down (power OFF)? [yes or no] yes
Contacting source site source_cs, please wait... done
Shutting down remote site source_cs .................................................. done
Sync with CLARiiON backend ...... done
STARTING an MV 'FAILOVER' operation.
Device group: mviewgroup ........... done
The MV 'FAILOVER' operation SUCCEEDED.
Failing over Devices ... done
Adding NBS access for server_2 ....... done
Adding NBS access for server_3 ....... done
Activating the target environment ... done

server_2 : going offline
rdf : going active
replace in progress ...done
failover activity complete

server_3 : going offline
rdf : going active
replace in progress ...done
failover activity complete
commit in progress (not interruptible)...done
commit in progress (not interruptible)...done
commit in progress (not interruptible)...done
commit in progress (not interruptible)...done
done

EXAMPLE #4
To restore, log in to the destination Control Station using `dradmin` account, as `root` user, and type:

```bash
# /nas/sbin/nas_mview -restore
```
Sync with CLARiiON backend ...... done
Validating mirror group configuration ...... done
Contacting source site source_cs, please wait... done
Running restore requires shutting down source site source_cs.
Do you wish to continue? [yes or no] yes
Shutting down remote site source_cs ....... done
Is source site source_cs ready for storage restoration? [yes or no] yes
Sync with CLARiiON backend ...... done
STARTING an MV 'RESUME' operation.
Device group: mviewgroup ............... done
The MV 'RESUME' operation SUCCEEDED.
Percent synchronized: 100
Updating device group ... done
Is source site ready for network restoration? [yes or no] yes
Restoring servers ...... done
Waiting for servers to reboot ...... done
Removing NBS access for server_2 .. done
Removing NBS access for server_3 .. done
Waiting for device group ready to failback .... done
Sync with CLARiiON backend ...... done
STARTING an MV 'FAILBACK' operation.
Device group: mviewgroup ............... done
The MV 'FAILBACK' operation SUCCEEDED.
Restoring remote site source_cs ...... failed
Error 5008: -1:Cannot restore source_cs. Please run restore on site source_cs.

Then on the Source Control Station, as the `root` user, type:

```bash
# /nasmcd/sbin/nas_mview -restore
```
Stopping NAS services. Please wait...

Powering on servers (please wait) ...... done
Sync with CLARiiON backend ...... done
STARTING an MV 'SUSPEND' operation.
Device group: mviewgroup ............... done
The MV 'SUSPEND' operation SUCCEEDED.
servers_2 : going standby
rdf : going active
replace in progress ... done
failover activity complete
servers_3 : going standby
rdf : going active
The nas Commands

replace in progress ...done
failover activity complete
commit in progress (not interruptible)...done
commit in progress (not interruptible)...done
Sync with CLARiiON backend ...... done
STARTING an MV 'RESUME' operation.
Device group: mviewgroup ............ done
The MV 'RESUME' operation SUCCEEDED.
Restarting NAS services ...... done
commit in progress (not interruptible)...done
commit in progress (not interruptible)...done
done
nas_pool

Manages the user-defined and system-defined storage pools for the system.

SYNOPSIS

nas_pool
    -list
    | -info {<name>|id=<id>|-all} [-storage <system_name>]
    | -size {<name>|id=<id>|-all} [-mover <mover>][\-slice \{y|n\}]
      [-storage <system_name>]
    | -create [-name <name>] [-acl <acl>] [-description <desc>]
      [-volumes <volume_name>,<volume_name>,...]
      [-default_slice_flag \{y|n\}]
    | -create [-name <name>] [-acl <acl>] [-description <desc>]
      [-default_slice_flag \{y|n\}]
      -size <integer>[M|G|T][\-storage <system_name>]
    | -template <system_pool_name> [-num_stripe_members <num>]
      [-stripe_size <num>]
    | -modify {<name>|id=<id>} [-storage <system_name>][-name <name>]
      [-acl <acl>][-description <desc>][-default_slice_flag \{y|n\}]
      [-is_dynamic \{y|n\}][-is_greedy \{y|n\}]
    | -delete {<name>|id=<id>} [-deep] [-storage <system_name>]
    | -xtend {<name>|id=<id>} [-storage <system_name>]
      [-volumes <volume_name>,<volume_name>,...]
    | -xtend {<name>|id=<id>} -size <integer> [M|G|T][-storage <system_name>]
    | -shrink {<name>|id=<id>} [-storage <system_name>] -volumes
      <volume_name>[,<volume_name>,...][-deep]

DESCRIPTION

    nas_pool creates, deletes, extends, shrinks, lists, displays, manages
    the access control level, and modifies a user-defined storage pool.

    nas_pool extends, shrinks, lists, displays, and modifies
    system-defined storage pools.

OPTIONS

- list
    Lists all storage pools on the system.

    - info {<name>|id=<id>|-all} [-storage <system_name>]
        Displays detailed information for the specified storage pool, or all
        storage pools. The -storage option can be used to differentiate pools
        when the same pool name is used in multiple storage systems.

    - size {<name>|id=<id>|-all}
        Displays the size for the specified storage pool, or all storage pools.
The nas Commands

[-mover <mover>]
Displays size information that is visible to the physical Data Mover or the virtual Data Mover (VDM).

[-slice {y|n}]
If y is typed, displays size information when volumes in the storage pool are sliced. If n is typed, displays size information when volumes in the storage pool are not sliced. The -slice option defaults to the value of default_slice_flag for the storage pool.

[-storage <system_name>]
Displays size information for members that reside on a specified storage system.

-create
Creates a user-defined storage pool.

[-name <name>]
Assigns a name to the new storage pool. If no name is specified, assigns one by default.

[-acl <acl>]
Sets an access control level value that defines the owner of the storage pool, and the level of access allowed for users and groups defined in the access control level table. The nas_acl command provides more information.

[-description <desc>]
Assigns a comment to the storage pool.

[-volumes <volume_name>[,<volume_name>,...]]
Designates the members to be added to the storage pool. The members can be any meta, slice, stripe, or disk volumes.

[-default_slice_flag {y|n}]
If set to y (default), then members of the storage pool might be sliced when space is allocated from the storage pool. If set to n, members of the storage pool will not be sliced when space is dispensed from the storage pool and the volumes specified cannot be built on a slice.

[-size <integer> {M|G|T}]
Creates a storage pool with the size specified. When specifying a size, type an integer between 1 and 1024, then specify T for terabytes, G for gigabytes (default), or M for megabytes.
The `nas` Commands

```
[-storage <system_name>]
Specifies the storage system on which one or more volumes will be created, to be added to the storage pool.

[-template <system_pool_name>]
Specifies a system pool name, required when the `-size` option is specified. The user pool will be created using the profile attributes of the specified system pool template.

[-num_stripe_members <num>]
Specifies the number of stripe members for user pool creation by size. The `-num_stripe_members` option works only when both `-size` and `-template` options are specified. It overrides the number of stripe members attribute of the specified system pool template.

[-stripe_size <num>]
Specifies the stripe size for user pool creation by size. The `-stripe_size` option works only when both `-size` and `-template` options are specified. It overrides the stripe size attribute of the specified system pool template.

-modify {<name>|id=<id>} [-storage <system_name>]
Modifies the attributes of the specified user-defined or system-defined storage pool. The `-storage` option can be used to differentiate pools when the same pool name is used in multiple storage systems.

Managing Volumes and File Systems with VNX Automatic Volume Management lists the available system-defined storage pools.

```
[-name <name>]
Changes the name of the storage pool to the new name.

[-acl <acl>]
Sets an access control level value that defines the owner of the storage pool, and the level of access allowed for users and groups defined in the access control level table. The `nas_acl` command provides more information.

[-description <desc>]
Changes the comment for the storage pool.

[-default_slice_flag {y|n}]
If set to y (default), then members of the storage pool might be sliced when space is dispensed from the storage pool. If set to n, members of the storage pool will not be sliced when space is dispensed from the storage pool and the volumes specified cannot be built on a slice.
```
The nas Commands

[-is_dynamic {y|n}]
Allows a system-defined storage pool to automatically extend or shrink member volumes.

---

Note: The -is_dynamic option is for system-defined storage pools only.

[-is_greedy {y|n}]
If set to y, then the system-defined storage pool attempts to create new member volumes before using space from existing member volumes. A storage pool that is not greedy (set to n), consumes all the space existing in the storage pool before trying to add additional member volumes. A y or n value must be specified when modifying a system-defined storage pool.

---

Note: The -is_greedy option is for system-defined storage pools only.

-delete {<name>|id=<id>} [-storage <system_name>]
Deletes a storage pool. Storage pools cannot be deleted if any members are in use. After deletion, the storage pool no longer exists on the system, however, members of the storage pool are not deleted. The -storage option can be used to differentiate pools when the same pool name is used in multiple storage systems.

[-deep]
Deletes the storage pool and also recursively deletes each member of the storage pool. Each storage pool member is deleted unless it is in use or is a disk volume.

-xtend {<name>|id=<id>} [-storage <system_name>]
-volumes <volume_name>[, <volume_name>,...]
Adds one or more unused volumes to a storage pool. The -storage option can be used to differentiate pools when the same pool name is used in multiple storage systems. If the default_slice_value is set to n, member volumes cannot contain slice volumes (for compatibility with TimeFinder/FS).

---

Note: Extending a storage pool by volume is for user-defined storage pools only.

-xtend {<name>|id=<id>} -size <integer> [M|G|T]
Extends the specified storage pool with one or more volumes of the size equal to or greater than the size specified. When specifying the volume by size, type an integer between 1 and 1024, then specify T for terabytes, G for gigabytes (default), or M for megabytes.
[\textbf{-storage } <\textsf{system\_name}>]\\ Specifies the storage system on which one or more volumes will be created, to be added to the storage pool.

\textbf{Note}: To successfully extend a system-defined storage pool by size, the \texttt{is\_dynamic} attribute must be set to \texttt{n}, and there must be enough available disk volumes to satisfy the request.

\textbf{-shrink} \{<\texttt{name}>|\texttt{id}=<\texttt{id}>\} [\textbf{-storage} <\textsf{system\_name}>] \textbf{-volumes} <\texttt{volume\_name}>,<\texttt{volume\_name},...>[\textbf{-deep}]\\ Shrinks the storage pool by the specified unused volumes. The -storage option can be used to differentiate pools when the same pool name is used in multiple storage systems. When the -deep option is used to shrink a user-defined storage pool, it removes the specified member volumes from the pool, and recursively deletes any unused volumes unless it is a disk volume. If the -deep option is not used to shrink a user-defined storage pool, the member volumes are left intact so that they can be reused. The \texttt{is\_dynamic} option must be set to \texttt{n} before shrinking system-defined storage pools.

\textbf{Note}: Shrinking of a system-defined storage pool by default deletes member volumes automatically. Specifying the -deep option on the system-defined storage pool shrink does not make any difference.

\textbf{SEE ALSO} Managing Volumes and File Systems with VNX Automatic Volume Management, Managing Volumes and File Systems for VNX Manually, Controlling Access to System Objects on VNX, Using TimeFinder/FS, NearCopy, and FarCopy on VNX for File, \texttt{fs\_timefinder}, \texttt{nas\_fs}, \texttt{nas\_volume}, and \texttt{nas\_slice}.

\textbf{STORAGE SYSTEM OUTPUT} VNX for block supports the following traditional system-defined storage pools: \texttt{clar\_r1}, \texttt{clar\_r5\_performance}, \texttt{clar\_r5\_economy}, \texttt{clar\_r6}, \texttt{clarata\_r3}, \texttt{clarata\_r6}, \texttt{clarata\_r10}, \texttt{clarata\_archive}, \texttt{cm\_r1}, \texttt{cm\_r5\_performance}, \texttt{cm\_r5\_economy}, \texttt{cm\_r6}, \texttt{cmata\_r3}, \texttt{cmata\_archive}, \texttt{cmata\_r6}, \texttt{cmata\_r10}, \texttt{clarsas\_archive}, \texttt{clarsas\_r6}, \texttt{clarsas\_r10}, \texttt{clarefd\_r5}, \texttt{clarefd\_r10}, \texttt{cmsas\_archive}, \texttt{cmsas\_r6}, \texttt{cmsas\_r10}, and \texttt{cmefd\_r5}.

A mapped pool was formerly called a thin or virtual pool.

Disk types when using VNX for block are \texttt{CLSTD}, \texttt{CLEFD}, \texttt{CLATA}, \texttt{MIXED} (indicates that tiers used in the pool contain multiple disk types), \texttt{Performance}, \texttt{Capacity}, and \texttt{Extreme\_performance} and for VNX for block involving mirrored disks are: \texttt{CMEFD}, \texttt{CMSTD}, \texttt{CMATA}, \texttt{CMATA\_archive}, \texttt{CMATA\_r6}, \texttt{CMATA\_r10}, \texttt{CMSAS\_archive}, \texttt{CMSAS\_r6}, \texttt{CMSAS\_r10}, and \texttt{CMefd\_r5}.
CMATA, Mirrored_mixed, Mirrored_performance, Mirrored_capacity, and Mirrored_extreme_performance.

Disk types when using VNX for block are CLSTD, CLEFD, and CLATA, and for VNX for block involving mirrored disks are: CMEFD, CMSTD, and CMATA.

VNX with a Symmetrix storage system support the following system-defined storage pools: symm_std, symm_std_rdf_src, symm_ata, symm_ata_rdf_src, symm_ata_rdf_tgt, symm_std_rdf_tgt, and symm_efd.

For user-defined storage pools, the difference in output is in the disk type. Disk types when using a Symmetrix are STD, R1STD, R2STD, BCV, R1BCV, R2BCV, ATA, R1ATA, R2ATA, BCVA, R1BCA, R2BCA, EFD, BCVMIXED, R1MIXED, R2MIXED, R1BCVMIXED, and R2BCVMIXED.

**EXAMPLE #1**

To create a storage pool with the name, *marketing*, with a description, with the following disk members, *d12, d13*, and with the default slice flag set to *y*, type:

```
$ nas_pool -create -name marketing -description 'Storage Pool' -volumes d12,d13 -default_slice_flag y
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the storage pool.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the storage pool.</td>
</tr>
<tr>
<td>description</td>
<td>Comment assigned to the storage pool.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value assigned to the storage pool.</td>
</tr>
<tr>
<td>in_use</td>
<td>Whether the storage pool is being used by a file system.</td>
</tr>
<tr>
<td>clients</td>
<td>File systems using the storage pool.</td>
</tr>
<tr>
<td>members</td>
<td>Volumes used by the storage pool.</td>
</tr>
<tr>
<td>default_slice_flag</td>
<td></td>
</tr>
<tr>
<td>is_user_defined</td>
<td></td>
</tr>
<tr>
<td>disk_type</td>
<td></td>
</tr>
<tr>
<td>server_visibility</td>
<td></td>
</tr>
<tr>
<td>template_pool</td>
<td></td>
</tr>
<tr>
<td>num_stripe_members</td>
<td></td>
</tr>
<tr>
<td>stripe_size</td>
<td></td>
</tr>
</tbody>
</table>

Where:

- **id**: ID of the storage pool.
- **name**: Name of the storage pool.
- **description**: Comment assigned to the storage pool.
- **acl**: Access control level value assigned to the storage pool.
- **in_use**: Whether the storage pool is being used by a file system.
- **clients**: File systems using the storage pool.
- **members**: Volumes used by the storage pool.
default_slice_flag | Allows slices from the storage pool.
is_user_defined   | User-defined as opposed to system-defined.
disk_type         | Type of disk contingent on the storage system attached. CLSTD, CLATA, CMSTD, CLEFD, CMEFD, CMATA, MIXED (indicates tiers used in the pool contain multiple disk types), Performance, Capacity, Extreme_performance, Mirrored_mixed, Mirrored_performance, Mirrored_capacity, and Mirrored_extreme_performance are for VNX for block, and STD, BCV, R1BCV, R2BCV, R1STD, R2STD, ATA, R1ATA, R2ATA, BCVA, R1BCA, R2BCA, EFD, BCVMIXED, R1MIXED, R2MIXED, R1BCVMIXED, and R2BCVMIXED are for Symmetrix.

server_visibility | Storage pool is visible to the physical Data Movers specified.
template_pool     | System pool template used to create the user pool. Only applicable to user pools created by size or if the last member volume is a stripe or both.
num_stripe_members | Number of stripe members used to create the user pool. Applicable to system pools and user pools created by size or if the last member volume is a stripe or both.
stripe_size       | Stripe size used to create the user pool. Applicable to system pools and user pools created by size or if the last member volume is a stripe or both.

**EXAMPLE #2**  
To change the description for the marketing storage pool to include a descriptive comment, type:

```
$ nas_pool -modify marketing -description 'Marketing Storage Pool'
```

id = 20
name = marketing
description = Marketing Storage Pool
acl = 0
in_use = False
clients =
members = d12,d13
default_slice_flag = True
is_user_defined = True
disk_type = CLSTD
server_visibility = server_2,server_3,server_4,server_5
template_pool = N/A
num_stripe_members = N/A
stripe_size = N/A

**EXAMPLE #3**  
To view the size information for the FP1 mapped pool, type:

```
$ nas_pool -size FP1
```

id = 40
name = FP1
used_mb = 0
avail_mb = 0
total_mb = 0
potential_mb = 2047
The nas Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>used_mb</td>
<td>Space in use by the storage pool specified.</td>
</tr>
<tr>
<td>avail_mb</td>
<td>Unused space still available in the storage pool.</td>
</tr>
<tr>
<td>total_mb</td>
<td>Total space in the storage pool (total of used and unused).</td>
</tr>
<tr>
<td>potential_mb</td>
<td>Available space that can be added to the storage pool.</td>
</tr>
</tbody>
</table>

Note: Each of the options used with the command `nas_pool - size` is filters for the output of the command. For example, if you specify a Data Mover, the output will reflect only the space to which the specified Data Mover has visibility. Physical used_mb, Physical avail_mb, and Physical total_mb are applicable for system-defined virtual AVM pools only.

**EXAMPLE #4**

To view the size information for the TP1 mapped pool which contains only virtual LUNs, type:

```
$ nas_pool -size TP1
```

```
id           = 40
name         = TP1
used_mb      = 0
avail_mb     = 0
total_mb     = 0
potential_mb = 2047
```

Physical storage usage in tp1 on FCNTR074200038:

```
used_mb = 0
avail_mb = 20470
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical used_mb</td>
<td>Used physical size of a storage system mapped pool in MB (some may be used by non-VNX hosts).</td>
</tr>
<tr>
<td>Physical avail_mb</td>
<td>Available physical size of a storage system mapped pool in MB.</td>
</tr>
</tbody>
</table>

Note: Physical used_mb and Physical avail_mb are applicable for system-defined AVM pools that contain virtual LUNs only.

**EXAMPLE #5**

For VNX storage system, to change the `is_greedy` and `is_dynamic` options for the system defined, `clar_r5_performance` storage pool, type:

```
$ nas_pool -modify clar_r5_performance -is_dynamic n
             -is_greedy y
```

```
id                   = 3
name                 = clar_r5_performance
description          = CLARiiON RAID5 4plus1
acl                  = 421
in_use               = False
```
EXAMPLE #6 For VNX for file with a Symmetrix storage system, to change the
-is_greedy and -is_dynamic options for the system-defined,
symm_std storage pool, type:

```
$ nas_pool -modify symm_std -is_dynamic y -is_greedy y
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the storage pool.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the storage pool.</td>
</tr>
<tr>
<td>description</td>
<td>Comment assigned to the storage pool.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value assigned to the storage pool.</td>
</tr>
<tr>
<td>in_use</td>
<td>Whether the storage pool is being used by a file system.</td>
</tr>
<tr>
<td>clients</td>
<td>File systems using the storage pool.</td>
</tr>
<tr>
<td>members</td>
<td>Disks used by the storage pool.</td>
</tr>
<tr>
<td>default_slice_flag</td>
<td>Allows slices from the storage pool.</td>
</tr>
<tr>
<td>is_user_defined</td>
<td>User-defined as opposed to system-defined.</td>
</tr>
<tr>
<td>thin</td>
<td>Indicates whether thin provisioning is enabled or disabled.</td>
</tr>
<tr>
<td>disk_type</td>
<td></td>
</tr>
<tr>
<td>server_visibility</td>
<td></td>
</tr>
<tr>
<td>volume_profile</td>
<td></td>
</tr>
<tr>
<td>is_dynamic</td>
<td></td>
</tr>
<tr>
<td>is_greedy</td>
<td></td>
</tr>
<tr>
<td>num_stripe_members</td>
<td></td>
</tr>
<tr>
<td>stripe_size</td>
<td></td>
</tr>
</tbody>
</table>
The nas Commands

<table>
<thead>
<tr>
<th>disk_type</th>
<th>Contingent on the storage system attached.</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_visibility</td>
<td>Storage pool is visible to the physical Data Movers specified.</td>
</tr>
<tr>
<td>volume_profile</td>
<td>Volume profile used.</td>
</tr>
<tr>
<td>is_dynamic</td>
<td>Whether the system-defined storage pool can add or remove volumes.</td>
</tr>
<tr>
<td>is_greedy</td>
<td>Indicates whether the system-defined storage pool will use new member volumes as needed.</td>
</tr>
<tr>
<td>template_pool</td>
<td>System pool template used to create the user pool. Only applicable to user pools created by size or if the last member volume is a stripe or both.</td>
</tr>
<tr>
<td>num Stripe Members</td>
<td>Number of stripe members used to create the user pool. Applicable to system pools and user pools created by size or if the last member volume is a stripe or both.</td>
</tr>
<tr>
<td>stripe size</td>
<td>Stripe size used to create the user pool. Applicable to system pools and user pools created by size or if the last member volume is a stripe or both.</td>
</tr>
</tbody>
</table>

**EXAMPLE #7**  To add the volumes, d7 and d8, to the **marketing** storage pool, type:

```bash
$ nas_pool -xtend marketing -volumes d7,d8
```

<table>
<thead>
<tr>
<th>id</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>marketing</td>
</tr>
<tr>
<td>description</td>
<td>Marketing Storage Pool</td>
</tr>
<tr>
<td>acl</td>
<td>0</td>
</tr>
<tr>
<td>in_use</td>
<td>False</td>
</tr>
<tr>
<td>clients</td>
<td></td>
</tr>
<tr>
<td>members</td>
<td>d12,d13,d7,d8</td>
</tr>
<tr>
<td>default_slice_flag</td>
<td>True</td>
</tr>
<tr>
<td>is_user_defined</td>
<td>True</td>
</tr>
<tr>
<td>thin</td>
<td>True</td>
</tr>
<tr>
<td>disk_type</td>
<td>CLSTD</td>
</tr>
<tr>
<td>server_visibility</td>
<td>server_2,server_3,server_4,server_5</td>
</tr>
<tr>
<td>template_pool</td>
<td>N/A</td>
</tr>
<tr>
<td>num Stripe Members</td>
<td>N/A</td>
</tr>
<tr>
<td>stripe size</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**EXAMPLE #8**  For VNX storage system, to extend the system-defined storage pool by a specified size with a specified storage storage system, type:

```bash
$ nas_pool -xtend clar_r5_performance -size 128M -storage APM00042000818
```

<table>
<thead>
<tr>
<th>id</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>clar_r5_performance</td>
</tr>
<tr>
<td>description</td>
<td>CLARiiON RAID5 4plus1</td>
</tr>
<tr>
<td>acl</td>
<td>1421, owner=nasadmin, ID=201</td>
</tr>
<tr>
<td>in_use</td>
<td>False</td>
</tr>
<tr>
<td>clients</td>
<td></td>
</tr>
<tr>
<td>members</td>
<td>v120</td>
</tr>
<tr>
<td>default_slice_flag</td>
<td>True</td>
</tr>
<tr>
<td>is_user_defined</td>
<td>False</td>
</tr>
<tr>
<td>thin</td>
<td>False</td>
</tr>
<tr>
<td>disk_type</td>
<td>CLSTD</td>
</tr>
<tr>
<td>server_visibility</td>
<td>server_2,server_3,server_4,server_5</td>
</tr>
<tr>
<td>template_pool</td>
<td>clar_r5_performance_vp</td>
</tr>
</tbody>
</table>
The nas Commands

EXAMPLE #9
For VNX storage system, to remove d7 and d8 from the marketing storage pool, type:

```
$ nas_pool -shrink marketing -volumes d7,d8
```

EXAMPLE #10
To list the storage pools, type:

```
$ nas_pool -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>inuse</th>
<th>acl</th>
<th>name</th>
<th>storage_system</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>n</td>
<td>421</td>
<td>clar_r1</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>n</td>
<td>421</td>
<td>clar_r5_performance</td>
<td>FCNTR074200038</td>
</tr>
<tr>
<td>4</td>
<td>n</td>
<td>421</td>
<td>clar_r5_economy</td>
<td>N/A</td>
</tr>
<tr>
<td>10</td>
<td>n</td>
<td>421</td>
<td>clarata_archive</td>
<td>FCNTR074200038</td>
</tr>
<tr>
<td>11</td>
<td>n</td>
<td>421</td>
<td>clarata_r3</td>
<td>N/A</td>
</tr>
<tr>
<td>20</td>
<td>n</td>
<td>0</td>
<td>marketing</td>
<td>FCNTR074200038</td>
</tr>
<tr>
<td>40</td>
<td>y</td>
<td>0</td>
<td>TP1</td>
<td>FCNTR074200038</td>
</tr>
<tr>
<td>41</td>
<td>y</td>
<td>0</td>
<td>FP1</td>
<td>FCNTR074200038</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the storage pool.</td>
</tr>
<tr>
<td>inuse</td>
<td>Whether the storage pool is being used by a file system.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value assigned to the storage pool.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the storage pool.</td>
</tr>
<tr>
<td>storage_system</td>
<td>Name of the storage system where the storage pool resides.</td>
</tr>
</tbody>
</table>

EXAMPLE #11
To display information about the user-defined storage pool called marketing, type:

```
$ nas_pool -info marketing
```

is_dynamic       = False
is_greedy        = True
num_stripe_members = 4
stripe_size      = 32768
The nas Commands

EXAMPLE #12
To display information about the system-defined clar_r5_performance storage pool, type:

```
$ nas_pool -info clar_r5_performance
```

id = 3
name = clar_r5_performance
description = CLARiiON RAID5 4plus1
acl = 1421, owner=nasadmin, ID=201
in_use = False
clients =
members = v120
default_slice_flag = True
is_user_defined = False
thin = False
disk_type = CLSTD
server_visibility = server_2,server_3,server_4,server_5
volume_profile = clar_r5_performance_vp
is_dynamic = False
is_greedy = True
num_stripe_members = 4
stripe_size = 32768

EXAMPLE #13
To display information about the system-defined engineer virtual pool, type:

```
$ nas_pool -info engineer
```

id = 40
name = engineer
description = Mapped Pool engineer on APM00084401666
acl = 0
in_use = True
clients = DA_BE_VIRT_FS,vp_test,vp_test1,vp_test12,cvpfs1,cvpfs3
members = v363
default_slice_flag = True
is_user_defined = False
thin = True
EXAMPLE #14 To display information about the mapped storage pool called **FP1** from a VNX for block, type:

```
$ nas_pool -info FP1
```

```
id                   = 40
name                 = FP1
description          = Mapped Pool on FCNTR074200038
acl                  = 0
in_use               = False
clients              =
members              =
default_slice_flag   = True
is_user_defined      = False
thin                 = True
tiering_policy       = Auto-tier
compressed           = False
mirrored             = False
disk_type            = Mixed
volume_profile       = FP1
is_dynamic           = True
is_greedy            = True
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>tiering_policy</td>
<td>Indicates the tiering policy in effect. If the initial tier and the tiering policy are the same, the values are: Auto-Tier, Highest Available Tier, Lowest Available Tier. If the initial tier and the tiering policy are not the same, the values are: Auto-Tier/No Data Movement, Highest Available Tier/No Data Movement, Lowest Available Tier/No Data Movement.</td>
</tr>
<tr>
<td>compressed</td>
<td>For VNX for block, indicates whether data is compressed. Values are: True, False, Mixed (indicates some of the LUNs, but not all, are compressed).</td>
</tr>
<tr>
<td>mirrored</td>
<td>Indicates whether the disk is mirrored.</td>
</tr>
</tbody>
</table>

EXAMPLE #15 To display information about the mapped storage pool called **SG0** from a Symmetrix storage system, type:

```
$ nas_pool -info SG0
```

```
id                   = 40
name                 = SG0
description          = Symmetrix Mapped Pool on 000192601245
acl                  = 0
in_use               = False
clients              =
```
**EXAMPLE #16**

To delete the storage pool, `marketing`, and each of the storage pool member volumes recursively, type:

```
$ nas_pool -delete marketing -deep
```

<table>
<thead>
<tr>
<th>id</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>marketing</td>
</tr>
<tr>
<td>description</td>
<td>Marketing Storage Pool</td>
</tr>
<tr>
<td>acl</td>
<td>0</td>
</tr>
<tr>
<td>in_use</td>
<td>False</td>
</tr>
<tr>
<td>clients</td>
<td></td>
</tr>
<tr>
<td>members</td>
<td></td>
</tr>
<tr>
<td>default_slice_flag</td>
<td>True</td>
</tr>
<tr>
<td>is_user_defined</td>
<td>True</td>
</tr>
<tr>
<td>thin</td>
<td>True</td>
</tr>
<tr>
<td>template_pool</td>
<td>N/A</td>
</tr>
<tr>
<td>numStripe_members</td>
<td>N/A</td>
</tr>
<tr>
<td>stripe_size</td>
<td>N/A</td>
</tr>
</tbody>
</table>
nas_quotas

Manages quotas for mounted file systems.

SYNOPSIS

nas_quotas
    -edit [-user|-group] {-mover <movername>|-fs <fs_name>} [-path <pathname>] [[-proto <proto_id>] | [-block <hard_limit>[:<soft_limit>]] [-inode <hard_limit>[:<soft_limit>]] <id> [<id>...]]
    | -edit -config {-mover <movername>|-fs <fs_name>} [-path <pathname>] [-option <options>]
    | -edit -tree -fs <fs_name> [[-proto <proto_id>]] [[-block <hard_limit>[:<soft_limit>]] [-inode <hard_limit>[:<soft_limit>]] [-comment <comment>]] <id> [<id>...]
    | -report [-user|-group] {-mover <movername>|-fs <fs_name>} [-path <pathname>]
    | -report -config {-mover <movername>|-fs <fs_name>} [-path <pathname>] [-option <options>]
    | -report -tree -fs <fs_name> [-path <pathname>] [<id> <id>...]
    | -on|-off|-clear [-user|-group|-both] {-mover <movername>|-fs <fs_name> [-path <pathname>] -all}
    | -on -tree -fs <fs_name> -path <pathname> [-comment <comment>]
    | -off -tree -fs <fs_name> -path <pathname>
    | -check -list|-clear -tree -fs <fs_name> [-path <pathname>]
    | -check [{-stop|-mode online|offline}] -fs <fs_name> [-path <pathname>]
    | -check { -stop | -status} -fs <fs_name> [-path <pathname>]
    | -quotadb { -info | -upgrade [-Force]} {-mover <movername>|-fs <fs_name>}

DESCRIPTION

nas_quotas edits quotas for mounted file systems, and displays a listing of quotas and disk usage at the file system level (by the user, group, or tree), or at the quota-tree level (by the user or group).

nas_quotas also turns quotas on and off, and clears quotas records for a file system, quota tree, or a Data Mover. When a Data Mover is specified, the action applies to all mounted file systems on the Data Mover.

nas_quotas also starts and stops quota database checks either online or offline for quota trees and file systems, and allows you to upgrade the quota database limits to the maximum limit value for a file system. When a Data Mover is specified, the action applies to all mounted file systems on the Data Mover.
CAUTION

Quotas should be turned on (enabled) before file systems go into a production environment. Enabling (or disabling, or clearing) quotas in a production environment is time consuming and the process may disrupt file system operation. CIFS clients are disconnected during these events and NFS clients receive a message that the server is not responding. However, once enabled, quotas can be changed at any time without impact.

OPTIONS

- **edit**  
  
  **[-user | -group]**  
  **[-mover <movername>| -fs <fs_name>  
  [-path <pathname>]]**  
  
  Sets the quota limits for users or groups on a specified Data Mover, mounted file system, or directory tree.

  For a user, the ID can be a user ID or UID, however, if NIS or the local password file on the Data Mover is available, a username can also be used.

  For a group, the ID can be a group ID or GID, however, if NIS or the local password file is available, a group name can also be used.

  Upon execution, a **vi** session (unless the EDITOR environment variable specifies otherwise) is opened to edit the quota configuration file. Changes to the file are applied when the **vi** session is saved and exited.

  
  
  
  **[-proto <proto_id>] |  
  [-block <hard_limit> [::<soft_limit>]]**

  Applies the quota configuration defined for the prototype user for each specified ID, and sets a hard and soft limit for storage (block) usage in kilobytes.

  
  
  
  **[-inode <hard_limit> [::<soft_limit>]][<id> [::<id> ...] [-block <hard_limit> [::<soft_limit>]]**

  Edits the inode (file count) limits and the block (storage in KBs) limits directly into the quota configuration file without opening an editing session.

  
  
  
  **-edit -config**  
  
  **[-mover <movername>| -fs <fs_name>  
  [-path <pathname>]]**

  Edits the default quota configuration for all users/groups currently without quotas or subsequently added to the specified Data Mover or file system or quota tree. Also edits the grace periods for soft quotas, and the conditions upon which to generate a quotas-event message to the system log.
The nas Commands

[-option <options>]
Specifies the following comma-separated options:

BGP=<integer>
Sets the block grace period in seconds.

IGP=<integer>
Sets the inode grace period in seconds.

DUBSL=<integer>
Sets the default user block soft limit in KB.

DUBHL=<integer>
Sets the default user block hard limit in KB.

DUISL=<integer>
Sets the default user inode soft limit.

DUIHL=<integer>
Sets the default user inode hard limit.

DGBSL=<integer>
Sets the default group block soft limit in KB.

DGBHL=<integer>
Sets the default group block hard limit in KB.

DGISL=<integer>
Sets the default group inode soft limit.

DGIIHL=<integer>
Sets the default group inode hard limit.

HLE={True|False}
Specifies whether the hard limit is enforced.

ESFCS={True|False}
Specifies the event for check start has been sent.

ESFCE={True|False}
Specifies the event for check end has been sent.

ESFBSSL={True|False}
Specifies that the event for block soft limits has been sent.

ESFBHLL={True|False}
Specifies that the event for block hard limits has been sent.

-edit -tree -fs <fs_name> [[-proto <proto_id>] | [-block <hard_limit> [:<soft_limit>]][-inode <hard_limit>[:<soft_limit>]]][-comment <comment>]}
The nas Commands

`<id> [<id>...]`
Edits the quota limits for trees (inodes or blocks used by a tree directory) where the `<id>` is the tree ID. This option can only be applied on each file system basis. The `-list` option to display the tree IDs.

The `-proto` option applies the quota configuration of the prototype tree for each specified tree ID, or sets a hard and soft limit for blocks. The `<proto_id>` must be a tree ID.

The `-inode` and `-block` options edit the inode/block limits for the tree directly in the quota configuration file without opening an editing session.

The `-comment` option associates a comment with the quota tree. The comment is delimited by single quotes. Comment length is limited to 256 bytes (represented as 256 ASCII characters or a variable number of Unicode multibyte characters) and cannot include single quotes (‘ ’), double quotes (“ ”), semicolons (;), NL (New Line), or FF (Form Feed).

`-report [-user | -group] {-mover <movername> | -fs <fs_name>} [-path <pathname>] [<id> <id> ...]
Displays a summary of disk usage and quotas for the user or group, including the number of files and space in kilobytes for the specified `<fs_name>`, or all file systems mounted on the specified `<movername>`, or for the specified quota tree. The `-edit` option provides more information for the usage of UIDs and GIDs.

---

**Note:** The nas_quotas can show report for maximum 1024 IDs at a time.

`-report -config {-mover <movername> | -fs <fs_name>} [-path <pathname>]`
Displays quota configuration information as viewed from the specified Data Mover, file system, or quota-tree level, including:

- Active quota policy
- Quota status (user/group quotas enabled or disabled)
- Grace period
- Default limits currently set for users/groups
- Hard-quota enforcement option setting (deny disk space enabled or disabled)
- Quota conditions that trigger event-logging
The nas Commands

-**report** -**tree** -**fs** <fs_name>[<id> <id>...] Displays the quota limits for a specified quota tree in a file system. The <id> is a tree ID.

**Note:** The <id> is either a user ID, a group ID, or a tree ID. If the quota type is not specified, the default is set to the ‘-user’ ID.

{**-on** | **-off** | **-clear**} [**-user** | **-group** | **-both**] {**-mover** <movername> | **-fs** <fs_name> | [**-path** <pathname>] | **-all**} Turns quotas on, off, and clears quotas for the user, group, or both (users and groups at once) on the <movername>, <fs_name>, <pathname>, for all users, or groups on all file systems on all Data Movers in the cabinet.

The **-clear** option permanently removes all quota records, deletes the quota configuration file, and turns quotas **off**.

**CAUTION**

While quotas are being turned on, off, or cleared, other operations to a file system may be disrupted. CIFS clients are disconnected during this execution.

-**on** -**tree** -**fs** <fs_name> -**path** <pathname> Turns on (enables) tree quotas so that quota tracking and hard-limit enforcement (if enabled) can occur. When enabling tree quotas, the directory must not exist; it is created in this tree-quota-enabling process.

**Note:** The quota path length (which VNX for file calculates as including the file system mountpoint) must be less than 1024 bytes. If Unicode is enabled on the selected Data Mover, -path accepts any characters defined by the Unicode 3.0 standard. Otherwise, it accepts only ASCII characters.

[**-comment** <comment>] The **-comment** option associates a comment with the quota tree. The comment is delimited by single quotes. Comment length is limited to 256 bytes (represented as 256 ASCII characters or a variable number of Unicode multibyte characters) and cannot include single quotes (‘ ’), double quotes (" "), semicolons (;), NL (New Line), or FF (Form Feed).

-**off** -**tree** -**fs** <fs_name> -**path** <pathname> Turns tree quotas off. When turning tree quotas off, the tree directory must be empty.
The nas Commands

{-list| -clear} -tree -fs <fs_name>
The -list option displays all active quota trees and their respective tree IDs used by -edit and -report with the specified file system.

Use the -tree -clear option to clear all the information from the database after you disable (turn off) quotas for all trees within a file system. Once cleared, the database information is not recoverable.

CAUTION
The -clear option deletes the usage and the limit information for tree quotas. The limits cannot be recovered.

-check -start [-mode online | offline] [-tree] -fs <fs_name> [-path <pathname>]
Starts a check of a quota database in online or offline mode for a tree quota or a file system quota. The default mode is online if the -mode option is not specified, and a quota check is run while the file system remains online.

-check [-stop | -status] -fs <fs_name> [-path <pathname>]
Stops or provides status of a file system quota database check that is in progress.

-quotadb { -info | -upgrade [-Force] | -mover <movername> | -fs <fs_name>}
Either displays status related to the quota database upgrade or starts an upgrade of the quota database for a specific file system or all file systems on a Data Mover.

The -info option displays the status related to the quota database limits upgrade.

Use the -upgrade option to perform an upgrade of the quota database. If the -Force option is not specified, you are in interactive mode while upgrading the quota database. If the -Force option is specified, you are in non-interactive mode while upgrading the quota database.

Use -mover <movername> to upgrade all mounted file systems’ quota databases on a Data Mover.

Use -fs <fs_name> to upgrade a specific file system’s quota database.

Note: Before the upgrade process runs, the Control Station displays the estimated upgrade time on the file system whose quota database will be upgraded, and also displays a warning message to notify users that the file...
system will be unavailable during the upgrade process. If users are in interactive mode, a dialog displays letting users choose whether they want to continue. If users are in non-interactive mode, after displaying the estimated upgrade time message and warning message, the upgrade process starts immediately.

**SEE ALSO**  *Using Quotas on VNX.*

**EXAMPLE #1**  To enable quotas on for users and groups of a file system, type:

```
$ nas_quotas -on -both -fs ufs1
```

done

**EXAMPLE #2**  To open a vi session to edit file system quotas on *ufs1* for the specified user, 1000, type:

```
$ nas_quotas -edit -user -fs ufs1 1000
```

Userid : 1000
fs ufs1 blocks (soft = 2000, hard = 3000) inodes (soft = 0, hard = 0)
~
~
~
~/tmp/EdP.agGQuIz" 2L, 84C written
done

**EXAMPLE #3**  To change the block limit and inode limit for a file system without opening up a vi session, type:

```
$ nas_quotas -edit -user -fs ufs1 -block 7000:6000 -inode 700:600 2000
```

done

**EXAMPLE #4**  To view a report of user quotas for *ufs1*, type:

```
$ nas_quotas -report -user -fs ufs1
```

Report for user quotas on filesystem ufs1 mounted on /ufs1

| User  | Bytes Used (1K) | Files |
|-------+----------------+-------|
|       | Used | Soft | Hard | Timeleft | Used | Soft | Hard | Timeleft |
| #1000 | 1328 | 2000 | 3000 |  | 54 | 0 | 0 |  |
| #2000 | 6992 | 6000 | 7000 | 7.0days | 66 | 600 | 700 |  |
| #5000 | 141592 | 0 | 0 |  | 516 | 0 | 0 |  |

done
EXAMPLE #5  To select user 300 as prototype user for ufs1, and assign other users
the same limits, type:

$ nas_quotas -group -edit -fs ufs1 -proto 300 301 302 303
done

EXAMPLE #6  To display the group quotas information for ufs1, type:

$ nas_quotas -report -group -fs ufs1

Report for group quotas on filesystem ufs1 mounted on /ufs1

<table>
<thead>
<tr>
<th>Group</th>
<th>Used</th>
<th>Soft</th>
<th>Hard</th>
<th>Timeleft</th>
<th>Used</th>
<th>Soft</th>
<th>Hard</th>
<th>Timeleft</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>296</td>
<td>0</td>
<td>0</td>
<td></td>
<td>12</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>#300</td>
<td>6992</td>
<td>6000</td>
<td>7000</td>
<td>7.0 days</td>
<td>67</td>
<td>600</td>
<td>700</td>
<td></td>
</tr>
<tr>
<td>#301</td>
<td>0</td>
<td>6000</td>
<td>7000</td>
<td></td>
<td>0</td>
<td>600</td>
<td>700</td>
<td></td>
</tr>
<tr>
<td>#302</td>
<td>0</td>
<td>6000</td>
<td>7000</td>
<td></td>
<td>0</td>
<td>600</td>
<td>700</td>
<td></td>
</tr>
<tr>
<td>#303</td>
<td>0</td>
<td>6000</td>
<td>7000</td>
<td></td>
<td>0</td>
<td>600</td>
<td>700</td>
<td></td>
</tr>
<tr>
<td>#32772</td>
<td>22296</td>
<td>0</td>
<td>0</td>
<td></td>
<td>228</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

done

EXAMPLE #7  To edit the default quota configuration for server_2, type:

$ nas_quotas -edit -config -mover server_2

File System Quota Parameters:
  fs "ufs1"
    Block Grace: (1.0 weeks)
    Inode Grace: (1.0 weeks)
    * Default Quota Limits:
      User: block (soft = 5000, hard = 8000) inodes (soft = 100, hard= 200)
      Group: block (soft = 6000, hard = 9000) inodes (soft = 200, hard= 400)
    Deny disk space to users exceeding quotas: (yes)
    * Generate Events when:
      Quota check starts: (no)
      Quota check ends: (no)
      soft quota crossed: (no)
      hard quota crossed: (no)

  fs "ufs2"
    Block Grace: (1.0 weeks)
    Inode Grace: (1.0 weeks)
    * Default Quota Limits:
      User: block (soft = 0, hard = 0) inodes (soft = 0, hard= 0)
      Group: block (soft = 0, hard = 0) inodes (soft = 0, hard= 0)
    Deny disk space to users exceeding quotas: (yes)
    * Generate Events when:
      Quota check starts: (no)
      Quota check ends: (no)
      soft quota crossed: (no)
      hard quota crossed: (no)
EXAMPLE #8  To open a vi session and edit the quotas for a file system, type:

```bash
$ nas_quotas -edit -config -fs ufs1
```

File System Quota Parameters:

```
fs "ufs1"
Block Grace: (1.0 weeks)
Inode Grace: (1.0 weeks)
* Default Quota Limits:
  User: block (soft = 5000, hard = 8000) inodes (soft = 100, hard= 200)
  Group: block (soft = 6000, hard = 9000) inodes (soft = 200, hard= 400)
Deny disk space to users exceeding quotas: (yes)
* Generate Events when:
  Quota check starts: (no)
  Quota check ends: (no)
  soft quota crossed: (yes)
  hard quota crossed: (yes)
```

EXAMPLE #9  To view the quota configuration for the file system, ufs1, type:

```bash
$ nas_quotas -report -config -fs ufs1
```

```
+------------------------------------------------------------------------------+
| Quota parameters for filesystem ufs1 mounted on /ufs1:                      |
+------------------------------------------------------------------------------+
| Quota Policy: blocks                                                        |
| User Quota: ON                                                             |
| Group Quota: ON                                                            |
| Block grace period: (1.0 weeks)                                            |
| Inode grace period: (1.0 weeks)                                           |
| Default USER quota limits:                                                 |
|   Block Soft: ( 5000), Block Hard: ( 8000)                                 |
|   Inode Soft: ( 100), Inode Hard: ( 200)                                  |
| Default GROUP quota limits:                                                |
|   Block Soft: ( 6000), Block Hard: ( 9000)                                 |
|   Inode Soft: ( 200), Inode Hard: ( 400)                                  |
| Deny Disk Space to users exceeding quotas: YES                              |
| Log an event when ...                                                      |
|   Block hard limit reached/exceeded: YES                                   |
|   Block soft limit (warning level) crossed: YES                            |
|   Quota check starts: NO                                                   |
|   Quota Check ends: NO                                                    |
+------------------------------------------------------------------------------+
```

done
EXAMPLE #10  To enable tree quotas on for ufs1, type:

$ nas_quotas -on -tree -fs ufs1 -path /tree1 -comment 'Tree #1'

done

EXAMPLE #11  To create a tree quota in a language that uses multibyte characters (such as Japanese), type:

$ nas_quotas -on -tree -fs fs_22 -path /<path_in_local_language_text> -comment <comment_in_local_language_text>

done

EXAMPLE #12  To list the tree quotas for ufs1, type:

$ nas_quotas -list -tree -fs ufs1

+----------------------------------------------------------------------------+
| Quota trees for filesystem ufs1 mounted on /ufs1:
| +-------+---------------------------------------------------------------------+
| TreeId| Quota tree path (Comment)                                           |
| +-------+---------------------------------------------------------------------+
| 1      | /tree1 (Tree #1)                                                    |
| 2      | /tree2 (Tree #2)                                                    |
| 3      | /<tree_path_in_local_language_text> (Tree #3)                        |
+-------+---------------------------------------------------------------------+

done

EXAMPLE #13  To edit or add a comment for a tree quota for ufs1, type:

$ nas_quotas -edit -tree -fs ufs1 -comment 'Quota for Tree1' 1

done

EXAMPLE #14  To edit tree quotas for ufs1, type:

$ nas_quotas -edit -tree -fs ufs1 1
treeid : 1
fs ufs1 blocks (soft = 6000, hard = 8000) inodes (soft = 200, hard = 300)
~
~ "/tmp/EdP.aiHKgh5" 2L, 85C written
done

EXAMPLE #15  To edit tree quotas for ufs1 and change the block and inodes, type:

$ nas_quotas -edit -tree -fs ufs1 -block 8000:6000 -inode 900:800 1

done

EXAMPLE #16  To edit tree quotas for ufs1 and apply the quota configuration of the prototype tree, type:
$ nas_quotas -edit -tree -fs ufs1 -proto 1 2

done

**EXAMPLE #17**  
To display any currently active trees on a file system, type:

```
$ nas_quotas -report -tree -fs ufs1
```

Report for tree quotas on filesystem ufs1 mounted on /ufs1

<table>
<thead>
<tr>
<th>Tree</th>
<th>Used</th>
<th>Soft</th>
<th>Hard</th>
<th>Timeleft</th>
<th>Used</th>
<th>Soft</th>
<th>Hard</th>
<th>Timeleft</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>384</td>
<td>6000</td>
<td>8000</td>
<td></td>
<td>3</td>
<td>800</td>
<td>900</td>
<td></td>
</tr>
<tr>
<td>#2</td>
<td>7856</td>
<td>6000</td>
<td>8000</td>
<td>7.0 days</td>
<td>60</td>
<td>800</td>
<td>900</td>
<td></td>
</tr>
</tbody>
</table>

done

**EXAMPLE #18**  
To enable tree quotas off, type:

```
$ nas_quotas -tree -off -fs ufs1 -path /tree1
```

done

**EXAMPLE #19**  
To enable quotas on for users and groups on tree quota, /tree3, of a file system, ufs1, type:

```
$ nas_quotas -on -both -fs ufs1 -path /tree3
```

done

**EXAMPLE #20**  
To open a vi session to edit file system quotas on quota tree, /tree3, on ufs1 for the specified user, 1000, type:

```
$ nas_quotas -edit -user -fs ufs1 -path /tree3 1000
```

Userid : 1000

fs ufs1 tree "/tree3" blocks (soft = 4000, hard = 6000) inodes (soft = 30, hard = 50)

~
~
~

"/tmp/EdP.aMdtIQR" 2L, 100C written
done

**EXAMPLE #21**  
To change the block limit and inode limit on quota tree, /tree3, on ufs1 for the specified user, 1000, without opening up a vi session, type:

```
$ nas_quotas -edit -user -fs ufs1 -path /tree3 -block 6000:4000 -inode 300:200 1000
```

done

**EXAMPLE #22**  
To view a report of user quotas on tree quota, /tree3, for ufs1, type:
The nas Commands

$ nas_quotas -report -user -fs ufs1 -path /tree3
Report for user quotas on quota tree /tree3 on filesystem ufs1 mounted on /ufs1

+-----------+----------------+----------------+-----------------+------------------+
|User       | Used | Soft | Hard | Timeleft| Used | Soft | Hard | Timeleft|
+-----------+-------+-------+-------+--------+-------+------+------+--------+
|           |  Used |  Soft |  Hard |Timeleft|  Used |  Soft |  Hard |Timeleft|
+-----------+-------+-------+-------+--------+-------+------+------+--------+
|#1000      |   2992|   4000|   6000|        |     34|   200|   300|        |
|#32768     |   9824|      0|      0|        |     28|     0|     0|        |
+-----------+-------+-------+-------+--------+-------+------+------+--------+
done

EXAMPLE #23
To open a vi session and edit the quota configuration for tree quota, /tree3, on a file system, ufs1, type:

$ nas_quotas -edit -config -fs ufs1 -path /tree3

Tree Quota Parameters:
fs "ufs1"
tree "/tree3"
Block Grace: (1.0 weeks)
Inode Grace: (1.0 weeks)
* Default Quota Limits:
  User: block (soft = 8000, hard = 9000) inodes (soft = 200, hard= 300)
  Group: block (soft = 8000, hard = 9000) inodes (soft = 300, hard= 400)
Deny disk space to users exceeding quotas: (yes)
* Generate Events when:
  Quota check starts:          (no)
  Quota check ends:            (no)
  soft quota crossed:          (yes)
  hard quota crossed:          (yes)

~
~
~
"/tmp/EdP.aDTOKeU" 14L, 508C written
done

EXAMPLE #24
To view the quota configuration for tree quota, /tree3, on file system, ufs1, type:

$ nas_quotas -report -config -fs ufs1 -path /tree3

+------------------------------------------------------------------------------+
| Quota parameters for tree quota /tree3 on filesystem ufs1 mounted on /ufs1: |
+------------------------------------------------------------------------------+
| Quota Policy: blocks          | User Quota: ON |
| User Quota: ON          | Group Quota: ON |
| Group Quota: ON       | Block grace period: (1.0 weeks) |
| Block grace period: (1.0 weeks) | Inode grace period: (1.0 weeks) |
| Default USER quota limits: | Block Soft: ( 8000), Block Hard: ( 9000) |
|                          | Inode Soft: ( 200), Inode Hard: ( 300) |

Default GROUP quota limits:
  Block Soft: (8000), Block Hard: (9000)
  Inode Soft: (300), Inode Hard: (400)
Deny Disk Space to users exceeding quotas: YES
Log an event when ...
  Block hard limit reached/exceeded: YES
  Block soft limit (warning level) crossed: YES
  Quota check starts: NO
  Quota Check ends: NO

EXAMPLE #25  To disable user quota and group quota on tree quota, /tree3, type:
              $ nas_quotas -off -both -fs ufs1 -path /tree3

done

EXAMPLE #26  To disable group quotas for ufs1, type:
              $ nas_quotas -off -group -fs ufs1

done

EXAMPLE #27  To clear all tree quotas for ufs1, type:
              $ nas_quotas -clear -tree -fs ufs1

done

EXAMPLE #28  To clear quotas for users and groups of a Data Mover, type:
              $ nas_quotas -clear -both -mover server_2

done

EXAMPLE #29  To start a tree quota check in quota tree /mktg-a/dir1 in file system ufs1 with the file system online, type:
              $ nas_quotas -check -start -mode online -tree -fs ufs1 /mktg-a/dir1

done

EXAMPLE #30  To stop a tree quota check in file system ufs1, type:
              $ nas_quotas -check -stop -fs ufs1

done

EXAMPLE #31  To view the status of a tree quota check in quota tree /mktg-a/dir1 in file system ufs1, type:
              $ nas_quotas -check -status -tree -fs ufs1 -path /mktg-a/dir1

Tree quota check on filesystem ufs1 and path /mktg-a/dir is running and is 60% complete.
Done
EXAMPLE #32  To list quota database limits for all file systems on a Data Mover, type:

$ nas_quotas -quotadb -info -mover server_2

Info 13421850365 : The quota limit on ufs0 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds. A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850365 : The quota limit on ufs1 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds. A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850365 : The quota limit on ufs2 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds. A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850365 : The quota limit on ufs3 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds. A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850366 : The quota limit on ufs4 is at 256 TB

EXAMPLE #33  To list quota database limits for file system ufs4, type:

$ nas_quotas -quotadb -info -fs ufs4

Info 13421850366 : The quota limit on ufs4 is at 256 TB

EXAMPLE #34  To upgrade all file systems on a Data Mover, in interactive mode, type:

$ nas_quotas -quotadb -upgrade -mover server_2

Info 13421850365 : The quota limit on ufs0 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds. A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850365 : The quota limit on ufs1 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds. A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850365 : The quota limit on ufs2 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds. A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850365 : The quota limit on ufs3 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds. A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.
Info 13421850365: The quota limit on ufs3 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds.
A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850366: The quota limit on ufs4 is at 256 TB.

Warning 17716861297: The file systems specified in the list above will not be accessible during the quota database upgrade, and a file system's CIFS share and NFS export also will not be accessible during the upgrade. The file systems shown above are listed in the order that the quota database conversion is performed, one by one sequentially. The estimated time (shown above) needed to upgrade the quota database may change based on the file system's quota configuration and I/O performance when the conversion is running.
Do you really want to upgrade the file system quota database now? [Y/N]: Y

Info 13421850367: quota db upgraded on ufs0
Info 13421850367: quota db upgraded on ufs1
Info 13421850367: quota db upgraded on ufs2
Error 13421850368: Timeout occurred when upgrading quota db on ufs3. The Quota db upgrade may still be in progress. Use the "-info" option to check status.
Info 13421850369: quota db already upgraded on ufs4

EXAMPLE #35  To list quota database limits for file system ufs3 after an upgrade has timed out, type:

$ nas_quotas -quotadb -info -fs ufs3

Info 13421850370: The quota limit on ufs3 is at 4TB. Upgrade is 48% complete.

EXAMPLE #36  To upgrade all file systems on a Data Mover, in non-interactive mode, type:

$ nas_quotas -quotadb -upgrade -Force -mover server_2

Info 13421850365: The quota limit on ufs0 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds.
A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850365: The quota limit on ufs1 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds.
A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850365: The quota limit on ufs2 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds.
A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Info 13421850365: The quota limit on ufs3 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds.
A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.
Info 13421850366: The quota limit on ufs4 is at 256 TB.
Warning 17716861297: The file systems specified in the list above will not be accessible during the quota database upgrade, and a file system's CIFS share and NFS export also will not be accessible during the upgrade. The file systems shown above are listed in the order that the quota database conversion is performed, one by one sequentially. The estimated time (shown above) needed to upgrade the quota database may change based on the file system's quota configuration and I/O performance when the conversion is running.

Example #37 To upgrade file system ufs3, in interactive mode, type:

```
$ nas_quotas -quotadb -upgrade -fs ufs3
```

Info 13421850365: The quota limit on ufs3 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds.
A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Warning 17716861297: The file systems specified in the list above will not be accessible during the quota database upgrade, and a file system's CIFS share and NFS export also will not be accessible during the upgrade. The file systems shown above are listed in the order that the quota database conversion is performed, one by one sequentially. The estimated time (shown above) needed to upgrade the quota database may change based on the file system's quota configuration and I/O performance when the conversion is running.

Do you really want to upgrade the file system quota database now[Y/N]: Y
Info 13421850367: quota db upgraded on ufs3
Done

Example #38 To upgrade file system ufs3, in non-interactive mode, type:

```
$ nas_quotas -quotadb -upgrade -Force -fs ufs3
```

Info 13421850365: The quota limit on ufs3 is at 4TB. The upgrade to 256 TB is estimated to take 5 seconds.
A total number of 1500 data blocks in the quota database will be converted at a speed of 300 blocks per second.

Warning 17716861297: The file systems specified in the list above will not be accessible during the quota database upgrade, and a file system's CIFS share and NFS export also will not be accessible during the upgrade. The file systems shown above are listed in the order that the quota database conversion is performed, one by one sequentially. The estimated time (shown above) needed to upgrade the quota database may change based on the file system's quota configuration and I/O performance when the conversion is running.

Info 13421850367: quota db upgraded on ufs3
Done
nas_rdf

Facilitates communication between two VNX systems. Its primary use is to manage VNX for file systems and define the relationships needed for disaster recovery in a Symmetrix Remote Data Facility (SRDF) environment. It is also used to establish communication between VNX for file systems and define the relationships needed for file system replication by using EMC VNX Replicator™ or TimeFinder/FS.

SYNOPSIS

```
nas_rdf
    -init
    -activate [-reverse] [-nocheck]
    -restore [-nocheck]
    -check [-all] [test, ...]
```

DESCRIPTION

**nas_rdf** establishes and manages relationships for Control Stations and Data Movers that physically reside in separate VNX for file cabinets.

For SRDF, **nas_rdf** initializes the VNX activates a failover to a destination VNX for file, or restores a source VNX. For Dynamic SRDF, **nas_rdf** activates a failover and reverses the storage system from a destination volume (R2) to a source volume (R1). Configuration details depend on the type of SRDF: active/passive or active/active’ SRDF/S for synchronous replication with disaster recovery, or active/passive SRDF/A for extended-distance, or asynchronous replication with a point-in-time replica.

**Note:** The apostrophe in active/active’ indicates that both sites have a source volume mirrored at the other site.

SRDF is supported only on a VNX attached to a Symmetrix system. Also, this command must be run from a primary Control Station in slot 0; it will report an error if run from a Control Station in slot 1.

**Note:** This command must be executed from the /nas/sbin directory, unless otherwise directed. Log in with your administrative username and password, and execute this functionality from root.
The nas Commands

OPTIONS

-init
Initializes a source or destination (target) VNX for SRDF/S or SRDF/A.

-activate [ -reverse ]
Initiates an SRDF failover from the source VNX for file to the destination. The -activate option is executed on the destination VNX at the discretion of the user. The -activate option sets each SRDF-protected volume on the source VNX as read-only, and each mirrored volume on the destination VNX is set as read-write. The SRDF standby Data Movers acquire the IP and MAC addresses, file systems, and export tables of their source counterparts. The -reverse option reverses SRDF direction by converting R2 volumes at destination site to R1 and synchronizing the destination and source sites. The -reverse option adds SYMCLI swap and establishes operations on the storage system after the normal activate operation is performed. When the -activate option is executed, an automatic, internal SRDF health check is performed before activating a failover. The -nocheck option allows you to skip this health check.

-restore
Restores a source VNX after a failover. The -restore option is initially executed on the destination VNX. The data on each destination volume is copied to the corresponding volume on the source VNX. On the destination VNX, services on each SRDF standby Data Mover are stopped. (NFS clients connected to these Data Movers see a “server unavailable” message; CIFS client connections time out.) Each volume on the source VNX is set as read-write, and each mirrored volume on the destination VNX is set as read-only.

Finally, nas_rdf -restore can be remotely executed on the source VNX to restore the original configuration. Each primary Data Mover reacquires its IP and MAC addresses, file systems, and export tables. When the -restore option is executed, an automatic, internal SRDF health check is performed before restoring source and destination VNX systems. The -nocheck option allows you to skip this health check.

-check { -all | <test>,… }
Runs SRDF health checks on the VNX. The -check option can be executed either by using the -all option or by specifying one or more of the following individual checks: SRDF standby Data Mover configuration check (r1_dm_config, r2_dm_config), SRDF session state check (r1_session, r2_session), Device group configuration check (r1_dev_group, r2_dev_group), Data Mover mirrored device
accessibility check (r1_dev_avail, r2_dev_avail), Symmetrix device state check (dev_not_normal), and SRDF restored state check (restored). In these checks, r1 represents the source side and r2 represents the destination side.

When the –all option is used, all the checks are performed automatically. If the -check option detects invalid configurations or state issues, it prints relevant warning messages with recommended actions so that the issues can be resolved before running the activate or restore options. You can use the –check option to perform health checks at any time.

**Note:** To run the –check option, you must log in to the VNX either as nasadmin and then switch (su) to root, or as rdfadmin and then switch (su) to root.

---

### SEE ALSO

Using SRDF/S with VNX for Disaster Recovery, Using SRDF/S with VNX, Using VNX Replicator, and nas_cel.

### EXAMPLE #1

To start the initialization process on a source VNX in an active/passive SRDF/S configuration, as root user, type:

```
# /nas/sbin/nas_rdf -init
```

Discover local storage devices ...
Discovering storage (may take several minutes)
done

### EXAMPLE #2

To initiate an SRDF failover from the source VNX to the destination, as a root user, type:

```
# /nas/sbin/nas_rdf -activate
```

Is remote site CELERRA completely shut down (power OFF)?
Do you wish to continue? [yes or no]: yes
Successfully pinged (Remotely) Symmetrix ID: 000187430809
Successfully pinged (Remotely) Symmetrix ID: 000190100559
Successfully pinged (Remotely) Symmetrix ID: 000190100582

Write Disable devices on SA at source (R1)..............Done.
Suspend RDF links.......................................Done.
Read/Write Enable devices on RA at target (R2)........Done.
Waiting for nbs clients to die ... done
Waiting for nbs clients to start ... done
fsck 1.35 (28-Feb-2004)
/dev/ndj1: recovering journal
/dev/ndj1: clean, 13780/231360 files, 233674/461860 blocks
Waiting for nbs clients to die ... done
Waiting for nbs clients to start ... done

id type acl slot groupID state name
The nas Commands

1  1  1000  2  0  server_2
2  4  1000  3  0  server_3
3  1  1000  4  0  server_4
4  4  1000  5  0  server_5

server_2:
server_2: going offline
rdf: going active
replace in progress ... done
failover activity complete
commit in progress (not interruptible)... done
done
server_3:
server_3: going offline
rdf: going active
replace in progress ... done
failover activity complete
commit in progress (not interruptible)... done
done
server_4:
Error 4003: server_4: standby is not configured
server_5:
Error 4003: server_5: standby is not configured
Suspend RDF links....................................... Done.
Merge device track tables between source and target....... Started.
Device: 045A in (0557,005)............................... Merged.
Merge device track tables between source and target....... Done.
Resume RDF links........................................ Started.
Resume RDF links........................................ Done.

EXAMPLE #3
To initiate an SRDF failover from the source VNX to the destination, without the SRDF health check, as a root user, type:

# /nas/sbin/nas_rdf --activate --nocheck

Skipping SRDF health check ....

Is remote site CELERRA completely shut down (power OFF)?
Do you wish to continue? [yes or no]: yes
Successfully pinged (Remotely) Symmetrix ID: 000187430809
Successfully pinged (Remotely) Symmetrix ID: 000190100559
Successfully pinged (Remotely) Symmetrix ID: 000190100582
Write Disable devices on SA at source (R1).............. Done.
Suspend RDF links........................................ Done.
Read/Write Enable devices on RA at target (R2)......... Done.
Waiting for nbs clients to die ... done
Waiting for nbs clients to start ... done
fsck 1.35 (28-Feb-2004)
/dev/ndj1: recovering journal
/dev/ndj1: clean, 13780/231360 files, 233674/461860 blocks
Waiting for nbs clients to die ... done
Waiting for nbs clients to start ... done
The nas Commands

<table>
<thead>
<tr>
<th>id</th>
<th>type</th>
<th>acl</th>
<th>slot</th>
<th>groupID</th>
<th>state</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1000</td>
<td>2</td>
<td>0</td>
<td>server_2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1000</td>
<td>3</td>
<td>0</td>
<td>server_3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1000</td>
<td>4</td>
<td>0</td>
<td>server_4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1000</td>
<td>5</td>
<td>0</td>
<td>server_5</td>
<td></td>
</tr>
</tbody>
</table>

**server_2:**
- going offline
- rdf: going active
- replace in progress ...done
- failover activity complete
- commit in progress (not interruptible)...done
- done

**server_3:**
- going offline
- rdf: going active
- replace in progress ...done
- failover activity complete
- commit in progress (not interruptible)...done
- done

**server_4:**
- Error 4003: server_4 : standby is not configured

**server_5:**
- Error 4003: server_5 : standby is not configured

**EXAMPLE #4**

To initiate a Dynamic SRDF failover from the source VNX to the destination, as a **root** user, type:

```
# /nas/sbin/nas_rdf -activate -reverse
```

Is remote site CELERRA completely shut down (power OFF)?
Do you wish to continue? [yes or no]: yes

Successfully pinged (Remotely) Symmetrix ID: 000280600118
- Write Disable devices on SA at source (R1)..............Done.
- Suspend RDF links...........................................Done.
- Read/Write Enable devices on RA at target (R2)........Done.

```
fsck 1.35 (28-Feb-2004)
/dev/sdj1: recovering journal
Clearing orphaned inode 37188 (uid=0, gid=0, mode=0100644, size=0)
```

```
/dev/sdj1: clean, 12860/219968 files, 194793/439797 blocks
```

<table>
<thead>
<tr>
<th>id</th>
<th>type</th>
<th>acl</th>
<th>slot</th>
<th>groupID</th>
<th>state</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1000</td>
<td>2</td>
<td>0</td>
<td>server_2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1000</td>
<td>3</td>
<td>0</td>
<td>server_3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2000</td>
<td>4</td>
<td>0</td>
<td>server_4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2000</td>
<td>5</td>
<td>0</td>
<td>server_5</td>
<td></td>
</tr>
</tbody>
</table>

**server_2:**
server_2 : going offline
rdf : going active
replace in progress ...done
failover activity complete
commit in progress (not interruptible)...done
done
server_3 :
server_3 : going offline
rdf : going active
replace in progress ...done
failover activity complete
commit in progress (not interruptible)...done
done
An RDF 'Swap Personality' operation execution is
in progress for device group '1R2_500_1'. Please wait...

Swap RDF Personality..............................Started.
Swap RDF Personality..............................Done.

The RDF 'Swap Personality' operation successfully executed for
device group '1R2_500_1'.

An RDF 'Incremental Establish' operation execution is
in progress for device group '1R2_500_1'. Please wait...

Suspend RDF links..............................Done.
Resume RDF links..............................Started.
Merge device track tables between source and target......Started.
Devices: 0009-000B .............................. Merged.
Devices: 0032-0034 .............................. Merged.
Devices: 0035-0037 .............................. Merged.
Devices: 0038-003A .............................. Merged.
Devices: 003B-003D .............................. Merged.
Devices: 003E-0040 .............................. Merged.
Devices: 0041-0043 .............................. Merged.
Devices: 0044-0046 .............................. Merged.
Devices: 0047-0049 .............................. Merged.

Merge device track tables between source and target......Done.
Resume RDF links..............................Done.

The RDF 'Incremental Establish' operation successfully initiated for
device group '1R2_500_1'.
EXAMPLE #5  To restore a source VNX after failover, as a root user, type:

```
# /nas/sbin/nas_rdf -restore
```

Is remote site CELERRA ready for Storage restoration?
Do you wish to continue? [yes or no]: yes
Contact Joker_R1_CS0 ... is alive

Restore will now reboot the source site control station.
Do you wish to continue? [yes or no]: yes

Device Group (DG) Name : 1R2_500_5
DG's Type : RDF2
DG's Symmetrix ID : 000190100557

<table>
<thead>
<tr>
<th>Target (R2) View</th>
<th>Source (R1) View</th>
<th>MODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>LI</td>
<td>ST</td>
</tr>
<tr>
<td>Standard</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Logical</td>
<td>T</td>
<td>R1 Inv</td>
</tr>
<tr>
<td>Device</td>
<td>Dev E Tracks</td>
<td>Tracks S</td>
</tr>
<tr>
<td>DEV001 045A RW</td>
<td>10 0 RW</td>
<td>045A WD</td>
</tr>
<tr>
<td>DEV002 045B RW</td>
<td>2054 0 NR</td>
<td>045B WD</td>
</tr>
<tr>
<td>DEV003 045C RW</td>
<td>0 0 NR</td>
<td>045C WD</td>
</tr>
<tr>
<td>DEV004 045D RW</td>
<td>0 0 NR</td>
<td>045D WD</td>
</tr>
<tr>
<td>DEV005 045E RW</td>
<td>1284 0 NR</td>
<td>045E WD</td>
</tr>
<tr>
<td>DEV006 045F RW</td>
<td>0 0 NR</td>
<td>045F WD</td>
</tr>
<tr>
<td>DEV007 0467 RW</td>
<td>0 0 NR</td>
<td>0467 WD</td>
</tr>
<tr>
<td>DEV008 0468 RW</td>
<td>2 0 NR</td>
<td>0468 WD</td>
</tr>
<tr>
<td>DEV009 0469 RW</td>
<td>0 0 NR</td>
<td>0469 WD</td>
</tr>
<tr>
<td>DEV010 046A RW</td>
<td>0 0 NR</td>
<td>046A WD</td>
</tr>
<tr>
<td>DEV011 046B RW</td>
<td>2 0 NR</td>
<td>046B WD</td>
</tr>
<tr>
<td>DEV012 046C RW</td>
<td>0 0 NR</td>
<td>046C WD</td>
</tr>
<tr>
<td>DEV013 046D RW</td>
<td>0 0 NR</td>
<td>046D WD</td>
</tr>
<tr>
<td>DEV014 046E RW</td>
<td>0 0 NR</td>
<td>046E WD</td>
</tr>
<tr>
<td>DEV015 046F RW</td>
<td>2 0 NR</td>
<td>046F WD</td>
</tr>
<tr>
<td>DEV016 0470 RW</td>
<td>0 0 NR</td>
<td>0470 WD</td>
</tr>
<tr>
<td>DEV017 0471 RW</td>
<td>2 0 NR</td>
<td>0471 WD</td>
</tr>
<tr>
<td>DEV018 0472 RW</td>
<td>0 0 NR</td>
<td>0472 WD</td>
</tr>
<tr>
<td>DEV019 0473 RW</td>
<td>0 0 NR</td>
<td>0473 WD</td>
</tr>
<tr>
<td>DEV020 0474 RW</td>
<td>0 0 NR</td>
<td>0474 WD</td>
</tr>
<tr>
<td>DEV021 0475 RW</td>
<td>0 0 NR</td>
<td>0475 WD</td>
</tr>
<tr>
<td>DEV022 0476 RW</td>
<td>0 0 NR</td>
<td>0476 WD</td>
</tr>
<tr>
<td>DEV023 0477 RW</td>
<td>2 0 NR</td>
<td>0477 WD</td>
</tr>
<tr>
<td>DEV024 0478 RW</td>
<td>2 0 NR</td>
<td>0478 WD</td>
</tr>
<tr>
<td>DEV025 0479 RW</td>
<td>0 0 NR</td>
<td>0479 WD</td>
</tr>
<tr>
<td>DEV026 047A RW</td>
<td>0 0 NR</td>
<td>047A WD</td>
</tr>
<tr>
<td>DEV027 047B RW</td>
<td>0 0 NR</td>
<td>047B WD</td>
</tr>
<tr>
<td>DEV028 047C RW</td>
<td>0 0 NR</td>
<td>047C WD</td>
</tr>
<tr>
<td>DEV029 047D RW</td>
<td>0 0 NR</td>
<td>047D WD</td>
</tr>
<tr>
<td>DEV030 047E RW</td>
<td>0 0 NR</td>
<td>047E WD</td>
</tr>
</tbody>
</table>
### The nas Commands

<table>
<thead>
<tr>
<th>Device</th>
<th>Mode</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV031</td>
<td>047F</td>
<td>RW</td>
</tr>
<tr>
<td>DEV032</td>
<td>0480</td>
<td>RW</td>
</tr>
<tr>
<td>DEV033</td>
<td>0481</td>
<td>RW</td>
</tr>
<tr>
<td>DEV034</td>
<td>0482</td>
<td>RW</td>
</tr>
<tr>
<td>DEV035</td>
<td>0483</td>
<td>RW</td>
</tr>
<tr>
<td>DEV036</td>
<td>0484</td>
<td>RW</td>
</tr>
<tr>
<td>DEV037</td>
<td>0485</td>
<td>RW</td>
</tr>
<tr>
<td>DEV038</td>
<td>0486</td>
<td>RW</td>
</tr>
<tr>
<td>DEV039</td>
<td>0487</td>
<td>RW</td>
</tr>
<tr>
<td>DEV040</td>
<td>0488</td>
<td>RW</td>
</tr>
<tr>
<td>DEV041</td>
<td>0489</td>
<td>RW</td>
</tr>
<tr>
<td>DEV042</td>
<td>048A</td>
<td>RW</td>
</tr>
<tr>
<td>DEV043</td>
<td>048B</td>
<td>RW</td>
</tr>
<tr>
<td>DEV044</td>
<td>048C</td>
<td>RW</td>
</tr>
<tr>
<td>DEV045</td>
<td>048D</td>
<td>RW</td>
</tr>
<tr>
<td>DEV046</td>
<td>048E</td>
<td>RW</td>
</tr>
<tr>
<td>DEV047</td>
<td>048F</td>
<td>RW</td>
</tr>
<tr>
<td>DEV048</td>
<td>0490</td>
<td>RW</td>
</tr>
<tr>
<td>DEV049</td>
<td>0491</td>
<td>RW</td>
</tr>
<tr>
<td>DEV050</td>
<td>0492</td>
<td>RW</td>
</tr>
<tr>
<td>DEV051</td>
<td>0493</td>
<td>RW</td>
</tr>
<tr>
<td>DEV052</td>
<td>0494</td>
<td>RW</td>
</tr>
<tr>
<td>DEV053</td>
<td>0495</td>
<td>RW</td>
</tr>
<tr>
<td>DEV054</td>
<td>0496</td>
<td>RW</td>
</tr>
<tr>
<td>DEV055</td>
<td>0497</td>
<td>RW</td>
</tr>
<tr>
<td>DEV056</td>
<td>0498</td>
<td>RW</td>
</tr>
<tr>
<td>DEV057</td>
<td>0499</td>
<td>RW</td>
</tr>
<tr>
<td>DEV058</td>
<td>049A</td>
<td>RW</td>
</tr>
<tr>
<td>DEV059</td>
<td>049B</td>
<td>RW</td>
</tr>
<tr>
<td>DEV060</td>
<td>049C</td>
<td>RW</td>
</tr>
<tr>
<td>DEV061</td>
<td>049D</td>
<td>RW</td>
</tr>
<tr>
<td>DEV062</td>
<td>049E</td>
<td>RW</td>
</tr>
<tr>
<td>DEV063</td>
<td>049F</td>
<td>RW</td>
</tr>
<tr>
<td>DEV064</td>
<td>04A0</td>
<td>RW</td>
</tr>
<tr>
<td>DEV065</td>
<td>04A1</td>
<td>RW</td>
</tr>
<tr>
<td>DEV066</td>
<td>04A2</td>
<td>RW</td>
</tr>
<tr>
<td>DEV067</td>
<td>04A3</td>
<td>RW</td>
</tr>
<tr>
<td>DEV068</td>
<td>04A4</td>
<td>RW</td>
</tr>
<tr>
<td>DEV069</td>
<td>04A5</td>
<td>RW</td>
</tr>
<tr>
<td>DEV070</td>
<td>04A6</td>
<td>RW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mode</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Failed Over</td>
</tr>
<tr>
<td>S</td>
<td>Failed Over</td>
</tr>
<tr>
<td>D</td>
<td>Failed Over</td>
</tr>
</tbody>
</table>

---

**Legend for MODES:**

- **Mode of Operation**: A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
- **Domino**: X = Enabled, . = Disabled
- **Adaptive Copy**: D = Disk Mode, W = WP Mode, . = ACp off

- Suspend RDF link(s) .................................. Done.
- Merge device track tables between source and target ....... Started.
- Devices: 0478-0489 in (0557,005) ...................... Merged.
- Devices: 048A-049B in (0557,005) ...................... Merged.
The nas Commands

Devices: 049C-04A6 in (0557,005)......................... Merged.
Merge device track tables between source and target.......Done.
Resume RDF link(s)........................................Started.
Resume RDF link(s)........................................Done.

Is remote site CELERRA ready for Network restoration?
Do you wish to continue? [yes or no]: yes
server_2 : done
server_3 : done
server_4 : Error 4003: server_4 : standby is not configured
server_5 : Error 4003: server_5 : standby is not configured

fsck 1.35 (28-Feb-2004)
/dev/ndj1: clean, 13836/231360 files, 233729/461860 blocks
Waiting for nbs clients to die ... done
Waiting for nbs clients to start ... done
Waiting for nbs clients to die ... done
Waiting for nbs clients to start ... done
Waiting for 1R2_500_5 access ... done
Write Disable device(s) on RA at target (R2)..............Done.
Suspend RDF link(s).......................................Done.
Merge device track tables between source and target......Started.
Devices: 045A-045F, 0467-0477 in (0557,005).............. Merged.
Devices: 0478-0489 in (0557,005)......................... Merged.
Devices: 048A-049B in (0557,005)......................... Merged.
Devices: 049C-04A6 in (0557,005)......................... Merged.
Merge device track tables between source and target.....Done.
Resume RDF link(s)........................................Started.
Resume RDF link(s)........................................Done.
Read/Write Enable device(s) on SA at source (R1).........Done.
Waiting for 1R2_500_5 sync ... done
Starting restore on remote site CELERRA ...
Waiting for nbs clients to start ... done
Waiting for nbs clients to start ... done
Suspend RDF link(s).......................................Done.
server_2 :
server_2 : going standby
rdf : going active
replace in progress ...done
failover activity complete
commit in progress (not interruptible)...done

server_3 :
server_3 : going standby
rdf : going active
replace in progress ...done
failover activity complete
commit in progress (not interruptible)...done
done
server_4 :
Error 4003: server_4 : standby is not configured
server_5 :
Error 4003: server_5 : standby is not configured
Resume RDF link(s).....................................................Started.
Resume RDF link(s).....................................................Done.
If the RDF device groups were setup to operate in ASYNCHRONOUS (SRDF/A) mode, now would be a good time to set it back to that mode.
Would you like to set device group 1R2_500_5 to ASYNC Mode? [yes or no]: no
done

EXAMPLE #6  To restore a source VNX after failover, without the SRDF health check, as a root user, type:

# /nas/sbin/nas_rdf -restore -nocheck

Skipping SRDF health check ....

Is remote site CELERRA ready for Storage restoration?
Do you wish to continue? [yes or no]: yes
Contact Joker_R1_CS0 ... is alive
Restore will now reboot the source site control station.
Do you wish to continue? [yes or no]: yes
Device Group (DG) Name : 1R2_500_5
DG's Type : RDF2
DG's Symmetrix ID : 000190100557
Target (R2) View Source (R1) View MODES
-------------------------------- ------------------------ ----- ------------
ST LI ST
Standard A N A
Logical T R1 Inv R2 Inv K T R1 Inv R2 Inv RDF Pair
Device Dev E Tracks Tracks S Dev E Tracks Tracks MDA STATE
-------------------------------- -- ------------------------ ----- ------------
DEV001 045A RW 10 0 RW 045A WD 0 0 S.. R1 Updated
DEV002 045B RW 2054 0 NR 045B WD 0 0 S.. Failed Over
DEV003 045C RW 0 0 NR 045C WD 0 0 S.. Failed Over
DEV004 045D RW 0 0 NR 045D WD 0 0 S.. Failed Over
DEV005 045E RW 1284 0 NR 045E WD 0 0 S.. Failed Over
DEV006 045F RW 0 0 NR 045F WD 0 0 S.. Failed Over
DEV007 0467 RW 0 0 NR 0467 WD 0 0 S.. Failed Over
DEV008 0468 RW 2 0 NR 0468 WD 0 0 S.. Failed Over
DEV009 0469 RW 0 0 NR 0469 WD 0 0 S.. Failed Over
DEV010 046A RW 0 0 NR 046A WD 0 0 S.. Failed Over
DEV011 046B RW 2 0 NR 046B WD 0 0 S.. Failed Over
DEV012 046C RW 0 0 NR 046C WD 0 0 S.. Failed Over
DEV013 046D RW 0 0 NR 046D WD 0 0 S.. Failed Over
DEV014 046E RW 0 0 NR 046E WD 0 0 S.. Failed Over
DEV015 046F RW 2 0 NR 046F WD 0 0 S.. Failed Over
DEV016 0470 RW 0 0 NR 0470 WD 0 0 S.. Failed Over
DEV017 0471 RW 2 0 NR 0471 WD 0 0 S.. Failed Over
DEV018 0472 RW 0 0 NR 0472 WD 0 0 S.. Failed Over
DEV019 0473 RW 0 0 NR 0473 WD 0 0 S.. Failed Over
DEV020 0474 RW 0 0 NR 0474 WD 0 0 S.. Failed Over
DEV021 0475 RW 0 0 NR 0475 WD 0 0 S.. Failed Over
DEV022 0476 RW 0 0 NR 0476 WD 0 0 S.. Failed Over
DEV023 0477 RW 2 0 NR 0477 WD 0 0 S.. Failed Over
DEV024 0478 RW 2 0 NR 0478 WD 0 0 S.. Failed Over
DEV025 0479 RW 0 0 NR 0479 WD 0 0 S.. Failed Over
DEV026 047A RW 0 0 NR 047A WD 0 0 S.. Failed Over
DEV027 047B RW 0 0 NR 047B WD 0 0 S.. Failed Over
DEV028 047C RW 0 0 NR 047C WD 0 0 S.. Failed Over
DEV029 047D RW 0 0 NR 047D WD 0 0 S.. Failed Over
DEV030 047E RW 0 0 NR 047E WD 0 0 S.. Failed Over
DEV031 047F RW 0 0 NR 047F WD 0 0 S.. Failed Over
DEV032 0480 RW 0 0 NR 0480 WD 0 0 S.. Failed Over
DEV033 0481 RW 0 0 NR 0481 WD 0 0 S.. Failed Over
DEV034 0482 RW 0 0 NR 0482 WD 0 0 S.. Failed Over
DEV035 0483 RW 0 0 NR 0483 WD 0 0 S.. Failed Over
DEV036 0484 RW 0 0 NR 0484 WD 0 0 S.. Failed Over
DEV037 0485 RW 0 0 NR 0485 WD 0 0 S.. Failed Over
DEV038 0486 RW 0 0 NR 0486 WD 0 0 S.. Failed Over
DEV039 0487 RW 0 0 NR 0487 WD 0 0 S.. Failed Over
DEV040 0488 RW 0 0 NR 0488 WD 0 0 S.. Failed Over
DEV041 0489 RW 0 0 NR 0489 WD 0 0 S.. Failed Over
DEV042 048A RW 0 0 NR 048A WD 0 0 S.. Failed Over
DEV043 048B RW 0 0 NR 048B WD 0 0 S.. Failed Over
DEV044 048C RW 0 0 NR 048C WD 0 0 S.. Failed Over
DEV045 048D RW 0 0 NR 048D WD 0 0 S.. Failed Over
DEV046 048E RW 0 0 NR 048E WD 0 0 S.. Failed Over
DEV047 048F RW 2 0 NR 048F WD 0 0 S.. Failed Over
DEV048 0490 RW 0 0 NR 0490 WD 0 0 S.. Failed Over
DEV049 0491 RW 0 0 NR 0491 WD 0 0 S.. Failed Over
DEV050 0492 RW 0 0 NR 0492 WD 0 0 S.. Failed Over
DEV051 0493 RW 0 0 NR 0493 WD 0 0 S.. Failed Over
DEV052 0494 RW 0 0 NR 0494 WD 0 0 S.. Failed Over
DEV053 0495 RW 0 0 NR 0495 WD 0 0 S.. Failed Over
DEV054 0496 RW 0 0 NR 0496 WD 0 0 S.. Failed Over
DEV055 0497 RW 2 0 NR 0497 WD 0 0 S.. Failed Over
DEV056 0498 RW 2 0 NR 0498 WD 0 0 S.. Failed Over
DEV057 0499 RW 0 0 NR 0499 WD 0 0 S.. Failed Over
DEV058 049A RW 0 0 NR 049A WD 0 0 S.. Failed Over
DEV059 049B RW 0 0 NR 049B WD 0 0 S.. Failed Over
DEV060 049C RW 0 0 NR 049C WD 0 0 S.. Failed Over
DEV061 049D RW 0 0 NR 049D WD 0 0 S.. Failed Over
DEV062 049E RW 0 0 NR 049E WD 0 0 S.. Failed Over
DEV063 049F RW 0 0 NR 049F WD 0 0 S.. Failed Over
DEV064 04A0 RW 0 0 NR 04A0 WD 0 0 S.. Failed Over
DEV065 04A1 RW 0 0 NR 04A1 WD 0 0 S.. Failed Over
DEV066 04A2 RW 0 0 NR 04A2 WD 0 0 S.. Failed Over
DEV067 04A3 RW 0 0 NR 04A3 WD 0 0 S.. Failed Over
DEV068 04A4 RW 0 0 NR 04A4 WD 0 0 S.. Failed Over
DEV069 04A5 RW 0 0 NR 04A5 WD 0 0 S.. Failed Over
DEV070 04A6 RW 0 0 NR 04A6 WD 0 0 S.. Failed Over
Total -------- -------- -------- --------
Track(s) 3366 0 0 0
MB(s) 105.2 0.0 0.0 0.0

Legend for MODES:
M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
D(omino) : X = Enabled, . = Disabled
A(daptive Copy) : D = Disk Mode, W = WP Mode, . = ACp off
Suspend RDF link(s)........................................ Done.
Merge device track tables between source and target....... Started.
Devices: 045A-045F, 0467-0477 in (0557,005)............ Merged.
Devices: 0478-0489 in (0557,005).......................... Merged.
Devices: 048A-049B in (0557,005).......................... Merged.
Devices: 049C-04A6 in (0557,005)......................... Merged.
Merge device track tables between source and target....... Done.
Resume RDF link(s)........................................ Started.
Resume RDF link(s)........................................ Done.
Is remote site CELERRA ready for Network restoration?
Do you wish to continue? [yes or no]: yes
server_2 : done
server_3 : done
server_4 :
Error 4003: server_4 : standby is not configured
server_5 :
Error 4003: server_5 : standby is not configured
fsck 1.35 (28-Feb-2004)
/dev/ndj1: clean, 13836/231360 files, 233729/461860 blocks
Waiting for nbs clients to die ... done
Waiting for nbs clients to start ... done
Waiting for nbs clients to die ... done
Waiting for nbs clients to start ... done
Waiting for 1R2_500_5 access ... done
Write Disable device(s) on RA at target (R2).............. Done.
Suspend RDF link(s)........................................ Done.
Merge device track tables between source and target....... Started.
Devices: 045A-045F, 0467-0477 in (0557,005)............ Merged.
Devices: 0478-0489 in (0557,005).......................... Merged.
Devices: 048A-049B in (0557,005).......................... Merged.
Devices: 049C-04A6 in (0557,005)......................... Merged.
Merge device track tables between source and target....... Done.
Resume RDF link(s)........................................ Started.
Resume RDF link(s)........................................ Done.
Read/Write Enable device(s) on SA at source (R1)......... Done.
Waiting for 1R2_500_5 sync ... done
Starting restore on remote site CELERRA ...
Waiting for nbs clients to start ... done
Waiting for nbs clients to start ... done
Suspend RDF link(s)........................................ Done.
server_2 :
server_2 : going standby
rdf : going active
replace in progress ... done
failover activity complete
commit in progress (not interruptible)... done
done
server_3 :
server_3 : going standby
rdf : going active
replace in progress ...done
failover activity complete
commit in progress (not interruptible)...done
done
server_4 :
Error 4003: server_4 : standby is not configured
server_5 :
Error 4003: server_5 : standby is not configured
Resume RDF link(s)........................................Started.
Resume RDF link(s)........................................Done.
If the RDF device groups were setup to operate in ASYNCHRONOUS ( SRDF/A ) mode,
now would be a good time to set it back to that mode.
Would you like to set device group 1R2_500_5 to ASYNC Mode ? [yes or no]: no
done

EXAMPLE #7  To restore a source VNX after failover, when using Dynamic SRDF, as
a root user, type:

# /nas/sbin/nas_rdf -restore
Is remote site CELERRA ready for Storage restoration?
Do you wish to continue? [yes or no]: yes
Contact eng17335 ... is alive

Restore will now reboot the source site control station.
Do you wish to continue? [yes or no]: yes

Device Group (DG) Name               : 1R2_500_1
DG's Type                            : RDF1
DG's Symmetrix ID                    : 000280600187    (Microcode Version: 5568)
Remote Symmetrix ID                  : 000280600118    (Microcode Version: 5568)
RDF (RA) Group Number                : 1 (00)

<table>
<thead>
<tr>
<th>Source (R1) View</th>
<th>Target (R2) View</th>
<th>MODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>LI</td>
<td>ST</td>
</tr>
<tr>
<td>Standard</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Logical</td>
<td>T R1 Inv R2 Inv</td>
<td>K</td>
</tr>
<tr>
<td>Device</td>
<td>Dev</td>
<td>E</td>
</tr>
<tr>
<td>Device  Dev</td>
<td>E Tracks Tracks</td>
<td>S</td>
</tr>
<tr>
<td>Device  Dev</td>
<td>E Tracks Tracks</td>
<td>MDA</td>
</tr>
<tr>
<td>Device  Dev</td>
<td>E Tracks Tracks</td>
<td>STATE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device</th>
<th>Dev</th>
<th>E Tracks</th>
<th>Tracks</th>
<th>S</th>
<th>MDA</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0056</td>
<td>0056</td>
<td>RW</td>
<td>0</td>
<td>0</td>
<td>RW</td>
<td>0030 WD</td>
</tr>
<tr>
<td>0057</td>
<td>0057</td>
<td>RW</td>
<td>0</td>
<td>0</td>
<td>RW</td>
<td>0031 WD</td>
</tr>
<tr>
<td>0032</td>
<td>0032</td>
<td>RW</td>
<td>0</td>
<td>0</td>
<td>RW</td>
<td>000C WD</td>
</tr>
<tr>
<td>.........</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0069</td>
<td>0069</td>
<td>RW</td>
<td>0</td>
<td>0</td>
<td>RW</td>
<td>005F WD</td>
</tr>
<tr>
<td>006A</td>
<td>006A</td>
<td>RW</td>
<td>0</td>
<td>0</td>
<td>RW</td>
<td>0060 WD</td>
</tr>
<tr>
<td>006B</td>
<td>006B</td>
<td>RW</td>
<td>0</td>
<td>0</td>
<td>RW</td>
<td>0061 WD</td>
</tr>
<tr>
<td>.........</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Track(s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB(s)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
The nas Commands

Legend for MODES: M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
D(omino) : X = Enabled, . = Disabled
A(daptive Copy) : D = Disk Mode, W = WP Mode, . = ACp off

Is remote site CELERRA ready for Network restoration?
Do you wish to continue? [yes or no]: yes
server_2 : done
tserver_3 : done
tserver_4 :
Error 4003: server_4 : standby is not configured
tserver_5 :
Error 4003: server_5 : standby is not configured

fsck 1.35 (28-Feb-2004)
/dev/sdj1: clean, 12956/219968 files, 188765/439797 blocks

An RDF 'Failover' operation execution is
in progress for device group '1R2_500_1'. Please wait...
  
  Write Disable device(s) on SA at source (R1)............Done.
  Suspend RDF link(s)......................................Done.
  Swap RDF Personality..................................Started.
  Swap RDF Personality..................................Done.
  Suspend RDF link(s)......................................Done.
  Read/Write Enable device(s) on SA at source (R1).........Done.
  Resume RDF link(s)......................................Started.
  Resume RDF link(s)......................................Done.
  Read/Write Enable device(s) on SA at target (R2).........Done.

The RDF 'Failover' operation successfully executed for
device group '1R2_500_1'.

Waiting for 1R2_500_1 sync ...done

Starting restore on remote site CELERRA ...
Suspend RDF link(s)......................................Done.
tserver_2 :
tserver_2 : going standby
t   rdf : going active
t   replace in progress ...done
   failover activity complete
t   commit in progress (not interruptible)...done
done
tserver_3 :
tserver_3 : going standby
t   rdf : going active
t   replace in progress ...done
   failover activity complete
t   commit in progress (not interruptible)...done
done
tserver_4 :

Error 4003: server_4: standby is not configured
server_5:
Error 4003: server_5: standby is not configured

done

**EXAMPLE #8** To run all available checks on a source VNX, as a root user, type:

```bash
#/nas/sbin/nas_rdf -check -all
```

--------------------- SRDF Health Checks ---------------------
SRDF: Checking device is normal.......................... Pass
SRDF: Checking system is restored........................ Pass
SRDF: Checking R1 SRDF session is Synch or Consistent.... Pass
SRDF: Checking R1 Data Mover configuration is valid...... Pass
SRDF: Checking R1 devices are available.................. Pass
SRDF: Checking R1 device group has all devices.......... Pass
SRDF: Checking R2 SRDF session is Synch or Consistent.... Pass
SRDF: Checking R2 Data Mover configuration is valid...... Pass
SRDF: Checking R2 devices are available.................. Pass
SRDF: Checking R2 device group has all devices.......... Pass

**EXAMPLE #9** To run one or more specific available checks on a source VNX, as a root user, type:

```bash
#/nas/sbin/nas_rdf -check r1_dev_group,r2_dev_group
```

--------------------- SRDF Health Checks ---------------------
SRDF: Checking R1 device group has all devices......... Pass
SRDF: Checking R2 device group has all devices......... Pass
The nas Commands

**nas_replicate**

Manages loopback, local, and remote VNX Replicator sessions.

**SYNOPSIS**

nas_replicate
- list [-id]
  | - info { -all | id=<sessionId> | <name> }
  |
- create <name>
  - source -fs {<fsName>|id=<fsId>}
    [-sav {<srcSavVolStoragePool>|id=<srcSavVolStoragePoolId>}] 
  - destination {-fs {id=<dstFsId>|<existing_dstFsName>} 
    | -pool {id=<dstStoragePoolId>|<dstStoragePool>} } [-vdm <dstVdmName>] 
    [-sav {id=<dstSavVolStoragePoolId>|<dstSavVolStoragePool>}] 
  - interconnect {<name>|id=<interConnectId>} 
    [-source_interface {ip=<ipAddr>|<nameServiceInterfaceName>}] 
    [-destination_interface {ip=<ipAddr>|<nameServiceInterfaceName>}] 
    [{-max_time_out_of_sync <maxTimeOutOfSync>|-manual_refresh}] 
    [-overwrite_destination][-tape_copy][-background]
  |
- create <name>
  - source -vdm <vdmName>
  - destination { -vdm <existing_dstVdmName> | -pool 
    {id=<dstStoragePoolId>|<dstStoragePool>} } 
  - interconnect {<name>|id=<interConnectId>} 
    [-source_interface {ip=<ipAddr>|<nameServiceInterfaceName>}] 
    [-destination_interface {ip=<ipAddr>|<nameServiceInterfaceName>}] 
    [{-max_time_out_of_sync <maxTimeOutOfSync>|-manual_refresh}] 
    [-overwrite_destination][-background]
  |
- create <name>
  - source -lun <lunNumber> -target <targetIqn>
  - destination -lun <lunNumber> -target <targetIqn>
  - interconnect {<name>|id=<interConnectId>} 
    [-source_interface {ip=<ipAddr>|<nameServiceInterfaceName>}] 
    [-destination_interface {ip=<ipAddr>|<nameServiceInterfaceName>}] 
    [{-max_time_out_of_sync <maxTimeOutOfSync>|-manual_refresh}] 
    [-overwrite_destination][-reverse][-full_copy][-background]
  |
- start {<name>|id=<sessionId>} 
  [-interconnect {<name>|id=<interConnectId>}] 
  [-source_interface {ip=<ipAddr>|<nameServiceInterfaceName>}] 
  [-destination_interface {ip=<ipAddr>|<nameServiceInterfaceName>}] 
  [{-max_time_out_of_sync <maxTimeOutOfSync>|-manual_refresh}] 
  [-overwrite_destination][-background]
  |
- modify {<name>|id=<sessionId>} {-name <new name> }
  [-source_interface {ip=<ipAddr>|<nameServiceInterfaceName>}] 
  [-destination_interface {ip=<ipAddr>|<nameServiceInterfaceName>}] 
  [{-max_time_out_of_sync <maxTimeOutOfSync>|-manual_refresh}]
The nas Commands

| -stop {<name>|id=<sessionId>} [-mode {source|destination|both}] [-background] |
| -delete {<name>|id=<sessionId>} [-mode {source|destination|both}] [-background] |
| -failover {<name>|id=<sessionId>} [-background] |
| -switchover {<name>|id=<sessionId>} |
| -reverse {<name>|id=<sessionId>} [-background] |
| -refresh {<name>|id=<sessionId>} [-source {<ckptName>|id=<ckptId>}] [-destination {<ckptName>|id=<ckptId>}] [-background] |

Note: This command manages replication sessions using VNX Replicator. For a one-time file system copy using VNX Replicator, use the nas_copy command. For ongoing file system, use the fs_ckpt command.

DESCRIPTION

nas_replicate creates, manages, or displays session information for ongoing VNX Replicator replication of a file system, Virtual Data Mover (VDM), or iSCSI LUN at a destination using an existing Data Mover interconnect. Each session handles a single source object and destination, and is assigned a globally unique ID, fixed for the life of the session.

In response to a potential disaster scenario, use nas_replicate to perform a failover of a specified replication session with possible data loss. The -switchover option switches over a replication relationship and performs synchronization of the source and destination without data loss. Use nas_replicate to also reverse the direction of a replication session or refresh the destination side with updates to the source based on a time-out of synchronization value or on demand.

OPTIONS

-list [-id]
Displays all configured (or stopped) replication sessions (V2 file system, VDM, iSCSI LUN, or copy sessions) on each Data Mover in the VNX for file cabinet. Each session is represented by either a name or a session ID that is generated automatically whenever a session is configured and is globally unique.

Use this option to obtain the session ID needed for another command. Since session IDs are lengthy, the session ID obtained from this command can be copied and pasted into the command.

-info {all|id=<sessionId>|<name>}
Displays the status of a specific configured (or stopped) replication
CREATING FILE SYSTEM
REPLICATION

The nas Commands

session (V2 file system, VDM, iSCSI LUN, or copy session, or the status of all replication sessions).

-**create** <name>
Assigns a name to the file system replication session. The name must be unique for each Data Mover pair, which is defined by the interconnect.

-**source -fs** {<fsName>|id=<fsId>} [-sav <srcSavVolStoragePoolId>|id=<srcSavVolStoragePoolId>]
Specifies the name or ID of the existing source file system to replicate. The source file system must be mounted as read-only or read/write.

**Note:** If the source file system is mounted to a VDM and the goal is to replicate a CIFS environment for disaster recovery (that is, replicate a VDM and the file systems mounted to the VDM), create a session to replicate the VDM first, before replicating a file system mounted to the VDM.

The **-sav** option allocates a storage pool for all subsequent checkpoints for the file system. By default, if checkpoint storage (the checkpoint SavVol) needs to be allocated for checkpoints of the file system, the command uses the same storage pool used to create the source file system.

-**destination** {-fs {<existing_dstFsName>|id=<dstFsId> | -pool <dstStoragePool>|id=<dstStoragePoolId> | -vdm <dstVdmName>}}[-sav {<dstSavVolStoragePool>|id=<dstSavVolStoragePoolId>}
Specifies an existing destination file system or the storage needed to create the destination file system. An existing destination file system must be mounted as read-only and the same size as the source. Specifying a storage pool or ID creates the read-only, destination file system automatically, using the same name and size as the source file system.

Specifying a pool with the **-vdm** option mounts the destination file system to an existing VDM as part of replication in a CIFS environment.
The `-sav` option allocates a storage pool for all subsequent checkpoints of the destination file system. By default, if destination checkpoint storage needs to be allocated for checkpoints, the command uses the same storage pool used to create the destination file system.

By default, the destination file system name will be the same as the source file system name. If a file system with the same name as the source file system already exists on the destination, the naming convention `<source_fs_name>_replica<#>` will be used. A number 1–4 is assigned according to how many replicas of that file system already exist.

`-interconnect {<name>|id=<interConnectId>}`

Specifies the local (source) side of an established Data Mover interconnect to use for this replication session.

Use the `nas_cel -interconnect -list` command on the source VNX for file to list the interconnects available to the replication sessions.

`[-source_interface
  {<nameServiceInterfaceName>|ip=<ipAddr>}]`  
Instructs the replication session to use a specific local interface defined for the interconnect on the source VNX instead of selecting the local interface supporting the lowest number of sessions (the default). If this local interface was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address. If you define an interface using an IP address, make sure that the destination interface uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

The `source_interfaces` field of the output from the `nas_cel -interconnect -info` command shows how the source interface was defined. This option does not apply to a loopback interconnect, which always uses 127.0.0.1.

If no source interface is specified, the system will select an interface. This ensures that the interface selected can communicate with the destination interface.

`[-destination_interface{<nameServiceInterfaceName>|ip=<ipaddr>}]`  
Instructs the replication session to use a specific peer interface
defined for the interconnect on the destination VNX instead of selecting the peer interface supporting the lowest number of sessions (the default). If this peer interface was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address. If you define an interface using an IP address, make sure that the source interface uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

The destination_interfaces field of the output from the nas_cel -interconnect -info command shows how the peer interface was defined. This option does not apply to a loopback interconnect, which always uses 127.0.0.1.

If no destination interface is specified, the system will select an interface. This ensures that the interface selected can communicate with the source interface.

[(-max_time_out_of_sync <maxTimeOutOfSync> | -manual_refresh)]
Specifies the time, in 1—1440 minutes (up to 24 hours), that the source and destination can be out of synchronization before an update occurs. If you do not specify a max_time_out_of_sync value, use the -manual_refresh option to indicate that the destination will be updated on demand using the nas_replicate -refresh command. If no option is selected, the refresh default time for a file system replication is 10 minutes.

[(-overwrite_destination)]
For an existing destination object, discards any changes made to the destination object and restores it from the established common base, thereby starting the replication session from a differential copy. If this option is not specified and the destination object contains different content than the established common base, an error is returned.

[(-tape_copy)]
For file system replication only, creates and stops the replication session to enable an initial copy using the physical tape backup and process instead of an initial copy over the network. Using VNX Replicator describes the procedures for performing a tape copy, which involves a manually issued
The nas Commands

backup to tape from the source file system, a restore from tape to the destination file system, and a start of the replication session.

`[-background]`
Executes the command in asynchronous mode. Use the `nas_task` command to check the status of the command.

**CREATING VDM REPLICATION**

**-create <name>**
Assigns a name to the VDM replication session. The name must be unique for each Data Mover pair, which is defined by the interconnect.

```bash
-source -vdm <vdmName> | [id=<VdmId>]
```
Specifies the name or ID of an existing VDM to replicate. This replicates the CIFS working environment information contained in the root file system of the VDM. The source VDM must be in a loaded read/write or mounted read-only state. The source VDM can be the source or destination VDM of another replication session.

**Note:** Any file system mounted to a VDM must be replicated using file system replication. VDM replication affects the VDM only.

```bash
-destination -vdm {<existing_dstVdmName> | id=<dstVdmId> | -pool {id=<dstStoragePoolId> | <dstStoragePool>}
```
Specifies either an existing destination VDM or the storage needed to create the destination VDM. An existing destination VDM must be mounted as read-only, the same size as the source, and not loaded. The destination VDM can be the source of another replication but cannot be the destination of another replication. Specifying a storage pool creates the destination VDM automatically, as read-only, using the same name and size as the source VDM.

```bash
-interconnect {<name> | id=<interConnectId>}
```
Specifies the local (source) side of an established Data Mover interconnect to use for this replication session.

Use the `nas_cel -interconnect -list` command on the source VNX to list the interconnects available to replication sessions. The `nas_cel -interconnect -create` command is executed twice, one from each side, to create an interconnect between a pair of Data
Movers (two local Data Movers for local replication, or one local and one remote, for remote replication). Loopback interconnects are created for each Data Mover and are named automatically.

```
[-source_interface {<nameServiceInterfaceName> | ip=<ipAddr>}]}
```

Instructs the replication session to use a specific local interface defined for the interconnect on the source VNX instead of selecting the local interface supporting the lowest number of sessions (the default). If this local interface was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address. If you define an interface using an IP address, make sure that the destination interface uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

The `source_interfaces` field of the output from the `nas_cel -interconnect -info` command shows how the source interface was defined. This option does not apply to a loopback interconnect, which always uses 127.0.0.1.

If no source interface is specified, the system will select an interface. This ensures that the interface selected can communicate with the destination interface.

```
[-destination_interface {<nameServiceInterfaceName> | ip=<ipaddr>}]}
```

Instructs the replication session to use a specific peer interface defined for the interconnect on the destination VNX instead of selecting the peer interface supporting the lowest number of sessions (the default). If this peer interface was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address. If you define an interface using an IP address, make sure that the source interface uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

The `destination_interfaces` field of the output from the `nas_cel -interconnect -info` command shows how the peer interface was defined. This option does not apply to a loopback interconnect, which always uses 127.0.0.1.
The nas Commands

If no destination interface is specified, the system will select an interface. This ensures that the interface selected can communicate with the source interface.

```bash
[[
  -max_time_out_of_sync <maxTimeOutOfSync>
  | -manual_refresh
]]
```

Specifies the time, in 1–1440 minutes (up to 24 hours), that the source and destination can be out of synchronization before an update occurs. If you do not specify a `max_time_out_of_sync` value, use the `-manual_refresh` option to indicate that the destination will be updated on demand using the `nas_replicate -refresh` command. If no option is selected, the refresh default time for a VDM replication is 5 minutes.

```bash
[-overwrite_destination]
```

For an existing destination object, discards any changes made to the destination object and restores it from the established common base, thereby starting the replication session from a differential copy. If this option is not specified, and the destination object contains different content than the established common base, an error is returned.

```bash
[-background]
```

Executes the command in asynchronous mode. Use the `nas_task` command to check the status of the command.

**CREATING iSCSI LUN REPICATION**

```bash
-create <name>
```

Assigns a name to the V2 iSCSI LUN replication session. The name must be unique for each Data Mover pair, which is defined by the interconnect.

```bash
-source -lun <lunNumber> -target <targetIqn>
```

Specifies the LUN number (up to 256) of an existing iSCSI LUN to replicate and the fully qualified name (IQN) of the iSCSI target with the source LUN.

```bash
-destination -lun {<lunNumber> -target <targetIqn>}
```

 Specifies an available iSCSI LUN and iSCSI target at the destination. The destination iSCSI LUN cannot be created automatically, unlike a destination file system or VDM. The destination iSCSI LUN must be the same size as the source, and it must be read-only and marked protected. For `-lun`, specify a LUN number up to 256. For `-target`, specify a fully qualified name (IQN) of the iSCSI target with the destination LUN.
The nas Commands

**-interconnect** {<name>|id=<interConnectId>}

Specifies the local (source) side of an established Data Mover interconnect to use for this replication session.

Use the `nas_cel -interconnect -list` command on the source VNX to list the interconnects available to replication sessions. The `nas_cel -interconnect -create` command is executed twice, one from each side, to create an interconnect between a pair of Data Movers (two local Data Movers for local replication, or one local and one remote for remote replication). Loopback interconnects are created for each Data Mover and are named automatically.

[-source_interface {<nameServiceInterfaceName>|ip=<ipAddr>}]  

Instructs the replication session to use a specific local interface defined for the interconnect on the source VNX a instead of selecting the local interface supporting the lowest number of sessions (the default). If this local interface was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address. If you define an interface using an IP address, make sure that the destination interface uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

The `source_interfaces` field of the output from the `nas_cel -interconnect -info` command shows how the source interface was defined. This option does not apply to a loopback interconnect, which always uses 127.0.0.1.

If no source interface is specified, the system will select an interface. This ensures that the interface selected can communicate with the destination interface.

[-destination_interface {<nameServiceInterfaceName>|ip=<ipAddr>}]  

Instructs the replication session to use a specific peer interface defined for the interconnect on the destination VNX instead of selecting the peer interface supporting the lowest number of sessions (the default). If this peer interface was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address. If you define an interface using an IP address, make sure that the source interface uses the same IPv4/IPv6 protocol.
IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

The destination Interfaces field of the output from the nas_cel -interconnect -info command shows how the peer interface was defined. This option does not apply to a loopback interconnect, which always uses 127.0.0.1.

If no destination interface is specified, the system will select an interface. This ensures that the interface selected can communicate with the source interface.

[-max_time_out_of_sync <maxTimeOutOfSync> | -manual_refresh]
Specifies the time, in 1–1440 minutes (up to 24 hours), that the source and destination can be out of synchronization before an update occurs. If you do not specify a max_time_out_of_sync value, use the -manual_refresh option to indicate that the destination will be updated on demand using the nas_replicate -refresh command. If no option is selected, the refresh default time for an iSCSI LUN replication is 10 minutes.

[-overwrite_destination]
For an existing destination object, discards any changes made to the destination object and restores it from the established common base, thereby starting the replication session from a differential copy. If this option is not specified, and the destination object contains different content than the established common base, an error is returned.

[-background]
Executes the command in asynchronous mode. Use the nas_task command to check the status of the command.

START OPTIONS

-start {<name>|id=<sessionId>}
From the source side only, specifies the name or session ID needed to start the replication session. A replication name is unique for each Data Mover pair; if a duplicate name is detected on the system, the session ID is required. To get the session ID, use nas_replicate -list.

[-interconnect {<name>|id=<interConnectId>}]}
Specifies an established source-side (local) Data Mover interconnect to use for the replication session. Use the nas_cel -interconnect -list command to list the interconnects available to replication sessions. The nas_cel -interconnect -create command
creates an interconnect between a pair of Data Movers (two local Data Movers for local replication, or one local and one remote, for remote replication). Loopback interconnects are created and named automatically, and always use IP address 127.0.0.1.

```
[-source_interface {<nameServiceInterfaceName> | ip=<ipaddr>}]  
As the source interface for the replication session, uses a specific local interface defined for the interconnect instead of any local interface defined for the interconnect (the default, which enables the software to select the interface supporting the lowest number of sessions). If this interface was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address (IPv4 or IPv6). If you define an interface using an IP address, make sure that the destination interface uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

[-destination_interface {<nameServiceInterfaceName> | ip=<ipaddr>}]  
As the destination interface for the replication session, uses a specific peer interface defined for the interconnect instead of any peer interface defined for the interconnect (the default, which enables the software to select the interface supporting the lowest number of sessions). If this interface was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address (IPv4 or IPv6). If you define an interface using an IP address, make sure that the source interface uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

[{-max_time_out_of_sync <maxtimeOutOfSync> | -manual_refresh}]  
Specifies the time, in 1–1440 minutes (up to 24 hours), that the source and destination can be out of synchronization before an update occurs. If you do not specify a max_time_out_of_sync value, use the -manual_refresh option to indicate that the destination will be updated on demand using the nas_replicate -refresh command. If no option is selected, the refresh default time for both file system replication and iSCSI LUN is 10 minutes, and 5 minutes for VDM replication sessions.
The nas Commands

[-overwrite_destination]
For an existing destination object, discards any changes made to the destination object and restores the destination object from the established, internal common base checkpoint, thereby starting the replication session from a differential copy. If this option is not specified and the destination object has different content than the established common base, an error is returned.

[-reverse]
Reverses the direction of the replication session when invoked from the new source side (the original destination). A reverse operation continues to use the established replication name or replication session ID. Use this option to restart replication after a failover or switchover.

[-full_copy]
For an existing destination object that contains content changes, performs a full copy of the source object to the destination object. If replication cannot be started from a differential copy using the -overwrite_destination option, omitting this option causes the command to return an error.

[-background]
Executes the command in asynchronous mode. Use the nas_task command to check the status of the command.

MODIFY OPTIONS

-modify {<name>|id=<sessionId>}
From the source side only, specifies the name or session ID of the replication session to be modified. If a duplicate name is detected on the system, the session ID (fixed for the life of the session) is required. To get the session ID, use nas_replicate -list.

Note: A session cannot be modified if a -stop, -delete, -reverse, -failover, -switchover, -create, or -start operation is running. However, once a -modify operation is running, any other operation is permitted.

[-name <newName>]
Renames the replication session to the new name specified. When renaming a session, note that the name must be unique for each Data Mover pair.

[-source_interface {<nameServiceInterfaceName>|ip=<ipaddr>}] 
Changes the source interface used for the session to another local interface from the list defined for the interconnect. If this interface
was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address (IPv4 or IPv6). If you change an IP address, make sure that the destination interface uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

```
[-destination_interface{nameServiceInterfaceName|ip=ipaddr}]
```

Changes the destination interface used for the session to another peer interface from the list defined for the interconnect. If this interface was defined for the interconnect using a name service interface name, specify the name service interface name; if it was defined using an IP address, specify the IP address (IPv4 or IPv6). If you change an IP address, make sure that the source interface uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

```
[-max_time_out_of_sync <maxTimeOutOfSync>]
[-manual_refresh]
```

Specifies the time, from 1—1440 minutes (up to 24 hours), that the source and destination can be out of synchronization before an update occurs. If you do not specify a `max_time_out_of_sync` value, use the `manual_refresh` option to indicate that the destination will be updated on demand using the `nas_replicate -refresh` command. If no option is selected, the refresh default time for both file system replication and iSCSI LUN is 10 minutes, and 5 minutes for VDM replication sessions.

**STOP OPTIONS**

```
-stop {<name>|id=<session_id>}
```

Executed from the Control Station on the source VNX, stops the specified replication session but retains the session’s configuration information. Any data transfer in progress is terminated immediately and the destination object is restored to a consistent state.

---

**Note:** A session cannot be stopped if the `-delete option` is already running for the session. Once a stop operation is in progress, only the options `-list`, `-info`, and the `nas_task` command are permitted.

```
[-mode {source|destination|both}]
```

When stopping a session handling a local or remote replication from the source side, the `-mode both` option immediately stops
both sides of the replication session. The **-mode source** option stops only the replication session on the source and ignores the other side of the replication relationship. If the destination side is not operational, the **-mode source** option is required to stop the session. From the destination side, only the **-mode destination** option can be issued. When stopping a session handling a loopback replication, you can specify any **-mode** option to stop the session.

[ **-background**]
Executes the command in asynchronous mode. Use the `nas_task` command to check progress.

**DELETE OPTIONS**

- **delete** {<name> | id=<session_id>}
Executed from the Control Station on the source VNX, cancels replication data transfer if it is in progress, and deletes the replication session specified by the **-mode options**.

[ **-mode** {source | destination | both}]
When deleting a local or remote replication session from the source side, the **-mode both** option deletes both sides of the replication session. The **-mode source** option immediately aborts only the replication session on the source and ignores the other side of the replication relationship. If the destination side is not operational, the **-mode source** option is required to delete the session. From the destination side, only the **-mode destination** option can be issued. When deleting a loopback replication, you can specify any **-mode** option to stop the session.

[ **-background**]
Executes the command in asynchronous mode. Use the `nas_task` command to check progress.

The execution of the **delete** option is asynchronous and can be delayed if there is a network problem. During the delete process, other operations on the replication session are not allowed.

**FAILOVER OPTIONS**

- **failover** {<name> | id=<session_id>}
In response to a potential disaster scenario, performs a failover of the specified replication session with possible data loss. Execute this command from the Control Station on the destination VNX only. This command cancels any data transfer that is in process and marks the destination object as read-write so that it can serve as the new source object. When the original source Data Mover becomes reachable, the source object is changed to read-only (and for an iSCSI LUN, protected mode).
The nas Commands

CAUTION

The execution of the failover operation is asynchronous and results in data loss if all the data was not transferred to the destination site prior to issuing the failover.

For an iSCSI LUN, the administrator needs to make sure the LUN masks are set correctly on the destination target, so that after failover, hosts can connect to the destination LUN.

If there are multiple sessions using the same source object, only one replication session can be failed over. After the selected session is failed over, the other sessions become inactive until the session is restarted or failed back.

[-background]
Executes the command in asynchronous mode. Use the nas_task command to check progress.

SWITCHOVER OPTIONS

-switchover {<name>|id=<session_id>}
For test or migration purposes, switches over the specified replication relationship and performs synchronization of the source and destination without data loss. Execute this command from the Control Station on the source VNX only. This command stops replication, mounts the source object as read-only, and marks the destination object as read-write so that it can act as the new source object.

Unlike a reverse operation, a switchover operation does not restart replication.

[-background]
Executes the command in asynchronous mode. Use the nas_task command to check progress.

REVERSE OPTIONS

-reverse {<name>|id=<session_id>}
If executed from the source side of a replication session, reverses the direction of the specified replication session without data loss. A reverse synchronizes the destination with the source, mounts the source object as read-only, stops replication, marks the destination object as read-write so that it can act as the new source object, then restarts replication in the reverse direction from a differential copy (using the same configuration parameters established originally for the session).
The nas Commands

[**-background**]
Executes the command in asynchronous mode. Use the **nas_task** command to check progress.

**REFRESH OPTIONS**

**-refresh**\(<name>|id=<session_id>\)
Updates the destination side of the specified replication session based on changes to the source side. Execute this command from the Control Station on the source side only. A refresh operation handles updates on demand; as an alternative, the **-max_time_out_of_sync** option performs an update automatically after a specified number of minutes.

If the data changes on the source are large, this command can take a long time to complete. Consider running this command in background mode.

\[**-source**\(<ckptName>|id=<ckptId>\) **-destination**\(<ckptName>|id=<ckptId>\)\]
Instructs the replication **-refresh** option to use a specific checkpoint on the source side and a specific checkpoint on the destination side.

Specifying source and destination checkpoints for the **-refresh** option is optional. However, if you specify a source checkpoint, you must also specify a destination checkpoint. Replication transfers the contents of the user-specified source checkpoint to the destination file system. This transfer can be either a full copy or a differential copy depending on the existing replication semantics. After the transfer, the replication internally refreshes the user specified destination checkpoint and marks the two checkpoints as common bases.

After the replication refresh operation completes successfully, both the source and destination checkpoints have the same view of their file systems. The replication continues to use these checkpoints as common bases until the next transfer is completed. After a user checkpoint is marked with a common base property, the property is retained until the checkpoint is refreshed or deleted. A checkpoint that is already paired as a common base with another checkpoint propagates its common base property when it is specified as the source in a replication refresh operation. This propagation makes it possible for file systems without a direct replication relationship to have common base checkpoints.
The nas Commands

[-background]

Executes the command in asynchronous mode. Use the nas_task command to check progress.

STORAGE SYSTEM OUTPUT

The number associated with the storage device is dependent on the attached storage system. VNX for block displays a prefix of APM before a set of integers, for example, APM00033900124-0019. For example, Symmetrix storage systems appear as 002804000190-003C. The outputs displayed in the examples use a VNX for block.

EXAMPLE #1

To list all the VNX Replicator sessions, type:

```
$ nas_replicate -list
```

```
Name           Type       Local Mover Interconnect     Celerra   Status
ufs1_rep1      filesystem server_3 -->NYs3_LAs2        cs110     OK
vdm1_rep1      vdm        server_3 -->NYs3_LAs2        cs110     OK
iscsi_rep1     iscsiLun   server_3 -->NYs3_LAs2        cs110     OK
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Either then name of the session or the globally unique session ID for the session, if there are duplicate names on the system.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of V2 replication session (ongoing file system (fs), copy, LUN, or VDM).</td>
</tr>
<tr>
<td>Source Mover</td>
<td>The source Data Mover for the session.</td>
</tr>
<tr>
<td>Interconnect</td>
<td>The name of the source-side interconnect used for the session.</td>
</tr>
<tr>
<td>Celerra</td>
<td>The name of the VNX for file.</td>
</tr>
<tr>
<td>Status</td>
<td>The status of the session (OK, Active, Idle, Stopped, Error, Waiting) Info, Critical.</td>
</tr>
</tbody>
</table>

EXAMPLE #2

To create a file system replication session ufs1_rep1 on the source file system ufs1 and destination pool clar_r5_performance on the interconnect NYs3_LAs2 using the specified source and destination IP addresses to be updated automatically every 5 minutes, type:

```
$ nas_replicate -create ufs1_rep1 -source -fs ufs1 -destination -pool clar_r5_performance -interconnect NYs3_LAs2 -source_interface ip=10.6.3.190 -destination_interface ip=10.6.3.173 -max_time_out_of_sync 5
```

OK

EXAMPLE #3

To display information for a replication session ufs1_rep1, type:

```
$ nas_replicate -info ufs1_rep1
```

```
ID                             = 184_APM00064600086_0000_173_APM00072901601_0000
Name                           = ufs1_rep1
Source Status                  = OK
Network Status                 = OK
```

Command Line Interface Reference for File 7.0
The nas Commands

Destination Status = OK
Last Sync Time = Thu Dec 13 14:47:16 EST 2007
Type = filesystem
Celerra Network Server = cs110
Dart Interconnect = NYs3_LAs2
Peer Dart Interconnect = 20004
Replication Role = source
Source Filesystem = ufs1
Source Data Mover = server_3
Source Interface = 10.6.3.190
Source Control Port = 0
Source Current Data Port = 0
Destination Filesystem = ufs1_replica3
Destination Data Mover = server_2
Destination Interface = 10.6.3.173
Destination Control Port = 5081
Destination Data Port = 8888
Max Out of Sync Time (minutes) = 5
Next Transfer Size (Kb) = 0
Latest Snap on Source =
Latest Snap on Destination =
Current Transfer Size (KB) = 0
Current Transfer Remain (KB) = 0
Estimated Completion Time =
Current Transfer is Full Copy = No
Current Transfer Rate (KB/s) = 76
Current Read Rate (KB/s) = 11538
Current Write Rate (KB/s) = 580
Previous Transfer Rate (KB/s) = 0
Previous Read Rate (KB/s) = 0
Previous Write Rate (KB/s) = 0
Average Transfer Rate (KB/s) = 6277
Average Read Rate (KB/s) = 0
Average Write Rate (KB/s) = 0

EXAMPLE #4
To create a VDM replication session vdm_rep1 on source VDM vdm1 and destination pool clar_r5_performance on the interconnect NYs3_LAs2 with the given source and destination IP addresses to be updated automatically every 5 minutes, type:

```
$ nas_replicate -create vdm1_rep1 -source -vdm vdm1 -destination -pool clar_r5_performance -interconnect NYs3_LAs2 -source_interface ip=10.6.3.190 -destination_interface ip=10.6.3.173 -max_time_out_of_sync 5
```

OK

EXAMPLE #5
To create an iSCSI replication session iscsi_rep1 with the source LUN number 0, and using specific interfaces to use for the interconnect, type:
$ nas_replicate -create iscsi_rep1 -source -lun 0 -target
    iqn.1992-05.com.emc:apm000646000860000-4 -destination
    -interconnect NYs3_LAs2 -source_interface ip=10.6.3.190
    -destination_interface ip=10.6.3.173
    -max_time_out_of_sync 60 -overwrite_destination

OK

EXAMPLE #6  To list existing replication sessions, type:

$ nas_replicate -list

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Local Mover</th>
<th>Interconnect</th>
<th>Celerra</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ufs1_repl</td>
<td>filesystem</td>
<td>server_3 --&gt;NYs3_LAs2</td>
<td>cs110</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>vdm1_repl</td>
<td>vdm</td>
<td>server_3 --&gt;NYs3_LAs2</td>
<td>cs110</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>iscsi_repl</td>
<td>iscsiLun</td>
<td>server_3 --&gt;NYs3_LAs2</td>
<td>cs110</td>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>

EXAMPLE #7  To manually synchronize source and destination for the replication session iscsi_rep1, type:

$ nas_replicate -refresh iscsi_rep1

OK

EXAMPLE #8  To manually synchronize source and destination for the replication session iscsi_rep1 by using user checkpoints on the source and the destination, type:

$ nas_replicate -refresh iscsi_rep1 -source id=101
    -destination id=102

OK

EXAMPLE #9  To stop replication on both source and destination for the replication session iscsi_rep1, type:

$ nas_replicate -stop iscsi_rep1 -mode both

OK

EXAMPLE #10  To start stopped replication session iscsi_rep1 on interconnect NYs3_LAs2, specify manual refresh and to overwrite the destination LUN performing a full copy, type:

$ nas_replicate -start iscsi_rep1 -interconnect NYs3_LAs2
    -manual_refresh -overwrite_destination -full_copy

OK

EXAMPLE #11  To display information for the iSCSI replication session iscsi_rep1, type:

$ nas_replicate -info iscsi_rep1

ID = fs188_T4_LUN0_APM00064600086_0000_fs188_T2_LUN0_APM00072901601_0000
Name = iscsi_repl
Source Status = OK
EXAMPLE #12  To display information for the VDM replication session `vdm_rep1`, type:

```bash
$ nas_replicate -info vdm1_rep1
```

```
ID                             = 278_APM00064600086_0000_180_APM00072901601_0000
Name                           = vdm1_rep1
Source Status                  = OK
Network Status                 = OK
Destination Status             = OK
Last Sync Time                 = Fri Dec 14 16:49:54 EST 2007
Type                           = vdm
Celerra Network Server         = cs110
Dart Interconnect              = NYs3_LAs2
Peer Dart Interconnect         = 20004
Replication Role               = source
Source Target                  = iqn.1992-05.com.emc:apm000646000860000-4
Source LUN                     = 0
Source Data Mover              = server_3
Source Interface               = 10.6.3.190
Source Control Port            = 0
Source Current Data Port       = 0
Destination LUN                = 0
Destination Data Mover         = server_2
Destination Interface          = 10.6.3.173
Destination Control Port       = 5081
Destination Data Port          = 8888
Max Out of Sync Time (minutes) = Manual Refresh
Application Data               =
Next Transfer Size (Kb)        = 0
Latest Snap on Source          =
Latest Snap on Destination     =
Current Transfer Size (KB)     = 0
Current Transfer Remain (KB)   = 0
Estimated Completion Time      =
Current Transfer is Full Copy  = No
Current Transfer Rate (KB/s)   = 78
Current Read Rate (KB/s)       = 12578
Current Write Rate (KB/s)      = 590
Previous Transfer Rate (KB/s)  = 0
Previous Read Rate (KB/s)      = 0
Previous Write Rate (KB/s)     = 0
Average Transfer Rate (KB/s)   = 6497
Average Read Rate (KB/s)       = 0
Average Write Rate (KB/s)      = 0
```
Peer Dart Interconnect = 20004
Replication Role = source
Source VDM = vdm1
Source Data Mover = server_3
Source Interface = 10.6.3.190
Source Control Port = 0
Source Current Data Port = 0
Destination VDM = vdm1
Destination Data Mover = server_2
Destination Interface = 10.6.3.173
Destination Control Port = 5081
Destination Data Port = 8888
Max Out of Sync Time (minutes) = 5
Next Transfer Size (Kb) = 0
Latest Snap on Source =
Latest Snap on Destination =
Current Transfer Size (KB) = 0
Current Transfer Remain (KB) = 0
Estimated Completion Time =
Current Transfer is Full Copy = No
Current Transfer Rate (KB/s) = 313
Current Read Rate (KB/s) = 19297
Current Write Rate (KB/s) = 469
Previous Transfer Rate (KB/s) = 0
Previous Read Rate (KB/s) = 0
Previous Write Rate (KB/s) = 0
Average Transfer Rate (KB/s) = 155
Average Read Rate (KB/s) = 0
Average Write Rate (KB/s) = 0

**EXAMPLE #13**  To change the session name *iscsi_rep1* to *iscsi_rep2*, and to change max time out of sync value to 90, type:

```
$ nas_replicate -modify iscsi_rep1 -name iscsi_rep2
-max_time_out_of_sync 90
```

**EXAMPLE #14**  To failover the replication session *ufs1_rep1*, type on destination:

```
$ nas_replicate -failover ufs1_rep1
```

OK
EXAMPLE #15  To start failed over replication in the reverse direction, type:

$ nas_replicate -start ufs1_rep1 -interconnect LAs2_NYs3
-reverse -overwrite_destination

OK

EXAMPLE #16  To reverse direction of the replication session ufs1_rep1, type:

$ nas_replicate -reverse ufs1_rep1

OK

EXAMPLE #17  To switch over the replication session ufs1_rep1 using the background option, type:

$ nas_replicate -switchover ufs1_rep1 -background

Info 26843676673: In Progress: Operation is still running. Check task id 4058 on the Task Status screen for results.

*** Comment: Use nas_task -info command to find out the status of background task.

EXAMPLE #18  To delete the replication session fs1_rep1 on both source and destination, type:

$ nas_replicate -delete fs1_rep1 -mode both

OK
EXAMPLE #14  To start failed over replication in the reverse direction, type:

```
$ nas_replicate -start ufs1_rep1 -interconnect LA2_NYs3
-reverse -overwrite_destination
```

OK

EXAMPLE #15  To reverse direction of the replication session ufs1_rep1, type:

```
$ nas_replicate -reverse ufs1_rep1
```

OK

EXAMPLE #16  To switch over the replication session ufs1_rep1 using the background option, type:

```
$ nas_replicate -switchover ufs1_rep1 -background
```

Info 26843676673: In Progress: Operation is still running. Check task id 4058 on the Task Status screen for results.

*** Comment: Use nas_task -info command to find out the status of background task.

EXAMPLE #17  To delete the replication session fs1_rep1 on both source and destination, type:

```
$ nas_replicate -delete fs1_rep1 -mode both
```

OK
The nas Commands

**nas_rp**

Manages VNX systems that use RecoverPoint and defines the relationships needed for cabinet level disaster recovery.

**SYNOPSIS**

```
nas_rp
 -rpa
  -add <rpaname> -ip <ip> -admin <rpaadminname> [-password <password>]
  -update {<rpaname>|id=<id>}
  -info {<rpaname>|id=<id>} [-version] [-verbose]
  -list

-cg
  -list
  -info {<cgname>|id=<id>}
  -suspend {<cgname>|id=<id>}
  -resume {<cgname>|id=<id>}
  -modify {<cgname>|id=<id>} -rpo <time_in_seconds>

-cabinetdr
  -info
  -init <cel_name>
  -failover
  -failback
```

**DESCRIPTION**

`nas_rp` manages VNX systems and cabinet level disaster recovery using RecoverPoint.

**OPTIONS**

- **-rpa** manages the configurations of RecoverPoint Appliances or RPA.
  - **-add <rpaname> -ip <ip> -admin <rpaadminname> [-password <password>]**
    Adds an RPA entry to the NAS database with the RPA’s IP address and administrator login credentials. The `-add` option can be run in the non-interactive mode with the `-password` option specified.
  - **-update {<rpaname>|id=<id>}**
    Repairs all SSH RSA key issues to make the CS-RPA communication go through based on SSH Key authentication.
  - **-info {<rpaname>|id=<id>} [-version] [-verbose]**
    Displays the properties of a specified RPA and the license settings.
  - **-list**
    Lists all the RPAs configured in the system.

- **-cg** manages the NAS Consistency Groups or CG.
The nas Commands

- **list**
  Lists all the NAS CGs discovered on the RPA.

- **info** {<cgname>|id=<id>}
  Displays the properties of a specified NAS CG, including the devices or LUNS that are being replicated.

- **suspend** {<cgname>|id=<id>}
  Suspends the data transfer for a specified NAS CG.

- **resume** {<cgname>|id=<id>}
  Resumes the suspended data transfer for a specified NAS CG.

- **modify** {<cgname> | id=<id>} -rpo <time_in_seconds>
  Modifies the RPO settings on a specified NAS CG to a user specified value. The RPO value must be greater than or equal to 1 minute.

- **cabinetdr** manages the activities needed for cabinet level disaster recovery.

  - **info**
    Displays the CG to be failed over, the state and transfer information of the CG, servers that have been configured as standby, and the state of these servers.

  - **init** <cel_name>
    Initializes a source or destination (target) Culham system.

- **failover**
  Initiates a RecoverPoint cabinet failover from the source VNX to the destination VNX. The -failover option is executed on the destination Control Station at the discretion of the user. The -failover option fails over the RecoverPoint NAS CGs. It sets each RecoverPoint-protected volume on the source VNX as read-only and each mirrored volume on the destination VNX as read-write.

- **failback**
  Restores a source VNX after a failover. The -failback option is initially executed on the destination Control Station. The data on each destination volume is copied to the corresponding volume on the source VNX. On the destination VNX, services on each standby Data Mover are stopped. (NFS clients connected to these Data Movers see a “server unavailable” message; CIFS client connections time out.) Each volume on the source VNX is set as read-write, and each mirrored volume on the destination VNX is set as read-only.
EXAMPLE #1 To add an RPA to the NAS database, type:

```
$ /nas/sbin/nas_rp -rpa -add rpa1 -ip 10.245.64.16 -admin admin -password admin
```
done

EXAMPLE #2 To fix all the SSH key related problems for an RPA, type:

```
$ /nas/sbin/nas_rp -rpa -update rpa1
```
done

EXAMPLE #3 To get detailed information about an RPA, type:

```
$ /nas/sbin/nas_rp -rpa -info rpa1
```
Name = rpa1
ID = 1
IP = 172.24.173.9
Administrator = admin
Installation ID = f442ea06c265ac910b153f855d98a8b86fe1f0
License =
Rts6q8aDCHx1l175rYKxwQs/NqZtosXmSwXoZIojaIVaXw60t9TLuMXkTD7/rAnVnEUL2wfqF+5eBMw0RS0wAA
Activation Code =
13dfacBpSohMgUIzhWgRJpYhbBvK1L56RnfU6CELkDoWXNa/PJB7whLptoiXfGsaSdTgaqtHs57UFP7M7wECwAA
Version = 3.2.SP2.P2(h.26)

EXAMPLE #4 To list all the RPAs configured in the system, type:

```
$ nas_rp -rpa -list
```
Id name ipaddress
1 rpa1 172.24.173.9

EXAMPLE #5 To list all the NAS CGs discovered on the RPA, type:

```
$ nas_rp -cg -list
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>RPA ID</th>
<th>Prod Copy</th>
<th>Remote Copy</th>
<th>Control LUN</th>
<th>CG</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>NASCG_dev10_dev13</td>
<td>4</td>
<td>Src_dev10</td>
<td>DR_dev13</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

EXAMPLE #6 To display the properties of a specified NAS CG, including the devices or LUNS that are being replicated, type:

```
$ nas_rp -cg -info id=13
```
id = 13
name = NASCG_dev10_dev13
rpa = rpa1
source copy = Src_dev10
remote copy = DR_dev13
source clar id = APM00102102333
remote clar id = APM00102400657
contains control luns = True
transfer state = ACTIVE
replication direction = remote -> local
role = Secondary
transfer mode = Async
rpo = SYSTEM

Replication sets

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Src LUN</th>
<th>Dst LUN</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>277</td>
<td>RSet 1</td>
<td>4</td>
<td>25</td>
<td>2147483648</td>
</tr>
<tr>
<td>278</td>
<td>RSet 10</td>
<td>119</td>
<td>119</td>
<td>214748364800</td>
</tr>
<tr>
<td>279</td>
<td>RSet 11</td>
<td>120</td>
<td>120</td>
<td>214748364800</td>
</tr>
<tr>
<td>280</td>
<td>RSet 12</td>
<td>101</td>
<td>101</td>
<td>214748364800</td>
</tr>
<tr>
<td>281</td>
<td>RSet 13</td>
<td>102</td>
<td>102</td>
<td>214748364800</td>
</tr>
<tr>
<td>282</td>
<td>RSet 14</td>
<td>103</td>
<td>103</td>
<td>214748364800</td>
</tr>
<tr>
<td>283</td>
<td>RSet 15</td>
<td>104</td>
<td>104</td>
<td>214748364800</td>
</tr>
<tr>
<td>284</td>
<td>RSet 16</td>
<td>105</td>
<td>105</td>
<td>214748364800</td>
</tr>
<tr>
<td>285</td>
<td>RSet 17</td>
<td>106</td>
<td>106</td>
<td>214748364800</td>
</tr>
<tr>
<td>286</td>
<td>RSet 18</td>
<td>107</td>
<td>107</td>
<td>214748364800</td>
</tr>
<tr>
<td>287</td>
<td>RSet 19</td>
<td>108</td>
<td>108</td>
<td>214748364800</td>
</tr>
<tr>
<td>288</td>
<td>RSet 2</td>
<td>0</td>
<td>18</td>
<td>11811160064</td>
</tr>
<tr>
<td>289</td>
<td>RSet 20</td>
<td>109</td>
<td>109</td>
<td>214748364800</td>
</tr>
<tr>
<td>290</td>
<td>RSet 21</td>
<td>110</td>
<td>110</td>
<td>214748364800</td>
</tr>
<tr>
<td>291</td>
<td>RSet 22</td>
<td>111</td>
<td>111</td>
<td>214748364800</td>
</tr>
<tr>
<td>292</td>
<td>RSet 23</td>
<td>112</td>
<td>112</td>
<td>214748364800</td>
</tr>
<tr>
<td>293</td>
<td>RSet 3</td>
<td>1</td>
<td>19</td>
<td>11811160064</td>
</tr>
<tr>
<td>294</td>
<td>RSet 4</td>
<td>113</td>
<td>113</td>
<td>214748364800</td>
</tr>
<tr>
<td>295</td>
<td>RSet 5</td>
<td>114</td>
<td>114</td>
<td>214748364800</td>
</tr>
<tr>
<td>296</td>
<td>RSet 6</td>
<td>115</td>
<td>115</td>
<td>214748364800</td>
</tr>
<tr>
<td>297</td>
<td>RSet 7</td>
<td>116</td>
<td>116</td>
<td>214748364800</td>
</tr>
<tr>
<td>298</td>
<td>RSet 8</td>
<td>117</td>
<td>117</td>
<td>214748364800</td>
</tr>
<tr>
<td>299</td>
<td>RSet 9</td>
<td>118</td>
<td>118</td>
<td>214748364800</td>
</tr>
</tbody>
</table>

**EXAMPLE #7**  To suspend data transfer for a specified NAS CG, type:

```
$ nas_rp -cg -suspend id=13
```

done

**EXAMPLE #8**  To resume the suspended data transfer for a specified NAS CG, type:

```
$ nas_rp -cg -resume id=13
```

done

**EXAMPLE #9**  To modify the RPO settings on a specified NAS CG to a user specified value, type:

```
$ nas_rp -cg -modify id=13 -rpo 300
```

done

**EXAMPLE #10**  To display the CG to be failed over, the state and transfer information of the CG, servers that have been configured as standby, and the state of these servers, type:

```
# nas_rp -cabinetdr -info
```

***** Consistency Group Configuration *****
name                  = NASCG_dev10_dev13
description           =
uid                   = 50:6:1:60:B0:60:26:BC:0:0:0:0:0:0:0:0
state                 = Consistent
role                  = Primary
condition             = Active
recovery policy       = Automatic
number of mirrors     = 16
mode                  = SYNC
owner                 = 0
mirrored disks        = root_disk,root_ldisk,d5,d8,d10,d11,d24,d25,d26,d27,d29,d30,d31,d32,d33,d39,
local clarid          = APM00053001552
remote clarid         = APM00053001549
mirror direction      = local -> remote

***** Servers configured with RPstandby *****

id        = 1
name      = server_2
acl       = 2000, owner=rpadmin, ID=500
type      = nas
slot      = 2
member_of = standby   = server_3, policy=auto
RDFstandby= slot=2
status    :
  defined = enabled
  actual = online, active

id        = 2
name      = server_3
acl       = 1000, owner=nasadmin, ID=201
type      = standby
slot      = 3
member_of = standbyfor= server_2
RDFstandby= slot=3
status    :
  defined = enabled
  actual = online, ready

***** Servers configured as standby *****

id        = 1
name      = server_2
acl       = 2000, owner=rpadmin, ID=500
type      = standby
slot      = 2
The nas Commands

member_of =
standbyfor=
status :
    defined = enabled
    actual = online, ready

id     = 2
name   = server_3
acl    = 2000, owner=rpadmin, ID=500
type   = standby
slot   = 3
member_of =
standbyfor=
status :
    defined = enabled
    actual = online, ready

EXAMPLE #11   To initialize a source or destination (target) VNX, type:

# /nas/sbin/nas_rp -cabinetdr -init new_york
Culham with RecoverPoint Disaster Recovery
Initializing new_york --> new_jersey
Contacting new_york for remote storage info
Local storage system: FNM00093600019
Remote storage system: FNM00094700042
Discovering storage on new_york (may take several minutes)
Setting security information for FNM00093600019
Discovering storage at 172.24.173.26 (may take several minutes)
Contacting new_york for remote storage info
Contacting new_york for server capabilities...
Analyzing server information...
Source servers available to be configured for remote DR
-------------------------------------------------------
1. server_2:new_york
2. server_3:new_york [ local standby ]
v. Verify standby server configuration
q. Quit initialization process
c. Continue initialization
Select a new_york server: 1
Destination servers available to act as remote standby
------------------------------------------------------
1. server_2:new_jersey
 server_3:new_jersey [ local standby ]
b. Back
Select a new_jersey server: 1
Source servers available to be configured for remote DR
-------------------------------------------------------
1. server_2:new_york [ remote standby is server_2:new_jersey ]
2. server_3:new_york [ local standby ]
v. Verify standby server configuration
q. Quit initialization process
c. Continue initialization
Select a new_york server: c
Standby configuration validated OK
Enter user information for managing remote site new_york
Username: dradmin
Password: ********** Retype your response to validate
Password: **********
Setting up server_2 on new_york
Rebooting server_2 on new_jersey as standby ... done
Setting acl for server_2 on new_jersey
Updating the Culham domain information
done

EXAMPLE #12  To activate a failover, log in to destination Control Station by using
the dradmin account, su to root, and type:

```bash
# /nas/sbin/nas_rp -cabinetdr -failover
```
Sync with CLARiiON backend ....... done
Validating consistency group configuration ...... done
Is source site new_york ready for complete shut down (power OFF)? [yes or no] yes
Contacting source site source_cs, please wait... done
Shutting down remote site source_cs ................................. done
Sync with CLARiiON backend ...... done
STARTING an RP 'FAILOVER' operation.
Device group: cg_new_york ............ done
The RP 'FAILOVER' operation SUCCEEDED.
Failing over Devices ... done
Adding NBS access for server_2 ........ done
Adding NBS access for server_3 ........ done
Activating the target environment ... done

done

EXAMPLE #13  To restore, log in to the destination Control Station by using the
dradmin account, su to root, and type:

```bash
# /nas/sbin/nas_rp -cabinetdr -failback
```
Sync with CLARiiON backend ...... done
Validating consistency group configuration ...... done
Contacting source site new_york, please wait... done
Running restore requires shutting down source site new_york.
Do you wish to continue? [yes or no] yes
Shutting down remote site new_york ........ done
Is source site new_york ready for storage restoration? [yes or no] yes
Sync with CLARiiON backend ...... done
STARTING an RP 'RESUME' operation.
Consistency group: cg_new_york ............ done
The RP 'RESUME' operation SUCCEEDED.
Percent synchronized: 100
Updating consistency group .... done
Is source site ready for network restoration? [yes or no] yes
Restoring servers ...... done
Waiting for servers to reboot ...... done
Removing NBS access for server_2 .. done
Removing NBS access for server_3 .. done
Waiting for consistency group ready to failback .... done
Sync with CLARiiON backend ...... done
STARTING an RP 'FAILBACK' operation.
Consistency group: cg_new_york ............ done
The RP 'FAILBACK' operation SUCCEEDED.
Restoring remote site cg_new_york, please wait... done
done
**nas_server**

Manages the Data Mover (server) table.

**SYNOPSIS**

```bash
nas_server
   -list [-all | -vdm]
   -delete <movername>
   -info { -all | <movername> | <slot_number> | id=<mover_id>
      | -vdm { -all | <vdm_name> | id=<vdm_id> }}
   -rename <old_movername> <new_movername>
   -acl <acl_value> <movername>
   [-name <name>] [-type <type>] -create <movername> [-setstate <state>]
      [-fs <fs_name> | pool=<pool>] [-option <options>]
   -vdm <vdm_name> -setstate <state> [ <movername> ] [-ConvertI18N]
   -vdm <vdm_name> -move <movername> [-ConvertI18N]
```

**DESCRIPTION**

`nas_server` manages the server tables for both physical and virtual Data Movers (VDMs), creates a VDM, sets an access control value for a physical Data Mover or VDM, renames a Data Mover and displays attributes for a specified Data Mover or all Data Movers, deletes a physical Data Mover entry from the server table, and deletes the VDM configuration for a Data Mover.

**OPTIONS**

- **-list**

Lists the Data Mover server table. The server table displays the ID, type, access control level value, slot number, group ID, state, and name of a Data Mover. Virtual Data Movers (VDMs) have their own server table and do not have a numeric reference in the general server table.

---

**Note:** The ID of the object is an integer and is assigned automatically. The name of the Data Mover might be truncated if it is too long for the display. To display the full name, use the `-info` option with the Data Mover ID.

Valid Data Mover types are:

1=nas  
2=not used  
3=not used  
4=standby  
5=not used
6=rdf

Note: The nas type is set automatically, vdm is set using nas_rp, rdf and standby are set up using server_ssh.

[-all | -vdm]
The -all option displays the physical Data Mover and VDMs. The -vdm option lists the VDMs only.

-delete <movername>
Deletes the specified physical Data Mover entry from the server table or deletes the VDM configuration. A Data Mover that is being deleted cannot contain mounted file systems.

Deleting a physical Data Mover requires the root command. Use /nas/sbin/rootnas_server to execute a delete.

-info { -all | <movername> | <slot_number> | id = <mover_id> }
Displays attributes for all physical Data Movers, or a Data Mover specified by its <movername>, <slot_number>, or <mover_id>.

-info -vdm { -all | <vdm_name> | id = <vdm_id> }
Displays attributes for all VDMs, or a specified VDM, including the network interfaces that are being used by the CIFS servers.

-rename <old_movername> <new_movername>
Changes the name of the physical Data Mover or the VDM to the specified name. The -create option provides more information for rules applicable to naming a Data Mover.

-acl <acl_value> <movername>
Sets an access control level value that defines the owner of the physical Data Mover or the Virtual Data Mover, and the level of access allowed for users and groups defined in the access control level table. The nas_acl command provides more information.

[-name <name>] [-type vdm] -create <movername>
Creates a VDM with an optional name for the specified physical or VDM. The movername is case-sensitive and supports the following characters: a through z, A through Z, 0 through 9, _(underscore), -(hyphen) though names may not start with a hyphen. The default type is nas.

[-setstate <state>]
Sets the Data Mover to loaded or mounted.
The loaded option installs the image of the VDM onto the physical Data Mover, but does not mount the non-root file systems. The mounted option mounts the root_fs as read-only, but the VDM image is not installed. The -setstate option is for use with replication.

Note: Before a VDM image is loaded, the image must be unloaded from the previous Data Mover, and the CIFS server must be joined using server_cifs. The network interfaces used by the CIFS servers on the VDM must be available on the destination Data Mover.

[-fs <fs_name> | pool=<pool>]
Specifies a file system or assigns a rule set known as a storage pool for the VDM root file system.

For the -fs option, the file system must be unmounted, clean (nas_fsck provides more information), and be either of type uxf or rawfs. For a loaded state VDM, only an uxf file system type can be used, but for mounted state VDM, both uxf and rawfs can be used. The file system to be provided as the VDM root file system is renamed to root_fs_vdm_<vdm_name>. This is deleted when the VDM is deleted.

The storage pool option assigns a rule set for the root file system of the VDM that contains automatically created volumes and defines the type of disk volumes used and how they are aggregated. Storage pools are system defined (storage pool description provides more information) or user defined. nas_pool provides a description of pool types.

[-option <options>]
Specifies the following comma-separated list of options:

fstype=rawfs
Specifies the file system type as rawfs when creating a mounted VDM.

-vdm <vdm_name> -setstate <state>
Sets the state of the VDM to loaded, mounted, tempunloaded, or permunloaded.

The loaded option installs the image of the VDM onto the physical Data Mover, but does not mount the non-root file systems. The mounted option mounts the root_fs read-only, but the VDM image is not installed.
The `tempunloaded` option, temporarily unloads the VDM image, while the `permunloaded` option permanently unloads the image.

```bash
[movername]
```
Specifies a physical Data Mover for the VDM.

```bash
[-ConvertI18N]
```
When loading the VDM image, forces the conversion of the I18N mode of the VDM’s root file system from ASCII to UNICODE. The I18N mode of the Data Mover can be either ASCII or UNICODE. The mode of the VDM must be the same as the physical Data Mover, for example, when performing the `-move` option, or when replicating.

This mode is used when the mode of the VDM root file system is different from that of the physical Data Mover.

```bash
-vdm <vdm_name> -move <movername>
```
Moves the image of the VDM onto the physical Data Mover, and mounts the non-root file systems.

**Note:** Before a VDM image is loaded, the image must be unloaded from the previous Data Mover, and the CIFS server must be joined using `server_checkup`. The network interfaces used by the CIFS servers on the VDM must be available on the destination Data Mover.

```bash
[-ConvertI18N]
```
When loading the VDM image, forces the conversion of the I18N mode of the VDM’s root file system from ASCII to UNICODE. The I18N mode of the Data Mover can be either ASCII or UNICODE. The mode of the VDM must be the same as the physical Data Mover, for example, when performing the `-move` option, or when replicating.

This mode is used when the mode of the VDM root file system is different from that of the physical Data Mover.

**SEE ALSO** Configuring Virtual Data Mover on VNX, Using International Character Sets for File, `nas_fs`, `nas_volume`, and `server_checkup`. 
VNX storage systems support the following system-defined storage pools: clar_r1, clar_r5_performance, clar_r5_economy, clar_r6, clarata_r3, clarata_r6, clarata_r10, clarata_archive, cm_r1, cm_r5_performance, cm_r5_economy, cm_r6, cmata_r3, cmata_archive, cmata_r6, cmata_r10, clarsas_archive, clarsas_r6, clarsas_r10, clarefd_r5, clarefd_r10, cmsas_archive, cmsas_r6, cmsas_r10, and cmefd_r5.

Disk types when using VNX for block are CLSTD, CLEFD, and CLATA, and for VNX for block involving mirrored disks are: CMEFD, CMSTD, and CMATA.

VNX with a Symmetrix storage system supports the following system-defined storage pools: symm_std, symm_std_rdf_src, symm_ata, symm_ata_rdf_src, symm_ata_rdf_tgt, symm_std_rdf_tgt, symm_ata_rdf_tgt, symm_std_rdf_tgt, and symm_efd.

For user-defined storage pools, the difference in output is in the disk type. Disk types when using a Symmetrix are: STD, R1STD, R2STD, BCV, R1BCV, R2BCV, ATA, R1ATA, R2ATA, BCVA, R1BCA, R2BCA, and EFD.

**EXAMPLE #1**

To list the physical Data Mover table, type:

```
$ nas_server -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>type</th>
<th>acl</th>
<th>slot</th>
<th>groupID</th>
<th>state</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1000</td>
<td>2</td>
<td></td>
<td>0</td>
<td>server_2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1000</td>
<td>3</td>
<td></td>
<td>0</td>
<td>server_3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1000</td>
<td>4</td>
<td></td>
<td>0</td>
<td>server_4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1000</td>
<td>5</td>
<td></td>
<td>0</td>
<td>server_5</td>
</tr>
</tbody>
</table>

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the Data Mover.</td>
</tr>
<tr>
<td>type</td>
<td>Type assigned to Data Mover.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value assigned to the Data Mover or VDM.</td>
</tr>
<tr>
<td>slot</td>
<td>Physical slot in the cabinet where the Data Mover resides.</td>
</tr>
<tr>
<td>groupID</td>
<td>ID of the Data Mover group.</td>
</tr>
<tr>
<td>state</td>
<td>Whether the Data Mover is enabled=0, disabled=1, failed over=2.</td>
</tr>
<tr>
<td>name</td>
<td>Name given to the Data Mover.</td>
</tr>
</tbody>
</table>
**EXAMPLE #2**  To list the physical Data Mover and VDM table, type:

```bash
$ nas_server -list -all
```

<table>
<thead>
<tr>
<th>id</th>
<th>type</th>
<th>acl</th>
<th>slot</th>
<th>groupID</th>
<th>state</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1000</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>server_2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1000</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>server_3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1000</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>server_4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1000</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>server_5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>id</th>
<th>acl</th>
<th>server</th>
<th>mountedfs</th>
<th>rootfs</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td></td>
<td>31</td>
<td>vdm_1</td>
</tr>
</tbody>
</table>

**EXAMPLE #1** provides a description of outputs for the physical Data Movers. The following table provides a description of the command output for the VDM table.

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the Data Mover.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value assigned to the Data Mover or VDM.</td>
</tr>
<tr>
<td>server</td>
<td>Server the VDM is loaded on.</td>
</tr>
<tr>
<td>mountedfs</td>
<td>File systems that are mounted on this VDM.</td>
</tr>
<tr>
<td>rootfs</td>
<td>ID number of the root file system.</td>
</tr>
<tr>
<td>name</td>
<td>Name given to the Data Mover or VDM.</td>
</tr>
</tbody>
</table>

**EXAMPLE #3**  To list the VDM server table, type:

```bash
$ nas_server -list -vdm
```

<table>
<thead>
<tr>
<th>id</th>
<th>acl</th>
<th>server</th>
<th>mountedfs</th>
<th>rootfs</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td></td>
<td>31</td>
<td>vdm_1</td>
</tr>
</tbody>
</table>

**EXAMPLE #4**  To list information for a Data Mover, type:

```bash
$ nas_server -info server_2
```

id = 1  
name = server_2  
acl = 1000, owner=nasadmin, ID=201  
type = nas  
slot = 2  
member_of =  
standby = server_5, policy=auto  
status :  
defined = enabled  
actual = online, ready

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the Data Mover.</td>
</tr>
<tr>
<td>name</td>
<td>Name given to the Data Mover.</td>
</tr>
</tbody>
</table>
The nas Commands

### EXAMPLE #5
To display detailed information for all servers, type:

```bash
$ nas_server -info -all
```

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>acl</th>
<th>type</th>
<th>slot</th>
<th>member_of</th>
<th>standby</th>
<th>status</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>server_2</td>
<td>1000, owner=nasadmin, ID=201</td>
<td>nas</td>
<td>2</td>
<td></td>
<td>server_5, policy=auto</td>
<td>defined = enabled</td>
<td>actual = online, active</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>server_3</td>
<td>1000, owner=nasadmin, ID=201</td>
<td>nas</td>
<td>3</td>
<td></td>
<td>server_5, policy=auto</td>
<td>defined = enabled</td>
<td>actual = online, ready</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>server_4</td>
<td>1000, owner=nasadmin, ID=201</td>
<td>nas</td>
<td>4</td>
<td></td>
<td>server_5, policy=auto</td>
<td>defined = enabled</td>
<td>actual = online, ready</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>server_5</td>
<td>1000, owner=nasadmin, ID=201</td>
<td>standby</td>
<td>5</td>
<td></td>
<td>server_4, server_2, server_3</td>
<td>defined = enabled</td>
<td>actual = online, ready</td>
</tr>
</tbody>
</table>

**acl**
Access control level value assigned to the Data Mover or VDM.

**type**
Type assigned to Data Mover.

**slot**
Physical slot in the cabinet where the Data Mover resides.

**member_of**
Group to which the Data Mover is a member.

**standby**
If the Data Mover has a local standby associated with it.

**status**
Whether the Data Mover is enabled or disabled, and whether it is active.
The nas Commands

status :
    defined = enabled
    actual = online, ready

EXAMPLE #4 provides a description of command outputs.

EXAMPLE #6 To display information for all VDMs, type:

$ nas_server -info -vdm -all

id        = 3
name      = vdm_1
acl       = 0
type      = vdm
server    = server_2
rootfs    = root_fs_vdm_1
I18N mode = UNICODE
mountedfs =
member_of =
status :
    defined = enabled
    actual = mounted

Interfaces to services mapping:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the Data Mover.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the Data Mover.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value assigned to the VDM.</td>
</tr>
<tr>
<td>type</td>
<td>For VDM server, the type is always VDM.</td>
</tr>
<tr>
<td>server</td>
<td>Server on which the VDM loaded.</td>
</tr>
<tr>
<td>rootfs</td>
<td>Root file system of the VDM.</td>
</tr>
<tr>
<td>I18N mode</td>
<td>L18N mode of the VDM. I18N mode is either ASCII or UNICODE.</td>
</tr>
<tr>
<td>mountedfs</td>
<td>File systems that are mounted on this VDM.</td>
</tr>
<tr>
<td>member_of</td>
<td>If it is a member of a cluster, then this field will show the cluster name.</td>
</tr>
<tr>
<td>status</td>
<td>Whether the VDM is enabled or disabled, and whether it can be loaded ready, loaded active, mounted, temporarily unloaded and permanently unloaded.</td>
</tr>
</tbody>
</table>

Interaces to services mapping | List of interfaces that are used for the services configured on this VDM. Currently, only CIFS service is provided, so this field lists all the interfaces used in the CIFS servers configured on this VDM. |
EXAMPLE #7 To create a mounted VDM named vdm_1 on server_2 using the storage pool, clar_r5_performance with a rawfs, type:

```bash
$ nas_server -name vdm_1 -type vdm -create server_2 -setstate mounted pool=clar_r5_performance -option fstype=uxfs
```

id = 3
name = vdm_1
acl = 0
type = vdm
server = server_2
rootfs = root_fs_vdm_1
I18N mode = UNICODE
mountedfs =
member_of =
status :
  defined = enabled
  actual = mounted
Interfaces to services mapping:

EXAMPLE #6 provides a description of command outputs.

EXAMPLE #8 To set the state of a vdm_1 to mounted, type:

```bash
$ nas_server -vdm vdm_1 -setstate mounted
```

id = 3
name = vdm_1
acl = 0
type = vdm
server = server_2
rootfs = root_fs_vdm_1
I18N mode = UNICODE
mountedfs =
member_of =
status :
  defined = enabled
  actual = mounted
Interfaces to services mapping:

EXAMPLE #6 provides a description of command outputs.

EXAMPLE #9 To move the image of vdm_1 onto server_4, type:

```bash
$ nas_server -vdm vdm_1 -move server_4
```

id = 3
name = vdm_1
acl = 0
type = vdm
server = server_4
rootfs = root_fs_vdm_1
I18N mode = UNICODE
mountedfs =
The nas Commands

member_of =
status :
   defined = enabled
   actual = loaded, ready
Interfaces to services mapping:

EXAMPLE #6 provides a description of command outputs.

EXAMPLE #10  To rename a Data Mover entry from server_2 to dm2, type:

   $ nas_server -rename server_2 dm2

   id        = 1
   name      = dm2
   acl       = 1000, owner=nasadmin, ID=201
   type      = nas
   slot      = 2
   member_of =
   standby   = server_5, policy=auto
   status    :
      defined = enabled
      actual = online, active

EXAMPLE #4 provides a description of command outputs.

EXAMPLE #11  To set the access control level for server_2, type:

   $ nas_server -acl 1432 server_2

   id        = 1
   name      = server_2
   acl       = 1432, owner=nasadmin, ID=201
   type      = nas
   slot      = 2
   member_of =
   standby   = server_5, policy=auto
   status    :
      defined = enabled
      actual = online, ready

Note: The value 1432 specifies nasadmin as the owner, gives users with an access level of at least observer read-only access, users with an access level of at least operator read/write access, and users with an access level of at least admin read/write/delete access.

EXAMPLE #4 provides a description of command outputs.
EXAMPLE #12  To delete vdm_1, type:

$ nas_server -delete vdm_1

id        = 3
name      = vdm_1
acl       = 0
type      = vdm
server    =
rootfs    = root_fs_vdm_1
I18N mode = UNICODE
mountedfs =
member_of =
status    :
  defined = enabled
  actual = permanently unloaded
Interfaces to services mapping:

EXAMPLE #6 provides a description of command outputs.

EXAMPLE #13  To delete a physical Data Mover using root command, type:

$ /nas/sbin/rootnas_server -delete server_3

id        = 2
name      = server_3
acl       = 0
type      = nas
slot      = 3
member_of =
standby   = server_5, policy=auto
status    :
  defined = disabled
  actual = boot_level=0

EXAMPLE #6 provides a description of command outputs.
nas_slice

Manages the slice table.

SYNOPSIS

nas_slice
   -list
   | -delete <slice_name>
   | -info {<slice_name>|id=<slice_id>}
   | -rename <old_name> <new_name>
   | [-name <name>] -create <volume_name> <size>[<offset>]

DESCRIPTION

nas_slice creates, deletes, and renames a slice. nas_slice also displays the slice table and displays the attributes of a specified slice.

OPTIONS

   -list
Lists the slice table.

   -delete <slice_name>
Deletes an unused slice. Slices cannot be used by a storage pool.

   -info {<slice_name>|id=<slice_id>}
Displays attributes for the slice as specified by the <slice_name> or <slice_id>.

   -rename <old_name> <new_name>
Changes the current name of a slice to the new name.

   [-name <name>] -create <volume_name> <size>[<offset>]
Creates a slice with the specified size in megabytes. Also assigns an optional name for the slice. Names are case-sensitive.

   [<offset>]
   Specifies the offset which is the distance in megabytes from the beginning of the volume to the start of a slice. If an offset is not specified, the system defaults to the next best-fit size.

SEE ALSO

Managing Volumes and File Systems with VNX Automatic Volume Management, Managing Volumes and File Systems for VNX Manually, nas_volume to manage metavolumes, stripe volumes, and disk volumes and nas_cel.
EXAMPLE #1

To display the slice table, type:

```
$ nas_slice -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>inuse</th>
<th>slice_of</th>
<th>offsetMB</th>
<th>sizeMB</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>134</td>
<td>root_dos</td>
</tr>
<tr>
<td>2</td>
<td>n</td>
<td>1</td>
<td>134</td>
<td>1</td>
<td>root_layout</td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td>1</td>
<td>135</td>
<td>16</td>
<td>root_slice_1</td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td>1</td>
<td>151</td>
<td>16</td>
<td>root_slice_2</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>1</td>
<td>167</td>
<td>16</td>
<td>root_slice_3</td>
</tr>
<tr>
<td>6</td>
<td>y</td>
<td>1</td>
<td>183</td>
<td>16</td>
<td>root_slice_4</td>
</tr>
<tr>
<td>7</td>
<td>y</td>
<td>1</td>
<td>199</td>
<td>16</td>
<td>root_slice_5</td>
</tr>
<tr>
<td>8</td>
<td>y</td>
<td>1</td>
<td>215</td>
<td>16</td>
<td>root_slice_6</td>
</tr>
<tr>
<td>9</td>
<td>y</td>
<td>1</td>
<td>231</td>
<td>16</td>
<td>root_slice_7</td>
</tr>
<tr>
<td>10</td>
<td>y</td>
<td>1</td>
<td>247</td>
<td>16</td>
<td>root_slice_8</td>
</tr>
<tr>
<td>11</td>
<td>y</td>
<td>1</td>
<td>263</td>
<td>16</td>
<td>root_slice_9</td>
</tr>
<tr>
<td>12</td>
<td>y</td>
<td>1</td>
<td>279</td>
<td>16</td>
<td>root_slice_10</td>
</tr>
<tr>
<td>13</td>
<td>y</td>
<td>1</td>
<td>295</td>
<td>16</td>
<td>root_slice_11</td>
</tr>
<tr>
<td>14</td>
<td>y</td>
<td>1</td>
<td>311</td>
<td>16</td>
<td>root_slice_12</td>
</tr>
<tr>
<td>15</td>
<td>y</td>
<td>1</td>
<td>327</td>
<td>16</td>
<td>root_slice_13</td>
</tr>
<tr>
<td>16</td>
<td>y</td>
<td>1</td>
<td>343</td>
<td>16</td>
<td>root_slice_14</td>
</tr>
<tr>
<td>17</td>
<td>y</td>
<td>1</td>
<td>359</td>
<td>16</td>
<td>root_slice_15</td>
</tr>
<tr>
<td>18</td>
<td>y</td>
<td>1</td>
<td>375</td>
<td>16</td>
<td>root_slice_16</td>
</tr>
<tr>
<td>19</td>
<td>n</td>
<td>1</td>
<td>391</td>
<td>1</td>
<td>root_log_1</td>
</tr>
<tr>
<td>20</td>
<td>n</td>
<td>1</td>
<td>392</td>
<td>1</td>
<td>root_log_2</td>
</tr>
<tr>
<td>21</td>
<td>n</td>
<td>1</td>
<td>393</td>
<td>1</td>
<td>root_log_3</td>
</tr>
<tr>
<td>22</td>
<td>n</td>
<td>1</td>
<td>394</td>
<td>1</td>
<td>root_log_4</td>
</tr>
<tr>
<td>23</td>
<td>n</td>
<td>1</td>
<td>395</td>
<td>1</td>
<td>root_log_5</td>
</tr>
<tr>
<td>24</td>
<td>n</td>
<td>1</td>
<td>396</td>
<td>1</td>
<td>root_log_6</td>
</tr>
<tr>
<td>25</td>
<td>n</td>
<td>1</td>
<td>397</td>
<td>1</td>
<td>root_log_7</td>
</tr>
<tr>
<td>26</td>
<td>n</td>
<td>1</td>
<td>398</td>
<td>1</td>
<td>root_log_8</td>
</tr>
<tr>
<td>27</td>
<td>n</td>
<td>1</td>
<td>399</td>
<td>1</td>
<td>root_log_9</td>
</tr>
<tr>
<td>28</td>
<td>n</td>
<td>1</td>
<td>400</td>
<td>1</td>
<td>root_log_10</td>
</tr>
<tr>
<td>29</td>
<td>n</td>
<td>1</td>
<td>401</td>
<td>1</td>
<td>root_log_11</td>
</tr>
<tr>
<td>30</td>
<td>n</td>
<td>1</td>
<td>402</td>
<td>1</td>
<td>root_log_12</td>
</tr>
<tr>
<td>31</td>
<td>n</td>
<td>1</td>
<td>403</td>
<td>1</td>
<td>root_log_13</td>
</tr>
<tr>
<td>32</td>
<td>n</td>
<td>1</td>
<td>404</td>
<td>1</td>
<td>root_log_14</td>
</tr>
</tbody>
</table>

Note: This is a partial listing of the slice table.

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the slice.</td>
</tr>
<tr>
<td>inuse</td>
<td>Whether the slice is used.</td>
</tr>
<tr>
<td>slice_of</td>
<td>Volume the slice is sliced from.</td>
</tr>
<tr>
<td>offsetMB</td>
<td>Distance in megabytes from the beginning of the disk to the start of the slice.</td>
</tr>
<tr>
<td>sizeMB</td>
<td>Slice size in megabytes.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the slice.</td>
</tr>
</tbody>
</table>
**EXAMPLE #2**  To create a slice named slv1 on volume d8 of 1024 MB in size, type:

```bash
$ nas_slice -name slv1 -create d8 1024
```

id        = 76
name      = slv1
acl       = 0
in_use    = False
slice_of  = d8
offset(MB)= 0
size (MB)= 1024
volume_name = slv1

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the slice.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the slice.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value assigned to the slice.</td>
</tr>
<tr>
<td>in_use</td>
<td>Whether the slice is used.</td>
</tr>
<tr>
<td>slice_of</td>
<td>Volume the slice is sliced from.</td>
</tr>
<tr>
<td>offset(MB)</td>
<td>Distance in Megabytes from the beginning of the disk to the start of the slice.</td>
</tr>
<tr>
<td>size (MB)</td>
<td>Slice size in Megabytes.</td>
</tr>
<tr>
<td>volume_name</td>
<td>Volume used to create the slice.</td>
</tr>
</tbody>
</table>

**EXAMPLE #3**  To create a slice volume of 1 GB (1024 MB) with an offset of 1 GB called slv2 from volume d8, type:

```bash
$ nas_slice -name slv2 -create d8 1024 1024
```

id        = 77
name      = slv2
acl       = 0
in_use    = False
slice_of  = d8
offset(MB)= 1024
size (MB)= 1024
volume_name = slv2

**EXAMPLE #2** provides a description of command outputs.

**EXAMPLE #4**  To display attributes of slv1, type:

```bash
$ nas_slice -info slv1
```

id        = 76
name      = slv1
acl       = 0
in_use    = False
slice_of  = d8
offset(MB)= 0
size (MB)= 1024
volume_name = slv1
EXAMPLE #2 provides a description of command outputs.

**EXAMPLE #5**

To rename a slice from `slv` to `slv1`, type:

```
$ nas_slice -rename slv slv1
```

id        = 76  
name      = slv1  
acl       = 0  
in_use    = False  
slice_of  = d8  
offset(MB)= 0  
size  (MB)= 1024  
volume_name = slv1

EXAMPLE #2 provides a description of command outputs.

**EXAMPLE #6**

To delete `slv1`, type:

```
$ nas_slice -delete slv1
```

id        = 76  
name      = slv1  
acl       = 0  
in_use    = False  
slice_of  = d8  
offset(MB)= 0  
size  (MB)= 1024

EXAMPLE #2 provides a description of command outputs.
nas_stats

Manages Statistics Groups.

SYNOPSIS

nas_stats
   -groups
   { -list
     | -info [-all] <statgroup_name> [, ...]
     | -create <statgroup_name>
     |     [-description "<description_line>"]
     |     {[<statpath_name>] <statgroup_name> [, ...]}
     | -modify <statgroup_name>
     |     [ [-rename <new_statgroup_name>]
     |     [-description "<description_line>"
     |     {[<statpath_name>] <statgroup_name> [, ...]}}
     | -add <statgroup_name>
     |     {[<statpath_name>] <statgroup_name> [, ...]}
     | -remove <statgroup_name>
     |     {[<statpath_name>] <statgroup_name> [, ...]}
     | -delete <statgroup_name> [-Force]
     | -database
     |     { -recover [-Force]
     |     -verify }

DESCRIPTION

nas_stats allows the user to manage Statistics Groups. There are two
types of Statistics Groups: System-defined and User-defined groups.
System-defined statistics groups are created as part of the installation
(or upgrade) process and include the following statistics, which
correspond to the summary and table collections used by
server_stats:

<table>
<thead>
<tr>
<th>System-defined statistics group name</th>
<th>server_stats collection name</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic-std</td>
<td>-summary basic</td>
</tr>
<tr>
<td>caches-std</td>
<td>-summary caches</td>
</tr>
<tr>
<td>cifs-std</td>
<td>-summary cifs</td>
</tr>
<tr>
<td>nfs-std</td>
<td>-summary nfs</td>
</tr>
<tr>
<td>cifsOps-std</td>
<td>-table cifs</td>
</tr>
<tr>
<td>diskVolumes-std</td>
<td>-table dvol</td>
</tr>
<tr>
<td>metaVolumes-std</td>
<td>-table fsvol</td>
</tr>
<tr>
<td>netDevices-std</td>
<td>-table net</td>
</tr>
<tr>
<td>nfsOps-std</td>
<td>-table nfs</td>
</tr>
</tbody>
</table>
Note: server_stats collection names are deprecated and will not be supported in future releases.

Statistics Groups can be created to include any combination of statpath names, displayed through the server_stats command, as well as other existing statgroup names.

Any Statistics Group name can be used with server_stats -monitor to collect statistics as defined in its member_stats list.

OPTIONS

-list
Lists system and user-defined Statistics Groups.

-info
Provides detailed information on all (or specified) Statistics Groups.

-create
Creates a statistics group and specifies what statpath names it includes. It also allows the nesting of statgroups by adding existing statgroups to new statgroups.

Statgroup names can be used with the -info request. A statgroup name is limited to 255 characters. Space, slash, back slash, quote, double quote, and comma are the illegal characters in it.

[-description]

The -description option is optional and defaults to the statgroup name. If the -description option is used, its argument must be enclosed in quotation marks.

-modify
Allows you to modify a statgroup’s member_stats list by specifying the new member statistics of the group, overriding the previous contents.

-add
Allows you to add statpath and existing statgroup names to a statgroup by specifying additional items to be appended to the statgroup’s list member_stats list.

-remove
Allows you to remove member statpath and statgroup names from a statgroup by specifying the items to remove from the statgroup’s member_stats list.
The nas Commands

- delete
  Allows you to delete a statgroup. However, this option does not delete any statsgroups that are members of the statgroup.

- recover
  Attempts to recover the latest uncorrupted copy of the Statistics Groups database from the NAS database backups. nas_stats searches through the available backups and restores the latest copy. In this event, NAS database backups do not contain a healthy version of the Statistics Groups; a new Statistics Groups database is installed. In the case of a new Statistics Groups database, all user-defined information is lost. NAS database backups run hourly and VNX maintains the last 12 backups.

    [-Force]

    Use the -Force option with the -recover option to skip the warning prompt.

- verify
  Checks the health status of the Statistics Groups database.

SEE ALSO server_stats

EXAMPLE #1 To list the system-defined and user-defined Statistics Groups, type:

$ nas_stats -groups -list

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>basic-std</td>
</tr>
<tr>
<td>System</td>
<td>basicCifs-std</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>User</td>
<td>basic</td>
</tr>
<tr>
<td>User</td>
<td>nfsNet</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

EXAMPLE #2 To provide detailed information on all (or specified) Statistics Groups, type:

$ nas_stats -groups -info

name                = basic-std
description         = The basic system-defined group.
type                = System-defined
member_stats        =
  kernel.cpu.utilization.cpuUtil,net.basic.inBytes,net.basic.outBytes,store.readBytes,store.writeBytes
member_elements     =
member_of           =
name            = basic3
description     = CPU and Memory
type            = User-defined
member_stats    = kernel.cpu.utilization.cpuUtil,kernel.memory.freeBytes
member_elements =
member_of       =

name            = caches-std
description     = The caches system-defined group.
type            = System-defined
member_stats    = fs.dnlc.hitRatio,fs.ofCache.hitRatio,kernel.memory.bufferCache.hitRatio
member_elements =
member_of       =

name            = cifs-std
description     = The cifs system-defined group.
type            = System-defined
member_stats    = cifs.global.basic.totalCalls,cifs.global.basic.reads,cifs.global.basic.readBytes,
cifs.global.basic.readAvgSize,cifs.global.basic.writes,cifs.global.basic.writeBytes,cifs.global.basic.writeAvgSize,cifs.global.usage.currentConnections,cifs.global.usage.currentOpenFiles
member_elements =
member_of       = newSG

name            = cifsOps-std
description     = The cifs table system-defined group.
type            = System-defined
member_stats    = cifs.smb1.op,cifs.smb2.op
member_elements =
member_of       =

name            = diskVolumes-std
description     = The disk volume table system-defined group.
type            = System-defined
member_stats    = store.diskVolume
member_elements =

name            = metaVolumes-std
description     = The meta volume table system-defined group.
type            = System-defined
member_stats    = store.logicalVolume.metaVolume
member_elements =
member_of       =

name            = netDevices-std
description     = The net table system-defined group.
type            = System-defined
member_stats    = net.device
member_elements =
member_of       =
**EXAMPLE #3**

To provide detailed information on all (or specified) Statistics Groups, type:

```bash
$ nas_stats -groups -info statsA
```

name = statsA
description = My group # 2
type = user-defined
member_stats = statpath1, statpath2, statpath3, statsC
member_elements =
member_of = statsB
EXAMPLE #4 To create a statistics group called basic3, type:

$ nas_stats -groups -create basic3 -description "CPU and Memory" kernel.cpu.utilization.cpuUtil,kernel.memory.freeBytes

'basic3' created successfully.

EXAMPLE #5 To create a statistics group called statgroup2, type:

$ nas_stats -groups -create statgroup2 statgroup1,nfs,net

'statgroup2' created successfully.

EXAMPLE #6 To use an existing statgroup, type:

$ nas_stats -groups -create statgroup1 -description "My first group" kernel.cpu.utilization.cpuUtil,net.basic.inBytes,net.basic.outBytes,store.readBytes,store.writeBytes

ERROR (13421969439): ‘statgroup1’ already exists.

EXAMPLE #7 To modify a statgroup by specifying the new contents of the group, overriding the previous contents, type:

$ nas_stats -groups -modify statgroup2 cifs,nfs-std

'statgroup2' modified successfully.

EXAMPLE #8 To modify the description of a statgroup, type:

$ nas_stats -groups -modify basic1 -description "My basic group"

'basic1' modified successfully.

EXAMPLE #9 To rename a user-defined statgroup, type:

$ nas_stats -groups -modify statgroup2 -rename basic2

'statgroup2' modified successfully.

EXAMPLE #10 To add to the member_stats list of a statgroup, type:

$ nas_stats -groups -add statgroup2 kernel.cpu.utilization.cpuUtil,statgroup1

Adding the following statistics:
... kernel.cpu.utilization.cpuUtil
... statgroup1
Statistics added to ‘statgroup2’ successfully.

EXAMPLE #11 To remove from the member_stats list of a statgroup, type:

$ nas_stats -groups -remove statgroup1 kernel.cpu.utilization.cpuUtil

Removing the following statistics:
... kernel.cpu.utilization.cpuUtil
Statistics removed from ‘statgroup1’ successfully.
**EXAMPLE #12** To delete a statgroup, type:

```bash
$ nas_stats -groups -delete statgroup1

'statgroup1' deleted successfully.
```

**EXAMPLE #13** To delete reference from other groups using `statgroupA`, type:

```bash
$ nas_stats -groups -delete statgroupA

'statgroupA' is used in group (s): mystats1, mystats2.
Clear 'statgroupA' from other groups? [Y/N] Y
'statgroupA' deleted successfully.
```

**EXAMPLE #14** To delete reference from other groups using `statgroupA` and the `-Force` option to skip the warning prompt, type:

```bash
$ nas_stats -groups -delete statgroupA -F

'statgroupA' is used in group (s): mystats1, mystats2.
'statgroupA' deleted successfully.
```

**EXAMPLE #15** To recover the latest healthy (uncorrupted) copy of a statgroup database from the NAS database backups, type:

```bash
$ nas_stats -groups -database -recover

Latest healthy database modified last on Tue Apr 7 17:29:06 EDT 2009.
Any updates performed after the latest backup will be lost. Continue? [Y/N] Y
The nas_stats command recover operation is completed successfully.
```

**EXAMPLE #16** To recover the latest healthy (uncorrupted) copy of the statgroup database from the NAS database backups using the `-Force` option to skip the warning prompt, type:

```bash
$ nas_stats -groups -database -recover -Force

Latest healthy database modified last on Tue Apr 7 17:29:06 EDT 2009.
The nas_stats command recover operation is completed successfully.
```

**EXAMPLE #17** To check the health status of the Statistics Groups database, type:

```bash
$ nas_stats -groups -database -verify

Database is healthy.
```
The nas Commands

**nas_storage**

Controls storage system access and performs some management tasks.

**SYNOPSIS**

```
nas_storage
   -list
   -info { -all | <name> | id=<storage_id> } [ -option <options> ]
   -rename <old_name> <new_name>
   -acl <acl_value> <name>
   -delete { <name> | id=<storage_id> } [ -spare <spindle-id> | -group <diskgroup-id> ]
   -failback { <name> | id=<storage_id> }
   -sync { -all | <name> | id=<storage_id> }
   -check { -all | <name> | id=<storage_id> }
   -modify { <name> | id=<storage_id> } -network { -spa | -spb } <IP>
   -modify { <name> | id=<storage_id> } -security [ -username <username> ] [ -password <password> ] [ -newpassword <new_password> ]
```

**DESCRIPTION**

**nas_storage** sets the name for a storage system, assigns an access control value, displays attributes, performs a health check, synchronizes the storage system with the Control Station, and performs a failback for VNX for block systems.

**OPTIONS**

**-list**

Displays a list of all attached storage systems available for the VNX.

**-info { -all | <name> | id=<storage_id> }**

Displays the configuration of the attached storage system.

[ -option <options> ]

Specifies a comma-separated list of options.

**sync={yes|no}**

Synchronizes the Control Station’s view with that of the storage system before displaying configuration information. Default=**yes**.

**Note:** Output from this command is determined by the type of storage system attached to the VNX.
The nas Commands

- **rename** <old_name> <new_name>
  Renames the current storage system name to a new name. By default, the storage system name is its serial number.

- **acl** <acl_value> <name>
  Sets an access control level value that defines the owner of the storage system, and the level of access allowed for users and groups defined in the access control level table (nas_acl provides information).

- **delete** {<name>|id=<storage_id> [-spare <spindle-id>|-group <diskgroup-id>]}
  Deletes an entry from the storage system table. The storage system can only be deleted after all disks on the storage system have been deleted using `nas_disk`. The storage system and disks can be rediscovered using the `server_devconfig` command. The `-spare` option deletes the hot spare disk from the hot spare pool on the VNX for block storage used by NAS. The `-group` option deletes the disk group specified. This deletes and unbinds the LUNs in the RAID groups used by VNX for file. If there are other LUNs in the RAID group not allocated to the VNX, the RAID group is not unbound. After removing the VNX LUNs, the RAID group is empty and it will be destroyed.

- **sync** {-all|<name>|id=<storage_id>}
  Synchronizes the Control Station’s view with that of the storage system.

- **check** {-all|<name>|id=<storage_id>}
  Performs a health check on the storage system to verify if it is configured for, and in a state to provide the level of high availability that is required.

  Use this option after making any management changes to your storage system (for example, changes to VNX for block array properties, such as enabling/disabling statistics polling).

For VNX for Block only

- **failback** {<name>|id=<storage_id>}
  Returns the storage system’s normal operating state by returning ownership of all disk volumes to their default storage processor.

  To verify that the storage system failed over, type the `-info` option. If the value appears as failed_over=True, then the system has failed over.

- **modify** {<name>|id=<storage_id>} -network {-spa | -spb} <IP>
  Modifies the IP address of the VNX for block in the VNX database.
The nas Commands

-modify {<name>|id=<storage_id>} -security [-username <username>] [-password <password>]

Updates the login information the VNX for file uses to authenticate with the VNX and changes the VNX username, or password if the VNX account is changed or the following error is reported:

Error 5010: APM00055105668: Storage API code=4651: SYMAPI_C_CLAR_NOT_PRIVILEGED

Operation denied by Clarion array - you are not privileged to perform the requested operation

[-newpassword <new_password>]

Assigns a new password to the username on the VNX for block.

Note: This operation is not supported for Symmetrix storage systems.

SEE ALSO

VNX System Operations, nas_rdf, nas_disk, and server_devconfig.

STORAGE SYSTEM OUTPUT

The number associated with the storage device is dependent on the attached storage system. VNX for block displays a prefix of APM before a set of integers, for example, APM00033900124-0019. For example, Symmetrix storage systems appear as 002804000190-003C.

EXAMPLE #1

For the VNX storage system, to list all attached storage systems, type:

$ nas_storage -list

id      acl                   name       serial_number
1         0         APM00042000818      APM00042000818

For the VNX with a Symmetrix storage system, to list all attached storage systems, type:

$ nas_storage -list

id      acl                   name       serial_number
1       0             000187940260        000187940260

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID number of the attached storage system.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value assigned to the attached storage system.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to the attached storage system.</td>
</tr>
<tr>
<td>serial_number</td>
<td>Serial number of the attached storage system.</td>
</tr>
</tbody>
</table>

EXAMPLE #2

For the VNX storage system, to display information for the attached storage system, type:

$ nas_storage -info APM00042000818
The nas Commands

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>1</td>
</tr>
<tr>
<td>arrayname</td>
<td>APM00042000818</td>
</tr>
<tr>
<td>name</td>
<td>APM00042000818</td>
</tr>
<tr>
<td>type</td>
<td>Clarion</td>
</tr>
<tr>
<td>model_type</td>
<td>RACKMOUNT</td>
</tr>
<tr>
<td>model_num</td>
<td>700</td>
</tr>
<tr>
<td>db_sync_time</td>
<td>1131986667 == Mon Nov 14 11:44:27 EST 2005</td>
</tr>
<tr>
<td>API_version</td>
<td>V6.0-629</td>
</tr>
<tr>
<td>num_disks</td>
<td>60</td>
</tr>
<tr>
<td>num_devs</td>
<td>34</td>
</tr>
<tr>
<td>num_pdevs</td>
<td>8</td>
</tr>
<tr>
<td>num_storage_grps</td>
<td>1</td>
</tr>
<tr>
<td>num_raid_grps</td>
<td>16</td>
</tr>
<tr>
<td>cache_page_size</td>
<td>8</td>
</tr>
<tr>
<td>wr_cache_mirror</td>
<td>True</td>
</tr>
<tr>
<td>low_watermark</td>
<td>60</td>
</tr>
<tr>
<td>high_watermark</td>
<td>80</td>
</tr>
<tr>
<td>unassigned_cache</td>
<td>0</td>
</tr>
<tr>
<td>is_local</td>
<td>True</td>
</tr>
<tr>
<td>failed_over</td>
<td>False</td>
</tr>
<tr>
<td>captive_storage</td>
<td>False</td>
</tr>
</tbody>
</table>

Active Software

- AccessLogix = -
- FLARE-Operating-Environment = 02.16.700.5.004
- NavisphereManager = -

Storage Processors

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP Identifier</td>
<td>A</td>
</tr>
<tr>
<td>signature</td>
<td>1057303</td>
</tr>
<tr>
<td>microcode_version</td>
<td>2.16.700.5.004</td>
</tr>
<tr>
<td>serial_num</td>
<td>LKE00040201171</td>
</tr>
<tr>
<td>prom_rev</td>
<td>3.30.0</td>
</tr>
<tr>
<td>agent_rev</td>
<td>6.16.0 (4.80)</td>
</tr>
<tr>
<td>phys_memory</td>
<td>3967</td>
</tr>
<tr>
<td>sys_buffer</td>
<td>773</td>
</tr>
<tr>
<td>read_cache</td>
<td>122</td>
</tr>
<tr>
<td>write_cache</td>
<td>3072</td>
</tr>
<tr>
<td>free_memory</td>
<td>0</td>
</tr>
<tr>
<td>raid3_mem_size</td>
<td>0</td>
</tr>
<tr>
<td>failed_over</td>
<td>False</td>
</tr>
<tr>
<td>hidden</td>
<td>False</td>
</tr>
<tr>
<td>network_name</td>
<td>spa</td>
</tr>
<tr>
<td>ip_address</td>
<td>172.24.102.5</td>
</tr>
<tr>
<td>subnet_mask</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>gateway_address</td>
<td>172.24.102.254</td>
</tr>
<tr>
<td>num_disk_volumes</td>
<td>20 - root_disk root_ldisk d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d16 d17 d18 d19 d20</td>
</tr>
</tbody>
</table>
Port Information

Port 1

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_status</td>
<td>UP</td>
</tr>
<tr>
<td>port_status</td>
<td>ONLINE</td>
</tr>
<tr>
<td>switch_present</td>
<td>True</td>
</tr>
<tr>
<td>sp_source_id</td>
<td>6373907</td>
</tr>
</tbody>
</table>

Port 2

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_status</td>
<td>UP</td>
</tr>
<tr>
<td>port_status</td>
<td>ONLINE</td>
</tr>
<tr>
<td>switch_present</td>
<td>True</td>
</tr>
<tr>
<td>sp_source_id</td>
<td>6373651</td>
</tr>
</tbody>
</table>

SP Identifier = B
signature = 1118484
microcode_version = 2.16.700.5.004
serial_num = LKE00041700812
prom_rev = 3.30.00
agent_rev = 6.16.0 (4.80)
phys_memory = 3967
sys_buffer = 773
read_cache = 122
write_cache = 3072
free_memory = 0
raid3_mem_size = 0
failed_over = False
hidden = False
network_name = spb
ip_address = 172.24.102.6
subnet_mask = 255.255.255.0
gateway_address = 172.24.102.254
num_disk_volumes = 0

Port Information

Port 1

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_status</td>
<td>UP</td>
</tr>
<tr>
<td>port_status</td>
<td>ONLINE</td>
</tr>
<tr>
<td>switch_present</td>
<td>True</td>
</tr>
<tr>
<td>switch_uid</td>
<td>10:0:8:0:88:A0:36:F3:20:3E:8:0:88:A0:36:F3</td>
</tr>
<tr>
<td>sp_source_id</td>
<td>6372883</td>
</tr>
</tbody>
</table>
Port 2

link_status = UP
port_status = ONLINE
switch_present = True
switch_uid = 10:0:8:0:88:A0:36:F3:20:3D:8:0:88:A0:36:F3
sp_source_id = 6372627

Storage Groups

name = ns704g-cs100
num_hbas = 18
num_devices = 24
shareable = True
hidden = False

Hosts

storage_processor = B
port = 1
server = server_4

storage_processor = A
port = 0
server = server_4

storage_processor = B
port = 0
server = server_2

...removed...

storage_processor = B
port = 1
server = server_3

uid = 20:0:0:0:C9:2B:98:77:10:0:0:0:C9:2B:98:77
storage_processor = B
port = 0

uid = 20:0:0:0:C9:2B:98:77:10:0:0:0:C9:2B:98:77
storage_processor = A
port = 0
### The nas Commands

**ALU** | **HLU**
---|---
0000 | 0000
0001 | 0001
0002 | 0002
0003 | 0003
0004 | 0004
0005 | 0005
0018 | 0018
0019 | 0019
0020 | 0020
0021 | 0021
0022 | 0022
0023 | 0023
0024 | 0024
0025 | 0025
0026 | 0026
0027 | 0027
0028 | 0028
0029 | 0029
0030 | 0030
0031 | 0031
0032 | 0032
0033 | 0033
0034 | 0034
0035 | 0035

#### Disk Groups

- **id**: 0000
- **storage profiles**: 2 - clar_r5_performance, cm_r5_performance
- **raid_type**: RAID5
- **logical_capacity**: 1068997528
- **num_spindles**: 5 - 0_0_0 0_0_1 0_0_2 0_0_3 0_0_4
- **num_luns**: 6 - 0000 0001 0002 0003 0004 0005
- **num_disk_volumes**: 6 - root_disk root_ldisk d3 d4 d5 d6
- **spindle_type**: FC
- **bus**: 0
- **raw_capacity**: 1336246910
- **used_capacity**: 62914560
- **free_capacity**: 1006082968
- **hidden**: False

<!-- ...removed... -->

- **id**: 2_0_14
- **product**: ST314670 CLAR146
- **revision**: 6A06
- **serial**: 3KS02RHM
- **capacity**: 280346624
The nas Commands

used_capacity         = 224222822
disk_group            = 0014
hidden                = False
type                  = FC
bus                   = 2
enclosure             = 0
slot                  = 14
vendor                = SEAGATE
remapped_blocks       = -1
state                 = ENABLED

For the VNX with a Symmetrix storage system, to display information for the attached storage system, type:

$ nas_storage -info 000187940260

id                    = 1
serial_number         = 000187940260
name                  = 000187940260
type                  = Symmetrix
ident                 = Symm6
model                 = 800-M2
microcode_version     = 5670
microcode_version_num = 16260000
microcode_date        = 03012004
microcode_patch_level = 69
microcode_patch_date  = 03012004
symmetrix_pwron_time  = 1130260200 == Tue Oct 25 13:10:00 EDT 2005
db_sync_time          = 1133215405 == Mon Nov 28 17:03:25 EST 2005
db_sync_bcv_time      = 1133215405 == Mon Nov 28 17:03:25 EST 2005
db_sync_rdf_time      = 1133215405 == Mon Nov 28 17:03:25 EST 2005
last_ipl_time         = 1128707062 == Fri Oct  7 13:44:22 EDT 2005
last_fast_ipl_time    = 1130260200 == Tue Oct 25 13:10:00 EDT 2005
API_version           = V6.0-629
cache_size            = 32768
cache_slot_count      = 860268
max_wr_pend_slots     = 180000
max_da_wr_pend_slots  = 90000
max_dev_wr_pend_slots = 6513
permacache_slot_count = 0
num_disks             = 60
num_symdevs           = 378
num_pdevs             = 10
sddf_configuration    = ENABLED
config_checksum       = 0x01ca544
num_powerpath_devs    = 0
config_crc            = 0x07e0ba1e6
is_local              = True

Physical Devices
/nas/dev/c0t0115s2  
/nas/dev/c0t0115s3
The nas Commands

/nas/dev/c0t0l15s4
/nas/dev/c0t0l15s6
/nas/dev/c0t0l15s7
/nas/dev/c0t0l15s8
/nas/dev/c16t0l15s2
/nas/dev/c16t0l15s3
/nas/dev/c16t0l15s4
/nas/dev/c16t0l15s8

### Director Table

<table>
<thead>
<tr>
<th>type</th>
<th>num</th>
<th>slot</th>
<th>ident</th>
<th>stat</th>
<th>scsi</th>
<th>vols</th>
<th>ports</th>
<th>p0_stat</th>
<th>p1_stat</th>
<th>p2_stat</th>
<th>p3_stat</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA</td>
<td>1</td>
<td>1</td>
<td>DF-1A</td>
<td>On</td>
<td>NA</td>
<td>21</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>DA</td>
<td>2</td>
<td>2</td>
<td>DF-2A</td>
<td>On</td>
<td>NA</td>
<td>8</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>DA</td>
<td>15</td>
<td>15</td>
<td>DF-15A</td>
<td>On</td>
<td>NA</td>
<td>21</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>DA</td>
<td>16</td>
<td>16</td>
<td>DF-16A</td>
<td>On</td>
<td>NA</td>
<td>8</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>DA</td>
<td>17</td>
<td>1</td>
<td>DF-1B</td>
<td>On</td>
<td>NA</td>
<td>8</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>DA</td>
<td>18</td>
<td>2</td>
<td>DF-2B</td>
<td>On</td>
<td>NA</td>
<td>21</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>DA</td>
<td>31</td>
<td>15</td>
<td>DF-15B</td>
<td>On</td>
<td>NA</td>
<td>152</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>DA</td>
<td>32</td>
<td>16</td>
<td>DF-16B</td>
<td>On</td>
<td>NA</td>
<td>165</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>FA</td>
<td>33</td>
<td>1</td>
<td>FA-1C</td>
<td>On</td>
<td>NA</td>
<td>0</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>FA</td>
<td>34</td>
<td>2</td>
<td>FA-2C</td>
<td>On</td>
<td>NA</td>
<td>0</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>FA</td>
<td>47</td>
<td>15</td>
<td>FA-15C</td>
<td>On</td>
<td>NA</td>
<td>0</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>FA</td>
<td>48</td>
<td>16</td>
<td>FA-16C</td>
<td>On</td>
<td>NA</td>
<td>0</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>FA</td>
<td>49</td>
<td>1</td>
<td>FA-1D</td>
<td>On</td>
<td>NA</td>
<td>0</td>
<td>2</td>
<td>On</td>
<td>On</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

**Note:** This is a partial listing due to the length of the outputs.

**EXAMPLE #3**  To rename a storage system, type:

```bash
$ nas_storage -rename APM00042000818 cx700_1
```

<table>
<thead>
<tr>
<th>id</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>serial_number</td>
<td>APM00042000818</td>
</tr>
<tr>
<td>name</td>
<td>cx700_1</td>
</tr>
<tr>
<td>acl</td>
<td>0</td>
</tr>
</tbody>
</table>

**EXAMPLE #4**  To set the access control level for the storage system `cx700_1`, type:

```bash
$ nas_storage -acl 1000 cx700_1
```

<table>
<thead>
<tr>
<th>id</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>serial_number</td>
<td>APM00042000818</td>
</tr>
<tr>
<td>name</td>
<td>cx700_1</td>
</tr>
<tr>
<td>acl</td>
<td>1000, owner=nasadmin, ID=201</td>
</tr>
</tbody>
</table>

**Note:** The value 1000 specifies `nasadmin` as the owner and gives read, write, and delete access only to `nasadmin`.

**EXAMPLE #5**  To change the existing password on the VNX for block, type:
$ nas_storage -modify APM00070204288 -security -username nasadmin -password nasadmin -newpassword abc

Changing password on APM00070204288

EXAMPLE #6  To avoid specifying passwords in clear text on the command line, type:

$ nas_storage -modify APM00070204288 -security
              -newpassword

Enter the Global CLARiiON account information
Username: nasadmin
Password: *** Retype your response to validate
Password: ***
New Password
Password: ******** Retype your response to validate
Password: ********
Changing password on APM00070204288
Done

EXAMPLE #7  To failback a VNX for block, type:

$ nas_storage -failback cx700_1

id       = 1
serial_number = APM00042000818
name     = cx700_1
acl      = 1000, owner=nasadmin, ID=201

EXAMPLE #8  To display information for a VNX for block and turn synchronization off, type:

$ nas_storage -info cx700_1 -option sync=no

id                    = 1
arrayname             = APM00042000818
name                  = cx700_1
type                  = Clariion
model_type            = RACKMOUNT
model_num             = 700
db_sync_time          = 1131986667 == Mon Nov 14 11:44:27 EST 2005
API_version           = V6.0-629
num_disks             = 60
num_devs              = 34
num_pdevs             = 8
num_storage_grps      = 1
num_raid_grps         = 16
cache_page_size       = 8
wr_cache_mirror       = True
low_watermark         = 60
high_watermark        = 80
unassigned_cache      = 0
is_local              = True
failed_over           = False
captive_storage       = False
Active Software

-AccessLogix = -
FLARE-Operating-Environment= 02.16.700.5.004
-NavisphereManager = -

Storage Processors

SP Identifier = A
signature = 1057303
microcode_version = 2.16.700.5.004
serial_num = LKE00040201171
prom_rev = 3.30.00
agent_rev = 6.16.0 (4.80)
phys_memory = 3967
sys_buffer = 773
read_cache = 122
write_cache = 3072
free_memory = 0
raid3_mem_size = 0
failed_over = False
hidden = False
network_name = spa
ip_address = 172.24.102.5
subnet_mask = 255.255.255.0
gateway_address = 172.24.102.254
num_disk_volumes = 20 - root_disk root_ldisk d3 d4 d5 d6 d7 d8 d9 d10
d11 d12 d13 d14 d15 d16 d17 d18 d19 d20

Port Information

Port 1

link_status = UP
port_status = ONLINE
switch_present = True
sp_source_id = 6373907

Port 2

link_status = UP
port_status = ONLINE
switch_present = True
sp_source_id = 6373651
The nas Commands

SP Identifier = B
signature = 1118484
microcode_version = 2.16.700.5.004
serial_num = LKE00041700812
prom_rev = 3.30.00
agent_rev = 6.16.0 (4.80)
phys_memory = 3967
sys_buffer = 773
read_cache = 122
write_cache = 3072
free_memory = 0
raid3_mem_size = 0
failed_over = False
hidden = False
network_name = spb
ip_address = 172.24.102.6
subnet_mask = 255.255.255.0
gateway_address = 172.24.102.254
num_disk_volumes = 0

Port Information

Port 1

link_status = UP
port_status = ONLINE
switch_present = True
switch_uid = 10:0:8:0:88:A0:36:F3:20:3E:8:0:88:A0:36:F3
sp_source_id = 6372883

<...removed...>

Port 2

link_status = UP
port_status = ONLINE
switch_present = True
switch_uid = 10:0:8:0:88:A0:36:F3:20:3D:8:0:88:A0:36:F3
sp_source_id = 6372627

Storage Groups

name = ns704g-cs100
num_hbas = 18
num_devices = 24
shareable = True
hidden = False
Hosts

storage_processor = B
port              = 1
server            = server_4

storage_processor = A
port              = 0
server            = server_4

storage_processor = B
port              = 0
server            = server_2

<...removed...>

storage_processor = B
port              = 1
server            = server_3

uid               = 20:0:0:0:C9:2B:98:77:10:0:0:0:C9:2B:98:77
storage_processor = B
port              = 0

uid               = 20:0:0:0:C9:2B:98:77:10:0:0:0:C9:2B:98:77
storage_processor = A
port              = 0

ALU     HLU
--------
0000 -> 0000
0001 -> 0001
0002 -> 0002
0003 -> 0003
0004 -> 0004
0005 -> 0005
0018 -> 0018
0019 -> 0019
0020 -> 0020
0021 -> 0021
0022 -> 0022
0023 -> 0023
0024 -> 0024
0025 -> 0025
0026 -> 0026
0027 -> 0027
0028 -> 0028
0029 -> 0029
Disk Groups

id = 0000
storage_profiles = 2 - clar_r5_performance,cm_r5_performance
raid_type = RAID5
logical_capacity = 1068997528
num_spindles = 5 - 0_0_0 0_0_1 0_0_2 0_0_3 0_0_4
num_luns = 6 - 0000 0001 0002 0003 0004 0005
num_disk_volumes = 6 - root_disk root_ldisk d3 d4 d5 d6
spindle_type = FC
bus = 0
raw_capacity = 1336246910
used_capacity = 62914560
free_capacity = 1006082968
hidden = False

<...removed...>

id = 0205
storage_profiles = 0
raid_type = SPARE
logical_capacity = 622868992
num_spindles = 1 - 0_1_0
num_luns = 1 - 0205
num_disk_volumes = 0
spindle_type = ATA
bus = 0
raw_capacity = 622868992
used_capacity = 622868992
free_capacity = 0
hidden = False

Spindles

id = 0_0_0
product = ST314670 CLAR146
revision = 6A06
serial = 3KS088SQ
capacity = 280346624
used_capacity = 12582912
disk_group = 0000
hidden = False
type = FC
bus = 0
The nas Commands

enclosure = 0
slot = 0
vendor = SEAGATE
remapped_blocks = -1
state = ENABLED

<...removed...>

id = 2_0_14
product = ST314670 CLAR146
revision = 6A06
serial = 3KS02RHM
capacity = 280346624
used_capacity = 224222822
disk_group = 0014
hidden = False
type = FC
bus = 2
enclosure = 0
slot = 14
vendor = SEAGATE
remapped_blocks = -1
state = ENABLED

Note: This is a partial display due to the length of the outputs.

EXAMPLE #9 To delete a storage system with no attached disks, type:
$ nas_storage -delete APM00035101740

id = 0
serial_number = APM00035101740
name = APM00035101740
acl = 0

EXAMPLE #10 To turn synchronization on for all systems, type:
$ nas_storage -sync -all

done
EXAMPLE #11  To perform a health check on the storage system, type:

$ nas_storage -check -all

Discovering storage (may take several minutes)
done

EXAMPLE #12  To set the access control level for the storage system APM00042000818, type:

$ nas_storage -acl 1432 APM00042000818

id = 1
serial_number = APM00042000818
name = APM00042000818
acl = 1432, owner=nasadmin, ID=201

Note: The value 1432 specifies nasadmin as the owner and gives users with an access level of at least observer read access only, users with an access level of at least operator read/write access, and users with an access level of at least admin read/write/delete access.

EXAMPLE #13  To modify the IP address of the VNX for block, type:

$ nas_storage -modify APM00072303347 -network -spa 10.6.4.225

Changing IP address for APM00072303347
Discovering storage (may take several minutes)
done
The nas Commands

nas_task

Manages in-progress or completed tasks.

SYNOPSIS

nas_task
   -list [-remote_system {<remoteSystemName>|id=<id>}] |
   -info {-all|<taskId>}
   [-remote_system {<remoteSystemName>|id=<id>}] |
   -abort <taskId>
   [-mover <moverName>][-remote_system {<remoteSystemName>|id=<id>}] |
   -delete <taskId>  [-remote_system {<remoteSystemName>|id=<id>}]}

DESCRIPTION

nas_task lists the tasks associated with commands currently in progress or completed, reports information about a particular task, aborts a task, or deletes a task. Each task can be uniquely identified by its task ID and the remote VNX system name or ID.

Use the nas_task command to monitor, abort, and delete long running tasks and tasks started in asynchronous mode.

OPTIONS

-list
Lists all local tasks that are in progress, or completed tasks that have not been deleted. For each task, lists the task ID, remote system name, a description of the task, and the task state (running, recovering, succeeded, or failed).

   -remote_system {<remoteSystemName>|id=<id>}
Lists local tasks initiated by the specified remote VNX system. Specify the remote system name or ID.

-info {-all|<taskId>}
Provides more detailed status information for all tasks or for a particular task. Displays the run time status, estimated completion time, and percent complete for running tasks. Displays the completion status and actual end time for completed tasks.

The taskID is the ID returned from a command run in the background mode or from the nas_task -list command.

Note: The ID of the task is an integer and is assigned automatically. The task ID is unique to the VNX.
The nas Commands

[-remote_system {<remoteSystemName> | id=<id>}]  
Provides more detailed status information of local tasks initiated by the specified remote VNX system. Specify the remote system name or remote system ID. The remote system name is returned from the nas_task -list command.

-abort <taskId>  
Aborts the specified task leaving the system in a consistent state. For example, it aborts a one-time copy in progress. This might take a long time to complete because a remote system may be unavailable or the network may be down. You should check the status of the task to verify that the task was aborted. This command can be executed from the source only.

CAUTION
This option might leave the system in an inconsistent state. Use caution when using this option.

[-mover <moverName>]  
Aborts a task running locally on the specified Data Mover. This command can be executed from the source or destination side. Use this command when the source and destination VNX systems cannot communicate. You should run this command on both sides.

[-remote_system {<remoteSystemName> | id=<id>}]  
Aborts a task that was initiated on a remote VNX leaving the source side intact. Specify the Data Mover to abort a task from the destination side. Specify the Data Mover and remote system name or remote system id along with the task id.

-delete id <taskId>  
Based on the task ID, deletes a completed task from the database on the Control Station.

[-remote_system {<remoteSystemName> | id=<id>}]  
Deletes a task that was initiated on a remote VNX. Specify the remote system name or remote system id along with the task id.

SEE ALSO  
Using VNX Replicator, nas_copy, nas_replicate, and nas_cel.

EXAMPLE #1  
To display detailed information about the task with taskID 4241, type:

$ nas_task -info 4241

Task Id = 4241
Celerra Network Server = cs100
The nas Commands

Task State             = Running
Percent Complete       = 95
Description            = Create Replication ufs1_replica1.
Originator             = nasadmin@cli.localhost
Start Time             = Mon Dec 17 14:21:35 EST 2007
Estimated End Time     = Mon Dec 17 19:24:21 EST 2007
Schedule               = n/a

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Id</td>
<td>Globally unique character string used as the identifier of the task.</td>
</tr>
<tr>
<td>VNX</td>
<td>When set, local.</td>
</tr>
<tr>
<td>Remote Task Id</td>
<td>When set, identifies a remote task.</td>
</tr>
<tr>
<td>State</td>
<td>Running, Recovering, Completed, or Failed. Running could be a combination of completed and failed.</td>
</tr>
<tr>
<td>Current Activity</td>
<td>Displays state property when available.</td>
</tr>
<tr>
<td>Percent Completed</td>
<td>Appears only when set and not complete.</td>
</tr>
<tr>
<td>Description</td>
<td>Appears if details are set.</td>
</tr>
<tr>
<td>Originator</td>
<td>User or host that initiated the task.</td>
</tr>
<tr>
<td>Start Time/End Time</td>
<td>The starting time and ending time (or status) for the task.</td>
</tr>
<tr>
<td>Estimated End Time</td>
<td>Appears instead of previous line when available and task is incomplete.</td>
</tr>
<tr>
<td>Schedule</td>
<td>The schedule in effect, or n/a for a task that is not a scheduled checkpoint refresh.</td>
</tr>
<tr>
<td>Response Statuses</td>
<td>Displayed list of messages, if any. A completed task should always have one.</td>
</tr>
</tbody>
</table>

**EXAMPLE #2**

To display the list of all tasks, type:

```
$ nas_task -list
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Task State</th>
<th>Originator</th>
<th>Start Time</th>
<th>escription</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>4241</td>
<td>Running</td>
<td>nasadmin@cli+</td>
<td>Mon Dec 17 14:21:35 EST 2007</td>
<td>Create Replication ufs1_r+</td>
<td>cs100</td>
</tr>
<tr>
<td>4228</td>
<td>Succeeded</td>
<td>nasadmin@cli+</td>
<td>Mon Dec 17 14:04:02 EST 2007</td>
<td>Delete task NONE: 4214.</td>
<td>cs100</td>
</tr>
<tr>
<td>4177</td>
<td>Failed</td>
<td>nasadmin@cli+</td>
<td>Mon Dec 17 13:59:26 EST 2007</td>
<td>Create Replication ufs1_r+</td>
<td>cs100</td>
</tr>
<tr>
<td>4150</td>
<td>Succeeded</td>
<td>nasadmin@cli+</td>
<td>Mon Dec 17 13:55:39 EST 2007</td>
<td>Delete task NONE: 4136.</td>
<td>cs100</td>
</tr>
<tr>
<td>4127</td>
<td>Succeeded</td>
<td>nasadmin@cli+</td>
<td>Mon Dec 17 11:38:32 EST 2007</td>
<td>Delete task NONE: 4113.</td>
<td>cs100</td>
</tr>
<tr>
<td>4103</td>
<td>Succeeded</td>
<td>nasadmin@cli+</td>
<td>Mon Dec 17 11:21:00 EST 2007</td>
<td>Delete task NONE: 4098.</td>
<td>cs100</td>
</tr>
<tr>
<td>4058</td>
<td>Succeeded</td>
<td>nasadmin@cli+</td>
<td>Fri Dec 14 16:43:23 EST 2007</td>
<td>Switchover Replication NONE.</td>
<td>cs100</td>
</tr>
<tr>
<td>2277</td>
<td>Succeeded</td>
<td>nasadmin@cli+</td>
<td>Fri Dec 14 16:42:08 EST 2007</td>
<td>Reverse Replication NONE.</td>
<td>cs110</td>
</tr>
<tr>
<td>2270</td>
<td>Succeeded</td>
<td>nasadmin@cli+</td>
<td>Fri Dec 14 16:40:29 EST 2007</td>
<td>Start Replication NONE.</td>
<td>cs110</td>
</tr>
</tbody>
</table>
EXAMPLE #1 provides a description of the outputs.

EXAMPLE #3 To abort task 4267 running locally on server_3, type:
$ nas_task -abort 4267 -mover server_3
OK

EXAMPLE #4 To delete the existing task 4267, type:
$ nas_task -delete 4267
OK
**nas_version**

Displays the software version running on the Control Station.

---

**SYNOPSIS**

```bash
nas_version
[-h| -l]
```

**DESCRIPTION**

`nas_version` displays the Control Station version in long form or short form. When used during a software upgrade, informs the user about the upgrade in progress.

**OPTIONS**

No arguments

- **-h**
  Displays command usage.

- **-l**
  Displays detailed software version information for the Control Station.

**EXAMPLE #1**

To display the software version running on the Control Station during a software upgrade, type:

```
$ nas_version
```

5.6.25-0

**EXAMPLE #2**

To display the system output during a software upgrade, type:

```
$ nas_version
```

5.6.19-0

Warning!! Upgrade is in progress from 5.6.19-0 to 5.6.20-0
Warning!! Please log off IMMEDIATELY if you are not upgrading the Celerra

**EXAMPLE #3**

To display the usage for nas_version, type:

```
$ nas_version -h
```

usage: /nas/bin/nas_version [-h|-l]
- h help
- l long_format

**EXAMPLE #4**

To display detailed software version information for the Control Station, type:

```
$ nas_version -l
```

<table>
<thead>
<tr>
<th>Name</th>
<th>emcnas</th>
<th>Relocations: /nas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>5.6.19</td>
<td>Vendor: EMC</td>
</tr>
<tr>
<td>Release</td>
<td>0</td>
<td>Build Date: Tue 19 Dec 2006 08:53:31 PM EST</td>
</tr>
<tr>
<td>Size</td>
<td>454239545</td>
<td>License: EMC Copyright</td>
</tr>
</tbody>
</table>
To display detailed software version information for the Control Station during a software upgrade, type:

$ nas_version -l

Name : emcnas
Version : 5.6.19
Release : 0
Size : 500815102
Signature : (none)
Packager : EMC Corporation
URL : http://www.emc.com
Summary : EMC nfs base install
Description : EMC nfs base install
Warning!!Upgrade is in progress from 5.6.19-0 to 5.6.20-0
Warning!!Please log off IMMEDIATELY if you are not upgrading the Celerra
nas_volume

Manages the volume table.

SYNOPSIS

nas_volume
   -list
   -delete <volume_name>
   -info [-size] {-all|<volume_name>} [-tree]
   -rename <old_name> <new_name>
   -size <volume_name>
   -acl <acl_value> <volume_name>
   -xtend <volume_name> {<volume_name>,...}
   [ -name <name> ] -create [-Stripe [<stripe_size>] | -Meta]
      [ -Force ] {<volume_name>,...}
   -Clone <volume_name> [{<svol>:<dvol>,...}][-option <options>]

DESCRIPTION

nas_volume creates metavolumes and stripe volumes and lists, renames, extends, clones, and deletes metavolumes, stripe, and slice volumes. nas_volume sets an access control value for a volume, and displays detailed volume attributes, including the total size of the volume configuration.

OPTIONS

-list
Displays the volume table.

Note: The ID of the object is an integer and is assigned automatically. The name of the volume may be truncated if it is more than 17 characters. To display the full name, use the -info option with the volume ID.

-delete <volume_name>
Deletes the specified volume.

-info [-size] {-all|<volume_name>} [-tree]
Displays attributes and the size for all volumes, or the specified <volume_name>. The -tree option recursively displays the volume set, that is, the list of component volumes for the specified volume or all volumes.

-rename <old_name> <new_name>
Changes the current name of a volume to a new name.

-size <volume_name>
Displays the total size in MB of the <volume_name>, including used and available space.
The nas Commands

- **acl** <acl_value> <volume_name>
  Sets an access control level value that defines the owner of the volume, and the level of access allowed for users and groups defined in the access control level table. The `nas_acl` command provides information.

- **xtend** <volume_name> {<volume_name>,...}
  Extends the specified metavolume by adding volumes to the configuration. The total size of the metavolume increases by the sum of all the volumes added.

  **Note:** Only metavolumes can be extended. The volume that was added remains in use until the original metavolume is deleted. Volumes containing mounted file systems cannot be extended using this option. The `nas_fs` command provides information to extend a volume that is hosting a mounted file system.

- **create** {<volume_name>,...}
  Creates a volume configuration from the specified volumes. Unless otherwise specified, volumes are automatically created as metavolumes.

    [-name <name>]
    Assigns a <name> to volume. If a name is not specified, one is assigned automatically. The name of a volume is case-sensitive.

    [-Stripe <stripe_size>|-Meta]
    Sets the type for the volume to be either a stripe volume or metavolume (default). If `-Stripe` is specified, a stripe size must be typed in multiples of 8192 bytes with a recommended size of 32,768 bytes (default) for NFS and CIFS environments. A 256 KB stripe size is recommended for EMC HighRoad® environments; while RAID 3 and 64 KB stripe size are recommended for ATA-based file systems.

    `nas_slice` provides information to create a slice volume.

    [-Force] {<volume_name>,...}
    Forces the creation of a volume on a mixed storage system.

- **Clone** <volume_name>
  Creates an exact clone of the specified <volume_name>. Volumes can be cloned from slice, stripe, or metavolumes. The name automatically assigned to the clone is derived from the ID of the volume.
The nas Commands

[{'svol': '<dvol>,...}]

Sets a specific disk volume set for the source volume and the destination volume. The size of the destination volume must be the same as the source volume.

- **option disktype=<type>**
  Specifies the type of disk to be created.

Disk types when using VNX for block are CLSTD, CLEFD, and CLATA, and for VNX for block involving mirrored disks are: CMEFD, CMSTD, and CMATA.

Disk types when using a Symmetrix are STD, R1STD, R2STD, BCV, R1BCV, R2BCV, ATA, R1ATA, R2ATA, BCVA, R1BCA, R2BCA, and EFD.

**SEE ALSO**


**EXAMPLE #1**

To list all volumes, type:

```
$ nas_volume -list
```

<table>
<thead>
<tr>
<th>id</th>
<th>inuse</th>
<th>type</th>
<th>acl</th>
<th>name</th>
<th>cltype</th>
<th>clid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>y</td>
<td>4</td>
<td>0</td>
<td>root_disk</td>
<td>0</td>
<td>1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51</td>
</tr>
<tr>
<td>2</td>
<td>y</td>
<td>4</td>
<td>0</td>
<td>root_ldisk</td>
<td>0</td>
<td>35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51</td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td>4</td>
<td>0</td>
<td>d3</td>
<td>1</td>
<td>76</td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td>4</td>
<td>0</td>
<td>d4</td>
<td>1</td>
<td>77</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>4</td>
<td>0</td>
<td>d5</td>
<td>1</td>
<td>78</td>
</tr>
<tr>
<td>6</td>
<td>y</td>
<td>4</td>
<td>0</td>
<td>d6</td>
<td>1</td>
<td>79</td>
</tr>
<tr>
<td>7</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>root_dos</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>n</td>
<td>1</td>
<td>0</td>
<td>root_layout</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>root_slice_1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>y</td>
<td>3</td>
<td>0</td>
<td>root_volume_1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>root_slice_2</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>y</td>
<td>3</td>
<td>0</td>
<td>root_volume_2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>root_slice_3</td>
<td>1</td>
<td>14</td>
</tr>
</tbody>
</table>

... 

**Note:** This is a partial listing due to the length of the outputs.
The nas Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the volume.</td>
</tr>
<tr>
<td>inuse</td>
<td>Whether the volume is used.</td>
</tr>
<tr>
<td>type</td>
<td>Type assigned to the volume. Available types are: 1=slice, 2=stripe, 3=meta, 4=disk, and 100=pool.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level assigned the volume.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to the volume.</td>
</tr>
</tbody>
</table>
| cltype  | The client type of the volume. Available values are:  
|         | • 0 — If the clid field is not empty then the client is a slice. 
|         | • 1 — The client is another volume (meta, stripe, volume_pool). 
|         | • 2 — The client is a file system.             |
| clid    | ID of the client.                              |

**EXAMPLE #2**  To create a metavolume named, mtv1, on disk volume, d7, type:

```bash
$ nas_volume -name mtv1 -create d7
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the volume.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to the volume.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value assigned to the volume.</td>
</tr>
<tr>
<td>in_use</td>
<td>Whether the volume is used.</td>
</tr>
<tr>
<td>type</td>
<td>Type assigned to the volume. Available types are: meta, stripe, slice, disk, and pool.</td>
</tr>
<tr>
<td>volume_set</td>
<td>Name assigned to the volume.</td>
</tr>
<tr>
<td>disks</td>
<td>Disks used to build a file system.</td>
</tr>
</tbody>
</table>

**EXAMPLE #3**  To display configuration information for mtv1, type:

```bash
$ nas_volume -info mtv1
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the volume.</td>
</tr>
<tr>
<td>name</td>
<td>Name assigned to the volume.</td>
</tr>
<tr>
<td>acl</td>
<td>Access control level value assigned to the volume.</td>
</tr>
<tr>
<td>in_use</td>
<td>Whether the volume is used.</td>
</tr>
<tr>
<td>type</td>
<td>Type assigned to the volume. Available types are: meta, stripe, slice, disk, and pool.</td>
</tr>
<tr>
<td>volume_set</td>
<td>Name assigned to the volume.</td>
</tr>
<tr>
<td>disks</td>
<td>Disks used to build a file system.</td>
</tr>
</tbody>
</table>
EXAMPLE #4  
To rename a mtv1 to mtv2, type:

$ nas_volume -rename mtv1 mtv2

id          = 146
name        = mtv2
acl         = 0
in_use      = False
type        = meta
volume_set  = d7
disks       = d7

EXAMPLE #5  
To create a stripe volume named, stv1, with a size of \(32768\) bytes on disk volumes d10, d12, d13, and d15, type:

$ nas_volume -name stv1 -create -Stripe 32768 d10,d12,d13,d15

id          = 147
name        = stv1
acl         = 0
in_use      = False
type        = stripe
stripe_size = 32768
volume_set  = d10,d12,d13,d15
disks       = d10,d12,d13,d15

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>stripe_size</td>
<td>Specified size of the stripe volume.</td>
</tr>
</tbody>
</table>

EXAMPLE #6  
To clone mtv1, type:

$ nas_volume -Clone mtv1

id          = 146
name        = mtv1
acl         = 0
in_use      = False
type        = meta
volume_set  = d7
disks       = d7

id          = 148
name        = v148
acl         = 0
in_use      = False
type        = meta
volume_set  = d8
disks       = d8
**EXAMPLE #7**
To clone the volume mtv1 and set the disk type to BCV, type:

```
$ /nas/sbin/rootnas_volume -Clone mtv1 -option
disktype=BCV
```

```
$ id          = 322
$ name        = mtv1
$ acl         = 0
$ in_use      = False
$ type        = meta
$ volume_set  = d87
$ disks       = d87
```

```
$ id          = 323
$ name        = v323
$ acl         = 0
$ in_use      = False
$ type        = meta
$ volume_set  = rootd99
$ disks       = rootd99
```

**EXAMPLE #8**
To extend mtv1 with mtv2, type:

```
$ nas_volume -xtend mtv1 mtv2
```

```
$ id          = 146
$ name        = mtv1
$ acl         = 0
$ in_use      = False
$ type        = meta
$ volume_set  = d7,mtv2
$ disks       = d7,d8
```

**EXAMPLE #9**
To display the size of mtv1, type:

```
$ nas_volume -size mtv1
```

```
$ total = 547418 avail = 547418 used = 0 ( 0% ) (sizes in MB)
```

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>Total size of the volume.</td>
</tr>
<tr>
<td>avail</td>
<td>Amount of unused space on the volume.</td>
</tr>
<tr>
<td>used</td>
<td>Amount of space used on the volume.</td>
</tr>
</tbody>
</table>
EXAMPLE #10  To set the access control level for the metavolume mtv1, type:

$ nas_volume -acl 1432 mtv1

id          = 125  
name        = mtv1  
acl         = 1432, owner=nasadmin, ID=201  
in_use      = False  
type        = meta  
volume_set  = d7,mtv2  
disks       = d7,d8

Note: The value 1432 specifies nasadmin as the owner and gives users with an access level of at least observer read access only, users with an access level of at least operator read/write access, and users with an access level of at least admin read/write/delete access.

EXAMPLE #11  To delete mtv2, type:

$ nas_volume -delete mtv1

id          = 146  
name        = mtv1  
acl         = 1432, owner=nasadmin, ID=201  
in_use      = False  
type        = meta  
volume_set  = d7,mtv2  
disks       = d7,d8
This chapter lists the VNX Command Set provided for managing, configuring, and monitoring Data Movers. The commands are prefixed with server and appear alphabetically. The command line syntax (Synopsis), a description of the options, and an example of usage are provided for each command.
server_archive

Reads and writes file archives, and copies directory hierarchies.

SYNOPSIS

server_archive <movername> [-cdnvN] -f <archive_file> [-J [p][w|d|u]]
[-I <client_dialect>]
[-e <archive_name>] [-s <replstr>] ...
[-T [<from_date>][,<to_date>]] [<pattern>] ...
server_archive <movername> -r [-cdiknuvDNYZ][-E <limit>]
[-J [w|d|u]][-C d|i|m][-I <client_dialect>]
[-f <file_name>][-e <archive_name>]
[-p <string>] ... [-s <replstr>] ...
[-T [<from_date>][,<to_date>]] [<pattern> ...]
server_archive <movername> -w [-dituvLNPX]
[-J [w|d|u]][-I <client_dialect>]
[-b <block_size>][-f <file_name>][-e <archive_name>]
[-x <format>][-B bytes][-s <replstr>] ...
[-T [<from_date>][,<to_date>]] /[c][m]] ... [-0][-1]<file> ...
server_archive -r -w [-dikintuvDLPXYZ]
[-J [w|d|u]][-C d|i|m]
[-p <string>] ... [-s <replstr>] ...
[-T [<from_date>][,<to_date>]] /[c][m]] ...<file> ...
<directory>

DESCRIPTION

server_archive reads, writes, and lists the members of an archive file, and copies directory hierarchies. The server_archive operation is independent of the specific archive format, and supports a variety of different archive formats.

Note: A list of supported archive formats can be found under the description of the -x option.

The presence of the -r and the -w options specifies the following functional modes: list, read, write, and copy.

-List (no arguments)
server_archive writes to standard output a table of contents of the members of the archive file read from archive, whose pathnames match the specified patterns.

Note: If no options are specified, server_archive lists the contents of the archive.
Read (-r)
**server_archive** extracts the members of the archive file read from the archive, with pathnames matching the specified patterns. The archive format and blocking are automatically determined on input. When an extracted file is a directory, the entire file hierarchy rooted at that directory is extracted.

**Note:** Ownership, access, and modification times, and file mode of the extracted files are discussed in more detail under the -p option.

Write (-w)
**server_archive** writes an archive containing the file operands to archive using the specified archive format. When a file operand is also a directory, the entire file hierarchy rooted at that directory is included.

Copy (-r -w)
**server_archive** copies the file operands to the destination directory. When a file operand is also a directory, the entire file hierarchy rooted at that directory is included. The effect of the copy is as if the copied files were written to an archive file and then subsequently extracted, except that there may be hard links between the original and the copied files. The -l option provides more information.

**CAUTION**
The destination directory must exist and must not be one of the file operands or a member of a file hierarchy rooted at one of the file operands. The result of a copy under these conditions is unpredictable.

While processing a damaged archive during a read or list operation, **server_archive** attempts to recover from media defects and searches through the archive to locate and process the largest number of archive members possible (the -E option provides more details on error handling).

**OPERANDS**
The directory operand specifies a destination directory pathname. If the directory operand does not exist, or it is not writable by the user, or it is not a directory name, **server_archive** exits with a non-zero exit status.

The pattern operand is used to select one or more pathnames of archive members. Archive members are selected using the pattern matching notation described by **fnmatch 3**. When the pattern...
operand is not supplied, all members of the archive are selected. When a pattern matches a directory, the entire file hierarchy rooted at that directory is selected. When a pattern operand does not select at least one archive member, `server_archive` writes these pattern operands in a diagnostic message to standard error and then exits with a non-zero exit status.

The file operand specifies the pathname of a file to be copied or archived. When a file operand does not select at least one archive member, `server_archive` writes these file operand pathnames in a diagnostic message to standard error and then exits with a non-zero exit status.

The archive_file operand is the name of a file where the data is stored (write) or read (read/list). The archive_name is the name of the streamer on which the data will be stored (write) or read (read/list).

**Note:** To obtain the device name, you can use `server_devconfig -scsi`.

**OPTIONS**

The following options are supported:

- **-r**
  Reads an archive file from archive and extracts the specified files. If any intermediate directories are needed to extract an archive member, these directories will be created as if `mkdir 2` was called with the bit-wise inclusive OR of S_IRWXU, S_IRWXG, and S_IRWXO, as the mode argument. When the selected archive format supports the specification of linked files and these files cannot be linked while the archive is being extracted, `server_archive` writes a diagnostic message to standard error and exits with a non-zero exit status at the completion of operation.

- **-w**
  Writes files to the archive in the specified archive format.

- **-0**
  With this option, a full referenced backup is performed with the time and date of launching put in a reference file. This reference file is an ASCII file and is located in `/etc/BackupDates`. The backup is referenced by the pathname of the files to back up and the time and date when the backup was created. This file is updated only if the backup is successful.

Backup files can be copied using the `server_file` command.
- <x>
Level x (x=1–9) indicates a backup of all files in a file system that have been modified since the last backup of a level smaller than the previous backup.

For example, a backup is performed for:

- Monday: level 0 = full backup
- Tuesday: level 3 = files modified since Monday
- Friday: level 5 = files modified since Tuesday
- Saturday: level 4 = files modified since Tuesday
- Sunday: level 4 = files modified since Tuesday

**Note:** If the backup type is not indicated, a full backup is performed automatically.

- b <block_size>
When writing an archive, blocks the output at a positive decimal integer number of bytes per write to the archive file. The <block_size> must be a multiple of 512 bytes with a maximum size of 40 kilobytes.

**Note:** To remain POSIX-compatible, do not exceed 32256 Bytes.

A <block_size> can end with k or b to specify multiplication by 1024 (1K) or 512, respectively. A pair of <block_size> can be separated by x to indicate a product. A specific archive device may impose additional restrictions on the size of blocking it will support. When blocking is not specified, the default for <block_size> is dependent on the specific archive format being used. The -x option provides more information.

- c
Matches all file or archive members except those specified by the pattern and file operands.

- d
Causes files of type directory being copied or archived, or archive members of type directory being extracted, to match only the directory file or archive member, and not the file hierarchy rooted at the directory.

- e <archive_name>
Specifies the archive name when it is streamed.
Note: To prevent the tape from rewinding at the end of command execution, use the -N option with the -e <archive_name> option.

-f <archive_file>
Specifies the archive name when it is a file.

Note: A single archive may span multiple files and different archive devices. When required, server_archive prompts for the pathname of the file or device of the next volume in the archive.

-i
Interactively renames files or archive members. For each archive member matching a pattern operand, or each file matching a file operand, server_archive prompts to /dev/tty giving the name of the file, its file mode, and its modification time. Then server_archive reads a line from /dev/tty. If this line is blank, the file or archive member is skipped. If this line consists of a single period, the file or archive member is processed with no modification to its name. Otherwise, its name is replaced with the contents of the line. Then server_archive immediately exits with a non-zero exit status if <EOF> is encountered when reading a response, or if /dev/tty cannot be opened for reading and writing.

-k
Does not allow overwriting existing files.

-l
Links files. In the copy mode (-r, -w), hard links are made between the source and destination file hierarchies whenever possible.

-I <client_dialect>
Allows filename information recovered from an archive to be translated into UTF-8.

-n
Selects the first archive member that matches each pattern operand. No more than one archive member is matched for each pattern. When members of type directory are matched, the file hierarchy rooted at that directory is also matched (unless -d is also specified).

-p <string>
Specifies one or more file characteristic options (privileges). The <string> option-argument is a string specifying file characteristics to be retained or discarded on extraction. The string consists of the specification characters a, e, m, o, and p. Multiple characteristics can
be concatenated within the same string and multiple -p options can be specified. The meaning of the specification characters is as follows:

- **a**
  Do not preserve file access times. By default, file access times are preserved whenever possible.

- **e**
  Preserve everything (default mode), the user ID, group ID, file mode bits, file access time, and file modification time.

  __Note:__ The e flag is the sum of the o and p flags.

- **m**
  Do not preserve file modification times. By default, file modification times are preserved whenever possible.

- **o**
  Preserve the user ID and group ID.

- **p**
  Preserve the file mode bits. This specification character is intended for a user with regular privileges who wants to preserve all aspects of the file other than the ownership. The file times are preserved by default, but two other flags are offered to disable this and use the time of extraction instead.

In the preceding list, preserve indicates that an attribute stored in the archive is given to the extracted file, subject to the permissions of the invoking process. Otherwise, the attribute of the extracted file is determined as part of the normal file creation action. If neither the e nor the o specification character is specified, or the user ID and group ID are not preserved for any reason, `server_archive` will not set the S_ISUID (setuid) and S_ISGID (setgid) bits of the file mode. If the preservation of any of these items fails for any reason, `server_archive` writes a diagnostic message to standard error.

  __Note:__ Failure to preserve these items will affect the final exit status, but will not cause the extracted file to be deleted.

If the file characteristic letters in any of the string option-arguments are duplicated, or in conflict with one another, the ones given last will take precedence. For example, if you specify `-p eme`, file modification times are still preserved.
-s <replstr>
Modifies the file or archive member names specified by the pattern or
<file> operand according to the substitution expression <replstr>
using the syntax of the ed utility regular expressions.

Note: The ed 1 manual page provides information.

Multiple -s expressions can be specified. The expressions are applied
in the order they are specified on the command line, terminating with
the first successful substitution. The optional trailing g continues to
apply the substitution expression to the pathname substring, which
starts with the first character following the end of the last successful
substitution.

The optional trailing p causes the final result of a successful
substitution to be written to standard error in the following format:

    <original pathname> >> <new pathname>

File or archive member names that substitute the empty string are not
selected and are skipped.

-t
Resets the access times of any file or directory read or accessed by
server_archive to be the same as they were before being read or
accessed by server_archive.

-u
Ignores files that are older (having a less recent file modification time)
than a pre-existing file, or archive member with the same name.
During read, an archive member with the same name as a file in a file
system is extracted if the archive member is newer than the file.
During copy, the file in the destination hierarchy is replaced by the
file in the source hierarchy, or by a link to the file in the source
hierarchy if the file in the source hierarchy is newer.
The server Commands

-v
During a list operation, produces a verbose table of contents using the format of the ls 1 utility with the -l option. For pathnames representing a hard link to a previous member of the archive, the output has the format:

<ls -l listing> == <link name>

For pathnames representing a symbolic link, the output has the format:

<ls -l listing> => <link name>

where <ls -l listing> is the output format specified by the ls 1 utility when used with the -l option. Otherwise, for all the other operational modes (read, write, and copy), pathnames are written and flushed a standard error without a trailing <newline> as soon as processing begins on that file or archive member. The trailing <newline> is not buffered, and is written only after the file has been read or written.

-x format
Specifies the output archive format, with the default format being ustar. The server_archive command currently supports the following formats:

- cpio
  The extended cpio interchange format specified in the -p1003.2 standard. The default blocksize for this format is 5120 bytes. Inode and device information about a file (used for detecting file hard links by this format) which may be truncated by this format is detected by server_archive and is repaired.

  Note: To be readable by server_archive, the archive must be built on another machine with the option -c (write header information in ASCII).

- bcpio
  The old binary cpio format. The default blocksize for this format is 5120 bytes.

  Note: This format is not very portable and should not be used when other formats are available.

  Inode and device information about a file (used for detecting file hard links by this format) which may be truncated by this format is detected by server_archive and is repaired.
sv4cpio
The System V release 4 cpio. The default blocksize for this format is 5120 bytes. Inode and device information about a file (used for detecting file hard links by this format) which may be truncated by this format is detected by server_archive and is repaired.

sv4crc
The System V release 4 cpio with file crc checksums. The default blocksize for this format is 5120 bytes. Inode and device information about a file (used for detecting file hard links by this format) which may be truncated by this format is detected by server_archive and is repaired.

tar
The old BSD tar format as found in BSD4.3. The default blocksize for this format is 10240 bytes. Pathnames stored by this format must be 100 characters or less in length. Only regular files, hard links, soft links, and directories will be archived (other file system types are not supported).

ustar
The extended tar interchange format specified in the -p1003.2 standard. The default blocksize for this format is 10240 bytes.

---

Note: Pathnames stored by this format must be 250 characters or less in length (150 for basename and 100 for <file_name>).

---

emctar
This format is not compatible with -p1003.2 standard. It allows archiving to a file greater than 8 GB. Pathnames stored by this format are limited to 3070 characters. The other features of this format are the same as ustar.

server_archive detects and reports any file that it is unable to store or extract as the result of any specific archive format restrictions. The individual archive formats may impose additional restrictions on use.

---

Note: Typical archive format restrictions include (but are not limited to) file pathname length, file size, link pathname length, and the type of the file.

---

-B bytes
Limits the number of bytes written to a single archive volume to bytes. The bytes limit can end with m, k, or b to specify multiplication by 1048576 (1M), 1024 (1K) or 512, respectively. A pair of bytes limits can be separated by x to indicate a product.
Note: The limit size will be rounded up to the nearest block size.

-C [d i m]
When performing a restore, this allows you to choose PAX behaviors on CIFS collision names.

  d: delete
  i: ignore
  m: mangle

-D
Ignores files that have a less recent file inode change time than a pre-existing file, or archive member with the same name. The -u option provides information.

Note: This option is the same as the -u option, except that the file inode change time is checked instead of the file modification time. The file inode change time can be used to select files whose inode information (such as uid, gid, and so on) is newer than a copy of the file in the destination directory.

-E limit
Has the following two goals:

- In case of medium error, to limit the number of consecutive read faults while trying to read a flawed archive to limit. With a positive limit, server_archive attempts to recover from an archive read error and will continue processing starting with the next file stored in the archive. A limit of 0 (zero) will cause server_archive to stop operation after the first read error is detected on an archive volume. A limit of "NONE" will cause server_archive to attempt to recover from read errors forever.

- In case of no medium error, to limit the number of consecutive valid header searches when an invalid format detection occurs. With a positive value, server_archive will attempt to recover from an invalid format detection and will continue processing starting with the next file stored in the archive. A limit of 0 (zero) will cause server_archive to stop operation after the first invalid header is detected on an archive volume. A limit of "NONE" will cause server_archive to attempt to recover from invalid format errors forever. The default limit is 10 retries.
**CAUTION**

Using this option with NONE requires extreme caution as server_archive may get stuck in an infinite loop on a badly flawed archive.

- **J**
  Backs up, restores, or displays CIFS extended attributes.
  
  **p:** displays the full path name for alternate names (for listing and archive only)
  **u:** specifies UNIX name for pattern search
  **w:** specifies M256 name for pattern search
  **d:** specifies M83 name for pattern search

- **L**
  Follows all symbolic links to perform a logical file system traversal.

- **N**
  Used with the `-e archive_name` option, prevents the tape from rewinding at the end of command execution.

- **P**
  Does not follow symbolic links.

**Note:** Performs a physical file system traversal. This is the default mode.

- **T** `[from_date] [, to_date] [/c] [/m]`
  Allows files to be selected based on a file modification or inode change time falling within a specified time range of from_date to to_date (the dates are inclusive). If only a from_date is supplied, all files with a modification or inode change time equal to or less than are selected. If only a to_date is supplied, all files with a modification or inode change time equal to or greater than will be selected. When the from_date is equal to the to_date, only files with a modification or inode change time of exactly that time will be selected.

When `server_archive` is in the write or copy mode, the optional trailing field `[c][m]` can be used to determine which file time (inode change, file modification or both) is used in the comparison. If neither is specified, the default is to use file modification time only. The `m` specifies the comparison of file modification time (the time when the file was last written). The `c` specifies the comparison of inode change time (the time when the file inode was last changed; for example, a change of owner, group, mode, and so on). When `c` and `m` are both
specified, then the modification and inode change times are both compared. The inode change time comparison is useful in selecting files whose attributes were recently changed, or selecting files which were recently created and had their modification time reset to an older time (as what happens when a file is extracted from an archive and the modification time is preserved). Time comparisons using both file times are useful when `server_archive` is used to create a time-based incremental archive (only files that were changed during a specified time range will be archived).

A time range is made up of six different fields and each field must contain two digits. The format is:

```
```

Where yy is the last two digits of the year, the first mm is the month (from 01 to 12), dd is the day of the month (from 01 to 31), hh is the hour of the day (from 00 to 23), the second mm is the minute (from 00 to 59), and ss is seconds (from 00 to 59). The minute field mm is required, while the other fields are optional, and must be added in the following order: hh, dd, mm, yy. The ss field may be added independently of the other fields. Time ranges are relative to the current time, so `-T 1234/cm` selects all files with a modification or inode change time of 12:34 P.M. today or later. Multiple `-T` time range can be supplied, and checking stops with the first match.

**-X**

When traversing the file hierarchy specified by a pathname, does not allow descending into directories that have a different device ID. The `st_dev` field as described in `stat` for more information about device IDs.

**-Y**

Ignores files that have a less recent file inode change time than a pre-existing file, or archive member with the same name.

**Note:** This option is the same as the `-D` option, except that the inode change time is checked using the pathname created after all the filename modifications have completed.

**-Z**

Ignores files that are older (having a less recent file modification time) than a pre-existing file, or archive member with the same name.
The server Commands

Note: This option is the same as the -u option, except that the modification time is checked using the pathname created after all the filename modifications have completed.

The options that operate on the names of files or archive members (-c, -i, -n, -s, -u, -v, -D, -T, -Y, and -Z) interact as follows.

When extracting files during a read operation, archive members are selected, based only on the user-specified pattern operands as modified by the -c, -n, -u, -D, and -T options. Then any -s and -i options will modify, in that order, the names of those selected files. Then the -Y and -Z options will be applied based on the final pathname. Finally, the -v option will write the names resulting from these modifications.

When archiving files during a write operation, or copying files during a copy operation, archive members are selected, based only on the user specified pathnames as modified by the -n, -u, -D, and -T options (the -D option applies only during a copy operation). Then any -s and -i options will modify, in that order, the names of these selected files. Then during a copy operation, the -Y and the -Z options will be applied based on the final pathname. Finally, the -v option will write the names resulting from these modifications.

When one or both of the -u or -D options are specified along with the -n option, a file is not considered selected unless it is newer than the file to which it is compared.

SEE ALSO  Using the server_archive Utility on VNX.

EXAMPLE #1  To archive the contents of the root directory to the device rst0, type:

```bash
$ server_archive <movername> -w -e rst0
```

EXAMPLE #2  To display the verbose table of contents for an archive stored in <file_name>, type:

```bash
$ server_archive <movername> -v -f <file_name>
```

EXAMPLE #3  To copy the entire olddir directory hierarchy to newdir, type:

```bash
$ server_archive <movername> -rw <olddir newdir>
```

EXAMPLE #4  To interactively select the files to copy from the current directory to dest_dir, type:

```bash
$ server_archive <movername> -rw -i <olddir dest_dir>
```
EXAMPLE #5  To extract all files from the archive stored in <file_name>, type:

$ server_archive <movername> -r -f <file_name>

EXAMPLE #6  To update (and list) only those files in the destination directory /backup that are older (less recent inode change or file modification times) than files with the same name found in the source file tree home, type:

$ server_archive <movername> -r -w -v -Y -Z home /backup

STANDARDS  The server_archive utility is a superset of the -p1003.2 standard.

Note: The archive formats bcpio, sv4cpio, sv4crc, and tar, and the flawed archive handling during list and read operations are extensions to the POSIX standard.

ERRORS  The server_archive command exits with one of the following system messages:

   All files were processed successfully.
   or
   An error occurred.

Whenever server_archive cannot create a file or a link when reading an archive, or cannot find a file when writing an archive, or cannot preserve the user ID, group ID, or file mode when the -p option is specified, a diagnostic message is written to standard error, and a non-zero exit status is returned. However, processing continues.

In the case where server_archive cannot create a link to a file, this command will not create a second copy of the file.

If the extraction of a file from an archive is prematurely terminated by a signal or error, server_archive may have only partially extracted a file the user wanted. Additionally, the file modes of extracted files and directories may have incorrect file bits, and the modification and access times may be wrong.

If the creation of an archive is prematurely terminated by a signal or error, server_archive may have only partially created the archive which may violate the specific archive format specification.

If while doing a copy, server_archive detects a file is about to overwrite itself, the file is not copied, a diagnostic message is written to standard error and when server_archive completes, it exits with a non-zero exit status.
server_arp

Manages the Address Resolution Protocol (ARP) table for Data Movers.

**SYNOPSIS**

```
server_arp {<movername>|ALL} <ip_addr> | -all | -delete <ip_addr> | -set <ip_addr> <physaddr>
```

**DESCRIPTION**

`server_arp` displays and modifies the IP-to-MAC address translation tables used by the ARP for the specified Data Mover.

The **ALL** option executes the command for all Data Movers.

**OPTIONS**

- `<ip_addr>`
  Displays the ARP entry for the specified IP address.

- `-all`
  Displays the first 64 of the current ARP entries.

- `-delete <ip_addr>`
  Deletes an ARP entry.

- `-set <ip_addr> <physaddr>`
  Creates an ARP entry with an IP address and physical address.

**EXAMPLE #1**

To create an ARP entry, type:

```
$ server_arp server_2 -set 172.24.102.20 00:D0:B7:82:98:E0
server_2 : added: 172.24.102.20 at 0:d0:b7:82:98:e0
```

**EXAMPLE #2**

To display all ARP entries for a specified Data Mover, type:

```
$ server_arp server_2 -all
server_2 :
172.24.102.254 at 0:d0:3:f9:37:fc
172.24.102.20 at 0:d0:b7:82:98:e0
172.24.102.24 at 0:50:56:8e:1d:5
128.221.253.100 at 0:4:23:a7:b1:35
```

**EXAMPLE #3**

To display an ARP entry specified by IP address, type:

```
$ server_arp server_2 172.24.102.20
```

```
server_2 : 172.24.102.20 at 0:d0:b7:82:98:e0
```
EXAMPLE #4  To delete an ARP entry, type:

    $ server_arp server_2 -delete 172.24.102.24
server_2 : deleted: 172.24.102.24 at 0:50:56:8e:1d:5
server_certificate

Manages VNX for file system’s Public Key Infrastructure (PKI) for the specified Data Movers.

SYNOPSIS

server_certificate {<movername>|ALL}
    -ca_certificate
        [-list]
        -info {-all|<certificate_id>}
        -import [-filename <path>]
        -delete {-all|<certificate_id>}
    -persona
        [-list]
        -info {-all|<persona_name>|id=<persona_id>}
        -generate {<persona_name>|id=<persona_id>} -key_size {2048|4096}
            [-cs_sign_duration <# of months>]
            [-cn|-common_name] <common_name>[;<common_name>]
            [-ou <org_unit>[;<org_unit>]]
            [-organization <organization>]
            [-location <location>]
            [-state <state>]
            [-country <country>]
            [-filename <output_path>]
        -clear {<persona_name>|id=<persona_id>} {-next|-current|-both}
        -import {<persona_name>|id=<persona_id>} [-filename <path>]

DESCRIPTION

server_certificate manages the use of public key certificates between Data Movers acting as either clients or servers. server_certificate -ca_certificate manages the Certificate Authority (CA) certificates the VNX uses to confirm a server’s identity when the Data Mover is acting as a client. server_certificate -persona manages the certificates presented by the Data Mover to a client application when the Data Mover is acting as a server as well as the certificates presented by the Data Mover to a server configured to require client authentication.

OPTIONS

-ca_certificate
Lists the CA certificates currently available on the VNX. The output from this command is identical to the output from the -list option.

-ca_certificate -list
Lists the CA certificates currently available on the VNX.

-ca_certificate -info {-all|<certificate_id>}
Displays the properties of a specified CA certificate or all CA certificates.
The server Commands

-ca_certificate -import [-filename <path>]
Imports a CA certificate. You can only paste text in PEM format at the command prompt. Specify -filename and provide a path to import a CA certificate in either DER or PEM format.

-ca_certificate -delete {-all | <certificate_id>}
Deletes a specified CA certificate or all CA certificates.

-persona
Lists the key sets and associated certificates currently available on the VNX. The output from this command is identical to the output from the -list option.

-persona -list
Lists the key sets and associated certificates currently available on the VNX.

-persona -info {-all | <persona_name> | id=<persona_id>}
Displays the properties of the key sets and associated certificates, including the text of a pending certificate request, of a specified persona or all personas.

-persona -generate {<persona_name> | id=<persona_id>}
-key_size <bits> { -cn | -common_name } <common_name> [ ; <common_name> ]
Generates a public/private key set along with a request to sign the certificate. Specify either the persona name or ID. The ID is automatically generated when the persona is created. You can determine the ID using the -list or -info options. The key size can be either 2048 or 4096 bits. Use either -cn or -common_name to specify the commonly used name. The common name is typically a hostname that describes the Data Mover with which the persona is associated. Multiple common names are allowed but must be separated by semicolons.

[ -cs_sign_duration # of months ]
Specifies the number of months the certificate is valid. A month is defined as 30 days. This option is valid only if the certificate will be signed by the Control Station. If this option is specified, you cannot save the request to a file using the -filename option.

[-ou <org_unit>[ ; <org_unit>]]
Identifies the organizational unit. Multiple organizational units are allowed but must be separated by semicolons.
The server Commands

[-organization <organization>]
Identifies the organization.

[-location <location>]
Identifies the physical location of the organizational unit.

[-state <state>]
Identifies the state where the organizational unit is located.

[-country <country>]
Identifies the country where the organization unit is located. This value is limited to two characters.

[-filename <output_path>]
Provides a path to where the request should be saved to a file. This option is valid only if the certificate will be signed by an external CA. If this option is specified, you cannot specify the number of months the certificate is valid using the -cs_sign_duration option.

-persona -clear {<persona_name>|id=<persona_id>}
{-next|-current|-both}
Deletes a key set and the associated certificate. You can delete the current key set and certificate, the next key set and certificate, or both.

-persona -import {<persona_name>|id=<persona_id>}
[-filename <path>]
Imports a CA-signed certificate. You can only paste text in PEM format at the command prompt. Specify -filename and provide a path to import a CA-signed certificate in either DER or PEM format.

SEE ALSO
nas_ca_certificate.

EXAMPLE #1
To import a CA certificate, specifying a filename and path, type:

$ server_certificate server_2 -ca_certificate -import
-filename "/tmp/ca_cert.pem"

done
EXAMPLE #2  To list all the CA certificates currently available on the VNX, type:

$ server_certificate ALL -ca_certificate -list

server_2 :
  id=1
  subject=O=Celerra Certificate Authority;CN=sorento
  issuer=O=Celerra Certificate Authority;CN=sorento
  expire=20120318032639Z
  id=2
  subject=O=Celerra Certificate Authority;CN=sorento
  issuer=O=Celerra Certificate Authority;CN=sorento
  expire=20120318032639Z

EXAMPLE #3  To list the properties of the CA certificate identified by certificate ID 2, type:

$ server_certificate server_2 -ca_certificate -info 2

server_2 :
  id=2
  subject = C=US;O=VeriSign, Inc.;OU=Class 3 Public Primary Certification Authority
  issuer = C=US;O=VeriSign, Inc.;OU=Class 3 Public Primary Certification Authority
  start   = 19960129000000Z
  expire  = 20280801235959Z
  signature alg. = md2WithRSAEncryption
  public key alg. = rsaEncryption
  public key size = 1024 bits
  serial number   = 70ba e41d 10d9 2934 b638 ca7b 03cc babf
  version         = 1

EXAMPLE #4  To generate a key set and certificate request to be sent to an external CA for the persona identified by the persona name default, type:

$ server_certificate server_2 -persona -generate default
-key_size 2048 -common_name division.xyz.com

server_2 :
  Starting key generation. This could take a long time ...
  done

EXAMPLE #5  To list all the key sets and associated certificates currently available on the VNX, type:

$ server_certificate ALL -persona -list

server_2 :
  id=1
  name=default
  next state=Request Pending
  request subject=CN=name;CN=1.2.3.4
The server Commands

server_3 :
id=1
name=default
next state=Not Available
CURRENT CERTIFICATE:
id=1
subject=CN=test;CN=1.2.3.4
expire=20070706183824Z
issuer=O=Celerra Certificate Authority;CN=eng173100

EXAMPLE #6 To list the properties of the key set and certificate identified by persona ID 1, type:

$ server_certificate server_2 -persona -info id=1

server_2 :
id=1
name=default
next state=Request Pending
Request subject=CN=name;CN=1.2.3.4
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIEZjCCAk4CAQAwITENMAsGA1UEAxMEbmFtZTEQMA4GA1UEAxMHMS4yLjMuNDCC
AiIwDQYJKoZIhvcNAQEBBQADgggABoBB7NaIcJzR7TZBfI2ztjuT1v87O76w67
/njUr2KbfWd81CC+ngUi33hbMuBR3FpsQNASYZUZg15+JexALH5jhBahd2ARKXBag
itQLhvxyKY0dEqEIGDfDdec7x+iyro2gbNxhrLdtkBuKrmOnuT5g2WXNZkNa/H7
KWv8JSwCvlw1WN/7Wn9aEBdizBBfer+ZdMPkGLbyb/EXV2nHABeWh3iKC6/cenRd
4K7k0F9F9gXVH1zzTeYvagMBAAGAqADANBgkqhkiG9w0BAQUFAAACAgEazSS4f
yf2Wn0mZ0LgsSBcVHPVEy+v/p/Au9INm9KDJP40K41UDU8tOGy09kC8E0k1Bum59
fyjt2T/3Rq5gvevCCEJhsVW3ZMnsyjDo6Ruc0HmuY4q+tuLL+dilsQnZGUxt8asw
d1hPeDzExA6o9cfmVZMst5QicfAmnBNr4Ba096+VA1g59fu/ChUlpvKWWMGXZi42s
7z+UdMBY04pEfjGli34Qof/z4K0SVN1CN3CEkw5TlSst8qA/E2JXX1LhbMYKWyYu
9/gsrHuwz1XMZx4SmTk9/RsEl1vy7fBztIoN8myFh0Nna849pypqls/yvhXZ/D
iDF6Tgky4RbNzuanRBSyijF4Tnpj/LJk8v3z3yFJ+3Dk0c8oz1BLUQdaxHcJgl	m/T4FsHa3JSS+d8CDA3uDPh4vvNcwP+4RBK+Dk6Eyq8eU9rVL7ShbacQCUX0NAAd
Ol+DQYFQ7Mcizm84L9srohV3JnIEKcjaPseB7S9KtHvVvs4q11Q5U2RjQppykZ
qpFInCbyDYgOcqlOorqNehV9F4h9fTszEdUY1UgLvtRj+FTE2Ik7nMK641wFtSO
LCial6kuysZg16SxncNH5gKHeQMVx9dnv+UyJ5VWx3aN12N0ZqaIDcPq75Em2E
aKjEd28c26FEavImn69sz0B8PHQV+6dPwywM=
-----END CERTIFICATE REQUEST-----
EXAMPLE #7  To generate a key set and certificate request that is automatically received by the Control Station for the persona identified by the persona name default, type:

```bash
$ server_certificate server_2 -persona -generate default -key_size 2048 -cs_sign_duration 12 -common_name division.xyz.com
```

server_2 :
Starting key generation. This could take a long time ...
done

EXAMPLE #8  To generate a key set and certificate request to be sent to an external CA specifying subject information, type:

```bash
$ server_certificate server_2 -persona -generate default -key_size 2048 -common_name division.xyz.com -ou QA -organization XYZ -location Bethesda -state Maryland -country US -filename /tmp/server_2.1.request.pem
```

server_2 :
Starting key generation. This could take a long time ...
done

EXAMPLE #9  To import a signed certificate and paste the certificate text, type:

```bash
$ server_certificate server_2 -persona -import default
```

server_2 : Please paste certificate data. Enter a carriage return and on the new line type ‘end of file’ or ‘eof’ followed by another carriage return.
server_cdms

Provides File Migration Service for VNX functionality for the specified Data Movers.

SYNOPSIS

server_cdms {<movername>|ALL} 
-connect <mgfs> -type {nfsv2|nfsv3} -path <localpath> 
-source <srcName>:/<srcPath>[-option <options>]

| -connect <mgfs> -type cifs -path <localpath> -netbios <netbios> -source \<srcServer>\.<domain>\<srcShare>\<srcPath> -admin [domain]\<admin_name> [-wins <wins>]
| -disconnect <mgfs> {-path <localpath>|-path <cid>|-all}
| -verify <mgfs> [-path {<localpath>|<cid>}]
| -Convert <mgfs>
| -start <mgfs> -path <localpath> [-Force] -log <logpath>  
[-include <include_path>][-exclude <exclude_path>]
| -halt <mgfs> -path <localpath>
| -info [<mgfs>][-state {START|STOP|ON_GOING|ERROR|SUCCEED|FAIL}]

DESCRIPTION

server_cdms establishes and removes connections to remote systems, and allows users to start on-access migration.

server_cdms creates an auto-migration process on the Data Mover to ensure that all data has been migrated from the remote system.

server_cdms also checks the state of the migrated file system (MGFS), all auto-migration processes, and the connection, and reports if all data has been migrated successfully.

CDMS supports NFSv2 and NFSv3 only.

The ALL option executes the command for all Data Movers.

OPTIONS

-connect <mgfs> -type {nfsv2|nfsv3} -path <localpath> -source <srcName>:/<srcPath>

Provides a connection for the VNX with the remote NFS server. The -type option specifies the protocol type to be used for communication with the remote NFS server. The directory <localpath> in the file system must be unique for that file system.

The -source option specifies the source file server name or IP address of the remote server as the <srcName> and the export path for migration. For example, nfs_server:/export/path

Note: After the -connect command completes, the file system must be exported.
The server Commands

[-option <options>]
Specifies the following comma-separated options:

[useRootCred={true | false}]
When the file system is mounted, true ensures that the MGFS reads from the source file server using root access UID=0, GID=0. This assumes that the source file server path is exported to allow root access from the specified Data Mover. When false (default), the MGFS uses the owner’s UID and GID to access data.

[proto={TCP | UDP}]
Sets the connection protocol type. The default is TCP.

[nfsPort=<port>]
Sets a remote NFS port number in case the Portmapper or RPC bind is not running, and the port is not the default of 2049.

[mntPort=<port>]
Sets a remote mount port number in case Portmapper or RPC bind is not running.

[mntVer={1 | 2 | 3}]
Sets the version used for mount protocol. By default, NFSv2 uses mount version 2, unless user specified version 1; NFSv3 uses mount version 3.

[localPort=<port>]
Sets the port number used for NFS services, if it needs to be different from the default. The default port number is always greater than 1024.

-connect <mgfs> -type cifs -path <localpath> -netbios <netbios> -source <srcServer>\<srcShare>\<srcPath> -admin <admin_name> -wins <wins>
Provides a connection for the VNX with the remote CIFS server as specified by its NetBIOS name. The directory <localpath> in the file system must be unique for that file system. The -source option specifies the source file server name of the remote server as the <srcName> and the share path for migration that is not at the root of the share. For example, \share\dir1...

The -source and -admin option strings must be enclosed by quotes when issued in a Linux shell.

The -admin option specifies an administrator for the file system. A password is asked interactively when the command is issued. The -wins option specifies an IP address for the WINS server.
Note: This is required only for Windows NT 4.0.

**-disconnect <mgfs> [-path <localpath> | -path <cid> | -all]**

Removes a connection without migrating the data. The `<localpath>` is not removed nor is any partially migrated data.

The administrator should manually remove this data before attempting a **-verify** or **-Convert** command. It may require the administrator to handle a partial migration of old data as well as potentially new data created by users.

It is recommended not to use the **-disconnect** option if the administrator has exported this directory for user access.

**-verify <mgfs>**

Checks that all data has completed the migration for the `<mgfs>`.

  [-path {<localpath> | <cid>}]  
  If the **-path** option is provided, it can check on a communication basis. If no path is provided, the system defaults to checking all connections on the file system.

**-Convert <mgfs>**

Performs a verify check on the entire file system, then changes the file system type from MGFS to UxFS. After the **-Convert** option succeeds, no data migration can be done on that file system.

**-start <mgfs> -path <localpath> [-Force] -log <logpath>**

Directs the Data Mover to migrate all files from the source file server to the VNX. The **-log** option provides detailed information on the state of the migration, and any failures that might occur. The `<localpath>` is the path where the migration thread is started. The **-Force** option is used if you need to start a migration thread a second time on the same `<localpath>` where a previous migration thread had already finished. For example, **-Force** would be needed to start a thread which had no include file (that is, to migrate all remaining files) on `<localpath>` where a thread with an include file had already been run.

  [-**include** <include_path>]  
  Starts the thread in the `<include_path>` which is the path of the file containing the specified directories.

  [-**exclude** <exclude_path>]  
  Excludes files or directories from migration. The `<include_path>` is the path of the file containing the specified directories.
-halt <mgfs> -path <localpath>
Stops a running thread, and halts its execution on the Data Mover. The <mgfs> is the name of the migration file system and the <localpath> is the full path where the migration thread was started.

The -start option resumes thread execution.

-info
Displays a status on the migration file system and the threads.

[<mgfs>]
Specifies the migration file system.

[-state {START | STOP | ON_GOING | ERROR | SUCCEED | FAIL}]
Displays only the threads that are in the state that is specified.

SEE ALSO VNX CDMS Version 2.0 for NFS and CIFS, server_export, server_mount, and server_setup.

EXAMPLE #1 To provide a connection for the migration file system to communicate with the remote NFS server, type:

$ server_cdms server_2 -connect ufs1 -type nfsv3 -path /nfsdir -source 172.24.102.144:/srcdir -option proto=TCP

server_2 : done

EXAMPLE #2 To provide a connection for the migration file system to communicate with the remote CIFS server, type:

$ server_cdms server_2 -connect ufs1 -type cifs -path /dstdir -netbios dm112-cge0 -source "\\winserver1.nasdocs.emc.com\srcdir" -admin "nasdocs.emc.com\administrator" -wins 172.24.102.25

server_2 : Enter Password:*******
done

EXAMPLE #3 To display a status on the migration file system, type:

$ server_cdms server_2

server_2 :
CDMS enabled with 32 threads.

ufs1:
path    = /nfsdir
cid     = 0
type    = NFSV3
source  = 172.24.102.144:/srcdir
options = proto=TCP
path    = /dstdir
cid     = 1
The server Commands

```
type   = CIFS
source = \\
\winserver1.nasdocs.emc.com\srcdir\n
netbios= DM112-CGE0.NASDOCS.EMC.COM
admin  = nasdocs.emc.com\administrator

When migration is started:

$ server_cdms server_2

server_2 :
CDMS enabled with 32 threads.

ufs1:
path    = /nfsdir
cid    = 0

when migration is started, protocol:

when migration is started, protocol:
type   = NFSV3
source = 172.24.102.144:/srcdir
options= proto=TCP
path    = /dstdir

source = \winserver1.nasdocs.emc.com\srcdir\n
netbios= DM112-CGE0.NASDOCS.EMC.COM
admin  = nasdocs.emc.com\administrator

threads:
    path      = /dstdir
    state   = ON_GOING
    log     = /
    cid     = NONE

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ufs1</td>
<td>Migration file system.</td>
</tr>
<tr>
<td>path</td>
<td>Directory in the local file system.</td>
</tr>
<tr>
<td>cid</td>
<td>Connection ID (0 through 1023).</td>
</tr>
<tr>
<td>type</td>
<td>Protocol type to be used to communicate with the remote server.</td>
</tr>
<tr>
<td>source</td>
<td>Source file server name or IP address of the remote server and the export path for migration.</td>
</tr>
<tr>
<td>options</td>
<td>Connection protocol type.</td>
</tr>
<tr>
<td>netbios</td>
<td>NetBIOS name of the remote CIFS server.</td>
</tr>
<tr>
<td>admin</td>
<td>Administrator for the file system.</td>
</tr>
<tr>
<td>threads</td>
<td>Currently existing migration threads.</td>
</tr>
<tr>
<td>state</td>
<td>Current status of migration threads.</td>
</tr>
<tr>
<td>log</td>
<td>Location of the log file that provides detailed information.</td>
</tr>
</tbody>
</table>

EXAMPLE #4  To direct server_2 to migrate all files from the source file server to the VNX, type:

$ server_cdms server_2 -start ufs1 -path /dstdir -log /

server_2 : done
EXAMPLE #5  To display information about migration with the specified status, type:

```
$ server_cdms server_2 -info ufs1 -state ON_GOING
```

```
server_2 :  
ufs1:  
path    = /nfsdir  
cid    = 0  
type   = NFSV3  
source = 172.24.102.144:/srcdir  
options= proto=TCP  
path    = /dstdir  
cid    = 1  
type   = CIFS  
source = \winserver1.nasdocs.emc.com\srcdir\  
netbios= DM112-CGE0.NASDOCS.EMC.COM  
admin  = nasdocs.emc.com\administrator  
threads:  
     path      = /dstdir  
     state   = ON_GOING  
     log     = /  
     cid     = NONE  
```

EXAMPLE #6  To stop data migration on server_2 for ufs1, type:

```
$ server_cdms server_2 -halt ufs1 -path /dstdir
```

```
solver_2 : done
```

EXAMPLE #7  To check that all data has completed the migration, type:

```
$ server_cdms server_2 -verify ufs1 -path /dstdir
```

```
solver_2 : done
```

EXAMPLE #8  To disconnect the path on server_2 for data migration, type:

```
$ server_cdms server_2 -disconnect ufs1 -path /nfsdir
```

```
solver_2 : done
```

EXAMPLE #9  To disconnect all paths for data migration, type:

```
$ server_cdms server_2 -disconnect ufs1 -all
```

```
solver_2 : done
```

EXAMPLE #10  To perform a verify check on ufs1, and then convert it to a uxf, type:

```
$ server_cdms server_2 -Convert ufs1
```

```
solver_2 : done
```
server_cepp

Manages the Common Event Publishing Agent (CEPA) service on the specified Data Mover.

SYNOPSIS

server_cepp {<movername>|ALL}
   -service { -start | -stop | -status | -info }
   | -pool { -info | -stats }

DESCRIPTION

server_cepp starts or stops the CEPA service on the specified Data Mover or all Data Movers and displays information on the status, configuration, and statistics for the service and the pool. The CEPA service is set up in the cepp.conf configuration file. The CEPA configuration is displayed using -service -status, but changes can only be made by directly editing the file with a text editor.

ALL executes the command for all Data Movers.

OPTIONS

-service { -start | -stop | -status | -info }
The -start option starts the CEPA service on the specified Data Mover. -stop stops the CEPA service, -status returns a message indicating whether the CEPA service has started or been stopped, and -info displays information about the CEPA service including key properties of the configured pool.

-pool { -info | -stats }
Displays properties or statistics for the CEPA pool on the specified Data Mover.

SEE ALSO

Using VNX Event Enabler

EXAMPLE #1
To start the CEPA service on a Data Mover, type:

done

EXAMPLE #2
To display the status of the CEPA service, type:

CEPP Stopped
EXAMPLE #3  To display the configuration of the CEPA service, type:

$ server_cepp server_2 -service -info

server_2:
CIFS share name = \DVBL\CHECK$
cifs_server = DVBL
heartbeat_interval = 15 seconds
ft level = 1
ft size = 1048576
ft location = /etc/cepp
msrpc user = OMEGA13$
msrpc client name = OMEGA13.CEE.LAB.COM

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFS share name</td>
<td>The name of the shared directory and CIFS server used to access files in the Data Movers.</td>
</tr>
<tr>
<td>cifs_server</td>
<td>CIFS server to access files.</td>
</tr>
<tr>
<td>heartbeat_interval</td>
<td>The time taken to scan each CEPA server.</td>
</tr>
<tr>
<td>ft level</td>
<td>Fault tolerance level assigned. This option is required. 0 (continue and tolerate lost events; default setting), 1 (continue and use a persistence file as a circular event buffer for lost events), 2 (continue and use a persistence file as a circular event buffer for lost events until the buffer is filled and then stop CIFS), or 3 (upon heartbeat loss of connectivity, stop CIFS).</td>
</tr>
<tr>
<td>ft location</td>
<td>Directory where the persistence buffer file resides relative to the root of a file system. If a location is not specified, the default location is the root of the file system.</td>
</tr>
<tr>
<td>ft size</td>
<td>Maximum size in MB of the persistence buffer file. The default is 1 MB and the range is 1 MB to 100 MB.</td>
</tr>
<tr>
<td>msrpc user</td>
<td>Name assigned to the user account that the CEPA service is running under on the CEE machine. For example, ceeuser.</td>
</tr>
<tr>
<td>msrpc client name</td>
<td>Domain name assigned if the msrpc user is a member of a domain. For example, domain.ceeuser.</td>
</tr>
<tr>
<td>pool_name</td>
<td>Name assigned to the pool that will use the specified CEPA options.</td>
</tr>
<tr>
<td>server_required</td>
<td>Displays availability of the CEPA server. If a CEPA server is not available and this option is yes, an error is returned to the requestor that access is denied. If a CEPA server is not available and this option is no, an error is not returned to the requestor and access is allowed.</td>
</tr>
<tr>
<td>access_checks_ignored</td>
<td>The number of CIFS requests processed when a CEPA server is not available and the server_required option is set to &quot;no.&quot; This option is reset when the CEPA server becomes available.</td>
</tr>
<tr>
<td>req_timeout</td>
<td>Time out in ms to send a request that allows access to the CEPA server.</td>
</tr>
<tr>
<td>retry_timeout</td>
<td>Time out in ms to retry the access request sent to the CEPA server.</td>
</tr>
</tbody>
</table>
The server Commands

**EXAMPLE #4**

To display information about the CEPA pool, type:

```
 server_cepp server_2 -pool -info
```

server_2 :
pool_name = pool1
server_required = yes
access_checks_ignored = 0
req_timeout = 5000 ms
retry_timeout = 25000 ms
pre_events = OpenFileNoAccess, OpenFileRead
post_events = CreateFile, DeleteFile
post_err_events = CreateFile, DeleteFile
CEPP Servers:
IP = 10.171.10.115, state = ONLINE, vendor = Unknown
...

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>pre_events</td>
<td>Sends notification before selected event occurs. An empty list indicates that no pre-event messages are generated.</td>
</tr>
<tr>
<td>post_events</td>
<td>Sends notification after selected event occurs. An empty list indicates that no post-event messages are generated.</td>
</tr>
<tr>
<td>post_err_events</td>
<td>Sends notification if selected event generates an error. An empty list indicates that no post-error-event messages are generated.</td>
</tr>
<tr>
<td>CEPP Servers</td>
<td>IP addresses of the CEPA servers; state of the CEPA servers; vendor software installed on CEPA servers.</td>
</tr>
</tbody>
</table>

**EXAMPLE #5**

To display statistics for the CEPA pool, type:

```
 server_cepp server_2 -pool -stats
```

server_2 :
pool_name = pool1

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Requests</th>
<th>Min(us)</th>
<th>Max(us)</th>
<th>Average(us)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenFileWrite</td>
<td>2</td>
<td>659</td>
<td>758</td>
<td>709</td>
</tr>
<tr>
<td>CloseModified</td>
<td>2</td>
<td>604</td>
<td>635</td>
<td>620</td>
</tr>
</tbody>
</table>

Total Requests = 4
Min(us) = 604
Max(us) = 758
Average(us) = 664
server_checkup

Checks the configuration parameters, and state of a Data Mover and its dependencies.

SYNOPSIS

server_checkup {<movername>|ALL} {[-test <component> [-subtest <dependency>]][-quiet][-full]} | -list | -info {<component>|all}

DESCRIPTION

server_checkup performs a sanity check of a specific Data Mover component and its dependencies by checking configuration parameters, and the current state of the component and dependencies.

A component is any basic feature that is available on the Data Mover, for example, CIFS. A dependency is a configuration component of a Data Mover that the proper operation of a Data Mover functionality (like CIFS) is depending upon. This configuration component can be owned by multiple Data Mover components. For example, proper operation of a CIFS service depends on correctly specified DNS, WINS, Antivirus, and so on.

server_checkup displays a report of errors and warnings detected in the specified Data Mover component and its dependencies.

OPTIONS

No arguments
Performs a sanity check of all the components and all their dependencies on the specified Data Mover or all Data Movers.

-test <component>
Performs a sanity check of a specific component and all of its dependencies.

[-subtest <dependency>]
Performs a sanity check of a specific component and its specified dependency only. If the dependency is not defined, executes the command for all the dependencies of the component.

-quiet
Displays only the number of errors and warnings for the sanity check.

-full
Provides a full sanity check of the specified Data Movers.
The server Commands

-list
Lists all available components that can be checked on a Data Mover.

-info <component>
Lists all dependencies of the specified component, with details of checks that can be performed on each dependency.

EXAMPLE #1
To list the available component in the Data Mover, type:

$ server_checkup server_2 -list

server_2 : done

REPV2
HTTPS
CIFS
FTPDS

EXAMPLE #2
To execute the check of the CIFS component, type:

$ server_checkup server_2 -test CIFS

server_2 :

------------------------------------Checks--------------------------------------
Component CIFS :
ACL       : Checking the number of ACL per file system.........................*Pass
Connection: Checking the load of TCP connections of CIFS.................. Pass
Credential: Checking the validity of credentials........................... Pass
DC        : Checking the connectivity and configuration of the DCs.........*Fail
DFS       : Checking the DFS configuration files and DFS registry......... Pass
DNS       : Checking the DNS configuration and connectivity to DNS servers. Pass
EventLog  : Checking the configuration of Windows Event Logs........... Pass
FS_Type   : Checking if all file systems are all DIR3 type.................. Pass
GPO       : Checking the GPO configuration.................................. Pass
HomeDir   : Checking the configuration of home directory share........... Pass
I18N      : Checking the I18N mode and the Unicode/UTF8 translation tables. Pass
Kerberos  : Checking machine password update for Kerberos................. Fail
LocalGrp  : Checking the local groups database configuration............. Fail
NIS       : Checking the connectivity to the NIS servers, if defined...... Pass
NTP       : Checking the connectivity to the NTP servers, if defined...... Pass
Ntxmap    : Checking the ntxmap configuration file.......................... Pass
Security  : Checking the CIFS security settings............................... Pass
Server    : Checking the CIFS files servers configuration.................. Pass
Share     : Checking the network shares database............................. Pass
SmbList   : Checking the range availability of SMB ID......................*Pass
Threads   : Checking for CIFS blocked threads............................... Pass
UM_Client : Checking for the connectivity to usermapper servers, if any... Pass
UM_Server : Checking the consistency of usermapper database, if primary..*Pass
UnsupOS   : Checking for unsupported client network OS.................... Pass
UnsupProto: Checking for unsupported client network protocols............ Pass
VC        : Checking the configuration to Virus Checker servers............ Pass
The server Commands

WINS      : Checking for the connectivity to WINS servers, if defined...... Pass

NB: a result with a '*' means that some tests were not executed. use -full to run them

-------------------------------CIFS : Kerberos Warnings--------------------------------
Warning 17451974742: server_2 : No update of the machine password of server 'DM102-CGE1'. hold.
--> Check the log events to find out the reason of this issue.

Warning 17451974742: server_2 : No update of the machine password of server 'DM102-CGE0'. hold.
--> Check the log events to find out the reason of this issue.

-------------------------------CIFS : LocalGrp Warnings---------------------------------
Warning 17451974726: server_2 : The local group 'Guests' of server 'DM102-CGE1' contains an unmapped member: S-1-5-15-60415a8a-335a7a0d-6b635f23-202. The access to some network resources may be refused.
--> According the configured resolver of your system (NIS, etc config files, usermapper, LDAP...), add the missing members.

-------------------------------CIFS : DC Errors---------------------------------
Error 13160939577: server_2 : pingdc failed due to NT error ACCESS_DENIED at step SAMR lookups
--> check server configuration and/or DC policies according to reported error.

Error 13160939577: server_2 : pingdc failed due to NT error ACCESS_DENIED at step SAMR lookups
--> check server configuration and/or DC policies according to reported error.

EXAMPLE #3 To execute only the check of the DNS dependency of the CIFS component, type:

$ server_checkup server_2 -test CIFS -subtest DNS

server_2 :
------------------------------------Checks--------------------------------------
Component CIFS :
DNS : Checking the DNS configuration and connectivity to DNS servers. Pass
**EXAMPLE #4**  To list the available dependencies of the CIFS component, type:

```
$ server_checkup server_2 -info CIFS
```

```
server_2:
done

COMPONENT : CIFS
DEPENDENCY : ACL
DESCRIPTION : Number of ACL per file system.
TESTS :
  In full mode, check if the number of ACL per file system doesn't exceed 90% of the maximum limit.

COMPONENT : CIFS
DEPENDENCY : Connection
DESCRIPTION : TCP connection number
TESTS :
  Check if the number of CIFS TCP connections doesn't exceed 80% of the maximum number.

COMPONENT : CIFS
DEPENDENCY : Credential
DESCRIPTION : Users and groups not mapped
TESTS :
  Check if all credentials in memory are mapped to a valid SID.

COMPONENT : CIFS
DEPENDENCY : DC
DESCRIPTION : Connectivity to the domain controllers
TESTS :
  Check the connectivity to the favorite DC (DCPing),
  In full mode, check the connectivity to all DC of the domain,
  Check if DNS site information are defined for each computer name,
  Check if the site of each computer name has an available DC,
  Check if trusted domain of each computer name can be reached,
  Check the ds.useDCldapPing parameter is enabled,
  Check the ds.useADSite parameter is enabled.

COMPONENT : CIFS
DEPENDENCY : DFS
DESCRIPTION : DFS service configuration on computer names
TESTS :
  Check the DFS service is enabled in registry if DFS metadata exists,
  Check the DFS metadata of each share with DFS flag are correct,
  Check if share names in DFS metadata are valid and have the DFS flag,
  Check if each DFS link is valid and loaded,
  Check in the registry if the WideLink key is enabled and corresponds to a valid share name.

COMPONENT : CIFS
DEPENDENCY : DNS
DESCRIPTION : DNS domain configuration
TESTS:
Check if each DNS domain has at least 2 defined servers,
Check the connectivity to each DNS server of each DNS domain,
Check if each DNS server of each DNS domain supports really the DNS service,
Check the ds.useDSFile parameter (automatic discovery of DC),
Check the ds.useDSFile parameter is enabled if the directoryservice file exists.

COMPONENT: CIFS
DEPENDENCY: EventLog
DESCRIPTION: Event Logs parameters on servers
TESTS:
Check if the pathnames of each event logs files are valid (application, system and security),
Check if the maximum file size of each event logs file doesn't exceed 1GB,
Check if the retention time of each event logs file doesn't exceed 1 month.

COMPONENT: CIFS
DEPENDENCY: FS_Type
DESCRIPTION: DIR3 mode of filesystems
TESTS:
Check if each file system is configured in the DIR3 mode.

COMPONENT: CIFS
DEPENDENCY: GPO
DESCRIPTION: GPO configuration on Win2K servers
TESTS:
Check if the size of the GPO cache file doesn't exceed 10% of the total size of the root file system,
Check the last modification date of the GPO cache file is up-to-date,
Check the cifs.gpo and cifs.gpoCache parameters have not been changed,

COMPONENT: CIFS
DEPENDENCY: HomeDir
DESCRIPTION: Home directory shares configuration
TESTS:
Check if the home directory shares configuration file exists, the feature is enabled,
Check if the home directory shares configuration file is optimized (40 lines maximum),
Check the syntax of the home directory shares configuration file.

COMPONENT: CIFS
DEPENDENCY: I18N
DESCRIPTION: Internationalization and translation tables
TESTS:
Check if computer name exists, the I18N mode is enabled,
Check the .etc_common file system is correctly mounted,
Check the syntax of the definition file of the Unicode characters,
Check the uppercase/lowercase conversion table of Unicode character is valid.

COMPONENT: CIFS
DEPENDENCY: Kerberos
DESCRIPTION : Kerberos configuration
TESTS :
Check the machine password update is enabled and up-to-date.

COMPONENT : CIFS
DEPENDENCY : LocalGrp
DESCRIPTION : Local groups and local users
TESTS :
Check the local group database doesn't contain more than 80% of the maximum number of servers,
Check if the servers in the local group database are all valid servers,
Check the state of the local group database (initialized and writable),
Check if the members of built-in local groups are all resolved in the domain,
Check the number of built-in local groups and built-in local users,
Check if the number of defined local users doesn't exceed 90% of the maximum number.

COMPONENT : CIFS
DEPENDENCY : NIS
DESCRIPTION : Network Information System (NIS) configuration
TESTS :
If NIS is configured, check at least 2 NIS servers are defined (redundancy check),
Check if each NIS server can be contacted on the network,
Check if each NIS server really supports the NIS service.

COMPONENT : CIFS
DEPENDENCY : NTP
DESCRIPTION : Network Time Protocol (NTP) configuration
TESTS :
If NTP is configured, check at least 2 NTP servers are defined (redundancy check),
Check if each NIS server can be contacted on the network,
If computer names exist, check if NTP is configured and is running.

COMPONENT : CIFS
DEPENDENCY : Ntxmap
DESCRIPTION : Checking the ntxmap.conf file.
TESTS :
Check the data consistency of the ntxmap configuration file.

COMPONENT : CIFS
DEPENDENCY : Security
DESCRIPTION : Security settings
TESTS :
If the I18N mode is enabled, check the share/unix security setting is not in use,
Discourage to use the share/unix security setting,
Check the cifs.checkAcl parameter is enabled if the security setting is set to NT.

COMPONENT : CIFS
DEPENDENCY : Server
DESCRIPTION : Files servers
TESTS :
Check if each CIFS server is configured with a valid IP interface,
Check if each computer name has joined its domain,
Check if each computer name is correctly registered in their DNS servers,
Check if the DNS servers have the valid IP addresses of each computer name,
Check if a DNS domain exists if at least one computer name exists,

COMPONENT : CIFS
DEPENDENCY : Share
DESCRIPTION : Network shares
TESTS :
Check the available size and i-nodes on the root file system are at least 10% of the total size,
Check the size of the share database doesn't exceed 30% of the total size of the root file system,
Check if the pathname of each share is valid and is available,
Check if each server in the share database really exists,
Check if the I18N mode is enabled, all the share names are UTF-8 compatible,
Check the list of ACL of each share contains some ACE,
Check the length of each share name doesn't exceed 80 Unicode characters.

COMPONENT : CIFS
DEPENDENCY : SmbList
DESCRIPTION : 64k UID, TID and FID limits
TESTS :
In full mode, check the 3 SMB ID lists (UID, FID and TID) don't exceed 90% of the maximum ID number.

COMPONENT : CIFS
DEPENDENCY : Threads
DESCRIPTION : Blocked threads and overload
TESTS :
Check CIFS threads blocked more than 5 and 30 seconds,
Check the maximum number of CIFS threads in use in the later 5 minutes doesn't exceed 90% of the total number,
Check the number of threads reserved for Virus Checker doesn't exceed 20% of the total number of CIFS threads.

COMPONENT : CIFS
DEPENDENCY : UM_Client
DESCRIPTION : Connectivity to the usermapper server
TESTS :
If usermapper servers are defined, check each server can be contacted,
Check if usermapper servers are defined, NIS is not simultaneously activated.

COMPONENT : CIFS
DEPENDENCY : UM_Server
DESCRIPTION : Primary usermapper server
TESTS :
If a primary usermapper is defined locally, check its database size doesn't exceed 30% of the total size,
Check if configuration file is in use, the filling rate of the ranges doesn't exceed 90%,
Check if configuration file is in use, 2 ranges do not overlap,
The server Commands

Check if secmap is enabled,
In full mode, check the SID/UID and SID/GID mappings and reverses are correct and coherent.

COMPONENT : CIFS
DEPENDENCY : UnsupOS
DESCRIPTION : Client OS not supported
TESTS :
Check for unsupported client network OS.

COMPONENT : CIFS
DEPENDENCY : UnsupProto
DESCRIPTION : Unsupported protocol commands detected
TESTS :
Check for unsupported client network protocol commands.

COMPONENT : CIFS
DEPENDENCY : VC
DESCRIPTION : Virus checker configuration
TESTS :
If VC is enabled, check the syntax of the VC configuration file,
Check if the VC 'enable' file and the VC configuration are compatible,
Check the number of VC servers. Make sure at least 2 servers are defined, for redundancy,
Check if there are offline VC servers,
Check if the VC high watermark has not been reached,
Check the connection of VC servers to the Data Mover.

COMPONENT : CIFS
DEPENDENCY : WINS
DESCRIPTION : WINS servers.
TESTS :
If NetBIOS names are defined, check if at least one WINS server is defined,
Check the number of WINS servers. Check if two servers are defined for redundancy,
Check if each WINS server can be contacted on the network,
Check these servers are really WINS servers,
Check if the NetBIOS are correctly registered on the servers.

EXAMPLE #5 To execute additional tests, type:

$ server_checkup server_2 -full

server_2 :

------------------------------------Checks--------------------------------------

Component REPV2 :
F_RDE_CHEC: Checking the F-RDE compatibilty of Repv2 sessions............. Fail

Component HTTPS :
HTTP : Checking the configuration of HTTP applications............... Pass
SSL : Checking the configuration of SSL applications................. Fail

Component CIFS :
ACL : Checking the number of ACLs per file system................... Pass
Connection: Checking the load of CIFS TCP connections.................. Pass
Credential: Checking the validity of credentials....................... Fail
DC : Checking the connectivity and configuration of Domain Controller Fail
DFS : Checking the DFS configuration files and DFS registry......... Fail
DNS : Checking the DNS configuration and connectivity to DNS servers. Fail
EventLog : Checking the configuration of Windows Event Logs.......... Fail
FS_Type : Checking if all file systems are in the DIR3 format........ Pass
GPO : Checking the GPO configuration.................................. Pass
HomeDir : Checking the configuration of home directory shares........ Fail
I18N : Checking the I18N mode and the Unicode/UTF8 translation tables. Fail
Kerberos : Checking password updates for Kerberos.................... Fail
LDAP : Checking the LDAP configuration................................ Pass
LocalGrp : Checking the database configuration of local groups........ Fail
NIS : Checking the connectivity to the NIS servers.................... Pass
NS : Checking the naming services configuration....................... Fail
NTP : Checking the connectivity to the NTP servers.................... Fail
Ntxmap : Checking the ntxmap configuration file....................... Pass
Secmap : Checking the SECMAP database.................................. Fail
Security : Checking the CIFS security settings........................ Pass
Server : Checking the CIFS file servers configuration............... Fail
Share : Checking the network shares database.......................... Fail
SmbList : Checking the range availability of SMB IDs................. Fail
Threads : Checking for CIFS blocked threads............................ Pass
UM_Client : Checking the connectivity to usermapper servers......... Pass
UM_Server : Checking the usermapper server database.................. Pass
UnsupOS : Checking for unsupported client network operating systems Pass
UnsupProto: Checking for unsupported client network protocols........ Pass
VC : Checking the configuration of Virus Checker servers............. Pass
WINS : Checking the connectivity to WINS servers..................... Fail

Component FTPDS :
FS_Type : Checking if all file systems are in the DIR3 format........ Pass
FTPD : Checking the configuration of FTPD............................. Fail
NIS : Checking the connectivity to the NIS servers.................... Pass
NS : Checking the naming services configuration....................... Fail
NTP : Checking the connectivity to the NTP servers.................... Fail
SSL : Checking the configuration of SSL applications.................. Fail

-------------------------------------HTTPS : SSL Warnings-------------------------------------

Warning 17456169084: server_2 : The SSL feature 'DHSM' can not get certificate from the persona default. Because this feature needs a certificate and a private key, it can not start,
The server Commands

--> Run the server_certificate command to generate a new key set and certificate for this persona. Or run the appropriate command (like server_http for instance) to set a correct persona for this SSL feature.

Warning 17456169084: server_2 : The SSL feature 'DIC' can not get certificate from the persona default. Because this feature needs a certificate and a private key, it can not start,
--> Run the server_certificate command to generate a new key set and certificate for this persona. Or run the appropriate command (like server_http for instance) to set a correct persona for this SSL feature.

Warning 17456169084: server_2 : The SSL feature 'DIC_S' can not get certificate from the persona default. Because this feature needs a certificate and a private key, it can not start,
--> Run the server_certificate command to generate a new key set and certificate for this persona. Or run the appropriate command (like server_http for instance) to set a correct persona for this SSL feature.

Warning 17456169084: server_2 : The SSL feature 'DIC_L' can not get certificate from the persona default. Because this feature needs a certificate and a private key, it can not start,
--> Run the server_certificate command to generate a new key set and certificate for this persona. Or run the appropriate command (like server_http for instance) to set a correct persona for this SSL feature.

Warning 17456169084: server_2 : The SSL feature 'DBMS_FILE_TRANSFER' can not get certificate from the persona default. Because this feature needs a certificate and a private key, it can not start,
--> Run the server_certificate command to generate a new key set and certificate for this persona. Or run the appropriate command (like server_http for instance) to set a correct persona for this SSL feature.

--------------------------CIFS : Credential Warnings----------------------------

Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS sanity check tests cannot be done as all CIFS servers are currently disabled on this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

------------------------------CIFS : DC Warnings--------------------------------

Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS sanity check tests cannot be done as all CIFS servers are currently disabled on this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

-----------------------------CIFS : DFS Warnings--------------------------------

Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS sanity check tests cannot be done as all CIFS servers are currently disabled on this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.
Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS sanity check tests cannot be done as all CIFS servers are currently disabled on this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS sanity check tests cannot be done as all CIFS servers are currently disabled on this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS sanity check tests cannot be done as all CIFS servers are currently disabled on this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS sanity check tests cannot be done as all CIFS servers are currently disabled on this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS sanity check tests cannot be done as all CIFS servers are currently disabled on this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

Warning 17456169044: server_2 : The Network Time Protocol subsystem (NTP) has been stopped or is not connected to its server. It may cause potential errors during Kerberos authentication (timeskew).
--> If the NTP service is not running, start it using the server_date command. If it is not connected, check the IP address of the NTP server and make sure the NTP service is up and running on the server. If needed, add another NTP server in the configuration of the Data Mover. Use the server_date command to manage the NTP service and the parameters on the Data Mover.

Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS sanity check tests cannot be done as all CIFS servers are currently disabled on this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

----------------------------CIFS : Server Warnings-------------------------------
Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS
sanity check tests cannot be done as all CIFS servers are currently disabled on
this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

----------------------------CIFS : Share Warnings-------------------------------
Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS
sanity check tests cannot be done as all CIFS servers are currently disabled on
this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

----------------------------CIFS : SmbList Warnings-------------------------------
Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS
sanity check tests cannot be done as all CIFS servers are currently disabled on
this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

----------------------------CIFS : WINS Warnings-------------------------------
Warning 17456168968: server_2 : The CIFS service is currently stopped. Many CIFS
sanity check tests cannot be done as all CIFS servers are currently disabled on
this Data Mover.
--> Start the CIFS server by executing the 'server_setup' command, and try again.

-----------------------------FTPDS : NTP Warnings-------------------------------
Warning 17456169044: server_2 : The Network Time Protocol subsystem (NTP) has
been stopped or is not connected to its server. It may cause potential errors
during Kerberos authentication (timeskew).
--> If the NTP service is not running, start it using the server_date command. If
it is not connected, check the IP address of the NTP server and make sure the NTP
service is up and running on the server. If needed, add another NTP server in the
configuration of the Data Mover. Use the server_date command to manage the NTP
service and the parameters on the Data Mover.

-----------------------------FTPDS : SSL Warnings-------------------------------
Warning 17456169084: server_2 : The SSL feature 'DHSM' can not get certificate
from the persona default. Because this feature needs a certificate and a private
key, it can not start.
--> Run the server_certificate command to generate a new key set and certificate
for this persona. Or run the appropriate command (like server_http for instance)
to set a correct persona for this SSL feature.
Warning 17456169084: server_2 : The SSL feature 'DIC' can not get certificate from the persona default. Because this feature needs a certificate and a private key, it can not start,
--> Run the server_certificate command to generate a new key set and certificate for this persona. Or run the appropriate command (like server_http for instance) to set a correct persona for this SSL feature.

Warning 17456169084: server_2 : The SSL feature 'DIC_S' can not get certificate from the persona default. Because this feature needs a certificate and a private key, it can not start,
--> Run the server_certificate command to generate a new key set and certificate for this persona. Or run the appropriate command (like server_http for instance) to set a correct persona for this SSL feature.

Warning 17456169084: server_2 : The SSL feature 'DIC_L' can not get certificate from the persona default. Because this feature needs a certificate and a private key, it can not start,
--> Run the server_certificate command to generate a new key set and certificate for this persona. Or run the appropriate command (like server_http for instance) to set a correct persona for this SSL feature.

Warning 17456169084: server_2 : The SSL feature 'DBMS_FILE_TRANSFER' can not get certificate from the persona default. Because this feature needs a certificate and a private key, it can not start,
--> Run the server_certificate command to generate a new key set and certificate for this persona. Or run the appropriate command (like server_http for instance) to set a correct persona for this SSL feature.

----------------------------------REPV2 : F_RDE_CHECK Errors----------------------------------

Error 13160415855: server_2 : For the Replication session: repl,
Data Mover version on the source fs: 5.6.47
Data Mover version on the destination fs: 5.5.5
Minimum required Data Mover version on the destination fs: 5.6.46

The Data Mover version on the destination file system is incompatible with the Data Mover version on the source file system. After data transfer, the data in the destination file system may appear to be corrupt, even though the data is in fact intact.

Upgrade the Data Mover where the destination file system resides to at least 5.6.46.

Error 13160415855: server_2 : For the Replication session:rsd1,
F-RDE version on the source fs: 5.6.46
F-RDE version on the destination fs: 5.5.5
Minimum required F-RDE version on the destination fs: 5.6.46
The F-RDE versions are incompatible.
After data transfer, the data in the dst FS may appear to be corrupt.
--> Upgrade the DataMover where the dst fs resides to atleast the version on the source.

Error 13160415855: server_2 : For the Replication session:rsd2, F-RDE version on the source fs: 5.6.46 F-RDE version on the destination fs: 5.5.5 Minimum required F-RDE version on the destination fs: 5.6.46 The F-RDE versions are incompatible. After data transfer, the data in the dst FS may appear to be corrupt. --> Upgrade the DataMover where the dst fs resides to atleast the version on the source.

Error 13160415855: server_2 : For the Replication session:rsd3, F-RDE version on the source fs: 5.6.46 F-RDE version on the destination fs: 5.5.5 Minimum required F-RDE version on the destination fs: 5.6.46 The F-RDE versions are incompatible. After data transfer, the data in the dst FS may appear to be corrupt. --> Upgrade the DataMover where the dst fs resides to atleast the version on the source.

-----------------------------HTTPS : SSL Errors---------------------------------

Error 13156876314: server_2 : The persona 'default' contains nor certificate neither private keys sets. So, this persona can not be used by a SSL feature on the Data Mover. --> Run the server_certificate command to generate a new key set and certificate for this persona.

-----------------------------CIFS : DNS Errors---------------------------------

Error 13161070637: server_2 : The DNS service is currently stopped and does not contact any DNS server. The CIFS clients may not be able to access the Data Mover on the network. --> Start the DNS service on the Data Mover, using the 'server_dns' command.

-----------------------------CIFS : NS Errors---------------------------------

Error 13156352011: server_2 : None of the naming services defined for the entity 'host' in nsswitch.conf is configured. --> Make sure each entity (e.g. host, passwd..) in the nsswitch.conf file contains naming services, (e.g. local files, NIS or usermapper), and make sure these services are configured. Use the corresponding commands like server_nis, server_dns or server_ldap to make sure they are configured.

Error 13156352011: server_2 : None of the naming services defined for the entity 'group' in nsswitch.conf is configured. --> Make sure each entity (e.g. host, passwd..) in the nsswitch.conf file contains naming services, (e.g. local files, NIS or usermapper), and make sure these services are configured. Use the corresponding commands like server_nis, server_dns or server_ldap to make sure they are configured.
Error 13156352011: server_2 : None of the naming services defined for the entity 'netgroup' in nsswitch.conf is configured.
--> Make sure each entity (e.g. host, passwd..) in the nsswitch.conf file contains naming services, (e.g. local files, NIS or usermapper), and make sure these services are configured. Use the corresponding commands like server_nis, server_dns or server_ldap to make sure they are configured.

-----------------------------FTPDS : FTPD Errors--------------------------------

Error 13156876314: server_2 : The persona 'default' contains nor certificate neither private keys sets. So, this persona can not be used by a SSL feature on the Data Mover.
--> Run the server_certificate command to generate a new key set and certificate for this persona.

------------------------------FTPDS : NS Errors---------------------------------

Error 13156352011: server_2 : None of the naming services defined for the entity 'host' in nsswitch.conf is configured.
--> Make sure each entity (e.g. host, passwd..) in the nsswitch.conf file contains naming services, (e.g. local files, NIS or usermapper), and make sure these services are configured. Use the corresponding commands like server_nis, server_dns or server_ldap to make sure they are configured.

Error 13156352011: server_2 : None of the naming services defined for the entity 'group' in nsswitch.conf is configured.
--> Make sure each entity (e.g. host, passwd..) in the nsswitch.conf file contains naming services, (e.g. local files, NIS or usermapper), and make sure these services are configured. Use the corresponding commands like server_nis, server_dns or server_ldap to make sure they are configured.

Error 13156352011: server_2 : None of the naming services defined for the entity 'netgroup' in nsswitch.conf is configured.
--> Make sure each entity (e.g. host, passwd..) in the nsswitch.conf file contains naming services, (e.g. local files, NIS or usermapper), and make sure these services are configured. Use the corresponding commands like server_nis, server_dns or server_ldap to make sure they are configured.

------------------------------FTPDS : SSL Errors--------------------------------

Error 13156876314: server_2 : The persona 'default' contains nor certificate neither private keys sets. So, this persona can not be used by a SSL feature on the Data Mover.
--> Run the server_certificate command to generate a new key set and certificate for this persona.

--------------------------------------------------------------------------------

Total : 14 errors, 25 warnings
EXAMPLE #6

To display only the number of errors and warnings for a Data Mover and dependency, type:

```
$ server_checkup server_2 -quiet
```

```
server_2 :
------------------------------------Checks--------------------------------------
Component REPV2 :
F_RDE_CHEC: Checking the F-RDE compatibility of Repv2 sessions............ Fail
Component HTTPS :
HTTP : Checking the configuration of HTTP applications.................. Pass
SSL : Checking the configuration of SSL applications.................... Pass
Component CIFS :
ACL : Checking the number of ACLs per file system...................... Pass
Connection: Checking the load of CIFS TCP connections....................... Pass
Credential: Checking the validity of credentials.......................... Fail
DC : Checking the connectivity and configuration of Domain Controller Fail
DFS : Checking the DFS configuration files and DFS registry............. Fail
DNS : Checking the DNS configuration and connectivity to DNS servers Fail
EventLog : Checking the configuration of Windows Event Logs............... Fail
FS_Type : Checking if all file systems are in the DIR3 format............ Pass
GPO : Checking the GPO configuration......................................... Pass
HomeDir : Checking the configuration of home directory shares............... Fail
I18N : Checking the I18N mode and the Unicode/UTF8 translation tables. Fail
Kerberos : Checking password updates for Kerberos........................ Fail
LDAP : Checking the LDAP configuration....................................... Pass
LocalGrp : Checking the database configuration of local groups............. Fail
NIS : Checking the connectivity to the NIS servers......................... Pass
NS : Checking the naming services configuration............................ Fail
NTP : Checking the connectivity to the NTP servers........................ Fail
Ntxmap : Checking the ntxmap configuration file.............................. Pass
Secmap : Checking the SECMAP database...................................... Fail
Security : Checking the CIFS security settings................................ Pass
Server : Checking the CIFS file servers configuration....................... Fail
Share : Checking the network shares database............................... Fail
SmbList : Checking the range availability of SMB IDs....................... Pass
Threads : Checking for CIFS blocked threads................................. Pass
UM_Client : Checking the connectivity to usermapper servers.............. Pass
UM_Server : Checking the usermapper server database....................... Pass
UnsupOS : Checking for unsupported client network operating systems... Pass
UnsupProto: Checking for unsupported client network protocols............ Pass
VC : Checking the configuration of Virus Checker servers................. Pass
WINS : Checking the connectivity to WINS servers............................ Fail

Component FTPDS :
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS_Type</td>
<td>Checking if all file systems are in the DIR3 format</td>
<td>Pass</td>
</tr>
<tr>
<td>FTPD</td>
<td>Checking the configuration of FTPD</td>
<td>Fail</td>
</tr>
<tr>
<td>NIS</td>
<td>Checking the connectivity to the NIS servers</td>
<td>Pass</td>
</tr>
<tr>
<td>NS</td>
<td>Checking the naming services configuration</td>
<td>Fail</td>
</tr>
<tr>
<td>NTP</td>
<td>Checking the connectivity to the NTP servers</td>
<td>Fail</td>
</tr>
<tr>
<td>SSL</td>
<td>Checking the configuration of SSL applications</td>
<td>Pass</td>
</tr>
</tbody>
</table>

NB: a result with a '*' means that some tests were not executed. Use `-full` to run them

---

Total: 12 errors, 14 warnings

---
server_cifs

Manages the CIFS configuration for the specified Data Movers or Virtual Data Movers (VDMs).

SYNOPSIS

server_cifs {<movername>|ALL} [options]

'options' can be one of the following:

- option {{audit [,user=<user_name>] [,client=<client_name>] [,full]}
  | homedir=[NO]}

- add netbios=<netbios_name>, domain=<domain_name> [,alias=<alias_name>] ...
  [,hidden=[y|n]] [[,interface=<if_name> [,wins=<ip>[:<ip>]]]...]
  [,local_users][-comment <comment>]

- add
  
  compname=<comp_name>, domain=<full_domain_name> [,alias=<alias_name>] ...
  [,hidden=[y|n]] [,authentication={kerberos|all}]
  [,netbios=<netbios_name>] [[,interface=<if_name> [,wins=<ip>[:<ip>]]]
  [,dns=<if_suffix>]...][,local_users][-comment <comment>]

- add standalone=netbios_name, workgroup=workgroup_name
  [,alias=<alias_name>]...[,hidden=[y|n]]
  [[,interface=<if_name> [,wins=<ip>[:<ip>]]]...][,local_users]
  [-comment <comment>]

- rename -netbios <old_name> <new_name>

- Join compname=<comp_name>, domain=<full_domain_name>, admin=<admin_name>
  [,ou=organizational_unit]
  [-option {reuse|resetserverpasswd|addservice=nfs}]

- Unjoin compname=<comp_name>, domain=<full_domain_name>, admin=<admin_name>

- add security={NT|UNIX|SHARE} [,dialect=<dialect_name>]
- add wins=<ip_addr> [,wins=<ip_addr>...]
- add usrmapper=<ip_addr> [,usrmapper=<ip_addr>...]
- Disable <interface> [,<interface>...]
- Enable <interface> [,<interface>...]
- delete netbios=<netbios_name> [-remove_localgroup]
  [,alias=<alias_name>]...[,interface=<if_name>]

- delete compname=<comp_name> [-remove_localgroup]
  [,alias=<alias_name>]...[,interface=<if_name>]

- delete wins=<ip_addr> [,wins=<ip_addr>...]
- delete usrmapper=<ip_addr> [,usrmapper=<ip_addr>...]
- delete standalone=netbios_name [-remove_localgroup]
  [,alias=<alias_name>]...[,interface=<if_name>]

- update {<share_name>|<path>} [mindirsize=<size>][force]
- Migrate {<fs_name> -acl | netbios_servername} -localgroup
  <src_domain> {:nb=<netbios> | :if=<interface>}
  <dst_domain> {:nb=<netbios> | :if=<interface>}

- Replace {<fs_name> -acl | netbios_servername} -localgroup
  {nb=<netbios> | :if=<interface>}

- stats [-full [-zero]] [-summary]

- setspn { -list [server=<full_comp_name>] }
  | -add <SPN> compname=<comp_name>, domain=<full_domain_name>,


admin=<admin_name>
| -delete <SPN> compname=<comp_name>,domain=<full_domain_name>,
  admin=<admin_name>
}

DESCRIPTION server_cifs manages the CIFS configuration for the specified <movername> which can be the physical Data Mover or VDMs.

Most command options are used with both VDMs and physical Data Movers, whereas others are only used with physical Data Movers. Options available for physical Data Movers only are:

- add security/dialect...
- add/delete usrmapper
- enable/disable interface

The ALL option executes the command for all Data Movers.

OPTIONS No arguments
Displays the CIFS protocol configuration. Certain inputs are not case-sensitive; however, variables may be automatically converted to uppercase.

<options>
CIFS options include:

- option audit
Audits the CIFS configuration by testing for live connections to a Data Mover.

[,user=<user_name>], [client=<client_name>]
[,full]
Audits the live connections created when the session is initiated by the specified <client_name> or audits the live connections for those owned by the specified <user_name>. The full option can be used to identify open files. The <client_name> can be a string or an IPV4 address and the <user_name> can be a string of maximum 20 characters.

- option homedir[=-NO]
Enables and disables (default) the home directory feature. The Data Mover reads information from the homedir map file.

- add netbios=<netbios_name>, domain=<domain_name>
Configures a Windows NT 4.0-like CIFS server on a Data Mover, assigning the specified <netbios_name> and <domain_name> to the server. The domain name is limited to 15 bytes.
CAUTION

Each NetBIOS name must be unique to the domain and the Data Mover.

[,alias=<alias_name>...]  
Assigns a NetBIOS alias to the <netbios_name> associated with the NetBIOS name. The <alias_name> must:

- Be unique on a Data Mover
- Be limited to 15 bytes
- Not begin with an @ (at sign) or - (dash) character
- Not include spaces, tab characters, or the following symbols: / \ ; ; = * + | [ ] ? < > "

[,hidden={y|n}]  
By default, the <netbios_name> is displayed in the Network Neighborhood. If hidden=y is specified, the <netbios_name> does not appear.

[[,interface=<if_name>[,wins=<ip>[:<ip>]]]...]  
Specifies a logical IP interface for the CIFS server in the Windows NT 4.0 domain and associates up to two WINS IP addresses with each interface. The interface name is case-sensitive.

Note: When configuring a CIFS server without any interfaces for a Data Mover, it becomes the default CIFS server and is available on all interfaces not used by other CIFS servers. The default CIFS server can be deleted at any time. It is recommended that IP interfaces should always be specified. VDMs do not have default CIFS servers.

[,local_users]  
Enables local user support that allows the creation of a limited number of local user accounts on the CIFS server. When this command executes, type and confirm a password that is assigned to the local Administrator account on the CIFS server. In addition to the Administrator account, a Guest account is also created. The Guest account is disabled by default. The Administrator account password must be changed before the Administrator can log in to the CIFS server.
After initial creation of the stand-alone server, the **local_users** option resets the local Administrator account password. The password can only be reset if it has not been changed through a Windows client. If the password has already been changed through Windows, the reset will be refused.

```
[-comment <comment>]
```

Assigns a comment to the configuration. The comment is delimited by quotes. Comment length is limited to 48 bytes (represented as 48 ASCII characters or a variable number of Unicode multibyte characters) and cannot include colons since they are recognized as delimiters.

```
-`add compname=<comp_name>, domain=<full_domain_name>`
```

Configures a CIFS server as the `<comp_name>` in the specified Windows Active Directory workgroup. A default NetBIOS name is automatically assigned to the `<comp_name>`. Since the default for `<netbios_name>` is derived from the `<comp_name>`, the `<comp_name>` must not contain any characters that are invalid for a `<netbios_name>`.

In the case of disjoint namespaces, you must use the fully qualified domain name for the `<comp_name>`. For example, for a disjoint namespace, you must always specify the fully qualified domain name (FQDN) with the computer name when joining a CIFS server to a domain, that is, dm112-cge0.emc.com, not just dm112-cge0.

The `<comp_name>` is limited to 63 bytes. The fully qualified domain name is limited to 155 bytes. The `<full_domain_name>` must contain a dot (.). There cannot be a @ (at sign) or - (dash) character. The name also cannot include spaces, tab characters, or the symbols: / \ : ; , = + | [ ] ? < > "

**CAUTION**

Each computer name must be unique to the domain and the Data Mover.

**Note:** Using International Character Sets for File provides details. Only Windows NT security mode can be configured when UTF-8 is enabled.

```
[.alias=<alias_name>...]
```

Assigns an alias to the NetBIOS name. The `<alias_name>` must:

- Be unique on a Data Mover
The server Commands

- Be limited to 15 bytes
- Not begin with an @ (at sign) or - (dash) character
- Not include spaces, tab characters, or the following symbols: / \ : ; , = * + | [ ] ? < > ”

[, hidden={y|n}]
By default, the computer name appears in the Network Neighborhood. If hidden=y is specified, then the computer name does not appear.

[, authentication={kerberos|all}]
Specifies the type of user authentication. The kerberos option limits the server usage to Kerberos authentication; the all option (default) allows both Kerberos and NTLM authentication.

[, netbios=<netbios_name>]
Specifies a <netbios_name> for the <comp_name> in place of the default. The default for <netbios_name> is assigned automatically and is derived from the first 15 bytes of the <comp_name>. The <netbios_name> cannot begin with an @ (at sign) or - (dash) character. The name also cannot include spaces, tab characters, or the symbols: / \ : ; , = * + | [ ] ? < > ”

[, interface=<if_name>[, wins=<ip>[:<ip>]]]...]
Specifies a logical IP interface for the CIFS server in the Active Directory domain and associates up to two WINS IP addresses with each interface. The interface name is case-sensitive.

Note: When configuring a CIFS server without any interfaces for a Data Mover, it becomes the default CIFS server and is available on all interfaces not used by other CIFS servers. The default CIFS server can be deleted at any time. It is recommended that IP interfaces should always be specified. VDMs do not have default CIFS servers.

[, dns=<if_suffix>]
Specifies a different DNS suffix for the interface for DNS updates. By default, the DNS suffix is derived from the domain. This DNS option does not have any impact on the DNS settings of the Data Mover.

[, local_users]
Enables local user support that allows the creation of a limited number of local user accounts on the CIFS server. When this command executes, type and confirm a password that is assigned to the local Administrator account on the CIFS server. In addition
to the Administrator account, a Guest account is also created. The Guest account is disabled by default. The Administrator account password must be changed before the Administrator account can log in to the CIFS server.

After initial creation of the stand-alone server, the local_users option resets the local Administrator account password. The password can only be reset if it has not been changed through a Windows client. If the password has already been changed through Windows, the reset will be refused.

[-comment <comment>]
Assigns a comment to the configuration. The comment is delimited by quotes. Comment length is limited to 48 bytes (represented as 48 ASCII characters or a variable number of Unicode multibyte characters) and cannot include colons since they are recognized as delimiters.

-add standalone=<netbios_name>, workgroup=<workgroup_name>
Creates or modifies a stand-alone CIFS server on a Data Mover, assigning the specified <netbios_name> and <workgroup_name> to the server. The NetBIOS and workgroup names are limited to 15 bytes. When creating a stand-alone CIFS server for the first time, the ,local_users option must be typed, or the command will fail. It is not required when modifying the CIFS server. A stand-alone CIFS server does not require any Windows domain infrastructure. A stand-alone server has local user accounts on the Data Mover and NTLM is used to authenticate users against the local accounts database.

CAUTION
Each NetBIOS name must be unique to the workgroup and the Data Mover.

[,alias=<alias_name>...] Assigns an alias to the NetBIOS name. The <alias_name> must:

- Be unique on a Data Mover
- Be limited to 15 bytes
- Not begin with an @ (at sign) or - (dash) character
- Not include spaces, tab characters, or the following symbols: / \ : ; , = * + | [ ] ? < > "
The server Commands

\[ hidden=(y|n) \]
By default, the <netbios_name> is displayed in the Network Neighborhood. If hidden=y is specified, the <netbios_name> does not appear.

\[ interface=<if_name>[,wins=<ip>[[:<ip>]]] \]
Specifies a logical IP interface for the CIFS server and associates up to two WINS IP addresses with each interface. The interface name is case-sensitive.

**Note:** When configuring a CIFS server without any interfaces for a Data Mover, it becomes the default CIFS server and is available on all interfaces not used by other CIFS servers. The default CIFS server can be deleted at any time. It is recommended that IP interfaces should always be specified. VDMs do not have default CIFS servers.

\[ local_users \]
Enables local user support that allows the creation of a limited number of local user accounts on the CIFS server. When this command executes, type and confirm a password that is assigned to the local Administrator account on the CIFS server. In addition to the Administrator account, a Guest account is also created. The Guest account is disabled by default. The Administrator account password must be changed before the Administrator can log in to the CIFS server.

After initial creation of the stand-alone server, the local_users option resets the local Administrator account password. The password can only be reset if it has not been changed through a Windows client. If the password has already been changed through Windows, the reset will be refused.

\[-comment <comment>\]
Assigns a comment to the configuration. The comment is delimited by quotes. Comment length is limited to 48 bytes (represented as 48 ASCII characters or a variable number of Unicode multibyte characters) and cannot include colons since they are recognized as delimiters.

\[-rename -netbios <old_name> <new_name>\]
Renames a NetBIOS name. For Windows Server, renames a Compname after the CIFS server is unjoined from the domain.

**Note:** Before performing a rename, the new NetBIOS name must be added to the domain using the Windows Server Users and Computers MMC snap-in.
-Join compname=<comp_name>, domain=<full_domain_name>, admin=<admin_name>

Creates an account for the CIFS server in the Active Directory. By default, the account is created under the domain root as ou=Computers,ou=EMC VNX.

**CAUTION**

Before performing a -Join, CIFS service must be started using server_setup.

The <comp_name> is limited to 63 bytes and represents the name of the server to be registered in DNS. The <full_domain_name> is the full domain name to which the server belongs. This means the name must contain at least one period (.). The <admin_name> is the logon name of the user with the right to create and manage computer accounts in the Organizational Unit that the CIFS server is being joined to. If a domain is given as part of the admin username it should be of the form: admin@FQDN. If no domain is given the admin user account is assumed to be part of the domain the CIFS Server is being joined to. The user is prompted to type a password for the admin account.

An Active Directory and a DNS can have the same domain name, or a different domain name (disjoint namespace). For each type of Active Directory and DNS domain relationship, specific VNX parameters and command values must be used. For example, for a disjoint namespace, you must always specify the fully qualified domain name (FQDN) with the computer name when joining a CIFS server to a domain, that is, dm112-cge0.emc.com, not just dm112-cge0.

**CAUTION**

Time services must be synchronized using server_date.

[,ou=<organizational_unit>]

Specifies the organizational unit or container where computer accounts are created in the Active Directory. By default, computer accounts are created in an organizational unit called Computers. The name must be in a valid distinguished name format, for example, ou="cn=My_mover". The name may contain multiple nested elements, such as ou="cn=comp:ou=mach". The colon (:) must be used as a separator for multiple elements. By default, ou=Computers,ou=EMC VNX is used. The organizational unit name is limited to 256 bytes.
The server Commands

[-option {reuse|resetserverpasswd|addservice=nfs}]

The **reuse** option reuses the existing computer account with the original principal or joins a CIFS server to the domain where the computer account has been created manually.

The **resetserverpasswd** option resets the CIFS server password and encryption keys on a domain controller. This option could be used for security reasons, such as changing the server password in the Kerberos Domain Controller.

The **addservice** option adds the NFS service to the CIFS server, making it possible for NFS users to access the Windows Kerberos Domain Controller. Before adding NFS service, the `<comp_name>` must already be joined to the domain, otherwise the command will fail.

- **Unjoin** compname=<comp_name>, domain=<full_domain_name>, admin=<admin_name>

Deletes the account for the CIFS server as specified by its `<comp_name>` from the Active Directory database. The user is prompted to type a password for the admin account.

- **add security**={NT|UNIX|SHARE}

Defines the user authentication mechanism used by the Data Mover for CIFS services. **NT** (default) security mode uses standard Windows domain based user authentication. The local password and group files, NIS, EMC Active Directory UNIX users and groups extension, or UserMapper are required to translate Windows user and group names into UNIX UIDs and GIDs. **NT** security mode is required for the Data Mover to run Windows 2000 or later native environments. Unicode should be enabled for **NT** security mode.

**CAUTION**

**EMC does not recommend the use of UNIX or SHARE security modes.**

For UNIX security mode, the client supplies a username and a plain-text password to the server. The server uses the local (password or group) file or NIS to authenticate the user. To use UNIX security mode, CIFS client machines must be configured to send user passwords to the Data Mover unencrypted in plain text. This requires a registry or security policy change on every CIFS client machine.
For VDM, UNIX and SHARE security modes are global to the Data Mover and cannot be set for each VDM. Unicode must not be enabled.

For SHARE security mode, clients supply a read-only or read-write password for the share. No user authentication is performed using SHARE security. Since this password is sent through the network in clear text, you must modify the Client Registry to allow for clear text passwords.

**CAUTION**

Before adding or changing a security mode, CIFS service must be stopped using `server_setup`, then restarted once options have been set.

```
[,dialect=<dialect_name>]
```

Specifies a dialect. Optimum dialects are assigned by default. Options include CORE, COREPLUS, LANMAN1 (default for UNIX and SHARE security modes), LANMAN2, and NT1 (which represents SMB1 and is the default for NT security mode), and SMB2.

```
-add wins=<ip_addr>[,wins=<ip_addr>...]
```

Adds the WINS servers to the CIFS configuration. The list of WINS servers is processed in the order in which they are added. The first one is the preferred WINS server. If after 1500 milliseconds, the first WINS server times out, the next WINS server on the list is used.

```
-add usrmapper=<ip_addr>[,usrmapper=<ip_addr>...]
```

Adds the IP address(es) of a secondary Usermapper hosts to the CIFS configuration. A single IP address can point to a primary or secondary Usermapper host. If you are using distributed Usermappers, up to eight subsequent IP addresses can point to secondary Usermapper hosts.

```
-Disable <interface> [<interface>,...]
```

Disables the specified IP interfaces for CIFS service. Interface names are case-sensitive. All unused interfaces should be disabled.

```
-Enable <interface> [<interface>,...]
```

Enables the specified IP interfaces for CIFS service. Interface names are case-sensitive.

```
-delete standalone=<netbios_name>
[-remove_localgroup] [,alias=<alias_name>...] [,interface=<if_name>]
```

The server Commands

Deletes the stand-alone CIFS server as identified by its NetBIOS name from the CIFS configuration of the Data Mover.

-\texttt{delete netbios=netbios\_name} \[\texttt{-remove\_localgroup}[\texttt{,alias=alias\_name}...][\texttt{,interface=if\_name}]\]

Deletes the CIFS server as identified by its NetBIOS name from the CIFS configuration of the Data Mover.

-\texttt{delete compname=comp\_name} \[\texttt{-remove\_localgroup}[\texttt{,alias=alias\_name}...][\texttt{,interface=if\_name}]\]

Deletes the CIFS server as identified by its compname from the CIFS configuration of the Data Mover. This does not remove the account from the Active Directory. It is recommended that an \texttt{-Unjoin} be executed prior to deleting the computer name.

\textbf{CAUTION}

The \texttt{-remove\_localgroup} option permanently deletes the local group information of the CIFS server from the permanent storage of the Data Mover. The alias and interface options delete the alias and the interface only, however, the CIFS server exists. The alias and interface options can be combined in the same delete command.

-\texttt{delete wins=ip\_addr}[\texttt{,wins=ip\_addr}...]

Deletes the WINS servers from the CIFS configuration.

-\texttt{delete usrmapper=ip\_addr}[\texttt{,usrmapper=ip\_addr}...]

Deletes the IP addresses of a secondary Usermapper hosts from the CIFS configuration.

-\texttt{update \{<share\_name>|<path>\}}

Updates the attributes and their CIFS names for \texttt{COMPAT} file systems. For every file system, CIFS maintains certain attributes for which there are no NFS equivalents. Updating CIFS attributes updates file attributes and CIFS names by searching the subdirectories of the defined share or path, generating a listing of Microsoft clients filenames (M8.3 and M256), and converting them to a format that CIFS supports. It is not necessary to use this command for \texttt{DIR3} file systems. Options include:
[**mindirsize**=<size>]
Updates the directories with the minimum size specified. Size must be typed in multiples of 512 bytes. A value of 0 ensures that all directories are rebuilt.

[**force**]
Forces a previous update to be overwritten.

---

**CAUTION**

The initial conversion of a directory can take considerable time when the directory contains a large number of files. Although the process is designed to take place in the background, an update should be run only during periods of light system usage.

---

-Migrate {<fs_name> -acl | <netbios_servername> -localgroup}<src_domain>{:nb=<netbios> | :if=<interface>;<dst_domain>{:nb=<netbios> | :if=<interface>}
Updates all security IDs (SIDs) from a <src_domain> to the SIDs of a <dst_domain> by matching the user and group account names in the source domain to the user and group account names in the destination domain. The interface that is specified in this option queries the local server, then its corresponding source and target Domain Controllers to look up each object’s SID.

If -acl is specified, all secure IDs in the ACL database are migrated for the specified file system.

The -localgroup option must be used to migrate the SID members of local group defined for the specified NetBIOS name.

On the source domain, an interface specified to issue a lookup of the SID is defined by either the NetBIOS name or the interface name. On the destination domain, an interface specified to issue a lookup of the SID is defined by either the NetBIOS name or the interface name.

-Replacement {<fs_name> -acl | <netbios_servername> -localgroup}{:nb=<netbios> | :if=<interface>}
Replaces the history SIDs from the old domain with the new SIDs in the new domain. An interface that can be specified to issue a lookup of the SIDs is defined by the interface name or the NetBIOS name.

The -localgroup option must be used to migrate the SID members of the local group defined for the specified NetBIOS name. When the -Replace option is used, the user or group migrated in the new domain keeps their old SID in addition to the new SID created in the new domain.
The `-localgroup` option does the same kind of migration for a specified NetBIOS name in the local groups (instead of the ACL in a file system for the history argument).

**-stats**
Displays all non-zero CIFS statistics, including SMB statistics.

[-full] [-zero]
Lists all SMB statistics or zeroes out the values for SMB statistics.

[-summary]
Displays a summary of SMB statistics.

**-setspn**

- `-list` [server=<full_comp_name>]
  
  | -add <SPN> compname=<comp_name>,
  |     domain=<full_domain_name>, admin=<admin_name>
  
  | -delete <SPN> compname=<comp_name>,
  |     domain=<full_domain_name>, admin=<admin_name>

Displays all SPNs for the specified FQDN server, both for the Data Mover and for the KDC Windows Active Directory entry. If no server is specified, then the SPNs for all joined CIFS Servers for the specified movername is displayed. The command fails if an error occurs. For example, unable to connect to the Active Directory, the specified server is not joined to the domain.

When the `-add` and `-delete` sub-options are used, the user is prompted for the password associated with the admin name. The SPN must be the full value to use, including the realm.

The `-add` sub-option attempts to add the specified SPN to both the Data Mover and Active Directory. The operation succeeds if the SPN is added to both the Data Mover and Active Directory. In an entry already exists in one of these places, it is not duplicated. Otherwise, the operation fails if an error occurs. For example, unable to connect to the Active Directory, the specified server is not joined to the domain, or incorrect admin password.

The `-delete` sub-option attempts to remove the specified SPN from both the Data Mover and Active Directory. The operation succeeds if the SPN is removed from both the Data Mover and Active Directory. If the entry has already been deleted, it is not considered an error. Otherwise, the operation fails if an error occurs. For example, unable to connect to the Active Directory, the specified server is not joined to the domain, or incorrect admin password.
The server Commands

SEE ALSO
Using EMC Utilities for the CIFS Environment, Managing a Multiprotocol Environment on VNX, Using VNX Replicator, Using International Character Sets on VNX for File, server_date, server_export, server_mount, and server_setup.

OUTPUT NOTE
The network interface that appears in the output is dependent on the type of network interface cards that are installed. Dates appearing in the output are in UTC format.

EXAMPLE #1
To display the number and names of open files on server_2, type:

```
$ server_cifs server_2 -o audit,full
```

```
AUDIT Ctx=0xdffcc404, ref=2, Client(fm-main07B60004) Port=36654/139
NS40_1[BRCSLAB] on if=cge0_new
CurrentDC 0xceeb604=W2K3PHYAD
Proto=NT1, Arch=UNKNOWN, RemBufsz=0xfffb, LocBufsz=0xffff, popupMsg=1
0 FNN in FNNlist NbUsr=1 NbCnx=0
Uid=0x3f NTcred(0xcf156a04 RC=1 NTLM Capa=0x401) 'BRCSLAB\gustavo' CHECKER
AUDIT Ctx=0xde05cc04, ref=2, XP Client(BRCSBARREGL1C) Port=1329/445
NS40_1[BRCSLAB] on if=cge0_new
CurrentDC 0xceeb604=W2K3PHYAD
Proto=NT1, Arch=Win2K, RemBufsz=0xffff, LocBufsz=0xffff, popupMsg=1
0 FNN in FNNlist NbUsr=1 NbCnx=2
Uid=0x3f NTcred(0xceeabc04 RC=3 NTLMSSP Capa=0x11001) 'BRCSLAB\gustavo'
CHECKER
Cnxp(0xceeeae04), Name=IPCS, cUid=0x3f Tid=0x3f, Ref=1, Aborted=0
readOnly=0, umask=22, opened files/dirs=0
Cnxp(0xde4e3204), Name=gustavo, cUid=0x3f Tid=0x41, Ref=1, Aborted=0
readOnly=0, umask=22, opened files/dirs=2
Fid=64, FNN=0x1b0648f0(FREE,0x0,0), FOF=0x0  DIR=\Notify commands received:
Event=0x17, wt=0, curSize=0x0, maxSize=0x20, buffer=0x0
Tid=0x41, Pid=0xb84, Mid=0xecc0, Uid=0x3f, size=0x20
Fid=73, FNN=0x1b019ed0(FREE,0x0,0), FOF=0xdf2ae504 (CHECK) FILE=\New Wordpad Document.doc
```

EXAMPLE #2
To configure CIFS service on server_2 with a NetBIOS name of dm110-cge0, in the NT4 domain NASDOCS, with a NetBIOS alias of dm110-cge0a1, hiding the NetBIOS name in the Network Neighborhood, with the interface for CIFS service as cge0, the WINS server as 172.24.102.25, and with the comment string EMC VNX, type:

```
$ server_cifs server_2 -add
netbios=dm110-cge0, domain=NASDOCS, alias=dm110-cge0a1,hidden=y, interface=cge0, wins=172.24.102.25 -comment "EMC Celerra"
```

server_2 : done

EXAMPLE #3
To enable the home directory on server_2, type:
The server Commands

$ server_cifs server_2 -option homedir

EXAMPLE #4 To add the WINS servers, 172.24.103.25 and 172.24.102.25, type:

$ server_cifs server_2 -add
wins=172.24.103.25,wins=172.24.102.25

EXAMPLE #5 To rename the NetBIOS name from dm110-cge0 to dm112-cge0, type:

$ server_cifs server_2 -rename -netbios dm110-cge0
dm112-cge0

EXAMPLE #6 To display the CIFS configuration for NT4 with Internal Usermapper, type:

$ server_cifs server_2

server_2 : done

server_2 : done

EXAMPLE #4

EXAMPLE #5

EXAMPLE #6

server_2 : done

server_2 : done

EXAMPLE #4

EXAMPLE #5

EXAMPLE #6

256 Cifs threads started
Security mode = NT
Max protocol = NT1
I18N mode = UNICODE
Home Directory Shares ENABLED, map=/etc/homedir
Usermapper auto broadcast enabled

Usermapper[0] = [127.0.0.1] state:active port:14640 (auto discovered)

Default WINS servers = 172.24.103.25:172.24.102.25
Enabled interfaces: (All interfaces are enabled)

Disabled interfaces: (No interface disabled)

DOMAIN NASDOCS RC=3
SID=S-1-5-15-99589f8d-9aa3a5f-338728a8-ffffffff
>DC=WINSERVER1(172.24.102.66) ref=2 time=0 ms

CIFS Server DM112-CGE0[NASDOCS] RC=2 (Hidden)
Alias(es): DM110-CGE0A1
Comment='EMC Celerra'
if=cge0 1=172.24.102.242 b=172.24.102.255 mac=0:60:16:4:35:4f
  wins=172.24.102.25
Password change interval: 0 minutes

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cifs threads started</td>
<td>Number of CIFS threads used when the CIFS service was started.</td>
</tr>
<tr>
<td>Security mode</td>
<td>User authorization mechanism used by the Data Mover.</td>
</tr>
<tr>
<td>Max protocol</td>
<td>Maximum dialect supported by the security mode.</td>
</tr>
<tr>
<td>I18N mode</td>
<td>L18N mode (unicode or ASCII).</td>
</tr>
</tbody>
</table>
The server Commands

### EXAMPLE #7

To display the CIFS configuration for NT4, type:

```bash
$ server_cifs server_2
```

server_2:
256 Cifs threads started
Security mode = NT
Max protocol = NT1
I18N mode = UNICODE
Home Directory Shares ENABLED, map=./etc/homedir
Usermapper auto broadcast suspended

Usermapper[0] = [172.24.102.20] state:available

Default WINS servers = 172.24.103.25:172.24.102.25
Enabled interfaces: (All interfaces are enabled)

Disabled interfaces: (No interface disabled)

DOMAIN NASDOCS RC=3
SID=S-1-5-15-99589f8d-9aa3a5f-338728a8-ffffffff
>DC=WINSERVER1(172.24.102.66) ref=2 time=0 ms

CIFS Server DM112-CGE0[NASDOCS] RC=2 (Hidden)
The server Commands

Alias(es): DM110-CGE0A1
Comment='EMC Celerra'
if=cge0 l=172.24.102.242 b=172.24.102.255 mac=0:60:16:4:35:4f
wins=172.24.102.25
Password change interval: 0 minutes

EXAMPLE #8 To add a Windows server using the compname dm112-cge0, in the
Active Directory domain nasdocs.emc.com, with a NetBIOS alias of
dm112-cge0a1, hiding the NetBIOS name in the Network
Neighborhood, with the interface for CIFS service as cge0, the WINS
servers as 172.24.102.25 and 172.24.103.25, in the DNS domain
nasdocs.emc.com, and with the comment string EMC VNX, type:

```
$ server_cifs server_2 -add
compname=dm112-cge0,domain=nasdocs.emc.com,alias=dm112-cge0a1,hidden=y,interface=cge0,wins=172.24.102.25:172.24.103.25,dns=nasdocs.emc.com -comment "EMC Celerra"
```

server_2 : done

EXAMPLE #9 To join dm112-cge0 into the Active Directory domain
nasdocs.emc.com, using the Administrator account, and to add this
server to Engineering\Computers organizational unit, type:

```
$ server_cifs server_2 -Join
compname=dm112-cge0,domain=nasdocs.emc.com,admin=adminis
trator,ou="ou=Computers:ou=Engineering"
```

done

EXAMPLE #10 To add the NFS service to the CIFS server to make it possible for NFS
users to access the Windows KDC, type:

```
$ server_cifs server_2 -Join
compname=dm112-cge0,domain=nasdocs.emc.com,admin=adminis
trator -option addservice=nfs
```

done

EXAMPLE #11 To enable the cge1 interface, type:

```
$ server_cifs server_2 -Enable cge1
```

done

EXAMPLE #12 To display CIFS information for a Data Mover in a Windows domain
with internal usermapper, type:

```
$ server_cifs server_2
```

256 Cifs threads started
Security mode = NT
Max protocol = NT
I18N mode = UNICODE
Home Directory Shares ENABLED, map=/.etc/homedir
Usermapper auto broadcast enabled

Usermapper[0] = [127.0.0.1] state:active (auto discovered)

Default WINS servers = 172.24.103.25:172.24.102.25
Enabled interfaces: (All interfaces are enabled)

Disabled interfaces: (No interface disabled)

Unused Interface(s):
   if=cge1 l=172.24.102.243 b=172.24.102.255 mac=0:60:16:4:35:4e

DOMAIN NASDOCS FQDN=nasdocs.emc.com SITE=Default-First-Site-Name RC=3
   SID=S-1-5-15-99589f8d-9aa3a5f-338728a8-ffffffff
   >DC=WINSERVER1(172.24.102.66) ref=3 time=1 ms (Closest Site)

CIFS Server DM112-CGE0[NASDOCS] RC=2 (Hidden)
   Alias(es): DM112-CGEA1
   Full computer name=dm112-cge0.nasdocs.emc.com realm=NASDOCS.EMC.COM
   Comment='EMC Celerra'
   if=cge0 l=172.24.102.242 b=172.24.102.255 mac=0:60:16:4:35:4f
   wins=172.24.102.25:172.24.103.25
   FQDN=dm112-cge0.nasdocs.emc.com (Updated to DNS)
   Password change interval: 30 minutes
   Last password change: Thu Oct 27 15:59:17 2005
   Password versions: 2

EXAMPLE #13 To display CIFS information for a Data Mover in a Windows domain, type:

$ server_cifs server_2

server_2 :
256 Cifs threads started
Security mode = NT
Max protocol = NT1
I18N mode = UNICODE
Home Directory Shares ENABLED, map=/.etc/homedir
Usermapper auto broadcast suspended

Usermapper[0] = [172.24.102.20] state:available

Default WINS servers = 172.24.103.25:172.24.102.25
Enabled interfaces: (All interfaces are enabled)

Disabled interfaces: (No interface disabled)

Unused Interface(s):
   if=cge1 l=172.24.102.243 b=172.24.102.255 mac=0:60:16:4:35:4e
The server Commands

DOMIAN NASDOCS FQDN=nasdocs.emc.com SITE=Default-First-Site-Name RC=3
SID=S-1-5-15-99589f8d-9aa3a5f-338728a8-ffffff
> DC=WINSERVER1(172.24.102.66) ref=3 time=1 ms (Closest Site)

CIFS Server DM112-CGE0[NASDOCS] RC=2 (Hidden)
Alias(es): DM112-CGEA1
Full computer name=dm112-cge0.nasdocs.emc.com realm=NASDOCS.EMC.COM
Comment='EMC Celerra'
  if=cge0 l=172.24.102.242 b=172.24.102.255 mac=0:60:16:4:35:4f
  wins=172.24.102.25:172.24.103.25
  FQDN=dm112-cge0.nasdocs.emc.com (Updated to DNS)
Password change interval: 30 minutes
Last password change: Thu Oct 27 16:29:21 2005
Password versions: 3, 2

EXAMPLE #14 To display CIFS information for a Data Mover when CIFS service is not started, type:

```
$ server_cifs server_2
```

server_2 :
Cifs NOT started
Security mode = NT
Max protocol = NT1
I18N mode = UNICODE
Home Directory Shares ENABLED, map=/etc/homedir
Usermapper auto broadcast suspended

Usermapper[0] = [172.24.102.20] state:available

Default WINS servers = 172.24.103.25:172.24.102.25
Enabled interfaces: (All interfaces are enabled)

Disabled interfaces: (No interface disabled)

Unused Interface(s):
  if=cge1 l=172.24.102.243 b=172.24.102.255 mac=0:60:16:4:35:4e

CIFS Server DM112-CGE0[NASDOCS] RC=2 (Hidden)
Alias(es): DM112-CGEA1
Full computer name=dm112-cge0.nasdocs.emc.com realm=NASDOCS.EMC.COM
Comment='EMC Celerra'
  if=cge0 l=172.24.102.242 b=172.24.102.255 mac=0:60:16:4:35:4f
  wins=172.24.102.25:172.24.103.25
  FQDN=dm112-cge0.nasdocs.emc.com (Updated to DNS)
Password change interval: 30 minutes
Last password change: Thu Oct 27 16:29:21 2005
Password versions: 3, 2

EXAMPLE #15 To add a Windows server named, dm112-cge0, in the Active Directory domain nasdocs.emc.com, with the interface for CIFS service as cge0, and enable local users support, type:
The server Commands

$ server_cifs server_2 -add
compname=dm112-cge0,domain=nasdocs.emc.com,interface=cge0,local_users

server_2 : Enter Password:********
Enter Password Again:********
done

EXAMPLE #16 To set a security mode to NT for a Data Mover, type:

$ server_cifs server_2 -add security=NT

server_2 : done

EXAMPLE #17 To disable a CIFS interface, type:

$ server_cifs server_2 -Disable cge1

server_2 : done

EXAMPLE #18 To display CIFS audit information for a Data Mover, type:

$ server_cifs server_2 -option audit

server_2 :

| AUDIT Ctx=0xad3d4820, ref=1, W2K3 Client(WINSERVER1) Port=1638/139 |
| DM112-CGE0[NASDOCS] on if=cge0 |
| CurrentDC 0xad407620=WINSERVER1 |
| Proto=NT1, Arch=Win2K, RemBufsz=0xffff, LocBufsz=0xffff |
| 0 FNN in FNNlist NbUsr=1 NbCnx=1 |
| Uid=0x3f NTcred(0xad406a20 RC=2 KERBEROS Capa=0x2) 'NASDOCS\administrator' |
| Cnxp(0xad3d5420), Name=IPC$, cUid=0x3f Tid=0x3f, Ref=1, Aborted=0 |
| readOnly=0, umask=22, opened files/dirs=1 |

| AUDIT Ctx=0xad43c020, ref=1, W2K3 Client(172.24.102.67) Port=1099/445 |
| DM112-CGE0[NASDOCS] on if=cge0 |
| CurrentDC 0xad407620=WINSERVER1 |
| Proto=NT1, Arch=Win2K, RemBufsz=0xffff, LocBufsz=0xffff |
| 0 FNN in FNNlist NbUsr=1 NbCnx=1 |
| Uid=0x3f NTcred(0xad362c20 RC=2 KERBEROS Capa=0x2) 'NASDOCS\user1' |
| Cnxp(0xaec21020), Name=IPC$, cUid=0x3f Tid=0x3f, Ref=1, Aborted=0 |
| readOnly=0, umask=22, opened files/dirs=2 |

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctx</td>
<td>Address in memory of the Stream Context.</td>
</tr>
<tr>
<td>ref</td>
<td>Reference counter of components using this context at this time.</td>
</tr>
<tr>
<td>Port</td>
<td>The client port and the Data Mover port used in the current TCP connection.</td>
</tr>
<tr>
<td>CurrentDC</td>
<td>Specify the address of the Domain Controller that is currently used.</td>
</tr>
<tr>
<td>Proto</td>
<td>Dialect level that is currently used.</td>
</tr>
<tr>
<td>Arch</td>
<td>Type of the client OS.</td>
</tr>
<tr>
<td>RemBufsz</td>
<td>Max buffer size negotiated by the client.</td>
</tr>
<tr>
<td>LocBufsz</td>
<td>Max buffer size we have negotiated.</td>
</tr>
</tbody>
</table>
## The server Commands

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNN/FNNlist</td>
<td>Number of blocked files that has not yet been checked by Virus Checker.</td>
</tr>
<tr>
<td>NbUsr</td>
<td>Number of sessions connected to the stream context (TCP connection).</td>
</tr>
<tr>
<td>NbCnx</td>
<td>Number of connections to shares for this TCP connection.</td>
</tr>
<tr>
<td>Uid/NTcred</td>
<td>User Id (this number is not related to the UNIX UID used to create a file), the credential address, and the type of authentication.</td>
</tr>
<tr>
<td>Cnxp/Name</td>
<td>Share connection address’ and the name of the share the user is connecting to.</td>
</tr>
<tr>
<td>cUid</td>
<td>User Id who has opened the connection first.</td>
</tr>
<tr>
<td>Tid</td>
<td>Tree Id (number which represents the share connection in any protocol request).</td>
</tr>
<tr>
<td>Aborted</td>
<td>Status of the connection.</td>
</tr>
<tr>
<td>readOnly</td>
<td>If the share connection is read only.</td>
</tr>
<tr>
<td>umask</td>
<td>A user file-creation mask.</td>
</tr>
<tr>
<td>opened files/dirs</td>
<td>Number of files or directories opened on this share connection.</td>
</tr>
</tbody>
</table>

**EXAMPLE #19**

To unjoin the computer `dm112-cge0` from the `nasdocs.emc.com` domain, type:

```
$ server_cifs server_2 -Unjoin
compname=dm112-cge0,domain=nasdocs.emc.com,admin=administrator
```

server_2 : Enter Password:********
done

**EXAMPLE #20**

To delete WINS servers, `172.24.102.25`, and `172.24.103.25`, type:

```
$ server_cifs server_2 -delete
wins=172.24.102.25,wins=172.24.103.25
```

server_2 : done

**EXAMPLE #21**

To delete a NetBIOS name, `dm112-cge0`, type:

```
$ server_cifs server_2 -delete netbios=dm112-cge0
```

server_2 : done

**EXAMPLE #22**

To delete the compname, `dm112-cge0`, type:

```
$ server_cifs server_2 -delete compname=dm112-cge0
```

server_2 : done

**EXAMPLE #23**

To delete the usermapper, `172.24.102.20`, type:

```
$ server_cifs server_2 -delete usrmapper=172.24.102.20
```

server_2 : done

**EXAMPLE #24**

To add and join a Windows server in disjoint DNS and Windows domains, type:

```
$ server_cifs server_2 -add
```
The server Commands

compname=dm112-cge0, domain=nasdocs.emc.com, interface=cge0, dns=eng.emc.com
-comment "EMC Celerra"

$ server_cifs server_2 -Join
compname=dm112-cge0.eng.emc.com, domain=nasdocs.emc.com, admin=Administrator

EXAMPLE #25 To add a Windows server using a delegated account from a trusted domain, type:

$ server_cifs server_2 -Join
compname=dm112-cge0, domain=nasdocs.emc.com, admin=delegateduser@it.emc.com

server_2 : Enter Password:********
done

EXAMPLE #26 To add a Windows server in the Active Directory domain using a pre-created computer account, type:

$ server_cifs server_2 -Join
compname=dm112-cge0, domain=nasdocs.emc.com, admin=administrat
or -option reuse

server_2 : Enter Password:********
done

EXAMPLE #27 To update the directory /ufs1/users with a new minimum directory size of 8192, type:

$ server_cifs server_2 -update /ufs1/users mindirsize=8192

server_2 : done

EXAMPLE #28 To migrate all SIDs in the ACL database for file system, ufs1, from the <src_domain>, eng.emc.com:nb=dm112-cge1:if=cge1 to the <dst_domain>, nasdocs.emc.com:nb=dm112-cge0:if=cge0, type:

$ server_cifs server_2 -Migrate ufs1 -acl
eng.emc.com:nb=dm112-cge1:if=cge1
nasdocs.emc.com:nb=dm112-cge0:if=cge0

server_2 : done

EXAMPLE #29 To migrate SIDs of members of the local group defined for the specified NetBIOS name, from the <src_domain>, eng.emc.com:nb=dm112-cge1:if=cge1 to the <dst_domain>, nasdocs.emc.com:nb=dm112-cge0:if=cge0, type:

$ server_cifs server_2 -Migrate dm112-cge1 -localgroup
eng.emc.com:nb=dm112-cge1:if=cge1
nasdocs.emc.com:nb=dm112-cge0:if=cge0

server_2 : done
EXAMPLE #30  To replace the SIDs for ufs1, type:

```
$ server_cifs server_2 -Replace ufs1 -acl
:nb=dm112-cge0:if=cge0
```

server_2 : done

EXAMPLE #31  To configure a stand-alone CIFS server on server_2 with a NetBIOS name of dm112-cge0, in the workgroup NASDOCS, with a NetBIOS alias of dm112-cge0a1, hiding the NetBIOS name in the Network Neighborhood, with the interface for CIFS service as cge0, the WINS servers as 172.24.102.25 and 172.24.103.25, and with enabled local users support, type:

```
$ server_cifs server_2 -add
standalone=dm112-cge0,workgroup=NASDOCS,alias=dm112-cge0a1,hidden=y,interface=cge0,wins=172.24.102.25:172.24.103.25,local_users
```

server_2 : Enter Password:********
Enter Password Again:********
done

EXAMPLE #32  To delete the stand-alone CIFS server, dm112-cge0, type:

```
$ server_cifs server_2 -delete standalone=dm112-cge0
```

server_2 : done

EXAMPLE #33  To display a summary of SMB statistics, type:

```
$ server_cifs server_2 -stats -summary
```

server_2 :
State info:
Open connection Open files
2               2

SMB total requests:
totalAllSmb totalSmb totalTrans2Smb totalTransNTSmb
10038           6593            3437            8

EXAMPLE #34  To display all non-zero CIFS statistics, type:

```
$ server_cifs server_2 -stats
```

server_2 :
SMB statistics:
proc ncalls %totcalls maxTime ms/call
Close 1305 7.96 46.21 2.16
Rename 2 0.01 0.81 0.50
Trans 314 1.91 0.77 0.08
Echo 21 0.13 0.01 0.00
ReadX 231 1.41 0.03 0.00
WriteX 3697 22.54 39.96 0.98
Trans2Prim 9375 57.16 34.27 0.46
TreeDisco 10 0.06 0.06 0.00
The server Commands

NegProt 29 0.18 0.42 0.24
SessSetupX 47 0.29 60.55 5.81
UserLogoffX 9 0.05 0.01 0.00
TreeConnectX 13 0.08 0.39 0.23
TransNT 8 0.05 0.01 0.00
CreateNTX 1338 8.16 47.11 0.81
CancelNT 1 0.01 0.03 0.00

Trans2 SMBs:
proc ncalls %totcalls maxTime ms/call
FindFirst 22 0.23 0.22 0.09
QFsInfo 3154 33.65 0.08 0.05
QPathInfo 1113 11.87 6.73 0.15
QFileInfo 2077 22.16 0.04 0.02
SetFileInfo 3007 32.08 34.26 1.28

NT SMBs:
proc ncalls %totcalls maxTime ms/call
NotifyChange 8 100.00 0.01 0.00

Performance info:
Read Re/s Write Wr/s All Ops/sec
231 231000.00 3697 1021.27 25783 1575.40

State info:
Open connection Open files
2 2

Shadow info:
Reads Writes Splits Extinsert Truncates
0 0 0 0 0

SMB total requests:
totalAllSmb totalSmb totalTrans2Smb totalTransNTSmb (unsupported)
25783 16400 9375 8 2

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc</td>
<td>Name of CIFS requests received.</td>
</tr>
<tr>
<td>ncalls</td>
<td>Number of requests received.</td>
</tr>
<tr>
<td>%totcalls</td>
<td>Percentage of this type of request compared to all requests.</td>
</tr>
<tr>
<td>maxTime</td>
<td>Maximum amount of time used.</td>
</tr>
<tr>
<td>ms/call</td>
<td>Average time in milliseconds taken to service calls.</td>
</tr>
<tr>
<td>failures</td>
<td>Number of times the call has failed.</td>
</tr>
<tr>
<td>Read</td>
<td>Total number of read operations.</td>
</tr>
<tr>
<td>Re/s</td>
<td>Number of read operations per second.</td>
</tr>
<tr>
<td>Write</td>
<td>Total number of write operations.</td>
</tr>
<tr>
<td>Wr/s</td>
<td>Number of write operations per second.</td>
</tr>
</tbody>
</table>
**EXAMPLE #35**

To reset to zero the values for all SMB statistics, type:

```
$ server_cifs server_2 -stats -zero
```

```
server_2 : done
```

**EXAMPLE #36**

To configure CIFS service in a language that uses multibyte characters (such as Japanese), type:

```
$ server_cifs server_2 -add
  compname=<computer_name_in_local_language_text>,
  domain=nasdocs.emc.com, -comment
  <comment_in_local_language_text>
```

```
server_2 : done
```
server_cifssupport

Provides support services for CIFS users.

SYNOPSIS

server_cifssupport {<movername> | ALL}
  -accessright
    {-name <name> [-domain <domain_name>]
     | -sid <SID>| -uname <unix_name>| -uid <user_id>}
    {-path <pathname>| -share <sharename>}
    [-policy {mixed|native|secure|nt|unix}]
    [-build [-admin <admin_name>]]
    [-netbios <netbios_servername>| -compname <comp_name>
     | -standalone <netbios_name>]
    -acl { -path <pathname>| -share <sharename> } [-verbose]
  
  -cred
    { -name <name> [-domain <domain_name>]
     | -sid <SID>| -uname <unix_name>| -uid <user_id>}
    [-build [-ldap][-admin <admin_name>]]
    [-netbios <netbios_servername>| -compname <comp_name>
     | -standalone <netbios_name>]
  
  -pingdc
    {-netbios <netbios_servername>| -compname <comp_name>}
    [-dc <netbios_Dcname>]
    [-verbose]}

  -secmap
  -list
    {-name <name> [-domain <domain_name>]
     -domain <domain_name>
     -sid <SID>
     -uid <user_id>
     -gid <group_id>}
  
  -create
    {-name <name> [-domain <domain_name>]} 
    | -sid <SID>}
  
  -verify
    {-name <name> [-domain <domain_name>]} 
    | -sid <SID>}
  
  -update
    {-name <name> [-domain <domain_name>]} 
    | -sid <SID>}
  
  -delete
    {-name <name> [-domain <domain_name>]} 
    | -sid <SID>}
  
  -export [-file <filename>]
  
  -import -file <filename>

  -report

  -migration
**DESCRIPTION**

**server_cifssupport** checks network connectivity between a CIFS server and domain controller, manages access rights, generates credentials, and manages secure mapping cache.

The **-accessright** option:

- Displays user access rights to a file, directory, or share in a Windows permission mask.
- Rebuilds and displays a credential for users of a file, directory, or share, who do not have a session opened in one of the CIFS servers.
- Without impact on the actual access-checking policy of a file system object, shows how user permissions are reset if you were to change the policy on that object.

The **-acl** option displays the access control list (ACL) of files, directories, or shares in plain text form.

The **-cred** option generates a credential containing all groups to which a user belongs, including local groups, without the user being connected to a CIFS server. This allows you to verify if user’s SIDs are being correctly mapped to UNIX UIDs and GIDs and to troubleshoot any user access control issues.

The **-pingdc** option checks the network connectivity between a CIFS server and a domain controller then verifies that a CIFS server can access and use the following domain controller services:

- IPC$ share logon
- Secure Channel when verifying domain users during NT LAN Manager (NTLM) authentication
- Local Security Authority (LSA) pipe information when mapping Windows SIDs to UNIX UIDs and GIDs
- SAMR (Remote Security Account Manager) pipe when merging a user’s UNIX and Windows groups together to create a credential
- Trusted domain information
- Privilege names for internationalization: **pingdc**

The **-secmap** option manages the secure mapping (secmap) cache. Secmap contains all mapping between SIDs and UID/GIDs used by a Data Mover or Virtual Data Mover (VDM). The Data Mover permanently caches all mappings it receives from any mapping mechanism (local files, NIS, iPlanet, Active Directory, and Usermapper) in the secmap database, making the response to
subsequent mapping requests faster and less susceptible to network problems. Reverse mapping provides better quota support.

**ACCESS RIGHT OPTIONS**

- `-accessright { -name <name> [-domain <domain_name>] | -sid <SID> | -uname <unix_name> | -uid <user_id> }`  
  `-path <pathname> | -share <sharename>`

Displays user access rights to a file, directory, or share in a Windows permission mask for the specified:

- Windows username and the optional domain to which the user belongs
- or
- `<SID>` which is the user’s Windows security identifier
- or
- `<unix_name>`
- or
- `<user_id>` which is the user’s UNIX identifier

The `-path` option specifies the path of the file or directory to check for user permissions, or the absolute path of the share to check for user permissions.

  `[ -policy { mixed | native | secure | nt | unix } ]`

  Specifies an access-checking policy for the specified file, directory, or share. This does not change the current access-checking policy, instead it helps you anticipate any access problems before actually resetting the policy on a file system object. `server_cifs` provides more information.

  `[ -build [-admin <admin_name>] ]`

Rebuilds a credential for a user of a file, directory, or share, who does not have a session opened in one of the CIFS servers. If `-build` is not specified, the system searches the known user credentials in cache. If none are found, an error message is generated. The `-admin` option specifies the name of an administrative user to use for creating the access right list. The password of the admin_name user is prompted when executing the command.

  `[ -netbios <netbios_servername> | -compname <comp_name> | -standalone <netbios_name> ]`

Indicates the CIFS server, as specified by its NetBIOS name or computer name to use when rebuilding the user credential.
The server Commands

The **-standalone** option specifies the stand-alone CIFS server, as specified by its name, to use when rebuilding a user credential.

**Note:** If no CIFS server is specified, the system uses the default CIFS server, which uses all interfaces not assigned to other CIFS servers on the Data Mover.

### ACL OPTIONS

- **-acl**
  - **-path** `<pathname>` | **-share** `<sharename>`
  - **-verbose**

Displays the ACL of a file, directory, or a share in plain text form. Windows or UNIX access control data are both displayed in their native forms. The **-verbose** option displays the ACE access rights mask in plain text form in addition to their native forms.

### CREDENTIAL OPTIONS

- **-cred**
  - **-name** `<name>` | **-domain** `<domain_name>` | **-sid** `<SID>` | **-uname** `<unix_name>` | **-uid** `<user_id>`
  - **-build** | **-ldap** | **-admin** `<admin_name>`

Generates a credential containing all of the groups to which a user belongs without being connected to a CIFS server. The credential is specified by the user’s:

- Windows username and the domain to which the user belongs
- Windows security identifier
- UNIX name
- UNIX identifier

Rebuilds a user credential. If **-build** is not specified, the system searches the known user credentials in cache. If none are found, an error message is generated. The **-ldap** option retrieves the user’s universal groups to be included in the credential. If none are found, no universal groups are incorporated into the credential. The **-admin** option indicates the name of an administrative user for creating the credential. The password of the `<admin_name>` is prompted when executing the command.

- **-netbios** `<netbios_servername>` | **-compname** `<comp_name>` | **-standalone** `<netbios_name>`

Indicates the CIFS server, as specified by its NetBIOS name or computer name to use when rebuilding the user credential.
The **standalone** option specifies the stand-alone CIFS server to use when rebuilding a user credential.

**Note:** If no CIFS server is specified, the system uses the default CIFS server, which uses all interfaces not assigned to other CIFS servers on the Data Mover.

---

**PINGDC OPTIONS**

- `pingdc {-netbios <netbios_servername> | -compname <comp_name>}`

  Checks the network connectivity for the CIFS server as specified by its NetBIOS name or by its computer name with a domain controller. Once connectivity is established, it verifies that a CIFS server can access and use the domain controller services.

  **Note:** An IP address can be used for the `<netbios_servername>` and the `<comp_name>`.

  

  ```
  [-dc <netbios_Dcname>]
  ```

  Indicates the domain controller to ping for network and resource connectivity with the CIFS server. If not specified, the command checks the domain controllers currently in use by the CIFS server.

  **Note:** An IP address can be used for the `<netbios_Dcname>`.

  ```
  [-verbose]
  ```

  Adds troubleshooting information to the command output.

---

**SECMAP OPTIONS**

- `secmap -list`

  Lists the secmap mapping entries.

- `secmap -list -name <name> -domain <domain_name>`

  Lists the secmap mapping entries with the specified name and domain name.

- `secmap -list -domain <domain_name>`

  Lists the secmap mapping entries with the specified domain name.

- `secmap -list -sid <SID>`

  Lists the secmap mapping entries with the specified SID.

- `secmap -list -uid <user_id>`

  Lists the secmap mapping entries with the specified UID (reverse mapping).
The server Commands

-secmap -list -gid <group_id>
Lists the secmap mapping entries with the specified GID (reverse mapping).

-secmap -create { -name <name> [ -domain <domain_name>] }
Creates the secmap mapping entry with the specified name and domain name.

-secmap -create -sid <SID>
Creates the secmap mapping entry with the specified SID.

-secmap -verify { -name <name> [ -domain <domain_name>] }
Checks the mapping entry stored in secmap with the specified name and optional domain name with what is currently available in the mapping sources. If a mapping has changed, it is marked.

-secmap -verify -sid <SID>
Checks the secmap mapping entry with the specified SID.

-secmap -update { -name <name> [ -domain <domain_name>] }
Updates the specified mapping entry stored in secmap with the mappings currently available in the mapping sources. Once this option is performed, force an update of the Data Mover’s file systems ACLs so that the new mappings are recognized.

-secmap -update -sid <SID>
Updates the secmap mapping entry with the specified SID.

-secmap -delete -name <name> [ -domain <domain_name>] 
Deletes the secmap mapping entry with the specified name and domain name.

-secmap -delete -sid <SID>
Deletes the secmap mapping entry with the specified SID.

-secmap -export [ -file <filename>] 
Exports the secmap mapping entry to the specified file.

---

Note: If no filename is specified, the secmap database is displayed on the screen.

-secmap -import -file <filename>
Imports secmap mapping entries from the specified file.
-secmap -report
Displays current secmap status, including database state, domains handled by secmap, and resource usage.

-secmap -migration
Displays secmap database migration information like start date and end date of the operation, and migration status.

**EXAMPLE #1**
To display user access rights to a file for user1, type:

```
$ server_cifssupport server_2 -accessright -name user1 -domain NASDOCS -path /ufs1/test/test.txt
```

server_2 : done

ACCOUNT GENERAL INFORMATIONS

Name : user1
Domain : NASDOCS
Path : /ufs1/test/test.txt
Allowed mask : 0x200a9
Action : List Folder / Read data
Action : Read Extended Attributes
Action : Traverse Folder / Execute File
Action : Read Attributes
Action : Read Permissions

**EXAMPLE #2**
To rebuild a credential for a user to a file using the SID, type:

```
$ server_cifssupport server_2 -accessright -sid S-1-5-15-b8e641e2-33f0942d-8f03a08f-1f4 -path /ufs1/test/test.txt -build -compname dm102-cge0
```

server_2 : done

ACCOUNT GENERAL INFORMATIONS

Path : /ufs1/test/test.txt
Allowed mask : 0x301ff
Action : List Folder / Read data
Action : Create Files / Write data
Action : Create Folders / Append Data
Action : Read Extended Attributes
Action : Write Extended Attributes
Action : Traverse Folder / Execute File
Action : Delete Subfolders and Files
Action : Read Attributes
Action : Write Attributes
Action : Delete
Action : Read Permissions

**EXAMPLE #3**
To display user access rights to a file for user1 with access-checking policy UNIX, type:
$ server_cifssupport server_2 -accessright -name user1 -domain NASDOCS -path /ufs1/test/test.txt -policy unix

server_2 : done

ACCOUNT GENERAL INFORMATIONS

Name : user1
Domain : NASDOCS
Path : /ufs1/test/test.txt
Allowed mask : 0x20089
Action : List Folder / Read data
Action : Read Extended Attributes
Action : Read Attributes
Action : Read Permissions

EXAMPLE #4  To rebuild a credential for user1 to a file using an administrative account, type:

$ server_cifssupport server_2 -accessright -name user1 -domain NASDOCS -path /ufs1/test/test.txt -build -admin administrator

server_2 : Enter Password:*******
done

ACCOUNT GENERAL INFORMATIONS

Name : user1
Domain : NASDOCS
Path : /ufs1/test/test.txt
Allowed mask : 0x200a9
Action : List Folder / Read data
Action : Traverse Folder / Execute File
Action : Read Attributes
Action : Read Permissions

EXAMPLE #5  To display the verbose ACL information of a file, type:

$ server_cifssupport server_2 -acl -path /ufs1/test/test.txt -verbose

server_2 : done

ACL DUMP REPORT

Path : /ufs1/test/test.txt
UID : 32770
GID : 32797
Rights : rw-r--r--
acl ID : 0x4
acl size : 174
owner SID : S-1-5-20-220
group SID : S-1-5-15-b8e641e2-33f0942d-8f03a08f-201
The server Commands

DACL

Owner : USER 32770 S-1-5-15-b8e641e2-33f0942d-8f03a08f-1f4
Access : ALLOWED 0x0 0x1f01ff RWXPDO
Rights : List Folder / Read data
         Create Files / Write data
         Create Folders / Append Data
         Read Extended Attributes
         Write Extended Attributes
         Traverse Folder / Execute File
         Delete Subfolders and Files
         Read Attributes
         Write Attributes
         Delete
         Read Permissions
         Change Permissions
         Take Ownership
         Synchronize

Owner : USER 32771 S-1-5-15-b8e641e2-33f0942d-8f03a08f-a59
Access : ALLOWED 0x0 0x1200a9 R-X---
Rights : List Folder / Read data
         Read Extended Attributes
         Traverse Folder / Execute File
         Read Attributes
         Read Permissions
         Synchronize

EXAMPLE #6 To display the access control level of a share, type:

$ server_cifssupport server_2 -acl -share ufs1

server_2 : done

ACL DUMP REPORT

Share : ufs1
UID : 0
GID : 1
Rights : rwxr-xr-x

EXAMPLE #7 To generate a credential for user1, type:

$ server_cifssupport server_2 -cred -name user1 -domain
NASDOCS

server_2 : done

ACCOUNT GENERAL INFORMATIONS

Name : user1
Domain : NASDOCS
Primary SID : S-1-5-15-b8e641e2-33f0942d-8f03a08f-a59
The server Commands

UID : 32771
GID : 32768
Authentication : KERBEROS
Credential capability : 0x2
Privileges : 0x8
System privileges : 0x2
Default Options : 0x2
NT administrator : False
Backup administrator : False
Backup : False
NT credential capability : 0x2

ACCOUNT GROUPS INFORMATIONS

<table>
<thead>
<tr>
<th>Type</th>
<th>UNIX ID</th>
<th>Name</th>
<th>Domain</th>
<th>SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>NT</td>
<td>32797</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT</td>
<td>32798</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT</td>
<td>4294967294</td>
<td></td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-1f4</td>
<td>S-1-1-0</td>
</tr>
<tr>
<td>NT</td>
<td>4294967294</td>
<td></td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-1f4</td>
<td>S-1-5-2</td>
</tr>
<tr>
<td>NT</td>
<td>4294967294</td>
<td></td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-1f4</td>
<td>S-1-5-b</td>
</tr>
<tr>
<td>NT</td>
<td>2151678497</td>
<td></td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-1f4</td>
<td>S-1-5-20-221</td>
</tr>
<tr>
<td>UNIX</td>
<td>32797</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIX</td>
<td>32798</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIX</td>
<td>4294967294</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIX</td>
<td>2151678497</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**EXAMPLE #8** To rebuild a user credential including the user’s universal groups for a user using SID, type:

```
$ server_cifssupport server_2 -cred -sid S-1-5-15-b8e641e2-33f0942d-8f03a08f-1f4 -build -ldap -compname dm102-cge0
```

server_2 : done

ACCOUNT GENERAL INFORMATIONS

Name : 
Domain : NASDOCS
Server : dm102-cge0
Primary SID : S-1-5-15-b8e641e2-33f0942d-8f03a08f-1f4
UID : 32770
GID : 32768
Authentication : NTLM
Credential capability : 0x0
Privileges : 0x7f
System privileges : 0x1
Default Options : 0xe
NT administrator : True
Backup administrator : True
Backup : False
NT credential capability : 0x0

ACCOUNT GROUPS INFORMATIONS

<table>
<thead>
<tr>
<th>Type</th>
<th>UNIX ID</th>
<th>Name</th>
<th>Domain</th>
<th>SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>NT</td>
<td>32794</td>
<td>Group Policy Cre</td>
<td>NASDOCS S-1-5-15-b8e641e2-33f0942d-8f03a08f-208</td>
<td></td>
</tr>
<tr>
<td>NT</td>
<td>32795</td>
<td>Schema Admins</td>
<td>NASDOCS S-1-5-15-b8e641e2-33f0942d-8f03a08f-206</td>
<td></td>
</tr>
<tr>
<td>NT</td>
<td>32796</td>
<td>Enterprise Admin</td>
<td>NASDOCS S-1-5-15-b8e641e2-33f0942d-8f03a08f-207</td>
<td></td>
</tr>
<tr>
<td>NT</td>
<td>32797</td>
<td>Domain Users</td>
<td>NASDOCS S-1-5-15-b8e641e2-33f0942d-8f03a08f-201</td>
<td></td>
</tr>
<tr>
<td>NT</td>
<td>32793</td>
<td>Domain Admins</td>
<td>NASDOCS S-1-5-15-b8e641e2-33f0942d-8f03a08f-200</td>
<td></td>
</tr>
<tr>
<td>NT</td>
<td>4294967294</td>
<td>Everyone</td>
<td></td>
<td>S-1-1-0</td>
</tr>
<tr>
<td>NT</td>
<td>4294967294</td>
<td>NETWORK</td>
<td>NT AUTHORITY</td>
<td>S-1-5-2</td>
</tr>
<tr>
<td>NT</td>
<td>4294967294</td>
<td>ANONYMOUS LOGON</td>
<td>NT AUTHORITY</td>
<td>S-1-5-7</td>
</tr>
<tr>
<td>NT</td>
<td>2151678496</td>
<td>Administrators</td>
<td>BUILTIN</td>
<td>S-1-5-20-220</td>
</tr>
<tr>
<td>NT</td>
<td>2151678497</td>
<td>Users</td>
<td>BUILTIN</td>
<td>S-1-5-20-221</td>
</tr>
<tr>
<td>NT</td>
<td>1</td>
<td>UNIX GID=0x1 &amp;ap</td>
<td></td>
<td>S-1-5-12-2-1</td>
</tr>
<tr>
<td>UNIX</td>
<td>32794</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIX</td>
<td>32795</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIX</td>
<td>32796</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIX</td>
<td>32797</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIX</td>
<td>32793</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EXAMPLE #9  
To check the network connectivity for the CIFS server with netbios dm102-cge0, type:

```
$ server_cifssupport server_2 -pingdc -netbios dm102-cge0
```

server_2 : done

PINGDC GENERAL INFORMATIONS

DC SERVER:
Netbios name : NASDOCSDC

CIFS SERVER:
Compname : dm102-cge0
Domain : nasdocs.emc.com

EXAMPLE #10  
To check the network connectivity between the domain controller and the CIFS server with compname dm102-cge0, type:

```
$ server_cifssupport server_2 -pingdc -compname dm102-cge0 -dc NASDOCSDC -verbose
```

server_2 : done

PINGDC GENERAL INFORMATIONS

DC SERVER:
Netbios name : NASDOCSDC

CIFS SERVER:
Compname : dm102-cge0
Domain : nasdocs.emc.com
EXAMPLE #11  To display the secmap mapping entries, type:

```
$ server_cifssupport server_2 -secmap -list
```

server_2 : done

SECMAP USER MAPPING TABLE

<table>
<thead>
<tr>
<th>UID</th>
<th>Origin</th>
<th>Date</th>
<th>Name</th>
<th>SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>32772</td>
<td>usernmapper</td>
<td>Tue Sep 18 19:08:40 2007</td>
<td>NASDOCS\user2</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-452</td>
</tr>
<tr>
<td>32771</td>
<td>usernmapper</td>
<td>Tue Sep 18 17:56:53 2007</td>
<td>NASDOCS\user1</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-a59</td>
</tr>
<tr>
<td>32770</td>
<td>usernmapper</td>
<td>Sun Sep 16 07:50:39 2007</td>
<td>NASDOCS\Administrator</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-1f4</td>
</tr>
</tbody>
</table>

SECMAP GROUP MAPPING TABLE

<table>
<thead>
<tr>
<th>GID</th>
<th>Origin</th>
<th>Date</th>
<th>Name</th>
<th>SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>32793</td>
<td>usernmapper</td>
<td>Wed Sep 12 14:16:18 2007</td>
<td>NASDOCS\Domain Admins</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-200</td>
</tr>
<tr>
<td>32797</td>
<td>usernmapper</td>
<td>Sun Sep 16 07:50:40 2007</td>
<td>NASDOCS\Domain Users</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-201</td>
</tr>
<tr>
<td>32799</td>
<td>usernmapper</td>
<td>Mon Sep 17 19:13:16 2007</td>
<td>NASDOCS\Domain Guests</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-202</td>
</tr>
<tr>
<td>32800</td>
<td>usernmapper</td>
<td>Mon Sep 17 19:13:22 2007</td>
<td>NASDOCS\Domain Computers</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-203</td>
</tr>
<tr>
<td>32795</td>
<td>usernmapper</td>
<td>Sun Sep 16 07:50:40 2007</td>
<td>NASDOCS\Schema Admins</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-206</td>
</tr>
<tr>
<td>32796</td>
<td>usernmapper</td>
<td>Sun Sep 16 07:50:40 2007</td>
<td>NASDOCS\Enterprise Admins</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-207</td>
</tr>
<tr>
<td>32794</td>
<td>usernmapper</td>
<td>Sun Sep 16 07:50:40 2007</td>
<td>NASDOCS\Group Policy Creator Owners</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-208</td>
</tr>
<tr>
<td>32798</td>
<td>usernmapper</td>
<td>Mon Sep 17 19:13:15 2007</td>
<td>NASDOCS\CERTSVC_DCOM_ACCESS</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-4e5</td>
</tr>
<tr>
<td>32801</td>
<td>usernmapper</td>
<td>Tue Sep 18 19:08:41 2007</td>
<td>NASDOCS\NASDOCS Group</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-45b</td>
</tr>
</tbody>
</table>

EXAMPLE #12  To display the secmap mapping entry for a user *user1* in a domain NASDOCS, type:

```
$ server_cifssupport server_2 -secmap -list -name user1 -domain NASDOCS
```

server_2 : done

SECMAP USER MAPPING TABLE

<table>
<thead>
<tr>
<th>UID</th>
<th>Origin</th>
<th>Date</th>
<th>Name</th>
<th>SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>32771</td>
<td>usernmapper</td>
<td>Tue Sep 18 17:56:53 2007</td>
<td>NASDOCS\user1</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-a59</td>
</tr>
</tbody>
</table>

EXAMPLE #13  To display the secmap mapping entry for a user with UID 32771, type:
$ server_cifssupport server_2 -secmap -list -uid 32771

server_2 : done

SECMAP USER MAPPING TABLE

<table>
<thead>
<tr>
<th>UID</th>
<th>Origin</th>
<th>Date</th>
<th>Name</th>
<th>SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>32771</td>
<td>usermapper</td>
<td>Tue Sep 18 17:56:53 2007</td>
<td>NASDOCS\user1</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-a59</td>
</tr>
</tbody>
</table>

**EXAMPLE #14** To create the secmap mapping entry for user3 in a domain NASDOCS, type:

$ server_cifssupport server_2 -secmap -create -name user3 -domain NASDOCS

server_2 : done

SECMAP USER MAPPING TABLE

<table>
<thead>
<tr>
<th>UID</th>
<th>Origin</th>
<th>Date</th>
<th>Name</th>
<th>SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>32773</td>
<td>usermapper</td>
<td>Tue Sep 18 19:21:59 2007</td>
<td>NASDOCS\user3</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-a3d</td>
</tr>
</tbody>
</table>

**EXAMPLE #15** To check the secmap mapping for user1 in a domain NASDOCS, type:

$ server_cifssupport server_2 -secmap -verify -name user1 -domain NASDOCS

server_2 : done

**EXAMPLE #16** To update the secmap mapping entry for a user using SID, type:

$ server_cifssupport server_2 -secmap -update -sid S-1-5-15-b8e641e2-33f0942d-8f03a08f-a3d

server_2 : done

**EXAMPLE #17** To delete the secmap mapping entry for user3, type:

$ server_cifssupport server_2 -secmap -delete -name user3 -domain NASDOCS

server_2 : done

**EXAMPLE #18** To display current secmap status, type:

$ server_cifssupport server_2 -secmap -report

server_2 : done

SECMAP GENERAL INFORMATIONS

Name            : server_2
State           : Enabled
Fs              : /
Used nodes      : 12
Used blocks     : 8192
The server Commands

**SECMAP MAPPED DOMAIN**

<table>
<thead>
<tr>
<th>Name</th>
<th>SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>NASDOCS</td>
<td>S-1-5-15-b8e641e2-33f0942d-8f03a08f-fffffffe</td>
</tr>
</tbody>
</table>

**EXAMPLE #19** To export the secmap mapping entries to the display, type:

```
$ server_cifssupport server_2 -secmap -export
```

server_2 : done

**SECMAP MAPPING RECORDS**

S-1-5-15-b8e641e2-33f0942d-8f03a08f-200:2:96:8019:8019:NASDOCS\Domain Admins
S-1-5-15-b8e641e2-33f0942d-8f03a08f-201:2:96:801d:801d:NASDOCS\Domain Users
S-1-5-15-b8e641e2-33f0942d-8f03a08f-202:2:96:801f:801f:NASDOCS\Domain Guests
S-1-5-15-b8e641e2-33f0942d-8f03a08f-203:2:96:8020:8020:NASDOCS\Domain Computers
S-1-5-15-b8e641e2-33f0942d-8f03a08f-206:2:96:801b:801b:NASDOCS\Schema Admins
S-1-5-15-b8e641e2-33f0942d-8f03a08f-207:2:96:801c:801c:NASDOCS\Enterprise Admins
S-1-5-15-b8e641e2-33f0942d-8f03a08f-208:2:96:801a:801a:NASDOCS\Group Policy Creator Owners
S-1-5-15-b8e641e2-33f0942d-8f03a08f-e45:2:96:801e:801e:NASDOCS\CERTSVC_DCOM_ACCESS
S-1-5-15-b8e641e2-33f0942d-8f03a08f-452:1:96:8004:8000:NASDOCS\user2
S-1-5-15-b8e641e2-33f0942d-8f03a08f-a59:1:96:8003:8000:NASDOCS\user1
S-1-5-15-b8e641e2-33f0942d-8f03a08f-45b:2:96:8021:8021:NASDOCS\NASDOCS Group
S-1-5-15-b8e641e2-33f0942d-8f03a08f-1f4:1:96:8002:8000:NASDOCS\Administrator

**EXAMPLE #20** To export the secmap mapping entries to a file, type:

```
$ server_cifssupport server_2 -secmap -export -file exportfile.txt
```

server_2 : done

**EXAMPLE #21** To import the secmap mapping entries from a file, type:

```
$ server_cifssupport server_2 -secmap -import -file exportfile.txt
```

server_2 :
Secmap import in progress : #
done
server_cpu

Performs an orderly, timed, or immediate halt or reboot of a Data Mover.

**SYNOPSIS**

```
server_cpu {<movername>|ALL} {halt|reboot[cold|warm]} [-monitor] <time>
```

**DESCRIPTION**

server_cpu performs an orderly halt or reboot of the specified Data Mover.

The **ALL** option executes the command for all Data Movers.

**OPTIONS**

- **-halt**
  Performs an orderly shutdown of a Data Mover for the VNX. To restart a Data Mover, perform a **-reboot**. For the NS series, a **-halt** causes a system reboot.

- **-reboot**
  Performs an orderly shutdown, and restarts a Data Mover. The default parameter of the **-reboot** option will be the **warm** parameter. In case the warm rebooting fails, the **-reboot** option will use the cold parameter to reboot the Data Mover.

  ![cold]
  A cold reboot or a hardware reset shuts down the Data Mover completely before restarting, including a Power on Self Test (POST).

  ![warm]
  A warm reboot or a software reset performs a partial shutdown of the Data Mover, and skips the POST after restarting. A software reset is faster than the hardware reset.

**CAUTION**

Performing a reboot for ALL Data Movers can be time consuming relative to the size of the mounted file system configuration.

- **-monitor**
  Polls and displays the boot status until completion of the halt or reboot.
<time>
Specifies the time when the Data Mover is to be halted or rebooted. Time is specified as {now | +<min> | <hour>:<min>}. The now option is used for an immediate shutdown or reboot.

After a power fail and crash recovery, the system reboots itself at power-up unless previously halted.

SEE ALSO  VNX System Operations.

EXAMPLE #1  To monitor an immediate reboot of server_2, type:

$ server_cpu server_2 -reboot -monitor now

server_2 : reboot in progress 0.0.0.0.0.0.0.0.0.0.0.3.3.3.3.3.3.4.done

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>DOS booted</td>
</tr>
<tr>
<td>2</td>
<td>SIB failed</td>
</tr>
<tr>
<td>3</td>
<td>Loaded</td>
</tr>
<tr>
<td>4</td>
<td>Configured</td>
</tr>
<tr>
<td>5</td>
<td>Contacted</td>
</tr>
<tr>
<td>7</td>
<td>Panicked</td>
</tr>
<tr>
<td>9</td>
<td>Reboot pending</td>
</tr>
</tbody>
</table>

EXAMPLE #2  To immediately halt server_2, type:

$ server_cpu server_2 -halt now

server_2 : done

EXAMPLE #3  To immediately reboot server_2, type:

$ server_cpu server_2 -reboot now

server_2 : done

EXAMPLE #4  To monitor a reboot of server_2, that is set to take place in one minute, type:

$ server_cpu server_2 -reboot -monitor +1

server_2 : reboot in progress ........3.3.3.3.3.3.3.4.done
server_date

Displays or sets the date and time for a Data Mover, and synchronizes time between a Data Mover and an external time source.

SYNOPSIS

server_date {<movername> | ALL} [+<format>] [<yymmddhhmm>[<ss>]]
| timesvc start ntp [-sync_delay] [-interval <hh>[:<mm>]] [<host>[<host>...]]
| timesvc update ntp
| timesvc stop ntp
| timesvc delete ntp
| timesvc set ntp
| timesvc stats ntp
| timesvc
| timezone [<timezonestr>]
| timezone -name <timezonename>

DESCRIPTION

server_date sets and displays the current date and time for the specified Data Movers.

The server_date timesvc commands control the synchronization of the Data Mover with external timing sources and gets and sets the time zone.

The ALL option executes the command for all Data Movers.

OPTIONS

No arguments
Displays the current date and time for the specified Data Mover.

+<format>
Displays the date information in the format specified by each field descriptor. Each field descriptor is preceded by percent and is replaced in the output by its corresponding value. A single percent is encoded by double percent (%%).

If the argument contains embedded blanks, it must be quoted.

The complete listing of all field descriptors can be viewed using the Linux strftime (3C) man page.

<yymmddhhmm>[<ss>]
Sets a two-digit number for the year, month, day, hour, minutes, and seconds in this order where <yy> is the year; the first <mm> is the month; <dd> is the day; <hh> is the hour (in 24-hour system); and the second <mm> is the minute, and <ss> is the second.
**timesvc start ntp** `<host> [<host>...]`
Starts time synchronization immediately between a Data Mover and a host, which is the IP address of the time server hosts, and adds an entry to the database. The host must be running the NTP protocol. Only four host entries are allowed.

Other options include:

- **-sync_delay**
  Indicates that the clock should not be synchronized when the time server is activated. Instead, when the first poll is taken, latency adjustments are handled slowly. This option is generally used if time service is started after the Data Mover has already started, or if synchronization is starting after other services have already started.

  **Note:** If **-sync_delay** is not typed, by default, the clock is set at Data Mover startup. The clock is synchronized after the first poll.

- **-interval** `<hh>[:<mm>]`
  Sets the delay in hours (or hours and minutes) between polls (default=1 hour which is entered 01 or 00:60). Interval is displayed in minutes.

**timesvc update ntp**
Immediately polls the external source and synchronizes the time on the Data Mover.

**timesvc stop ntp**
Stops timing synchronization between the Data Mover and an external timing host for the NTP protocol, and does not remove the entry from the database.

  **Note:** A stop of time services takes about 12 seconds. If time service is restarted within this time, a "busy" status message is returned.

**timesvc delete ntp**
Stops time synchronization and deletes the NTP protocol from the database.

**timesvc set ntp**
Immediately polls the external source and synchronizes the time on the Data Mover without slewing the clock.

**timesvc stats ntp**
Displays the statistical information of time synchronization for the
Network Time Protocol such as time differences between the Data Mover and the time server. Also provides information about the current state of NTP service on the Data Mover.

**time svc**
Displays the current time service configuration.

**timezone**
Displays the current time zone on the specified Data Mover.

```
<timezonestr>
Sets the current time zone on the specified Data Mover. The <timezonestr> is a POSIX style time zone specification with the following formats:

<std><offset> (no daylight savings time)
<std><offset><dst>[offset],start[/time],end[/time] (adjusts for daylight savings time)
```

**Note:** The Linux man page for tzset provides information about the format.

**timezone -name <timezonename>**
Sets the time zone on the Data Mover to the specified <timezonename>. The <timezonename> is in Linux style time zone specification. A list of valid Linux timezones is located in the /usr/share/zoneinfo directory. The third column in the table in this file lists the valid timezones.

**Note:** The timezone -name option does not reset time on the Data Mover to the specified <timezonename> time.

SEE ALSO
Configuring Time Services on VNX, `server_dns`, and `server_nis`.

**EXAMPLE #1**
To display the current date and time on a Data Mover, type:

```
$ server_date server_2
server_2 : Thu Jan  6 16:55:09 EST 2005
```

**EXAMPLE #2**
To customize the display of the date and time on a Data Mover, type:

```
$ server_date server_2 "+%Y-%m-%d %H:%M:%S"
```
The server Commands

EXAMPLE #3  To start time synchronization between a Data Mover and an external source, type:

```
$ server_date server_2 timesvc start ntp -interval 06:00 172.24.102.20
```

server_2 : done

EXAMPLE #4  To set the time service without slewing the clock, type:

```
$ server_date server_2 timesvc set ntp
```

server_2 : done

EXAMPLE #5  To display statistical information, type:

```
$ server_date server_2 timesvc stats ntp
```

server_2 :
Time synchronization statistics since start:
hits= 2, misses= 0, first poll hit= 2, miss= 0
Last offset: 0 secs, 0 usecs
Current State: Running, connected, interval=360
Time sync hosts:
  0 1 172.24.102.20

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>hits</td>
<td>When a client sends a request to the server requesting the current time, if there is a reply, that is a hit.</td>
</tr>
<tr>
<td>misses</td>
<td>No reply from any of the time servers.</td>
</tr>
<tr>
<td>first poll hit</td>
<td>First poll hit which sets the first official time for the Data Mover.</td>
</tr>
<tr>
<td>miss</td>
<td>First poll miss.</td>
</tr>
<tr>
<td>Last offset</td>
<td>Time difference between the time server and the Data Mover.</td>
</tr>
<tr>
<td>Current State</td>
<td>State of the time server.</td>
</tr>
<tr>
<td>Time sync hosts</td>
<td>IP address of the time server.</td>
</tr>
</tbody>
</table>

EXAMPLE #6  To update time synchronization between a Data Mover and an external source, type:

```
$ server_date server_2 timesvc update ntp
```

server_2 : done

EXAMPLE #7  To get the time zone on the specified Data Mover, type:

```
$ server_date server_2 timezone
```

server_2 : Local timezone: GMT

EXAMPLE #8  To set the time zone to Central Time for a Data Mover when you do not have to adjust for daylight savings time, type:

```
$ server_date server_2 timezone CST6
```

server_2 : done
EXAMPLE #9  To set the time zone to Central Time and adjust for daylight savings time for a Data Mover, type:

```
$ server_date server_2 timezone CST6CDT5,M4.1.0,M10.5.0
```

```
server_2 : done
```

EXAMPLE #10  To set the time zone to Central Time and adjust the daylight savings time for a Data Mover using the Linux method, type:

```
$ server_date server_2 timezone -name America/Chicago
```

```
server_2 : done
```

EXAMPLE #11  To display the time service configuration for a Data Mover, type:

```
$ server_date server_2 timesvc
```

```
Timeservice State

| time:            Thu Jan  6 17:04:28 EST 2005 |
|------------------|------------------------------------------|
| type:            ntp                         |
| sync delay:      off                         |
| interval:        360                         |
| hosts:           172.24.102.20,             |

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>Date and time known to the Data Mover.</td>
</tr>
<tr>
<td>type</td>
<td>Time service protocol configured on the Data Mover.</td>
</tr>
<tr>
<td>sync delay</td>
<td>Whether sync delay is on or off.</td>
</tr>
<tr>
<td>interval</td>
<td>Time interval between polls.</td>
</tr>
<tr>
<td>hosts</td>
<td>Specifies the IP address of the time server.</td>
</tr>
</tbody>
</table>
```

EXAMPLE #12  To stop time services for a Data Mover, type:

```
$ server_date server_2 timesvc stop
```

```
server_2 : done
```

EXAMPLE #13  To delete the time service configuration for a Data Mover, type:

```
$ server_date server_2 timesvc delete ntp
```

```
server_2 : done
```

EXAMPLE #14  To set the timezone on a Data Mover to Los Angeles, type:

```
$ server_date server_2 timezone -n America/Los_Angeles
```

```
server_2 : done
```
server_dbms

Enables backup and restore of databases, displays database environment statistics.

SYNOPSIS

server_dbms {<movername> | ALL}
{ -db
  { -list [<db_name>]
    -delete <db_name>
    -check [<db_name>]
    -repair [<db_name>]
    -compact [<db_name>]
    -fullbackup -target <pathname>
    -incrbckup -previous <pathname> -target <pathname>
    -restore [<db_name>] -source <pathname>
    -stats [<db_name> [-table <name>]] [-reset]]
  | -service -stats [transaction | memory | log | lock | mutex] [-reset]
}

DESCRIPTION

server_dbms provides recovery of media failure or application corruption, displays database information, checks application database consistency, and fixes inconsistencies.

The ALL option executes the command for all Data Movers.

OPTIONS

- db -list [ <db_name> ]
  Gets the list of all application databases and their status. If <db_name> is specified, displays the list of all tables belonging to that database.

- db -delete <db_name>
  Deletes the target application database.

  Note: This command will fail if the target database is not closed.

- db -check [ <db_name> ]
  Checks the consistency of the target database at application level.

- db -repair [ <db_name> ]
  Fixes the application level inconsistencies in the database.

- db -compact [ <db_name> [-table <name>]]
  Frees up disc space by compacting the target environment or database.
-db -fullbackup -target <pathname>
Performs an online full backup of the VDM database environment.
The target parameter specifies the location to copy the database files.
The <pathname> specifies the local path of the database environment
on the Control Station.

-db -incrbackup -previous <pathname> -target <pathname>
Downloads the transactional logs from the VDM and replays them on
a copy of the previous VDM backup specified by previous <pathname>.
The -target option specifies the location to which the database files
are copied.

-db -restore [<db_name>] -source <pathname>
Restores the environment or database specified by <db_name>.
source <pathname> specifies the location for the backup of the
environment/database to be restored.

Note: The database must be closed before the command is executed.

-db -stats [<db_name> [ -table <name>]][-reset]
Displays statistics related to the specified databases and tables. If
-reset is specified, resets the statistics.

-service -stats [transaction|memory|log|lock|mutex][-reset]
Displays transaction, memory, logging, locking or mutex statistics of
the VDM database environment. If -reset is specified, resets all or
specified statistics.

Note: For this command to be executed, the VDM on which the target
environment resides, must be up.

EXAMPLE #1
To get the list of all application databases and their status, type:

$ server_dbms server_3 -db -list

server_3 : done
BASE NAME : Secmap
Version : 1
Comment : CIFS Secure mapping database.
This is a cache of the sid to uid/gid mapping of the VDM.
This database is part of the CIFS application.
It can closed with the command server_setup
Size : 16384
Modification time : Fri May 25 09:58:21 2007
The server Commands

Creation time : Fri May 25 09:58:21 2007
TABLE NAME : Mapping
Version : 1
Comment : Sid to uid/gid mapping table with one secondary key on xid ((1,uid) &amp; (2,gid))
Size : 16384
Modification time : Fri May 25 09:58:21 2007
Creation time : Fri May 25 09:58:21 2007
BASE NAME : V4NameSpace
Version : 1
Comment : NFSv4 namespace database, this represents the pseudofs and referrals.
Size : 8192
Modification time : Tue Jun 5 08:57:12 2007
Creation time : Tue Jun 5 08:57:12 2007
TABLE NAME : pseudofs
Version : 1
Comment : Pseudofs-table, this holds the export tree heirarchy
Size : 8192
Modification time : Mon Jun 11 11:06:23 2007
Creation time : Mon Jun 11 11:06:23 2007
BASE NAME : Usermapper
Version : 1
Comment : Usermapper database. It allows to assign a new uid or gid to a given SID.
Size : 57344
Modification time : Tue Jun 12 09:14:31 2007
Creation time : Tue Jun 12 09:14:31 2007
TABLE NAME : aliases
Version : 1
Comment : This table allows to retrieve a domain name from one of his aliases
Size : 8192
Modification time : Tue Jun 12 09:14:31 2007
Creation time : Tue Jun 12 09:14:31 2007
TABLE NAME : idxname
Version : 1
Comment : Store the reverse mapping uid/gid to sid.
Size : 8192
Modification time : Tue Jun 12 09:14:31 2007
Creation time : Tue Jun 12 09:14:31 2007
TABLE NAME : usrmapuc
Version : 1
Comment : Store the uid &amp; gid ranges allocations for domains.
Size : 8192
Modification time : Tue Jun 12 09:14:31 2007
Creation time : Tue Jun 12 09:14:31 2007
TABLE NAME : usrgrpmapnamesid
Comment : Store the mapping user.domain -> SID.
Size : 8192
Modification time : Tue Jun 12 09:14:31 2007
Creation time : Tue Jun 12 09:14:31 2007
TABLE NAME : usrmapgrpc
Version : 1
Comment : Store the mapping SID -> (gid, name).
Size : 8192
Modification time : Tue Jun 12 09:14:31 2007
Creation time : Tue Jun 12 09:14:31 2007
TABLE NAME : groupmapnamesid
Version : 1
Comment : Store the mapping group.domain -> SID.
Size : 8192
Modification time : Tue Jun 12 09:14:31 2007
Creation time : Tue Jun 12 09:14:31 2007

EXAMPLE #2   To display Secmap statistics, type:
$ server_dbms server_3 -db -stats Secmap

server_3 : done
STATISTICS FOR DATABASE : Secmap
TABLE : Mapping
NAME VALUE COMMENT
magic 340322 Magic number.
version 9 Table version number.
metaflags 0 Metadata flags.
nkeys 14 Number of unique keys.
ndata 14 Number of data items.
pagesize 4096 Page size.
minkey 2 Minkey value.
re_len 0 Fixed-length record length.
re_pad 32 Fixed-length record pad.
levels 1 Tree levels.
int_pg 0 Internal pages.
leaf_pg 1 Leaf pages.
dup_pg 0 Duplicate pages.
over_pg 0 Overflow pages.
empty_pg 0 Empty pages.
free 0 Pages on the free list.
int_pgfree 0 Bytes free in internal pages.
leaf_pgfree 2982 Bytes free in leaf pages.
dup_pgfree 0 Bytes free in duplicate pages.
over_pgfree 0 Bytes free in overflow pages.

EXAMPLE #3   To display statistics of the VDM database environment, type:
$ server_dbms server_3 -service -stats

STATISTICS FOR MODULE : LOG
NAME VALUE COMMENT
magic 264584 Log file magic number.
version             12                     Log file version number.
mode                0                      Log file mode.
lg_bsize            32768                  Log buffer size.
lg_size             5242880                Log file size.
record              96                     Records entered into the log.
w_bytes             16001                  Bytes to log.
w_mbytes            0                      Megabytes to log.
w_bytes            0                      Bytes to log since checkpoint.
w_mbytes           0                      Megabytes to log since checkpoint.
wcount              31                     Total writes to the log.
wcount_fill         0                      Overflow writes to the log.
rcount              137                    Total I/O reads from the log.
scount              31                     Total syncs to the log.
region_wait         0                      Region lock granted after wait.
region_nowait       0                      Region lock granted without wait.
cur_file            3                      Current log file number.
cur_offset          16001                  Current log file offset.
disk_file           3                      Known on disk log file number.
disk_offset         16001                  Known on disk log file offset.
regsize             98304                 Region size.
maxcommitperflush   1                      Max number of commits in a flush.
mincommitperflush   1                      Min number of commits in a flush.

STATISTICS FOR MODULE : LOCK

NAME                VALUE                  COMMENT
last_id             91                     Last allocated locker ID.
cur_maxid           2147483647             Current maximum unused ID.
maxlocks            1000                   Maximum number of locks in table.
maxlockers          1000                   Maximum num of lockers in table.
maxobjects          1000                   Maximum num of objects in table.
nmodes              9                      Number of lock modes.
nlocks              20                     Current number of locks.
maxnlocks           21                     Maximum number of locks so far.
nlockers            49                     Current number of lockers.
maxnlockers         49                     Maximum number of lockers so far.
nobjects            20                     Current number of objects.
maxnobjects         21                     Maximum number of objects so far.
nrequests           65711                  Number of lock gets.
nreleases           65691                  Number of lock puts.
nupgrade            0                      Number of lock upgrades.
n downgrade         20                    Number of lock downgrades.
lock_wait           0                      Lock conflicts w/ subsequent wait.
lock_nowait         0                      Lock conflicts w/o subsequent wait.
ndeadlocks          0                      Number of lock deadlocks.
locktimeout         0                      Lock timeout.
nlocktimeouts       0                      Number of lock timeouts.
txntimeout          0                      Transaction timeout.
ntxntimeouts        0                      Number of transaction timeouts.
region_wait         0                      Region lock granted after wait.
region_nowait       0                      Region lock granted without wait.
regsize             352256                 Region size.
### STATISTICS FOR MODULE : TXN

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>last_ckp</td>
<td>3/15945</td>
<td>lsn of the last checkpoint.</td>
</tr>
<tr>
<td>time_ckp</td>
<td>Fri Aug 3 09:38:36 2007</td>
<td>time of last checkpoint.</td>
</tr>
<tr>
<td>last_txnid</td>
<td>0x8000001a</td>
<td>last transaction id given out.</td>
</tr>
<tr>
<td>maxtxns</td>
<td>20</td>
<td>maximum txns possible.</td>
</tr>
<tr>
<td>naborts</td>
<td>0</td>
<td>number of aborted transactions.</td>
</tr>
<tr>
<td>nbegins</td>
<td>26</td>
<td>number of begun transactions.</td>
</tr>
<tr>
<td>ncommits</td>
<td>26</td>
<td>number of committed transactions.</td>
</tr>
<tr>
<td>nactive</td>
<td>0</td>
<td>number of active transactions.</td>
</tr>
<tr>
<td>nsnapshot</td>
<td>0</td>
<td>number of snapshot transactions.</td>
</tr>
<tr>
<td>nrestores</td>
<td>0</td>
<td>number of restored transactions after recovery.</td>
</tr>
<tr>
<td>maxnactive</td>
<td>2</td>
<td>maximum active transactions.</td>
</tr>
<tr>
<td>maxnsnapshot</td>
<td>0</td>
<td>maximum snapshot transactions.</td>
</tr>
<tr>
<td>region_wait</td>
<td>0</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>region_nowait</td>
<td>0</td>
<td>Region lock granted without wait.</td>
</tr>
<tr>
<td>regsize</td>
<td>16384</td>
<td>Region size.</td>
</tr>
</tbody>
</table>

### STATISTICS FOR MODULE : MPOOL

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>gbytes</td>
<td>0</td>
<td>Total cache size: GB.</td>
</tr>
<tr>
<td>bytes</td>
<td>10487684</td>
<td>Total cache size: B.</td>
</tr>
<tr>
<td>ncache</td>
<td>1</td>
<td>Number of caches.</td>
</tr>
<tr>
<td>regsize</td>
<td>10493952</td>
<td>Region size.</td>
</tr>
<tr>
<td>mmapsize</td>
<td>0</td>
<td>Maximum file size for mmap.</td>
</tr>
<tr>
<td>maxopenfd</td>
<td>0</td>
<td>Maximum number of open fd's.</td>
</tr>
<tr>
<td>maxwrite</td>
<td>0</td>
<td>Maximum buffers to write.</td>
</tr>
<tr>
<td>maxwrite_sleep</td>
<td>0</td>
<td>Sleep after writing max buffers.</td>
</tr>
<tr>
<td>map</td>
<td>0</td>
<td>Pages from mapped files.</td>
</tr>
<tr>
<td>cache_hit</td>
<td>65672</td>
<td>Pages found in the cache.</td>
</tr>
<tr>
<td>cache_miss</td>
<td>36</td>
<td>Pages not found in the cache.</td>
</tr>
<tr>
<td>page_create</td>
<td>0</td>
<td>Pages created in the cache.</td>
</tr>
<tr>
<td>page_in</td>
<td>36</td>
<td>Pages read in.</td>
</tr>
<tr>
<td>page_out</td>
<td>2</td>
<td>Pages written out.</td>
</tr>
<tr>
<td>ro_evict</td>
<td>0</td>
<td>Clean pages forced from the cache.</td>
</tr>
<tr>
<td>rw_evict</td>
<td>0</td>
<td>Dirty pages forced from the cache.</td>
</tr>
<tr>
<td>page_trickle</td>
<td>0</td>
<td>Pages written by memp_trickle.</td>
</tr>
<tr>
<td>pages</td>
<td>36</td>
<td>Total number of pages.</td>
</tr>
<tr>
<td>page_clean</td>
<td>36</td>
<td>Clean pages.</td>
</tr>
<tr>
<td>page_dirty</td>
<td>0</td>
<td>Dirty pages.</td>
</tr>
<tr>
<td>hash_buckets</td>
<td>1031</td>
<td>Number of hash buckets.</td>
</tr>
<tr>
<td>hash_searches</td>
<td>65744</td>
<td>Total hash chain searches.</td>
</tr>
<tr>
<td>hash_longest</td>
<td>1</td>
<td>Longest hash chain searched.</td>
</tr>
<tr>
<td>hash_examined</td>
<td>65672</td>
<td>Total hash entries searched.</td>
</tr>
<tr>
<td>hash_nowait</td>
<td>0</td>
<td>Hash lock granted with nowait.</td>
</tr>
<tr>
<td>hash_wait</td>
<td>0</td>
<td>Hash lock granted after wait.</td>
</tr>
<tr>
<td>hash_max_nowait</td>
<td>0</td>
<td>Max hash lock granted with nowait.</td>
</tr>
<tr>
<td>hash_max_wait</td>
<td>0</td>
<td>Max hash lock granted after wait.</td>
</tr>
</tbody>
</table>
The server Commands

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>region_nowait</td>
<td>0</td>
<td>Region lock granted with nowait.</td>
</tr>
<tr>
<td>region_wait</td>
<td>0</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>mvcc_frozen</td>
<td>0</td>
<td>Buffers frozen.</td>
</tr>
<tr>
<td>mvcc_thawed</td>
<td>0</td>
<td>Buffers thawed.</td>
</tr>
<tr>
<td>mvcc_freed</td>
<td>0</td>
<td>Frozen buffers freed.</td>
</tr>
<tr>
<td>alloc</td>
<td>123</td>
<td>Number of page allocations.</td>
</tr>
<tr>
<td>alloc_buckets</td>
<td>0</td>
<td>Buckets checked during allocation.</td>
</tr>
<tr>
<td>alloc_max_buckets</td>
<td>0</td>
<td>Max checked during allocation.</td>
</tr>
<tr>
<td>alloc_pages</td>
<td>0</td>
<td>Pages checked during allocation.</td>
</tr>
<tr>
<td>alloc_max_pages</td>
<td>0</td>
<td>Max checked during allocation.</td>
</tr>
<tr>
<td>io_wait</td>
<td>0</td>
<td>Thread waited on buffer I/O.</td>
</tr>
</tbody>
</table>

STATISTICS FOR MODULE : MUTEX

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mutex_align</td>
<td>4</td>
<td>Mutex alignment.</td>
</tr>
<tr>
<td>mutex_tas_spins</td>
<td>1</td>
<td>Mutex test-and-set spins.</td>
</tr>
<tr>
<td>mutex_cnt</td>
<td>3254</td>
<td>Mutex count.</td>
</tr>
<tr>
<td>mutex_free</td>
<td>1078</td>
<td>Available mutexes.</td>
</tr>
<tr>
<td>mutex_inuse</td>
<td>2176</td>
<td>Mutexes in use.</td>
</tr>
<tr>
<td>mutex_inuse_max</td>
<td>2176</td>
<td>Maximum mutexes ever in use.</td>
</tr>
<tr>
<td>region_wait</td>
<td>0</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>region_nowait</td>
<td>0</td>
<td>Region lock granted without wait.</td>
</tr>
<tr>
<td>regsize</td>
<td>278528</td>
<td>Region size.</td>
</tr>
</tbody>
</table>
server_devconfig

Queries, saves, and displays the SCSI over Fibre Channel device configuration connected to the specified Data Movers.

SYNOPSIS

server_devconfig {<movername>|ALL} 
  | -create -scsi [<chain_number>] {-disks|-nondisks|-all} 
    [-discovery {y|n}] [-monitor {y|n}] [-Force {y|n}] 
  | -list -scsi [<chain_number>] {-disks|-nondisks|-all} 
  | -probe -scsi [<chain_number>] {-disks|-nondisks|-all} 
  | -rename <old_name> <new_name>

DESCRIPTION

server_devconfig queries the available storage system device and tape device configuration, and saves the device configuration into the Data Mover’s database. server_devconfig renames the device name, and lists SCSI devices.

CAUTION

It is recommended that all Data Movers have the same device configuration. When adding devices to the device table for a single Data Mover only, certain actions such as standby failover will not be successful unless the standby Data Mover has the same disk device configuration as the primary Data Mover.

The ALL option executes the command for all Data Movers.

OPTIONS

- create -scsi [<chain_number>] {-disks|-nondisks|-all} 
  Queries SCSI devices and saves them into the device table database on the Data Mover. The <chain_number> specifies a SCSI chain number.

  The -disks option limits operations to disks. The -nondisks option limits operations to non-disks such as tapes, juke boxes, and gatekeeper devices. The -all option permits all SCSI devices (disks and non-disks).

  Note: The -create option modifies VNX for lock LUN names to the VNX_<vnx-hostname>_<lun-id>_<vnx-dvol-name> format, if the LUNs use the default Unisphere name.
The server Commands

CAUTION
The time taken to complete this command might be lengthy, dependent on the number and type of attached devices.

[-discovery {y|n}]
Enables or disables the storage discovery operation.

CAUTION
Disabling the -discovery option should only be done under the direction of an EMC Customer Service Engineer.

[-monitor {y|n}]
Displays the progress of the query and discovery operations.

[-Force {y|n}]
Overrides the health check failures and changes the storage configuration.

CAUTION
High availability can be lost when changing the storage configuration. Changing the storage configuration should only be done under the direction of an EMC Customer Service Engineer.

-list -scsi [<chain_number>] {-disks|-nondisks|-all}
Lists the SCSI device table database that has been saved on the Data Mover. The <chain_number> specifies a SCSI chain number.

Note: Fibre Channel devices appear as SCSI devices. Therefore, chain numbers might be different for Fibre Channel.

The -disks option limits operations to disks. The -nondisks option limits operations to non-disks such as tapes, juke boxes, and gatekeeper devices. The -all option permits all SCSI devices (disks and non-disks).

-probe -scsi [<chain_number>] {-disks|-nondisks|-all}
Queries and displays the SCSI devices without saving them into the database. The <chain_number> specifies a SCSI chain number.
The server Commands

Note: Fibre Channel devices appear as SCSI devices, therefore, chain numbers may be different for Fibre Channel.

The -disks option limits operations to disks. The -nondisks option limits operations to non-disks such as tapes, juke boxes, and gatekeeper devices. The -all option permits all SCSI devices (disks and non-disks).

-rename <old_name> <new_name>
Renames the specified non-disk from the <old_name> to <new_name>. The -rename option is available for non-disks only.

SEE ALSO
VNX System Operations, nas_disk, and nas_storage.

STORAGE SYSTEM OUTPUT
The number associated with the storage device is dependent on the attached storage system. VNX for block displays a prefix of APM before a set of integers, for example, APM00033900124-0019. For example, Symmetrix storage systems appear as 002804000190-003C.

EXAMPLE #1
For the VNX storage system, to list all devices, type:

$ server_devconfig server_2 -list -scsi -all

server_2:
Scsi Disk Table

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>Director</th>
<th>Port</th>
<th>stor_id</th>
<th>stor_dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>root_disk</td>
<td>c0t0l0</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0000</td>
</tr>
<tr>
<td>root_disk</td>
<td>c16t0l0</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0000</td>
</tr>
<tr>
<td>root_ldisk</td>
<td>c0t0l1</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0001</td>
</tr>
<tr>
<td>root_ldisk</td>
<td>c16t0l1</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0001</td>
</tr>
<tr>
<td>d3</td>
<td>c0t0l2</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0002</td>
</tr>
<tr>
<td>d3</td>
<td>c16t0l2</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0002</td>
</tr>
<tr>
<td>d4</td>
<td>c0t0l3</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0003</td>
</tr>
<tr>
<td>d4</td>
<td>c16t0l3</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0003</td>
</tr>
<tr>
<td>d5</td>
<td>c0t0l4</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0004</td>
</tr>
<tr>
<td>d5</td>
<td>c16t0l4</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0004</td>
</tr>
<tr>
<td>d6</td>
<td>c0t0l5</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0005</td>
</tr>
<tr>
<td>d6</td>
<td>c16t0l5</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0005</td>
</tr>
<tr>
<td>d7</td>
<td>c0t1l0</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0010</td>
</tr>
<tr>
<td>d7</td>
<td>c16t1l0</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0010</td>
</tr>
<tr>
<td>d8</td>
<td>c16t1l1</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0011</td>
</tr>
<tr>
<td>d8</td>
<td>c0t1l1</td>
<td></td>
<td></td>
<td>APM00043807043</td>
<td>0011</td>
</tr>
</tbody>
</table>

Scsi Device Table

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>type</th>
<th>info</th>
</tr>
</thead>
<tbody>
<tr>
<td>gk01</td>
<td>c0t0l0</td>
<td>disk</td>
<td>5 020700000000APM00043807043</td>
</tr>
<tr>
<td>ggk01</td>
<td>c0t1l0</td>
<td>disk</td>
<td>5 020710000100APM00043807043</td>
</tr>
<tr>
<td>gk161</td>
<td>c16t1l1</td>
<td>disk</td>
<td>5 020711000110APM00043807043</td>
</tr>
</tbody>
</table>
For the VNX with a Symmetrix storage system, to list all the devices in the SCSI table, type:

```
$ server_devconfig server_2 -list -scsi -all
```

```
server_2 :  

Scsi Disk Table

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>Director</th>
<th>Port</th>
<th>num</th>
<th>type</th>
<th>num</th>
<th>sts</th>
<th>stor_id</th>
<th>stor_dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>root_disk</td>
<td>c0t010</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0000</td>
</tr>
<tr>
<td>root_disk</td>
<td>c16t010</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0000</td>
</tr>
<tr>
<td>root_ldisk</td>
<td>c0t011</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0001</td>
</tr>
<tr>
<td>root_ldisk</td>
<td>c16t011</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0001</td>
</tr>
<tr>
<td>d3</td>
<td>c0t110</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0006</td>
</tr>
<tr>
<td>d3</td>
<td>c16t110</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0006</td>
</tr>
<tr>
<td>d4</td>
<td>c0t111</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0007</td>
</tr>
<tr>
<td>d4</td>
<td>c16t111</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0007</td>
</tr>
<tr>
<td>d5</td>
<td>c0t112</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0008</td>
</tr>
<tr>
<td>d5</td>
<td>c16t112</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0008</td>
</tr>
<tr>
<td>d6</td>
<td>c0t113</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0009</td>
</tr>
<tr>
<td>d6</td>
<td>c16t113</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0009</td>
</tr>
<tr>
<td>d7</td>
<td>c0t114</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>000A</td>
</tr>
<tr>
<td>d7</td>
<td>c16t114</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>000A</td>
</tr>
<tr>
<td>&lt;... removed ...&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d377</td>
<td>c1t816</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>017C</td>
</tr>
<tr>
<td>d377</td>
<td>c17t816</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>017C</td>
</tr>
<tr>
<td>rootd378</td>
<td>c1t817</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0180</td>
</tr>
<tr>
<td>rootd378</td>
<td>c17t817</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0180</td>
</tr>
<tr>
<td>rootd379</td>
<td>c1t818</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0181</td>
</tr>
<tr>
<td>rootd379</td>
<td>c17t818</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0181</td>
</tr>
<tr>
<td>rootd380</td>
<td>c1t819</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0182</td>
</tr>
<tr>
<td>rootd380</td>
<td>c17t819</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0182</td>
</tr>
<tr>
<td>rootd381</td>
<td>c1t8110</td>
<td>16C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0183</td>
</tr>
<tr>
<td>rootd381</td>
<td>c17t8110</td>
<td>01C</td>
<td>FA</td>
<td>0</td>
<td>On</td>
<td>0</td>
<td>On</td>
<td>000187940268</td>
<td>0183</td>
</tr>
</tbody>
</table>

Scsi Device Table

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>type</th>
<th>info</th>
<th>stor_id</th>
<th>stor_dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>gk01</td>
<td>c0t0115</td>
<td>disk</td>
<td>56706817D480</td>
<td>000187940268</td>
<td>0000</td>
</tr>
<tr>
<td>gk0161</td>
<td>c16t0115</td>
<td>disk</td>
<td>56706817D330</td>
<td>000187940268</td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:** This is a partial display due to the length of the output.
The server Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>A unique name for each device in the chain.</td>
</tr>
<tr>
<td>addr</td>
<td>SCSI chain, target, and LUN information.</td>
</tr>
<tr>
<td>Director num</td>
<td>Director number. This output is applicable for Symmetrix storage systems only.</td>
</tr>
<tr>
<td>type</td>
<td>Device type, as specified in the SCSI spec. for peripherals. This output is applicable for Symmetrix storage systems only.</td>
</tr>
<tr>
<td>Port num</td>
<td>Port number. This output is applicable for Symmetrix storage systems only.</td>
</tr>
<tr>
<td>sts</td>
<td>Indicates the port status. Possible values are: On, Off, WD (write disabled), and NA. This output is applicable for Symmetrix storage systems only.</td>
</tr>
<tr>
<td>stor_id</td>
<td>Storage system ID.</td>
</tr>
<tr>
<td>stor_dev</td>
<td>Storage system device ID.</td>
</tr>
</tbody>
</table>

**EXAMPLE #2** For the VNX, to list all SCSI-attached non-disk devices, type:

```
$ server_devconfig server_2 -list -scsi -nondisks
```

server_2 :

Scsi Device Table

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>type</th>
<th>info</th>
</tr>
</thead>
<tbody>
<tr>
<td>gk01</td>
<td>c0t0l0</td>
<td>disk</td>
<td>5 020700000000APM00043807043</td>
</tr>
<tr>
<td>ggk01</td>
<td>c0t1l0</td>
<td>disk</td>
<td>5 020710001000APM00043807043</td>
</tr>
<tr>
<td>gk161</td>
<td>c16t1l1</td>
<td>disk</td>
<td>5 020711001100APM00043807043</td>
</tr>
</tbody>
</table>

For the VNX with a Symmetrix storage system, to list all SCSI-attached non-disk devices, type:

```
$ server_devconfig server_2 -list -scsi -nondisks
```

server_2 :

Scsi Device Table

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>type</th>
<th>info</th>
</tr>
</thead>
<tbody>
<tr>
<td>gk01</td>
<td>c0t0l15</td>
<td>disk</td>
<td>56706817D480 000187940268</td>
</tr>
<tr>
<td>gk161</td>
<td>c16t0l15</td>
<td>disk</td>
<td>56706817D330 000187940268</td>
</tr>
</tbody>
</table>

For info=56706817D480, the following breakdown applies:

<table>
<thead>
<tr>
<th>5670</th>
<th>68</th>
<th>17D</th>
<th>48</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symm code</td>
<td>Last 2 digits in the Symm S/N</td>
<td>Symm Device ID#</td>
<td>Symm SA #</td>
<td>SA Port # (0=a, 1=b)</td>
</tr>
</tbody>
</table>

**EXAMPLE #3** To rename a device, type:

```
$ server_devconfig server_2 -rename gk161 gk201
```

server_2 : done
EXAMPLE #4  For the VNX, to discover SCSI disk devices, without saving them to the database table, type:

```
$ server_devconfig server_2 -probe -scsi -disks
```

```
server_2 :
SCSI disk devices :
chain= 0, scsi-0
  stor_id= APM00043807043  celerra_id= APM000438070430000
  tid/lun= 0/0 type= disk sz= 11263 val= 1 info= DGC RAID 5 02070000000000NI
  tid/lun= 0/1 type= disk sz= 11263 val= 2 info= DGC RAID 5 02070100010001NI
  tid/lun= 0/2 type= disk sz= 2047 val= 3 info= DGC RAID 5 02070200020002NI
  tid/lun= 0/3 type= disk sz= 2047 val= 4 info= DGC RAID 5 02070300030003NI
  tid/lun= 0/4 type= disk sz= 2047 val= 5 info= DGC RAID 5 02070400040004NI
  tid/lun= 0/5 type= disk sz= 2047 val= 6 info= DGC RAID 5 02070500050005NI
  tid/lun= 0/6 type= disk sz= 2047 val= 7 info= DGC RAID 5 02070600060006NI
  tid/lun= 1/0 type= disk sz= 245625 val= 8 info= DGC RAID 5 02070700070007NI
  tid/lun= 1/1 type= disk sz= 0 val= -5 info= DGC RAID 5 02070800080008NI
  tid/lun= 1/2 type= disk sz= 273709 val= 9 info= DGC RAID 5 02070900090009NI
  tid/lun= 1/3 type= disk sz= 0 val= -5 info= DGC RAID 5 02070A000A000ANI
  tid/lun= 1/4 type= disk sz= 273709 val= 10 info= DGC RAID 5 02070B000B000BNI
  tid/lun= 1/5 type= disk sz= 0 val= -5 info= DGC RAID 5 02070C000C000CNI
  tid/lun= 1/6 type= disk sz= 273709 val= 11 info= DGC RAID 5 02070D000D000DNI
  tid/lun= 1/7 type= disk sz= 0 val= -5 info= DGC RAID 5 02070E000E000ENI
  tid/lun= 1/8 type= disk sz= 273709 val= 12 info= DGC RAID 5 02070F000F000FNI
  chain= 1, scsi-1 : no devices on chain
chain= 2, scsi-2 : no devices on chain
chain= 3, scsi-3 : no devices on chain
chain= 4, scsi-4 : no devices on chain
chain= 5, scsi-5 : no devices on chain
chain= 6, scsi-6 : no devices on chain
chain= 7, scsi-7 : no devices on chain
chain= 8, scsi-8 : no devices on chain
chain= 9, scsi-9 : no devices on chain
chain= 10, scsi-10 : no devices on chain
chain= 11, scsi-11 : no devices on chain
chain= 12, scsi-12 : no devices on chain
chain= 13, scsi-13 : no devices on chain
chain= 14, scsi-14 : no devices on chain
chain= 15, scsi-15 : no devices on chain
```

For the VNX with a Symmetrix storage system, to discover SCSI disk devices, without saving them to the database table, type:

```
$ server_devconfig server_2 -probe -scsi -disks
```

```
server_2 :
SCSI disk devices :
chain= 0, scsi-0
  stor_id= 000190102173  celerra_id= 0001901021730041
  tid/lun= 0/0 type= disk sz= 11507 val= 1 info= 577273041291SI00041
  tid/lun= 0/1 type= disk sz= 11507 val= 2 info= 577273042291SI00042
```
The server Commands

Note: This is a partial listing due to the length of the output.
EXAMPLE #5  To discover and save all SCSI devices, type:

$ server_devconfig server_2 -create -scsi -all

Discovering storage (may take several minutes)
server_2 : done

EXAMPLE #6  To discover and save all non-disk devices, type:

$ server_devconfig server_2 -create -scsi -nondisks

Discovering storage (may take several minutes)
server_2 : done

EXAMPLE #7  To save all SCSI devices with the discovery operation disabled, and
display information regarding the progress, type:

$ server_devconfig ALL -create -scsi -all -discovery n
-mirror y

server_2 :
server_2:
chain 0 ..........
chain 16 ..... done
server_3 :
server_3:
chain 0 ..........
chain 16 ..... done
server_4 :
server_4:
chain 0 ..........
chain 16 ..... done
server_5 :
server_5:
chain 0 ..........
chain 16 ..... done
**server_df**

Reports free and used disk space and inodes for mounted file systems on the specified Data Movers.

**SYNOPSIS**

```
server_df {<movername>|ALL} [-inode] [<pathname>|<fs_name>]
```

**DESCRIPTION**

The `server_df` command reports the amount of used and available disk space for a Data Mover or file system, how much of a file system’s total capacity has been used, and the number of used and free inodes.

The **ALL** option executes the command for all Data Movers.

**OPTIONS**

No arguments

Displays the amount of disk space in kilobytes used by file systems.

- **inode**

Reports used and free inodes.

```
server_df [-inode] [<pathname>|<fs_name>]
```

Gets file system information. If `<fs_name>` specified, gets information for file system only.

**SEE ALSO**

`Managing Volumes and File Systems for VNX Manually`, `nas_disk`, and `nas_fs`.

**EXAMPLE #1**

To display the amount of used and available disk space on a Data Mover, type:

```
$ server_df server_2
```

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>kbytes</th>
<th>used</th>
<th>avail</th>
<th>capacity</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>ufs1</td>
<td>1075686032</td>
<td>477816</td>
<td>1075208216</td>
<td>0%</td>
<td>/ufs1</td>
</tr>
<tr>
<td>ufs4</td>
<td>101683184</td>
<td>584</td>
<td>101682600</td>
<td>0%</td>
<td>/nmfs1/ufs4</td>
</tr>
<tr>
<td>ufs2</td>
<td>206515184</td>
<td>600</td>
<td>206514584</td>
<td>0%</td>
<td>/nmfs1/ufs2</td>
</tr>
<tr>
<td>nmfs1</td>
<td>308198368</td>
<td>1184</td>
<td>308197184</td>
<td>0%</td>
<td>/nmfs1</td>
</tr>
<tr>
<td>root_fs_common</td>
<td>13624</td>
<td>5264</td>
<td>8360</td>
<td>39%</td>
<td>/.etc_common</td>
</tr>
<tr>
<td>root_fs_2</td>
<td>114592</td>
<td>760</td>
<td>113832</td>
<td>1%</td>
<td>/</td>
</tr>
</tbody>
</table>

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filesystem</td>
<td>Name of the file system.</td>
</tr>
<tr>
<td>kbytes</td>
<td>Total amount of space in kilobytes for the file system.</td>
</tr>
<tr>
<td>used</td>
<td>Amount of kilobytes used by the file system.</td>
</tr>
</tbody>
</table>
EXAMPLE #2  To display the amount of disk space and the amount of free and unused inodes on a Data Mover, type:

```
$ server_df server_2 -inode
```

```
system_2 :
Filesystem      inodes  used  avail   capacity Mounted on
ufs1            131210494 140  131210354  0%    /ufs1
ufs4            25190398 10   25190388  0%    /nmfs1/ufs4
ufs2            25190398 11   25190387  0%    /nmfs1/ufs2
nmfs1           50380796 21   50380775  0%    /nmfs1
root_fs_common  21822    26   21796   0%    /.etc_common
root_fs_2       130942   66   130876   0%    /
```

### Value | Definition
--- | ---
avail | Amount of space in kilobytes available for the file system.
capacity | Percentage capacity that is used.
Mounted on | Mount point of the file system.

EXAMPLE #3  To display the amount of disk space and the amount of free and unused inodes on a file system, type:

```
$ server_df server_2 -inode ufs1
```

```
system_2 :
Filesystem      inodes  used  avail   capacity Mounted on
ufs1            131210494 140  131210354  0%    /ufs1
```

**server_dns**

Manages the Domain Name System (DNS) lookup server configuration for the specified Data Movers.

**SYNOPSIS**

```
server_dns {<movername>|ALL} [-protocol {tcp|udp}] <domainname> {<ip_addr>,...}
| [-delete <domainname>]
| [-option {start|stop|flush|dump}]
```

**DESCRIPTION**

`server_dns` provides connectivity to the DNS lookup servers for the specified Data Movers to convert hostnames and IP addresses. Up to three DNS lookup servers are supported for each domain on the Data Mover.

`server_dns` also provides the ability to clear the cache that has been saved on the Data Mover as a result of the DNS lookup servers.

The **ALL** option executes the command for all Data Movers.

**OPTIONS**

No arguments

Displays the DNS configuration.

```
-protocol {tcp|udp} <domainname> {<ip_addr>,...}
```

Sets the protocol for the DNS lookup servers (*udp* is the default).

```
<domainname> {<ip_addr>,...}
```

Creates a list of up to three IP addresses to be used as the DNS lookup servers for the specified `<domainname>`.

```
-delete <domainname>
```

Deletes the DNS lookup servers in the DNS domain name.

```
-option {start|stop|flush|dump}
```

The `start` option activates the link for the DNS lookup servers. The `stop` option halts access to the DNS lookup servers. After DNS service has been halted, the `flush` option can be used to clear the cache that has been saved on the Data Mover, and the `dump` option displays the DNS cache.

**SEE ALSO**

*Configuring VNX Naming Services* and `server_nis`.

**EXAMPLE #1**

To connect to a DNS lookup server, type:

```
$ server_dns server_2 prod.emc.com 172.10.20.10
server_2 : done
```
The server Commands

EXAMPLE #2  To display the DNS configuration, type:

server_2 :
DNS is running.
prod.emc.com
proto:udp server(s):172.10.20.10

EXAMPLE #3  To change the protocol to TCP from UDP, type:

server_2 : done

EXAMPLE #4  To halt access to the DNS lookup servers, type:

server_2 : done

EXAMPLE #5  To flush the cache on a Data Mover, type:

server_2 : done

EXAMPLE #6  To dump the DNS cache, type:

server_2 :
DNS cache size for one record type: 64
DNS cache includes 6 item(s):
dm102-cge0.nasdocs.emc.com
  Type:A         TTL=184 s  dataCount:1
  172.24.102.202 (local subnet)
---
winserver1.nasdocs.emc.com
  Type:A         TTL=3258 s  dataCount:1
  172.24.103.60
---
ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.nasdocs.emc.com
  Type:SRV       TTL=258 s  dataCount:1
  priority:0      weight:100    port:389   server:winserver1.nasdocs.emc.com
---
kberberos._tcp.Default-First-Site-Name._sites.dc._msdcs.nasdocs.emc.com
  Type:SRV       TTL=258 s  dataCount:1
  priority:0      weight:100    port:88    server:winserver1.nasdocs.emc.com
---
Expired item(s): 2

EXAMPLE #7  To delete the DNS lookup servers, type:

server_2 : done

$ server_dns server_2
$ server_dns server_2 -protocol tcp prod.emc.com
$ server_dns server_2 -option stop
$ server_dns server_2 -option flush
$ server_dns server_2 -option dump
$ server_dns server_2 -delete prod.emc.com
server_export

Exports file systems, and manages access on the specified Data Movers for NFS and CIFS clients.

SYNOPSIS

server_export {<movername>|ALL}

operations on all cifs and/or nfs entries:

| [-Protocol {cifs|nfs}] -list -all
| [-Protocol {cifs|nfs}] -all
| [-Protocol {cifs|nfs}] -unexport [-perm] -all

nfs operations per entry:

| -list <pathname>
| [-Protocol nfs [-name <name>]][-ignore][-option <options>]
  [-comment <comment>] <pathname>
| -unexport [-perm] <pathname>

cifs operations per entry:

| -list -name <sharename> [-option <options>]
| -name <sharename> [-ignore][-option <options>][-comment <comment>] <pathname>
| -unexport -name <sharename> [-option <options>]

DESCRIPTION

server_export provides user access by exporting an NFS pathname, or creating a CIFS share. Allows specification of multiple clients identified by hostnames or network and subnet addresses separated by a colon.

server_export removes access by unexporting an NFS pathname, deleting a CIFS share, and displaying the exported entries and available shares for the specified Data Mover.

The ALL option executes the command for all Data Movers.

Note: NFSv4 does not support the -alias option.

GENERAL OPTIONS FOR CIFS AND NFS OPERATIONS

No arguments
Displays all exported NFS entries and CIFS shares.

[-Protocol {cifs|nfs}] -list -all
Lists all exported entries as defined by the protocol. The default is NFS.

[-Protocol {cifs|nfs}] -all
Exports all entries on a Data Mover as defined by the protocol. The default is NFS.
The server Commands

[-Protocol {cifs|nfs}] -unexport [-perm] -all
Unexports all entries as defined by the protocol. By default, unexports are permanent for CIFS, and temporary for NFS, unless -perm is specified. If -perm is specified, removes all entries from the export table. When the entry is temporarily unexported, clients are denied access to the entry until it is re-exported or the system is rebooted, but the entries are not removed from the export table. The default is NFS.

FOR NFS OPERATIONS

-list <pathname>
Lists a specific NFS entry. If there are extra spaces in the <pathname>, the entire pathname must be enclosed by quotes. By using the server_export command, IPv6 addresses can be specified and the hosts configured with these addresses can mount and access file systems over NFS.

Note: If you are configuring an IPv6 address for ro, rw, access, and root, it must be enclosed in [ ] or square brackets. This is to properly handle the colon used to separate entries. Link local addresses are not supported.

-Protocol nfs [-name <name>] <pathname>
Exports an NFS <pathname> by default as read-write for everyone. If specified, assigns an optional file system name for the <name>. Pathname length is limited to 1024 bytes (represented as 1024 ASCII characters or a variable number of Unicode multibyte characters), and must be enclosed by quotes, if spaces are used. Name length is limited to 255 bytes.

Note: In a nested mount file system hierarchy, users can export the mount point path of the component file system. Subdirectories of the component file system cannot be exported. In a multilevel file system hierarchy, users can export any part of a file system independent of existing exports.

-ignore <pathname>
Overwrites previous options and comments in the export table for the entry.

-comment <comment> <pathname>
Adds a comment for the specified NFS export entry. The comment is displayed when listing the exported entries.

-option <options> <pathname>
Specifies the following comma-separated options:
### The server Commands

**sec=** [sys | krb5 | krb5i | krb5p] : <mode> [, <mode>, … ]

Specifies a user authentication or security method with an access mode setting. The **sys** (default) security option specifies AUTH_SYS security. The access mode can be one, or a combination of the following: **ro**, **rw=**, **ro=**, **root=**, **access=**, **anon=**, **webroot**, **public**.

If the **sec** option is specified, it must always be the first option specified in the string.

**krb5** security specifies Kerberos user and data authentication.

**krb5i** checks for the integrity of the data by adding a signature to each NFS packet and **krb5p** encrypts the data before sending it over the network.

For **krb5**, **krb5i**, and **krb5p** security, the access mode can be one, or a combination of the following: **ro**, **rw=**, **ro=**, **root=**, **access=**.

**ro**

Exports the <pathname> for all NFS clients as read-only.

**ro=**<client>[:<client>]...

Exports the <pathname> for the specified NFS clients as read-only.

**Note**: If **<client>** is an IPv6 address, it must be enclosed in square brackets or [ ].

**ro=**<client>[:-<client>]...

Excludes the specified NFS clients from ro privileges. Clients must be preceded with dash (-) to specify exclusion.

**Note**: If **<client>** is an IPv6 address, it must be enclosed in square brackets or [ ].

**rw=**<client>[:<client>]...

Exports the <pathname> as read-mostly for the specified NFS clients. Read-mostly means exported read-only to most machines, but read-write to those specified. The default is read-write to all.

**Note**: If **<client>** is an IPv6 address, it must be enclosed in square brackets or [ ].
**The server Commands**

- **$\texttt{rw}=<\text{client}>::<\text{client}>...**
  Excludes the specified NFS clients from rw privileges. The description of read-mostly provides information. Clients must be preceded with - (dash) to specify exclusion.

  **Note:** If $<\text{client}>$ is an IPv6 address, it must be enclosed in square brackets or [ ].

- **$\texttt{root}=<\text{client}>::<\text{client}>...**
  Provides root privileges for the specified NFS clients. By default, no host is granted root privilege.

  **Note:** If $<\text{client}>$ is an IPv6 address, it must be enclosed in square brackets or [ ].

- **$\texttt{root}=<-\text{client}>::<-\text{client}>...**
  Excludes the specified NFS clients from root privileges. Clients must be preceded with - (dash) to specify exclusion.

  **Note:** If $<\text{client}>$ is an IPv6 address, it must be enclosed in square brackets or [ ].

- **$\texttt{anon}=<$uid$>**
  If a request comes from an unknown user, the UID should be used as the effective user ID. Root users (UID =0) are considered “unknown” by the NFS server unless they are included in the root option. The default value for anon=$<\text{uid}>$ is the user “nobody.” If the user “nobody” does not exist, then the value 65534 is used.

  **CAUTION**
  Using anon=0 is not recommended for security concerns.

- **$\texttt{access}=<\text{client}>::<\text{client}>...**
  Provides mount access for the specified NFS clients.

  **Note:** If $<\text{client}>$ is an IPv6 address, it must be enclosed in square brackets or [ ].
access=<client>[:<client>]...
Excludes the specified NFS clients from access even if they are part of a subnet or netgroup that is allowed access. Clients must be preceded with - (dash) to specify exclusion.

Note: If <client> is an IPv6 address, it must be enclosed in square brackets or [ ].

nfsv4only
Specifies that the NFS export can be accessed only when a client is using NFSv4.

vlan=<vlanid>[,<vlanid>,...]
Specifies that all hosts belonging to the VLAN ID will have access to the exported file system. Hosts on other VLANs will be denied.

The VLANs are separated by : or colon, just as any other server_export option values.

Clients can be a hostname, netgroup, subnet, or IP address and must be colon-separated, without spaces. A subnet is an IP address/netmask (for example, 168.159.50.0/255.255.255.0). A hostname is first checked for in the Data Mover’s local hosts database, then on the NIS (host database) or DNS server (if enabled). A netgroup is searched in the local netgroup database and then on the NIS server’s netgroup database. If the client name does not exist in any case, then access is denied.

Note: Netgroups are supported. The hosts and netgroup files can be created on the Control Station using your preferred method (for example, with an editor, or by copying from another node), then copied to the Data Mover.

-unexport  [-perm] <pathname>
Temporarily unexports a <pathname> unless -perm is specified. If -perm is specified, removes the entry from the export table.

FOR CIFS OPERATIONS

-list -name <sharename>
Displays the specified CIFS share.

[-option <options>]
Specifies the following comma-separated list of options:
[netbios=<netbios_name>]

When the share has an associated NetBIOS name that name is required to locate the entry. Multiple CIFS entries can have same <sharename> when belonging to a different NetBIOS name.

-name <sharename> [-ignore][-option <options>]

Creates a CIFS share. Share name length is limited to 12 ASCII characters unless Unicode is enabled, in which case the limit is 80 multibyte characters. Share names cannot include the following characters: /, \, %, ", NUL (Null character), STX (start of header), SOT (start of text), and LF (line feed). Share names can contain spaces and other nonalphanumeric characters, but must be enclosed by quotes if spaces are used. Share names cannot begin with a - (hyphen). Share names are case-sensitive.

Comment length is limited to 256 bytes (represented as 256 ASCII characters or a variable number of Unicode multibyte characters). A comment cannot include the following characters: NUL (Null character), STX (start of header), and SOT (start of text). Comments can contain spaces and other nonalphanumeric characters, but must be enclosed by quotes if spaces are used. Pathname length is limited to 1024 bytes.

The -ignore option overwrites the previous options and comment in the export table.

[-option <options>]

Specifies the following comma-separated options:

ro

Exports the <pathname> for CIFS clients as read-only.

rw=<client>[::<client>]...

Creates the share for CIFS clients as read-mostly. Read-mostly means shared read-only to most clients, but read-write to those specified. By default, the <pathname> is shared read-write to all. A client may be either a <user_name> or <group_name>.

Note: If <client> is an IPv6 address, it must be enclosed in [ ] or square brackets.

umask=<mask>

Specifies a user file-creation mask for the umask allowing NFS permissions to be determined for the share.
Note: The value of the umask is specified in octal and is XORed with the
permissions of 666 for files and 777 for directories. Common values
include 002, which gives complete access to the group, and read (and
directory search) access to others or 022 (default) which gives read (and
directory search), but not write permission to the group and others.

user=<default_user>
When using share level access (server_checkup provides
information), specifies a <default_user> which must be entered as
a character string. The user must be defined in the Data Mover’s
password file. There is a 20 character limit for the username.

group=<default_group>
When using share level access (server_checkup provides
information), indicates a <default_group> which must be entered
as a character string. There is a 256 character limit for group
names.

ropasswd=<share_passwd>
When using share level access (server_checkup provides
information), creates a read-only password to allow clients access
to the share. Passwords can be viewed in the list of shared entries.

rwpasswd=<share_rw_passwd>
When using share level access (server_checkup provides
information), creates a read-write password to allow clients
access to the share. Passwords are displayed in the list of shared
entries.

Note: Users from any client machine who know the value of the
ropasswd or rwpasswd can access the share for read-only and read-write
operations.

maxusr=<maxusr>
Sets the maximum number of simultaneous users permitted for a
share.

netbios=<netbiosName>[,,netbios=<netbiosName>]...
Associates a share on a single domain with one or more NetBIOS
names created with server_checkup. By default, if a NetBIOS
name is not specified for a share, the share is visible to all
NetBIOS names.
-comment
Adds a comment for the specified CIFS share. The comment is displayed when listing the shared entries.

-unexport  -name <sharename>
Permanently removes access to a share by removing the entry from the export table.

[-option <options>]
Specifies the following comma-separated options:

netbios=<netbios_name>
When the share has an associated NetBIOS name the NetBIOS name is required to locate the entry. This is required because multiple CIFS entries can have same <sharename> when belonging to different NetBIOS name.

SEE ALSO Configuring NFS on VNX, Managing Volumes and File Systems for VNX Manually, server_checkup, and server_mount.

EXAMPLE #1 To export a specific NFS entry, type:

$ server_export server_2 -Protocol nfs /ufs1
server_2 : done

EXAMPLE #2 To export an NFS entry and overwrite existing settings, type:

$ server_export server_2 -Protocol nfs -ignore -option access=172.24.102.0/255.255.255.0,root=172.24.102.240 -comment 'NFS Export for ufs1' /ufs1
server_2 : done

EXAMPLE #3 To export NFS entry dir1, a subdirectory of the exported entry /ufs1 in a multilevel file system hierarchy, type:

$ server_export server_2 -Protocol nfs /ufs1/dir1
server_2 : done

EXAMPLE #4 To assign a name to a NFS export, type:

$ server_export server_2 -Protocol nfs -name nasdocsfs /ufs1
server_2 : done

EXAMPLE #5 To export an NFS entry using Kerberos authentication, type:

$ server_export server_2 -Protocol nfs -option sec=krb5:ro,root=172.24.102.240,access=172.24.102.0/255.255.255.0 /ufs2
server_2 : done
EXAMPLE #6  To export an NFS entry for NFSv4 only, type:

```bash
$ server_export server_2 -Protocol nfs -option nfsv4only /ufs1
```

server_2 : done

EXAMPLE #7  To list all NFS entries, type:

```bash
$ server_export server_2 -Protocol nfs -list -all
```

server_2 :
export "/ufs2" sec=krb5 ro root=172.24.102.240 access=172.24.102.0/255.255.255.0
export "/ufs1" name="/nasdocsfs" access=172.24.102.0/255.255.255.0
root=172.24.102.240 nfsv4only comment="NFS Export for ufs1"
export "/" anon=0

EXAMPLE #8  To list NFS entries for the specified path, type:

```bash
$ server_export server_2 -list /ufs1
```

server_2 :
export "/ufs1" name="/nasdocsfs" access=172.24.102.0/255.255.255.0
root=172.24.102.240 nfsv4only comment="NFS Export for ufs1"

EXAMPLE #9  To temporarily unexport an NFS entry, type:

```bash
$ server_export server_2 -Protocol nfs -unexport /ufs2
```

server_2 : done

EXAMPLE #10  To export all NFS entries, type:

```bash
$ server_export server_2 -Protocol nfs -all
```

server_2 : done

EXAMPLE #11  To export a specific NFS entry in a language that uses multibyte characters (such as Japanese), type:

```bash
$ server_export server_2 -Protocol nfs /<nfs_entry_in_local_language_text>
```

server_2 : done

EXAMPLE #12  To permanently unexport an NFS entry, type:

```bash
$ server_export server_2 -unexport -perm /ufs1
```

server_2 : done

EXAMPLE #13  To permanently unexport all NFS entries, type:

```bash
$ server_export server_2 -Protocol nfs -unexport -perm -all
```

server_2 : done

EXAMPLE #14  To provide access to a CIFS share, type:
The server Commands

$ server_export server_2 -name ufs1 /ufs1

server_2 : done

**EXAMPLE #15** To create a CIFS share and overwrite existing settings, type:

$ server_export server_2 -name ufs1 -ignore -option ro,umask=027,maxusr=200,netbios=dm112-cge0 -comment 'CIFS share' /ufs1

server_2 : done

**EXAMPLE #16** To create a CIFS share in a language that uses multibyte characters (such as Japanese), type:

$ server_export server_2 -P cifs -name <name_in _local _language_text> -comment <comment_in local_language_text> /accounting

server_2 : done

**EXAMPLE #17** To list all CIFS entries, type:

$ server_export server_2 -Protocol cifs -list

server_2 :
share "ufs1" "/ufs1" ro umask=027 maxusr=200 netbios=DM112-CGE0 comment="CIFS share"
share "ufs2" "/ufs2" umask=022 maxusr=4294967295

**EXAMPLE #18** To display a specific CIFS share, type:

$ server_export server_2 -list -name ufs1 -option netbios=dm112-cge0

server_2 :
share "ufs1" "/ufs1" ro umask=027 maxusr=200 netbios=DM112-CGE0 comment="CIFS share"

**EXAMPLE #19** To export all CIFS entries, type:

$ server_export server_2 -Protocol cifs -all

server_2 : done

**EXAMPLE #20** To list all NFS and CIFS entries, type:

$ server_export server_2

server_2 :
export "/ufs2" sec=krb5 ro root=172.24.102.240 access=172.24.102.0/255.255.255.0
export "/ufs1" nfsv4only
export "/" anon=0
share "ufs2" "/ufs2" umask=022 maxusr=4294967295
share "ufs1" "/ufs1" ro umask=027 maxusr=200 netbios=DM112-CGE0 comment="CIFS share"
The server Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>export</td>
<td>A file system entry to be exported.</td>
</tr>
<tr>
<td>sec</td>
<td>Security mode for the file system.</td>
</tr>
<tr>
<td>ro</td>
<td>File system is to be exported as read-only.</td>
</tr>
<tr>
<td>root</td>
<td>IP address with root access.</td>
</tr>
<tr>
<td>access</td>
<td>Access is permitted for those IP addresses.</td>
</tr>
<tr>
<td>share</td>
<td>Entry to be shared.</td>
</tr>
<tr>
<td>ro</td>
<td>File system is to be shared as read-only.</td>
</tr>
<tr>
<td>umask</td>
<td>User creation mask.</td>
</tr>
<tr>
<td>maxuser</td>
<td>Maximum number of simultaneous users.</td>
</tr>
<tr>
<td>netbios</td>
<td>NetBIOS name for the share.</td>
</tr>
<tr>
<td>comment</td>
<td>Comment specified for the share.</td>
</tr>
</tbody>
</table>

**EXAMPLE #21** To permanently unexport all CIFS and NFS entries, type:

```bash
$ server_export server_2 -unexport -perm -all
```

server_2 : done

**EXAMPLE #22** To delete a CIFS share, type:

```bash
$ server_export server_2 -unexport -name ufs1 -option netbios=dm112-cge0
```

server_2 : done

**EXAMPLE #23** To delete all CIFS shares, type:

```bash
$ server_export server_2 -Protocol cifs -unexport -all
```

server_2 : done

**EXAMPLE #24** To export a file system for NFS that specifies an IPv4 and IPv6 address, type:

```bash
$ server_export server_2 -Protocol nfs -option access=172.24.108.10:[1080:0:0:8:800:200C:417A] /fs1
```

server_2 : done

**EXAMPLE #25** To verify that the file system was exported, type:

```bash
$ server_export server_2 -list /fs1
```

server_2 :

export "/fs1" access=172.24.108.10:[1080:0:0:8:800:200C:417A]

**EXAMPLE #26** To export a file system for NFS that specifies two IPv6 addresses, type:
Example #27

To verify that the file system was exported, type:

```
$ server_export server_2 -list /fs1
```

server_2:
export "/fs1" rw=[1080:0:0:8:80:200C:417A]:[1080:0:0:8:800:200C:417B]
**server_file**

Copies files between the Control Station and the specified Data Movers.

**SYNOPSIS**

```
server_file {<movername>|ALL} {-get|-put} <src_file> <dst_file>
```

**DESCRIPTION**

`server_file` copies the source file from the specified Data Mover (or Control Station) to the destination file on the Control Station (or specified Data Mover). The `<src_file>` indicates the source file, and the name `<dst_file>` indicates destination file. By default, if a directory is not specified on the Data Mover, the `/etc` directory is used.

The **ALL** option executes the command for all Data Movers.

**OPTIONS**

- **-get** `<src_file>` `<dst_file>`
  Copies the source file on Data Mover to destination file on Control Station. Both the `<src_file>` and `<dst_file>` may be full pathnames.

- **-put** `<src_file>` `<dst_file>`
  Copies source file on the Control Station to the destination file on the Data Mover. Both the `<src_file>` and `<dst_file>` must be full pathnames.

**CAUTION**

This command overwrites existing files of the same name without notification. Use care when copying files.

**EXAMPLE #1**

To copy a file from the Control Station to a Data Mover, type:

```
$ server_file server_2 -put passwd passwd
server_2 : done
```

**EXAMPLE #2**

To copy a file from the Data Mover to the Control Station, type:

```
$ server_file server_2 -get passwd /home/nasadmin/passwd
server_2 : done
```
server_ftp

Configures the FTP server configuration for the specified Data Movers.

SYNOPSIS

server_ftp {<movername>|ALL}
-service {-status|-start|-stop|{-stats [-all|-reset]}}
| -info
| -modify
  [-controlport <controlport>]
  [-dataport <dataport>]
  [-defaultdir <path>]
  [-homedir {enable|disable}]
  [-keepalive <keepalive>]
  [-highwatermark <highwatermark>]
  [-lowwatermark <lowwatermark>]
  [-deniedusers [<path>]]
  [-welcome [<path>]]
  [-motd [<path>]]
  [-timeout <timeout>]
  [-maxtimeout <maxtimeout>]
  [-readsize <readsize>]
  [-writesize <writesize>]
  [-maxcnx <maxcnx>]
  [-umask <umask>]
  [-sslcontrol {no|allow|require|requireforauth}]
  [-ssldata {allow|require|deny}]
  [-sslpersona {anonymous|default|<persona_name>}]
  [-sslprotocol {default|ssl3|tls1|all}]
  [-sslcipher {default|<cipherlist}>]
  [-sslcontrolport <sslcontrolport>]
  [-ssldataport <ssldataport>]

DESCRIPTION

server_ftp configures the ftp daemon. Optional SSL security support is available. The modifications are performed when the ftp daemon is stopped and are reflected after restart of the ftp daemon. There is no need to reboot the Data Mover for the changes to be reflected.

OPTIONS

server_ftp {<movername>|ALL}
Sends a request to the Data Mover to get all the parameters of the ftp daemon.

ALL option executes the command for all Data Movers.

-service {-status|-start|-stop|{-stats [-all|-reset]}}
The server Commands

- **status**
  Retrieves the current status of the ftp daemon.

- **start**
  Starts the ftp daemon. The `start` option persists after the daemon is rebooted.

- **stop**
  Stops the ftp daemon.

- **stats [all | reset]**
  Displays the statistics of the ftp daemon. The reset option resets all the ftp server statistics. The `all` option displays detailed statistics.

- **info**
  Retrieves all the parameters for the ftp daemon along with its current status.

- **modify**
  Modifies the ftp daemon configuration. The ftp daemon has to be stopped to carry out the changes. The modifications are taken into account when the service is restarted.

  **-controlport** `<controlport>`
  Sets the local tcp port for control connections. By default, the port is 21. When control port is set to 0, it disables unsecure ftp usage and validates port for implicit secure connection on SSL port (default 990).

  **Note:** This default port can be changed using the sslcontrolport option.

  **-dataport** `<dataport>`
  Sets the local tcp port for active data connections. By default, the port is 20. When `<dataport>` is set to 0, the port is allocated dynamically by the server in active mode.

  **-defaultdir** `<path>`
  Sets the default user directory when the user home directory is not accessible. This option replaces "ftpd.defaultdir." By default, "/" is used.

  **-homedir {enable | disable}**
  Restricts or allows user access to their home directory tree. When enabled the user is allowed access to their home directory only. If
the user home directory is not accessible, access is denied. During connection the user is denied access to data outside of their home directory space. By default, this feature is disabled.

Note: Using FTP on VNX provides more information about how the home directory of a user is managed.

- **umask** `<umask>`
  Defines the mask to set the mode bits on file or directory creation. By default the mask is 027, which means that `xrw-r----` mode bits are assigned.

- **keepalive** `<keepalive>`
  Sets TCP keepalive value for the ftp daemon. This value is given in seconds. By default, the value is 60. The value 0 disables the TCP keepalive option. The maximum value is 15300 (255 minutes).

- **highwatermark** `<highwatermark>`
  Sets TCP high watermark value (amount of data stored without knowledge of the client) for the ftp daemon. By default, the value is 65536. The minimum value is 8192, and the maximum value is 1048576 (1 MB).

**CAUTION**
Do not modify this parameter without a thorough knowledge of the impact on FTP client behavior.

- **lowwatermark** `<lowwatermark>`
  Sets TCP low watermark value (amount of the data to be added, after the highwatermark has been reached and new data can be accepted from the client) for the ftp daemon. The minimum value is 8192, maximum value is 1048576 (1 MB), and default value is 32768.

**CAUTION**
Do not modify this parameter without a thorough knowledge of the impact on FTP client behavior.
-deniedusers  <deniedusers_file>
Denies FTP access to specific users on a Data Mover. Specifies the path and text file containing the list of usernames to be denied access. Places each username on a separate line. By default, all users are allowed.

-welcome  <welcome_file>
Specifies path of the file to be displayed on the welcome screen. For example, this file can display a login banner before the user is requested for authentication data. By default, no welcome message is displayed.

-motd  <motd_file>
Specifies path of the file to be displayed on the welcome screen. Users see a welcome screen ("message of the day") after they successfully log in. By default, no message of the day is displayed.

-timeout  <timeout>
Specifies the default inactivity time-out period (when not set by the client). The value is given in seconds. After the specified time if there is no activity, the client is disconnected from the server and will have to re-open a connection. By default, the <timeout> value is 900 seconds. The minimum value is 10 seconds, and the maximum value is 7200.

-maxtimeout  <maxtimeout>
Sets the maximum time-out period allowed by the client. The value is given in seconds and any value larger than maximum time-out period is not allowed. By default, the <maxtimeout> value is 7200 seconds. The minimum value is 10 seconds, and the maximum value is 7200.

-readsize  <readsize>
Sets the size for reading files from the disk. The value must be greater than 8192, and a multiple of 8K. By default, the <readsize> is 8192 bytes. The minimum value is 8192, and the maximum value is 1048576 (1 MB).

-writesize  <writesize>
Sets the size for writing files from the disk. The value must be greater than 8192, and a multiple of 8K. By default, the <writesize> is 49152 (48 KB). The minimum value is 8192, and the maximum value is 1048576 (1 MB).
**-maxcnx** `<maxcnx>`
Sets the maximum number of control connections the ftp daemon will support. By default, the `<maxcnx>` value is set to 65535 (64K-1). The minimum value is 1, and the maximum value is 65535 (64K-1).

**-sslcontrol** `{no|allow|require|requireforauth}`
Uses SSL for the ftp control connection depending on the attributes specified. By default, SSL is disabled. The `no` option disables SSL control. The `allow` option specifies that SSL is enabled, but the user can still connect without SSL. The `require` option specifies that SSL is required for the connection. The `requireforauth` option specifies that SSL is required for authentication. The control path goes back to unsecure after this authentication. When the client is behind a firewall, this helps the firewall to filter the ftp commands requiring new port access.

**Note:** Before the server can be configured with SSL, the Data Mover must be set up with a private key and a public certificate. This key and certificate are identified using a `persona`. In addition, the necessary Certificate Authority (CA) certificates used to identify trusted servers must be imported into the Data Mover. Use the system’s PKI feature to manage the use of certificates prior to configuring SSL operation.

**-ssldata** `{no|allow|require}`
Uses SSL for the data connection depending on the attributes specified. The `no` option disables SSL. The `allow` option specifies that SSL is enabled, but the user can also transfer data without SSL. The `require` option specifies that SSL is required for data connection. The `ssldata` value cannot be set to `allow` or `require` if `sslcontrol` is set to `no`. By default, SSL is disabled.

**Note:** These options are set on the server but are dependent on ftp client capabilities. Some client capabilities may be incompatible with server settings. *Using FTP on VNX* provides information on validating compatibility.

**-sslpersona** `{anonymous|default|<persona_name>}`
Specifies the persona associated with the Data Mover. Personas are used to identify the private key and public certificate used by SSL. The default value specified is `default` (each Data Mover is configured with a persona named default). The `anonymous` value specifies that SSL can operate without using a certificate. This implies that the communication between client and server is encrypted and data integrity is guaranteed.
The server Commands

Note: Use `server_certificate` to configure the persona before using `server_ftp`.

```
-sslprotocol {default|ssl3|tls1|all}
```

Specifies the SSL protocol version that the ftp daemon on the server accepts:

- `ssl3` — Only SSLv3 connections
- `tls1` — Only TLSv1 connections
- `all` — Both SSLv3 and TLSv1 connections
- `default` — Uses the value set in the `ssl.protocol` parameter which, by default, is 0 (SSLv3 and TLSv1)

```
-sslcipher {default|<cipherlist>}
```

Specifies the SSL cipher suite. The value of default is the value set in the `ssl.cipher` parameter. This value means that all ciphers are supported by VNX except the Anonymous Diffie-Hellman, NULL, and SSLv2 ciphers and that the supported ciphers are sorted by the size of the encryption key.

```
-sslcontrolport <sslcontrolport>
```

Sets the implicit control port for FTP connections over SSL. By default, the port is 990. To disable implicit FTP connections over SSL, the `<sslcontrolport>` must be set to 0.

```
-ssldataport <ssldataport>
```

Sets the local tcp port for active data connections using implicit FTP connections over SSL. By default, the port is 899. If the ssldataport is set to 0, the Data Mover will use a port allocated by the system.

**SEE ALSO**

“`server_certificate`”.

**EXAMPLE #1**

To retrieve all the parameters for the ftp daemon and its status, type:

```
$ server_ftp server_2 -info
```

```
FTP started
=========
controlport          21
dataport             20
defaultdir /         .etc/ftpd/pub
homeditr             disable
umask               027

tcp keepalive        1 minute
tcp high watermark   65536 bytes
tcp low watermark    32768 bytes
```
readsize 8192 bytes
writesize 49152 bytes
denied users file path /.etc/ftpd/conf/ftpusers
welcome file path /.etc/ftpd/conf/welcome
motd file path /.etc/ftpd/conf/motd

session timeout 900 seconds
max session timeoutQ 7200 seconds

Security Options
================
sslpersona default
sslprotocol default
sslcipher default

FTP over TLS explicit Options
------------------------------
sslcontrol SSL require for authentication
ssldata allow SSL

FTP over SSL implicit Options
-----------------------------
sslcontrolport 990
ssldataport 989

EXAMPLE #2  To display the statistics of the ftp daemon, type:

$ server_ftp server_2 -service -stats

<table>
<thead>
<tr>
<th>Login Type</th>
<th>Successful</th>
<th>Failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anonymous</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Unix</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CIFS</td>
<td>7</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data transfers</th>
<th>Count</th>
<th>min</th>
<th>average</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write Bin</td>
<td>10</td>
<td>10.00</td>
<td>19.00</td>
<td>20.00</td>
</tr>
<tr>
<td>Read Bin</td>
<td>0</td>
<td>-----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Write ASCII</td>
<td>2</td>
<td>1.00</td>
<td>1.50</td>
<td>2.00</td>
</tr>
<tr>
<td>Read ASCII</td>
<td>0</td>
<td>-----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>SSL Write Bin</td>
<td>5</td>
<td>5.00</td>
<td>17.00</td>
<td>18.00</td>
</tr>
<tr>
<td>SSL Read Bin</td>
<td>15</td>
<td>7.00</td>
<td>25.00</td>
<td>35.00</td>
</tr>
<tr>
<td>SSL Write ASCII</td>
<td>0</td>
<td>-----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>SSL Read ASCII</td>
<td>0</td>
<td>-----</td>
<td>----</td>
<td>----</td>
</tr>
</tbody>
</table>
The server Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput (MBytes/sec)</td>
<td>Throughput is calculated using the size of the file (Mbytes) divided by the duration of the transfer (in seconds).</td>
</tr>
<tr>
<td>average</td>
<td>Average is the average of the throughputs (sum of the throughputs divided by the number of transfers).</td>
</tr>
<tr>
<td>Data transfers</td>
<td>Defines the type of transfer.</td>
</tr>
<tr>
<td>Count</td>
<td>Number of operations for a transfer type.</td>
</tr>
<tr>
<td>min</td>
<td>Minimum time in milliseconds required to execute the operation (with regards to Data Mover).</td>
</tr>
<tr>
<td>max</td>
<td>Maximum time in milliseconds required to execute the operation (with regards to Data Mover).</td>
</tr>
</tbody>
</table>

EXAMPLE #3 To display the statistics of the ftp daemon with details, type:

```
$ server_ftp server_2 -service -stats -all
```

<table>
<thead>
<tr>
<th>Commands</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER</td>
<td>23</td>
</tr>
<tr>
<td>PASS</td>
<td>23</td>
</tr>
<tr>
<td>QUIT</td>
<td>23</td>
</tr>
<tr>
<td>PORT</td>
<td>45</td>
</tr>
<tr>
<td>EPRT</td>
<td>10</td>
</tr>
<tr>
<td>...</td>
<td>....</td>
</tr>
<tr>
<td>PEAT</td>
<td>23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SITE Commands</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMASK</td>
<td>0</td>
</tr>
<tr>
<td>IDLE</td>
<td>10</td>
</tr>
<tr>
<td>CHMOD</td>
<td>0</td>
</tr>
<tr>
<td>HELP</td>
<td>0</td>
</tr>
<tr>
<td>BANDWIDTH</td>
<td>0</td>
</tr>
<tr>
<td>KEEPALIVE</td>
<td>10</td>
</tr>
<tr>
<td>PASV</td>
<td>56</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OPTS Commands</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTF8</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Login Type</th>
<th>Successful</th>
<th>Failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anonymous</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Unix</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CIFS</td>
<td>7</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connections</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non secure</td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td>10</td>
</tr>
</tbody>
</table>
Data

Explicit SSL
---------
Control Auth  3
Control  8
Data  20

Implicit SSL
---------
Control  0
Data  0

<table>
<thead>
<tr>
<th>Data transfers</th>
<th>Count</th>
<th>min</th>
<th>average</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write Bin</td>
<td>10</td>
<td>10.00</td>
<td>19.00</td>
<td>20.00</td>
</tr>
<tr>
<td>Read Bin</td>
<td>0</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Write ASCII</td>
<td>2</td>
<td>1.00</td>
<td>1.50</td>
<td>2.00</td>
</tr>
<tr>
<td>Read ASCII</td>
<td>0</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>SSL Write Bin</td>
<td>5</td>
<td>5.00</td>
<td>17.00</td>
<td>18.00</td>
</tr>
<tr>
<td>SSL Read Bin</td>
<td>15</td>
<td>7.00</td>
<td>25.00</td>
<td>35.00</td>
</tr>
<tr>
<td>SSL Write ASCII</td>
<td>0</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>SSL Read ASCII</td>
<td>0</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commands</td>
<td>FTP protocol command name.</td>
</tr>
<tr>
<td>Count</td>
<td>Number of commands received by Data Mover.</td>
</tr>
<tr>
<td>SITE Commands</td>
<td>Class of command in FTP protocol.</td>
</tr>
<tr>
<td>POTS Commands</td>
<td>Class of command in FTP protocol.</td>
</tr>
</tbody>
</table>

**EXAMPLE #4** To retrieve the status of the ftp daemon, type:

```
$ server_ftp server_3 -service -status
```

server_3 : done

State : running

**EXAMPLE #5** To start the ftp daemon, type:

```
$ server_ftp server_2 -service -start
```

server_2 : done

**EXAMPLE #6** To stop the ftp daemon, type:

```
$ server_ftp server_2 -service -stop
```

server_2 : done
EXAMPLE #7  To set the local tcp port for the control connections, type:

```bash
$ server_ftp server_2 -modify -controlport 256
```

```
server_2 : done

FTPD CONFIGURATION
===================
State : stopped
Control Port : 256
Data Port : 20
Default dir : /
Home dir : disable
Keepalive : 1
High watermark : 65536
Low watermark : 32768
Timeout : 900
Max timeout : 7200
Read size : 8192
Write size : 49152
Umask : 27
Max connection : 65535
```

SSL CONFIGURATION
=================
Control channel mode : disable
Data channel mode : disable
Persona : default
Protocol : default
Cipher : default
Control port : 990
Data port : 989

EXAMPLE #8  To set the local tcp port for active data connections, type:

```bash
$ server_ftp server_2 -modify -dataport 257
```

```
server_2 : done

FTPD CONFIGURATION
===================
State : stopped
Control Port : 256
Data Port : 257
Default dir : /
Home dir : disable
Keepalive : 1
High watermark : 65536
Low watermark : 32768
Timeout : 900
Max timeout : 7200
Read size : 8192
Write size : 49152
Umask : 27
```
Max connection : 65535

SSL CONFIGURATION
==================
Control channel mode : disable
Data channel mode : disable
Persona : default
Protocol : default
Cipher : default
Control port : 990
Data port : 989

EXAMPLE #9  To change the default directory of a user when his home directory is not accessible, type:

$ server_ftp server_2 -modify -defaultdir /big

EXAMPLE #10  To allow users access to their home directory tree, type:

$ server_ftp server_2 -modify -homedir enable
## The server Commands

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>stopped</td>
</tr>
<tr>
<td>Control Port</td>
<td>256</td>
</tr>
<tr>
<td>Data Port</td>
<td>257</td>
</tr>
<tr>
<td>Default dir</td>
<td>/big</td>
</tr>
<tr>
<td>Home dir</td>
<td>enable</td>
</tr>
<tr>
<td>Keepalive</td>
<td>1</td>
</tr>
<tr>
<td>High watermark</td>
<td>65536</td>
</tr>
<tr>
<td>Low watermark</td>
<td>32768</td>
</tr>
<tr>
<td>Timeout</td>
<td>900</td>
</tr>
<tr>
<td>Max timeout</td>
<td>7200</td>
</tr>
<tr>
<td>Read size</td>
<td>8192</td>
</tr>
<tr>
<td>Write size</td>
<td>49152</td>
</tr>
<tr>
<td>Umask</td>
<td>27</td>
</tr>
<tr>
<td>Max connection</td>
<td>65535</td>
</tr>
</tbody>
</table>

### SSL CONFIGURATION

- **Control channel mode**: disable
- **Data channel mode**: disable
- **Persona**: default
- **Protocol**: default
- **Cipher**: default
- **Control port**: 990
- **Data port**: 989

### EXAMPLE #11

To restrict user access to their home directory tree, type:

```
$ server_ftp server_2 -modify -homedir disable
```

`server_2`: done

### FTPD CONFIGURATION

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>stopped</td>
</tr>
<tr>
<td>Control Port</td>
<td>256</td>
</tr>
<tr>
<td>Data Port</td>
<td>257</td>
</tr>
<tr>
<td>Default dir</td>
<td>/big</td>
</tr>
<tr>
<td>Home dir</td>
<td>disable</td>
</tr>
<tr>
<td>Keepalive</td>
<td>1</td>
</tr>
<tr>
<td>High watermark</td>
<td>65536</td>
</tr>
<tr>
<td>Low watermark</td>
<td>32768</td>
</tr>
<tr>
<td>Timeout</td>
<td>900</td>
</tr>
<tr>
<td>Max timeout</td>
<td>7200</td>
</tr>
<tr>
<td>Read size</td>
<td>8192</td>
</tr>
<tr>
<td>Write size</td>
<td>49152</td>
</tr>
<tr>
<td>Umask</td>
<td>27</td>
</tr>
<tr>
<td>Max connection</td>
<td>65535</td>
</tr>
</tbody>
</table>

### SSL CONFIGURATION

- **Control channel mode**: disable
- **Data channel mode**: disable
- **Persona**: default
Protocol : default
Cipher : default
Control port : 990
Data port : 989

**EXAMPLE #12**

To set the default umask for creating a file or a directory by means of the ftp daemon, type:

```
$ server_ftp server_2 -modify -umask 077
```

server_2 : done

**FTPD CONFIGURATION**

```
State : stopped
Control Port : 256
Data Port : 257
Default dir : /big
Home dir : disable
Keepalive : 1
High watermark : 65536
Low watermark : 32768
Timeout : 900
Max timeout : 7200
Read size : 8192
Write size : 49152
Umask : 77
Max connection : 65535
```

**SSL CONFIGURATION**

```
Control channel mode : disable
Data channel mode : disable
Persona : default
Protocol : default
Cipher : default
Control port : 990
Data port : 989
```

**EXAMPLE #13**

To set the TCP keepalive for the ftp daemon, type:

```
$ server_ftp server_2 -modify -keepalive 120
```

server_2 : done

**FTPD CONFIGURATION**

```
State : stopped
Control Port : 256
Data Port : 257
Default dir : /big
Home dir : disable
Keepalive : 120
High watermark : 65536
```
The server Commands

Low watermark : 32768
Timeout : 900
Max timeout : 7200
Read size : 8192
Write size : 49152
Umask : 77
Max connection : 65535

SSL CONFIGURATION
=================
Control channel mode : disable
Data channel mode : disable
Persona : default
Protocol : default
Cipher : default
Control port : 990
Data port : 989

EXAMPLE #14
To set the TCP high water mark for the ftp daemon, type:

$ server_ftp server_2 -modify -highwatermark 90112

server_2 : done

FTPD CONFIGURATION
==================
State : stopped
Control Port : 256
Data Port : 257
Default dir : /big
Home dir : disable
Keepalive : 120
High watermark : 90112
Low watermark : 32768
Timeout : 900
Max timeout : 7200
Read size : 8192
Write size : 49152
Umask : 77
Max connection : 65535

SSL CONFIGURATION
=================
Control channel mode : disable
Data channel mode : disable
Persona : default
Protocol : default
Cipher : default
Control port : 990
Data port : 989
EXAMPLE #15  To set the TCP low water mark for the ftp daemon, type:

```
$ server_ftp server_2 -modify -lowwatermark 32768
```

server_2 : done

FTPD CONFIGURATION
===================
State                      : stopped
Control Port               : 256
Data Port                  : 257
Default dir                : /big
Home dir                   : disable
Keepalive                  : 120
High watermark             : 90112
Low watermark              : 32768
Timeout                    : 900
Max timeout                : 7200
Read size                  : 8192
Write size                 : 49152
Umask                      : 77
Max connection             : 65535

SSL CONFIGURATION
================
Control channel mode       : disable
Data channel mode          : disable
Persona                    : default
Protocol                   : default
Cipher                     : default
Control port               : 990
Data port                  : 989

EXAMPLE #16  To restrict FTP server access to specific users, type:

```
$ server_ftp server_2 -modify -deniedusers
/.etc/mydeniedlist
```

server_2 : done

FTPD CONFIGURATION
===================
State                      : stopped
Control Port               : 256
Data Port                  : 257
Default dir                : /big
Home dir                   : disable
Keepalive                  : 120
High watermark             : 90112
Low watermark              : 32768
Denied users conf file     : /.etc/mydeniedlist
Timeout                    : 900
Max timeout                : 7200
Read size                  : 8192
Write size  : 49152
Umask      : 77
Max connection : 65535

SSL CONFIGURATION
=================
Control channel mode : disable
Data channel mode    : disable
Persona               : default
Protocol              : default
Cipher                : default
Control port          : 990
Data port             : 989

EXAMPLE #17  To set the path of the file displayed before the user logs in, type:

$ server_ftp server_2 -modify -welcome
   /.etc/mywelcomefile

server_2 : done

FTPD CONFIGURATION
==================
State                              : stopped
Control Port                       : 256
Data Port                          : 257
Default dir                        : /big
Home dir                           : disable
Keepalive                          : 120
High watermark                     : 90112
Low watermark                      : 32768
Welcome file                       : /.etc/mywelcomefile
Timeout                            : 900
Max timeout                        : 7200
Read size                          : 8192
Write size                         : 49152
Umask                              : 77
Max connection                     : 65535

SSL CONFIGURATION
=================
Control channel mode : disable
Data channel mode    : disable
Persona               : default
Protocol              : default
Cipher                : default
Control port          : 990
Data port             : 989
The server Commands

**server_http**

Confirms the HTTP configuration file for independent services, such as VNX FileMover, for the specified Data Movers.

**SYNOPSIS**

```
server_http {<movername>|ALL}
   -info [<feature>]
   -service <feature> {-start|-stop}
   -service [<feature>] -stats [-reset]
   -modify <feature>
      [-threads <threads>]
      [-users {valid|<user>,<user>,...}]
      [-hosts <ip>,<ip>,...]
      [-port <port_number>]
      [-timeout <max_idle_time>]
      [-maxrequests <maxrequests>]
      [-authentication {none|basic|digest}]
      [-realm <realm_name>]
      [-ssl {required|off}]
      [-sslpersona {anonymous|default|<persona_name>}]}
      [-sslprotocol {default|ss13|ts1|all}]
      [-sslcipher {default|<cipherlist}>]
   | -append <feature>
      [-users {valid|<user>,<user>,...}]
      [-hosts <ip>,<ip>,...]
   | -remove <feature>
      [-users {valid|<user>,<user>,...}]
      [-hosts <ip>,<ip>,...]
```

**DESCRIPTION**

`server_http` manages user and host access to HTTP servers for independent services such as FileMover.

**ALL** option executes the command for all Data Movers.

**OPTIONS**

- **-info [<feature>]**
  Displays information about the specified feature or all features including server status, port, threads, requests allowed, timeout, access control, and SSL configuration.

- **-service <feature> {-start|-stop}**
  Stops or starts the HTTP server for the specified feature.

- **-service [<feature>] -stats [-reset]**
  Lists the usage statistics of the HTTP server for the specified feature or all features. If `-reset` is specified, statistics are reset to zero.
-modify <feature>
Displays the current HTTP protocol connection for the specified feature. When issued with options, -modify sets the HTTP protocol connection for the specified option. Any options previously set will be overwritten.

[-threads <threads>]
Sets the number of threads (default=20) for incoming service requests. The minimum value is 4, the maximum 99. The HTTP threads are started on the Data Mover at boot time.

[-users [valid|<user>[,<user>,<user>...]]]
Allows the users who correctly authenticate as defined in the Data Mover passwd file (server_user provides more information) to execute commands for the specified <feature>.

If valid is entered, all users in the passwd file are allowed to use digest authentication. A comma-separated list of users can also be given. If no users are given, digest authentication is turned off.

[-hosts <ip>[,<ip>,<ip>...]]
Specifies hosts by their IP addresses that are allowed to execute commands for the specified <feature>.

[-port <port_number>]
Specifies the port on which the HTTP server listens for incoming service requests. By default, the HTTP server instance for FileMover listens on port 5080.

[-timeout <max_idle_time>]
Specifies the maximum time the HTTP server waits for a request before disconnecting from the client. The default value is 60 seconds.

[-maxrequests <max_requests>]
Specifies the maximum number of requests allowed. The default value is 300 requests.

[-authentication {none|basic|digest}]
Specifies the authentication method. none disables user authentication, allowing for anonymous access (that is, no authentication). basic authentication uses a clear text password. digest authentication uses a scripted password. The default value is digest authentication.
The server Commands

[-realm <realm_name>]
Specifies the realm name. This information is required when authentication is enabled (that is, the -authentication option is set to basic or digest). The default realm name for FileMover is DHSM_authorization.

[-ssl {required|off}]  
Specifies whether the HTTP server runs in secure mode, that is, only accepts data receive on encrypted SSL sessions. The default value is off.

---

Note: Before the HTTP server can be configured with SSL, the Data Mover must be set up with a private key and public certificate. This key and certificate are identified using a persona. In addition, the necessary Certificate Authority (CA) certificates to identify trusted servers must be imported into the Data Mover. Use the system’s PKI feature to manage the use of certificates prior to configuring SSL operation.

[-sslpersona {default|anonymous|<persona_name>}]  
Specifies the persona associated with the Data Mover. Personas are used to identify the private key and public certificate used by SSL. The default value is default (each Data Mover is currently configured with a single persona named default). anonymous specifies that SSL can operate without using a certificate.

[-sslprotocol {default|ssl3|tls1|all}]  
Specifies the SSL protocol version the HTTPS server accepts:

- ssl3 — Only SSLv3 connections
- tls1 — Only TLSv1 connections
- all — Both SSLv3 and TLSv1 connections
- default — Uses the value set in the ssl.protocol parameter which, by default, is 0 (SSLv3 and TLSv1)

[-sslcipher {default|<cipherlist>}]  
Specifies the SSL cipher suite. The value of default is the value set in the ssl.cipher parameter which, by default, is ALL:!ADH:!SSLv2:@STRENGTH. This value means that all ciphers are supported by VNX except the Anonymous Diffie-Hellman, NULL, and SSLv2 ciphers and that the supported ciphers are sorted by the size of the encryption key.

-append <feature> [-users {valid|<user> [, <user> , <user> ... ]} [-hosts <ip> [, <ip> , <ip> ... ]}]

Adds the specified users or hosts to the list of those who can execute commands for the specified <feature> without having to re-enter the
existing list. The **users** and **hosts** descriptions provide information. If users or hosts are not specified, displays the current HTTP configuration.

```
server Commands
```

```
remove <feature> [-users {valid|<user>[,<user>,<user>...]}][-hosts <ip>[,<ip>,<ip>...]]
```

Removes the specified users and hosts from the list of those who can execute commands for the specified `<feature>` without impacting others in the list. The **users** and **hosts** descriptions provide information. If users or hosts are not specified, displays the current HTTP configuration.

**SEE ALSO**


**EXAMPLE #1**

To display information about the HTTP protocol connection for the FileMover service, type:

```
$ server_http server_2 -info dhsm
```

```
server_2 : done

DHSM FACILITY CONFIGURATION
Service name : EMC File Mover service
Comment : Service facility for getting DHSM attributes
Active : False
Port : 5080
Threads : 16
Max requests : 300
Timeout : 60 seconds

ACCESS CONTROL
Allowed IPs : any
Authentication : digest ,Realm : DHSM_Authorization
Allowed user : nobody

SSL CONFIGURATION
Mode : OFF
Persona : default
Protocol : default
Cipher : default
```

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service name</td>
<td>Name of the FileMover service.</td>
</tr>
<tr>
<td>active</td>
<td>Whether VNX FileMover is enabled or disabled on the file system.</td>
</tr>
<tr>
<td>port</td>
<td>TCP port of the FileMover service.</td>
</tr>
<tr>
<td>threads</td>
<td>Number of threads reserved for the FileMover service.</td>
</tr>
<tr>
<td>max requests</td>
<td>Maximum number of HTTP requests the service allows to keep the connection alive.</td>
</tr>
<tr>
<td>timeout</td>
<td>The time in seconds until which the service is kept alive after a period of no activity.</td>
</tr>
<tr>
<td>allowed IPs</td>
<td>List of client IP addresses that are allowed to connect to the service.</td>
</tr>
</tbody>
</table>
The server Commands

<table>
<thead>
<tr>
<th>authentication</th>
<th>The HTTP authentication method used by the service.</th>
</tr>
</thead>
<tbody>
<tr>
<td>allowed user</td>
<td>Users allowed to connect to the service.</td>
</tr>
<tr>
<td>mode</td>
<td>The SSL mode.</td>
</tr>
<tr>
<td>persona</td>
<td>Name of the persona associated with the Certificate for establishing a secure connection.</td>
</tr>
<tr>
<td>protocol</td>
<td>The level of SSL protocol used for the service.</td>
</tr>
<tr>
<td>cipher</td>
<td>The cipher suite the service is negotiating, for establishing a secure connection with the client.</td>
</tr>
</tbody>
</table>

**EXAMPLE #2** To display statistical information about the HTTP protocol connection for the FileMover service, type:

$ server_http server_2 -service dhsm -stats

server_2 : done

Statistics report for HTTPD facility DHSM :

- Thread activity
  - Maximum in use count : 0
- Connection
  - IP filtering rejection count : 0
- Request
  - Authentication failure count : 0
- SSL
  - Handshake failure count : 0

**EXAMPLE #3** To configure an HTTP protocol connection for FileMover using SSL, type:

$ server_http server_2 -modify dhsm -ssl required

server_2 : done

**EXAMPLE #4** To modify the threads option of the HTTP protocol connection for FileMover, type:

$ server_http server_2 -modify dhsm -threads 40

server_2 : done

**DHSM FACILITY CONFIGURATION**

- Service name : EMC File Mover service
- Comment : Service facility for getting DHSM attributes
- Active : False
- Port : 5080
- Threads : 40
- Max requests : 300
- Timeout : 60 seconds

**ACCESS CONTROL**

- Allowed IPs : any
- Authentication : digest ,Realm : DHSM_Authorization
- Allowed user : nobody

**SSL CONFIGURATION**

- Mode : OFF
- Persona : default
- Protocol : default
The server Commands

Cipher : default

**EXAMPLE #5** To allow specific users to manage the HTTP protocol connection for FileMover, type:

```shell
$ server_http server_2 -modify dhsm -users valid -hosts 10.240.12.146
```

server_2 : done

**EXAMPLE #6** To add specific users who can manage the existing HTTP protocol connection for FileMover, type:

```shell
$ server_http server_2 -append dhsm -users user1,user2,user3
```

server_2 : done

**EXAMPLE #7** To add a specific user who can manage the existing HTTP protocol connection for FileMover, type:

```shell
$ server_http server_2 -append dhsm -users user4 -hosts 172.24.102.20,172.24.102.21
```

server_2 : done

**EXAMPLE #8** To remove the specified users and hosts so they can no longer manage the HTTP connection for FileMover, type:

```shell
$ server_http server_2 -remove dhsm -users user1,user2 -hosts 10.240.12.146
```

server_2 : done
**server_ifconfig**

Manages the network interface configuration for the specified Data Movers.

**SYNOPSIS**

```bash
server_ifconfig {<movername>|ALL} -all [-ip4|-ip6] -delete <if_name>
-create -Device <device_name> -name <if_name> -protocol { IP <ipv4_addr> <ipmask> <ipbroadcast> | IP6 <ipv6_addr> [/PrefixLength]} -<if_name> {up|down|ipsec|noipsec | mtu=<MTUbytes> [vlan=<vlanID>]} -vlan=<vlanID> [mtu=<MTUbytes>]
```

**DESCRIPTION**

**server_ifconfig** creates a network interface, assigns an IP address to a network interface, enables and disables an interface, sets the MTU size and the VLAN ID, and displays network interface parameters for the specified Data Mover.

**server_ifconfig** is used to define the network address of each interface existing on a machine, to delete and recreate an interface’s address and operating parameters.

The **ALL** option executes the command for all Data Movers.

**OPTIONS**

**-all** [-ip4|-ip6]
Displays parameters for all configured interfaces. The **-ip4** option displays all ipv4 only interfaces and the **-ip6** option displays all ipv6 only interfaces.

**-delete** <if_name>
Deletes a network interface configuration. However, the autogenerated link local interfaces cannot be deleted.

**-create** -Device <device_name> -name <if_name> -protocol IP <ipv4_addr> <ipmask> <ipbroadcast> | IP6 <ipv6_addr> [/PrefixLength] Creates a network interface configuration on the specified device with the specified name and assigns a protocol to the interface. The <if_name> must not contain a colon (:).

Available protocols are:
IP <ipv4_addr> <ipmask> <ipbroadcast>
| IP6 <ipv6_addr> [/PrefixLength]

IPv4 assigns the IP protocol with the specified IP address, mask, and broadcast address. The IP address is the address of a particular interface. Multiple interfaces are allowed for each device, each identified by a different IP address. The IP mask includes the network part of the local address and the subnet, which is taken from the host field of the address. For example, 255.255.255.0 would be a mask for a Class C network. The IP broadcast is a special destination address that specifies a broadcast message to a network. For example, x.x.x.255 is the broadcast address for a Class C network.

IP6 assigns the IPv6 address and prefix length. When prefix length is not specified, the default value of 64 is used.

When creating the first IPv6 interface with a global unicast address on a broadcast domain, the system automatically creates an associated IPv6 link-local interface. Similarly, when deleting the last remaining IPv6 interface on a broadcast domain, the system automatically deletes the associated IPv6 link-local interface.

For CIFS users, when an interface is created, deleted, or marked up or down, use the server_setup command to stop and then restart the CIFS service in order to update the CIFS interface list.

<if_name> up
Allows the interface to receive and transmit data, but does not enable the physical port. Interfaces are marked up automatically when initially setting up the IP address.

<if_name> down
Stops data from being transmitted through that interface. If possible, the interface is reset to disable reception as well. This does not automatically disable routes using the interface.

<if_name> ipsec|noipsec
ipsec sets the network interface to IPSec enabled, and ready to process IPSec packets. noipsec sets the network interface to IPSec disabled.

Note: VNX does not support the IPSec service.

<if_name> mtu=<MTUbytes>
Resets the maximum transmission unit (MTU) size in bytes for the specified interface. By default, the MTU is automatically set depending on the type of network interface card installed.
Regardless of whether you have Ethernet or Gigabit Ethernet, the initial default MTU size is 1500 bytes. To take advantage of the capacity of Gigabit Ethernet, the MTU size can be increased up to 9000 bytes if your switch supports jumbo frames. Jumbo frames should be used only when the entire infrastructure, including client NICs, supports them.

For UDP, it is important that both the client and server use the same MTU size. TCP negotiates the MTU size when the connection is initialized. The switch’s MTU must be greater than or equal to the host’s MTU.

**Note:** The MTU size specified here is for the interface. The MTU size specified in `server_netstat` applies to the device and is automatically set.

```
<if_name>  vlan=<vlanID>
```

Sets the ID for the virtual LAN (VLAN). Valid inputs are 0 (default) to 4094. When a VLAN ID other than 0 is set, the interface only accepts packets tagged with that specified ID. Outbound packets are also tagged with the specified ID.

**Note:** IEEE 802.1Q VLAN tagging is supported. VLAN tagging is not supported on ana interfaces.

**SEE ALSO**  `Configuring and Managing Networking on VNX` and `Configuring and Managing Network High Availability on VNX`, `server_netstat`, `server_setup`, and `server_sysconfig`.

**FRONT-END OUTPUT**  The network device name is dependent on the front end of the system (for example, NS series Data Mover, 514 Data Movers, 510 Data Movers, and so on) and the network device type. NS series and 514 Data Movers network device name display a prefix of `cge`, for example, `cge0`. 510 or earlier Data Movers display a prefix of `ana` or `ace`, for example, `ana0`, `ace0`. Internal network devices on a Data Mover are displayed as `el30`, `el31`.

**EXAMPLE #1**  To display parameters of all interfaces on a Data Mover, type:

```
$ server_ifconfig server_2 -all
```

```
server_2 :
  loop protocol=IP device=loop
    inet=127.0.0.1 netmask=255.0.0.0 broadcast=127.255.255.255
    UP, loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0 netname=localhost
cge0 protocol=IP device=cge0
    inet=172.24.102.238 netmask=255.255.255.0 broadcast=172.24.102.255
```
EXAMPLE #2   To create an IP interface for Gigabit Ethernet, type:

   $ server_ifconfig server_2 -create -Device cge1 -name cge1 -protocol IP 172.24.102.239 255.255.255.0 172.24.102.255

   server_2 : done

EXAMPLE #3   To create an interface for network device cge0 with an IPv6 address with a nondefault prefix length on server_2, type:

   $ server_ifconfig server_2 -create -Device cge0 -name cge0_int1 -protocol IP6 3ffe:0000:3c4d:0015:0435:0200:0300:ED20/48

   server_2 : done

EXAMPLE #4   To create an interface for network device cge0 with an IPv6 address on server_2, type:

   $ server_ifconfig server_2 -create -Device cge0 -name cge0_int1 -protocol IP6 3ffe:0000:3c4d:0015:0435:0200:0300:ED20

   server_2 : done

EXAMPLE #5   To verify that the settings for the cge0_int1 interface for server_2 are correct, type:

   $ server_ifconfig server_2 cge0_int1

       cge0_int1 protocol=IP6 device=cge0
       inet=3ffe:0:3c4d:15:435:200:300:ed20 prefix=48

       UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:c:5:5

       Note: The bold item in the output highlights the nondefault 48-bit prefix.

EXAMPLE #6   To verify that the interface settings for server_2 are correct, type:

   $ server_ifconfig server_2 -all

       el31 protocol=IP device=cge6
       inet=128.221.253.2 netmask=255.255.255.0 broadcast=128.221.253.255

       UP, ethernet, mtu=1500, vlan=0, macaddr=0:60:16:4:29:87

       el31 protocol=IP device=fxp0
       inet=128.221.252.2 netmask=255.255.255.0 broadcast=128.221.252.255

       UP, ethernet, mtu=1500, vlan=0, macaddr=8:0:1b:43:7e:b8 netname=localhost

       el30 protocol=IP device=cge6
       inet=128.221.253.2 netmask=255.255.255.0 broadcast=128.221.253.255

       UP, ethernet, mtu=1500, vlan=0, macaddr=0:60:16:4:11:a6 netname=localhost
The server Commands

inet=128.221.253.2 netmask=255.255.255.0 broadcast=128.221.253.255
UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:d:30:b2 netname=localhost
loop6 protocol=IP6 device=loop
  inet=::1 prefix=128
  UP, Loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0 netname=localhost
loop protocol=IP device=loop
  inet=127.0.0.1 netmask=255.0.0.0 broadcast=127.255.255.255
  UP, Loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0 netname=localhost
cge0_int1 protocol=IP6 device=cge0
  inet=3ffe:0:3c4d:15:435:200:300:ed20 prefix=64
  UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:c:2:5

cge0_0000_11 protocol=IP6 device=cge0
  inet=fe80::260:16ff:fe0c:205 prefix=64
  UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:c:2:5

Note: The first bold item in the output highlights the default 64-bit prefix. The second and third bold items highlight the link-local name and address that are automatically generated when you configure a global address for cge0. The automatically created link local interface name is made by concatenating the device name with the four digit VLAN ID between 0 and 4094. Note that the interface you configured with the IPv6 address 3ffe:0:3c4d:15:435:200:300:ed20 and the address with the link-local address fe80::260:16ff:fe0c:205 share the same MAC address. The link-local address is derived from the MAC address.

EXAMPLE #7  To verify that the interface settings for server_2 are correct, type:

$ server_ifconfig server_2 -all

server_2 :
cge0_int2 protocol=IP device=cge0
  inet=172.24.108.10 netmask=255.255.255.0 broadcast=172.24.108.255
  UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:c:2:5
cge0_int1 protocol=IP6 device=cge0
  inet=3ffe:0:3c4d:15:435:200:300:ed20 prefix=64
  UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:c:2:5
cge0_0000_11 protocol=IP6 device=cge0
  inet=fe80::260:16ff:fe0c:205 prefix=64
  UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:c:2:5
e130 protocol=IP device=mge0
  inet=128.221.252.2 netmask=255.255.255.0 broadcast=128.221.252.255
  UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:d:30:b1 netname=localhost
e131 protocol=IP device=mge1
  inet=128.221.253.2 netmask=255.255.255.0 broadcast=128.221.253.255
  UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:d:30:b2 netname=localhost
loop6 protocol=IP6 device=loop
  inet=::1 prefix=128
  UP, Loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0 netname=localhost
loop protocol=IP device=loop
  inet=127.0.0.1 netmask=255.0.0.0 broadcast=127.255.255.255
  UP, Loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0 netname=localhost
The server Commands

Note: The bold items in the output highlight the IPv4 interface, cge0_int2, and the IPv6 interface, cge0_int1.

**EXAMPLE #8**
To disable an interface, type:

```bash
$ server_ifconfig server_2 cge0_int2 down
```

server_2 : done

**EXAMPLE #9**
To enable an interface, type:

```bash
$ server_ifconfig server_2 cge0_int2 up
```

server_2 : done

**EXAMPLE #10**
To reset the MTU for Gigabit Ethernet, type:

```bash
$ server_ifconfig server_2 cge0_int2 mtu=9000
```

server_2 : done

**EXAMPLE #11**
To set the ID for the Virtual LAN, type:

```bash
$ server_ifconfig server_2 cge0_int1 vlan=40
```

server_2 : done

**EXAMPLE #12**
To verify that the VLAN ID in the interface settings for server_2 are correct, type:

```bash
$ server_ifconfig server_2 -all
```

```
server_2 :
cge0_int1 protocol=IP6 device=cge0
    inet=3ffe:0:3c4d:15:435:200:300:ed20 prefix=64
        UP, Ethernet, mtu=1500, **vlan=40**, macaddr=0:60:16:c:2:5

cge0_0040_11 protocol=IP6 device=cge0
    inet=fe80::260:16ff:fe0c:205 prefix=64
        UP, Ethernet, mtu=1500, vlan=40, macaddr=0:60:16:c:2:5

cge0_int2 protocol=IP device=cge0
    inet=172.24.108.10 netmask=255.255.255.0 broadcast=172.24.108.255
        UP, Ethernet, mtu=1500, vlan=20, macaddr=0:60:16:c:2:5

el30 protocol=IP device=mge0
    inet=128.221.252.2 netmask=255.255.255.0 broadcast=128.221.252.255
        UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:d:30:b1 netname=localhost

el31 protocol=IP device=mge1
    inet=128.221.253.2 netmask=255.255.255.0 broadcast=128.221.253.255
        UP, Ethernet, mtu=1500, vlan=0, macaddr=0:60:16:d:30:b2 netname=localhost

loop6 protocol=IP6 device=loop
    inet::1 prefix=128
        UP, Loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0 netname=localhost

loop protocol=IP device=loop
    inet=127.0.0.1 netmask=255.0.0.0 broadcast=127.255.255.255
        UP, Loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0 netname=localhost
```
Note: The bold items in the output highlight the VLAN tag.
Note that the link-local address uses the VLAN tag as part of its name.

EXAMPLE #13
To delete an IP interface, type:

$ server_ifconfig server_2 -delete cge1_int2

server_2 : done

Note: The autogenerated link local interfaces cannot be deleted.
The server Commands

server_ip

Manages the IPv6 neighbor cache and route table for the VNX.

SYNOPSIS

server_ip {ALL|<mover>}
  -neighbor {
    -list [<v6addr> [-interface <ifname>]]
    -create <v6addr> -lladdress <macaddr> [-interface <ifname>]
    -delete {-all|<v6addr> [-interface <ifname>]} }

  -route {
    -list
    -create {
      -destination <destination> -interface <ifname>
      -default -gateway <v6gw> [-interface <ifname>]
    }
    -delete {
      -destination <destination>
      -default -gateway <v6gw> [-interface <ifname>]
      -all
    }
  }

DESCRIPTION server_ip creates, deletes, and lists the neighbor cache and route tables.

OPTIONS server_ip {<movername> | ALL}
Sends a request to the Data Mover to get IPv6 parameters related to the IPv6 routing table and neighbor cache.

ALL option executes the command for all Data Movers.

-neighbor [-list | -create | -delete]
Lists, creates, or deletes the neighbor cache entries from the neighbor cache table.

-list
Displays the neighbor cache entries.

-create
Creates a neighbor cache table entry with the specified details.

-delete
Deletes the specified neighbor cache table entries or all entries.

-route [-list | -create | -delete]
Lists, creates, or deletes entries in the IPv6 route table.
The server Commands

- **list**
  Displays the IPv6 route table.

- **create**
  Creates a route table entry with the specified details.

- **delete**
  Deletes the specified route table entries.

**EXAMPLE #1**
To view a list of neighbor cache entries on Data Mover **server_2**, type:

```
$ server_ip server_2 -neighbor -list
```

server_2:

<table>
<thead>
<tr>
<th>Address</th>
<th>Link layer address</th>
<th>Interface</th>
<th>Type</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>fe80::204:23ff:fead:4fd4</td>
<td>0:4:23:ad:4f:d4</td>
<td>cge1_0000_ll</td>
<td>host</td>
<td>STALE</td>
</tr>
<tr>
<td>fe80::216:9cff:fe15:c00</td>
<td>0:16:9c:15:c:0</td>
<td>cge1_0000_ll</td>
<td>router</td>
<td>STALE</td>
</tr>
<tr>
<td>fe80::216:9cff:fe15:c00</td>
<td>0:16:9c:15:c:0</td>
<td>cge4_0000_ll</td>
<td>router</td>
<td>STALE</td>
</tr>
<tr>
<td>fe80::216:9cff:fe15:c00</td>
<td>0:16:9c:15:c:0</td>
<td>cge3_2998_ll</td>
<td>router</td>
<td>STALE</td>
</tr>
<tr>
<td>fe80::216:9cff:fe15:c00</td>
<td>0:16:9c:15:c:0</td>
<td>cge2_2442_ll</td>
<td>router</td>
<td>STALE</td>
</tr>
<tr>
<td>3ffe::1</td>
<td>0:16:9c:15:c:10</td>
<td>cge3_0000_ll</td>
<td>router</td>
<td>REACHABLE</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>The neighbor IPv6 address.</td>
</tr>
<tr>
<td>Link layer address</td>
<td>The link layer address of the neighbor.</td>
</tr>
<tr>
<td>Interface</td>
<td>Interface name of the interface connecting to the neighbor.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of neighbor. The neighbor can be either host or router.</td>
</tr>
<tr>
<td>State</td>
<td>The state of the neighbor such as REACHABLE, INCOMPLETE, STALE, DELAY, or PROBE.</td>
</tr>
</tbody>
</table>

**EXAMPLE #2**
To view a list of neighbor cache entries for a specific IP address on Data Mover **server_2**, type:

```
$ server_ip server_2 -neighbor -list fe80::216:9cff:fe15:c00
```

server_2:

<table>
<thead>
<tr>
<th>Address</th>
<th>Link layer address</th>
<th>Interface</th>
<th>Type</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>fe80::216:9cff:fe15:c00</td>
<td>0:16:9c:15:c:0</td>
<td>cge1_0000_ll</td>
<td>router</td>
<td>STALE</td>
</tr>
<tr>
<td>fe80::216:9cff:fe15:c00</td>
<td>0:16:9c:15:c:0</td>
<td>cge4_0000_ll</td>
<td>router</td>
<td>STALE</td>
</tr>
<tr>
<td>fe80::216:9cff:fe15:c00</td>
<td>0:16:9c:15:c:0</td>
<td>cge3_2998_ll</td>
<td>router</td>
<td>STALE</td>
</tr>
<tr>
<td>fe80::216:9cff:fe15:c00</td>
<td>0:16:9c:15:c:0</td>
<td>cge2_2442_ll</td>
<td>router</td>
<td>STALE</td>
</tr>
</tbody>
</table>
**EXAMPLE #3**  To view a list of neighbor cache entries for a specific IP address and interface type, on Data Mover `server_2`, type:

```
$ server_ip server_2 -neighbor -list
fe80::216:9cff:fe15:c00 -interface cge1_0000_ll
```

server_2:

<table>
<thead>
<tr>
<th>Address</th>
<th>Link layer address</th>
<th>Interface</th>
<th>Type</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>fe80::216:9cff:fe15:c00</td>
<td>0:16:9c:15:c:0</td>
<td>cge1_0000_ll</td>
<td>router</td>
<td>STALE</td>
</tr>
</tbody>
</table>

**EXAMPLE #4**  To add an entry to the neighbor cache for a global unicast IPv6 address, on Data Mover `server_2`, type:

```
$ server_ip server_2 -neighbor -create 2002:8c8:0:2310::2 -lladdress 0:16:9c:15:c:15
```

OK

**EXAMPLE #5**  To add an entry to the neighbor cache for a link local IPv6 address, on Data Mover `server_2`, type:

```
$ server_ip server_2 -neighbor -create fe80::2 -lladdress 0:16:9c:15:c:12 -interface cge1v6
```

OK

**EXAMPLE #6**  To delete an entry from the neighbor cache for a global unicast IPv6 address, on Data Mover `server_2`, type:

```
$ server_ip server_2 -neighbor -delete 2002:8c8:0:2310::2:ac18:f401
```

OK

**EXAMPLE #7**  To delete an entry from the neighbor cache for a link local IPv6 address, on all Data Movers, type:

```
$ server_ip ALL -neighbor -delete fe80::1 -interface cge1v6
```

OK

**EXAMPLE #8**  To delete entries from the neighbor cache on Data Mover `server_2` type:

```
$ server_ip server_2 -neighbor -delete -all
```

OK

**EXAMPLE #9**  To view a list of route table entries on Data Mover `server_2`, type:

```
$ server_ip server_2 -route -list
```
The server Commands

server_2:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Interface</th>
<th>Expires (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002:8c8:0:2310::/64</td>
<td>cge1v6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2002:8c8:0:2311::/64</td>
<td>cge1v6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2002:8c8:0:2312::/64</td>
<td>cge1v6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2002:8c8:0:2313::/64</td>
<td>cge1v6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>fe80::260:16ff:fe05:1bdd</td>
<td>cge1_0000_ll</td>
<td>1785</td>
</tr>
<tr>
<td>default</td>
<td>fe80::260:16ff:fe05:1bdc</td>
<td>cge1_0000_ll</td>
<td>1785</td>
</tr>
<tr>
<td>default</td>
<td>2002:8c8:0:2314::1</td>
<td>cge4v6</td>
<td>0</td>
</tr>
<tr>
<td>selected default</td>
<td>fe80::260:16ff:fe05:1bdd</td>
<td>cge1_0000_ll</td>
<td>1785</td>
</tr>
</tbody>
</table>

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination</td>
<td>The prefix of the destination or the default route entry. There can be multiple default routes, but only one is active and shown as &quot;selected default&quot;. The default sorting of the destination column displays the default routes at the bottom of the list and the selected default at the end of the list.</td>
</tr>
<tr>
<td>Gateway</td>
<td>The default gateway for default route entries. This value is blank for prefix destination entries.</td>
</tr>
<tr>
<td>Interface</td>
<td>Interface name of the interface used for the route.</td>
</tr>
<tr>
<td>Expires</td>
<td>The time until the route entry is valid. Zero denotes route is permanent and does not have an expiry.</td>
</tr>
</tbody>
</table>

**EXAMPLE #10**

To add a default route table entry on the Data Mover server_2 to the destination network with the specified prefix, type:

```
$ server_ip server_2 -route -create -destination 2002:8c8:0:2314::/64 -interface cge4v6
```

OK

**EXAMPLE #11**

To add a default route table entry on the Data Mover server_2 through the specified gateway, type:

```
$ server_ip server_2 -route -create -default -gateway 2002:8c8:0:2314::1
```

OK

**EXAMPLE #12**

To add a default route table entry on the Data Mover server_2 through the specified gateway using the link-local interface, type:

```
$ server_ip server_2 -route -create -default -gateway fe80::1 -interface cge1v6
```

OK
EXAMPLE #13  To delete an entry from the route table with an IPv6 prefix route destination for all Data Movers, type:

```
$ server_ip ALL -route -delete -destination 2002:8c8:0:2314::/64
OK
```

EXAMPLE #14  To delete an entry from the route table for a global unicast IPv6 address, on Data Mover server_2, type:

```
$ server_ip server_2 -route -delete -default -gateway 2002:8c8:0:2314::1
OK
```

EXAMPLE #15  To delete an entry from the route table for a link local IPv6 address, on Data Mover server_2, type:

```
$ server_ip server_2 -route -delete -default -gateway fe80::1 -interface cge1v6
OK
```

EXAMPLE #16  To delete all entries from the IPv6 route table on Data Mover server_2 type:

```
$ server_ip server_2 -route -delete -all
OK
```
**server_iscsi**

Manages the iSCSI configuration for the specified Data Movers.

**SYNOPSIS**

```
server_iscsi {<movername>|ALL} -target <target_options>
   -lun <lun_options>
   -mask <lun_mask_options>
   -ns <name_service_options>
   -service { -start | -stop | -status }
   -snap <snap_options>
   -help
```

**DESCRIPTION**

Manages iSCSI targets, logical unit numbers (LUNs), and snapshots; performs name service management (iSNS) and iSCSI service management.

To get usage for all options, type `-help`, or to get options for target, LUN, LUN mask, name service, or snapshot management, type `-target`, `-lun`, `-mask`, `-ns`, or `-snap`, respectively after the `<movername>`.

The `ALL` option executes the command for all Data Movers.

**TARGET OPTIONS**

```
-target {
   -alias <alias_name> [-Q <iscsi_name>]
   -create [<pg_tag>:np=<np_list> [<pg_tag]:np=<np_list>]]
   -delete <alias_name>
   -rename <old_alias_name> <new_alias_name>
   -bind <alias_name> <pg_tag>:np=<np_list>
   -unbind <alias_name> [<pg_tag]:np=<np_list>]
   -info <alias_name> [ -all ]
   -stat <alias_name> [ -all ]
   -list
```

To display target options, type `-target`.

```
-target -alias <alias_name> [-Q <iscsi_name>]
-target -create [<pg_tag>:np=<np_list> [<pg_tag>:np=<np_list>]]
```

Creates an iSCSI target with the specified alias on the Data Mover with an optional portal group tag and a list of network portals.

The `-alias <alias_name>` option assigns a name to the new iSCSI target for local identification. The `<alias_name>` is used for designating a specific iSCSI target in other commands. The
<alias_name> is not used for authentication. The -Q <iscsi_name> option specifies an iSCSI name string that must be entered as either an iSCSI Qualified Name (“iqn.”) or in IEEE EUI-64 format (“eui.”). If no name is entered, an iSCSI-compliant target name is generated automatically.

The iSCSI standard requires that an iSCSI name be globally unique and in a specific format. To avoid entering illegal names, do not use the -Q option; instead, let the VNX generates a legal iSCSI name.

For international support, both the target name (given by -alias) and the iSCSI name (given by -Q) support strings of Unicode characters in UTF-8 encoding. The <pg_tag>, which is the portal group tag, identifies the portal group within an iSCSI node. The <pg_tag> is an integer within the range of 0 to 65535.

The <np_list> is a comma-separated list of network portals. A network portal in a target is identified by its IP address and its listening TCP port. The format of a network portal is {<IPv4Addr>[::<port>] | <IPv6Addr> | <[IPv6Addr]:port> }. IPv6 addresses need to be enclosed in square brackets if a port is specified; the brackets do not signify optional content. If unspecified, port 3260 is used.

**-target -delete <alias_name>**
Removes a target specified by its <alias_name> from a Data Mover.

**CAUTION**
If one or more initiators are logged in to the target, or if all logical units have not been removed, the -delete option will fail. The -delete option should be performed with caution when no initiators are logged in to the target. The preconditions listed above are provided as a safeguard to prevent a system administrator from deleting the wrong target.

**-target -rename <old_alias_name> <new_alias_name>**
Changes the <alias_name> of the target.

**-target -bind <alias_name> <pg_tag>:np=<np_list>**
Adds target portal group tags or network portals to an existing target specified by the <alias_name>. If the portal group tag number does not exist, a new portal group is created and bound to the target.

**-target -unbind <alias_name>{<pg_tag>|np=<np_list>}**
Removes a portal group tag and network portals from an existing target.
The server Commands

Note: In case the portal group has only one network portal, then removing this network portal also removes the portal group and the associated sessions.

-target -info {<alias_name> | -all}
Displays attributes such as alias, iSCSI name, serial number, portal group tag, network portal, and details about logical units and connected initiators for all targets or the specified target.

-target -stat {<alias_name> | -all}
Displays statistical attributes such as initiator login/logout statistics and iSCSI session statistics for all targets or the specified target.

-target -list
Displays a listing of all targets.

LUN OPTIONS

-lun {
    -number <lun_number>
    -create <target_alias_name> -size <size>[M|G|T] -fs <fs_name>
    [-vp {yes|no}][-readonly {yes|no}]
    -modify <lun_number> -target <target_alias_name>
        -readonly {yes [-Force] | no}
    -extend <target_alias_name> -size <size>[M|G|T]
    -info {<lun_number> | -all} [-target <target_alias_name>]
    -stat {<lun_number> | -all} [-target <target_alias_name>]
    -list [-target <target_alias_name>]}

To display LUN options, type -lun. The VNX supports a maximum of 256 LUNs.

-lun -number <lun_number> -create
    <target_alias_name> -size <size> -fs <fs_name>
Creates a new logical unit number (LUN) and assigns it to an existing target specified by the <target_alias_name>. The <lun_number> can be an integer within the range of 0 to 255, but must be unique on the target or the command fails. The <size> specifies the size of the LUN in megabytes (M, the default), gigabytes (G), or terabytes (T). Maximum LUN size is 2 TB minus 1 MB. The <fs_name> specifies the name of a mounted file system that will serve as the storage pool for the LUN. When creating regular LUNs (the default LUN type), the mounted file system must be able to support the size specified.
The server Commands

[\texttt{-vp \{yes|no\}}]

The \texttt{-vp yes} option creates a virtually provisioned iSCSI LUN instead of a regular LUN. Regular LUNs reserve space on the file system for the entire LUN at the time of creation, but virtually provisioned LUNs do not reserve space. Virtually provisioned LUNs can be created larger than the underlying file system; however, when data is written to the virtually provisioned LUN, the file system must have sufficient free space to contain the data. The default is \texttt{no}.

\begin{caution}

\textbf{CAUTION}

When using virtually provisioned LUNs, monitoring the file system space usage is very important. Attempting to write data to a virtually provisioned LUN without sufficient free space in the underlying file system can result in client file system corruption on the iSCSI host system.

\end{caution}

[\texttt{-readonly \{yes|no\}}]

The \texttt{-readonly yes} option creates a protected, read-only LUN that is available as a destination in an iSCSI LUN replication relationship. By default, the iSCSI LUN is created as read-write. The default is \texttt{no}.

\texttt{-lun -modify <lun_number> -target <target_alias_name> -readonly \{yes [-Force]|no\}}

Modifies a LUN by changing it to read-only or read/write. The LUN cannot be a promoted snapshot. The \texttt{-readonly no} option converts the specified read-only LUN into a read/write LUN (for example, to change a replication destination LUN into a copy of the source LUN). The \texttt{-readonly yes} option converts the specified read/write LUN to read-only (for example, to prepare a replication source LUN for failback). If the LUN’s mask grants access to an iSCSI initiator, and that initiator is logged in to the target that contains the LUN, the \texttt{-readonly yes} option fails unless the \texttt{-Force} option is included.

\texttt{-lun -extend <target_alias_name> -size <size> [M|G|T]}  

Extends an existing LUN by the specified number of megabytes (M, the default), gigabytes (G), or terabytes (T).

\texttt{-lun -delete <lun_number> -target <target_alias_name>}

Deletes a LUN from a target as specified by the \texttt{<target_alias_name>}. 


The server Commands

- **CAUTION**
The LUN must not be in use when a delete is performed. Deleting a LUN deletes all of the data stored on the LUN.

-lun -info {<lun_number>|-all} [-target <target_alias_name>]
Displays the size, space usage, and file system information where the LUN resides, for the specified LUN on the specified target, or on all LUNs.

-lun -stat {<lun_number>|-all} [-target <target_alias_name>]
Displays statistical information such as size, LUN type, file system location, replication relationship (if any), maximum extension size, and SCSI CDB statistics for the specified LUN or all LUNs on the specified target.

-lun -list [-target <target_alias_name>]
Displays information on all LUNs, or to those belonging to a specified target.

- **MASK OPTIONS**

-mask {
- list [<target_alias_name>]
| -info <target_alias_name>
  {-initiator <initiator_name>|-all}
| -set <target_alias_name> -initiator <initiator_name>
  {-grant <access_list>|-deny <access_list>}
| -clear <target_alias_name> -initiator <initiator_name>}

To display mask options, type -mask.

-mask -list [<target_alias_name>]
Displays a list of existing LUN masks on the existing target on the specified Data Mover.

-mask -info <target_alias_name> {-initiator <initiator_name>|-all}
Returns LUN mask information for the specified target, or all targets including the alias name, iSCSI name, portal group information, default LUN mask information, and list of existing LUNs. The <initiator_name> is the iSCSI name of the initiator to which the mask command is applied.
The server Commands

`-mask -set <target_alias_name> -initiator <initiator_name>`
Sets the initiator access options (grant or deny or both) for the LUN mask on the specified `<target_alias_name>`. The `<initiator_name>` is the iSCSI name of the initiator to which the mask is applied. Outside of a clustered environment, do not grant multiple initiators access to the same LUN. Conflicts can occur if more than one initiator tries writing to the LUN. If your environment requires multiple initiators to access the same LUN, create a mask for each initiator through a separate command.

`-grant <access_list>`
Allows the initiator access to all LUNs specified in `<access_list>`. The `<access_list>` is a LUN number or a comma-separated list of LUN numbers. Access-list examples:

grant 2, 4, 6
grant 6-8

`-deny <access_list>`
Denies the initiator access to all LUNs specified in `<access_list>`. Access-list example: deny 1, 3-5, 9.

`-mask -clear <target_alias_name> -initiator <initiator_name>`
Clears the existing LUN mask for the initiator.

NAME SERVICE OPTIONS
Performs name service client management. Before modifying the `-ns` server or port, iSCSI service must be stopped, then restarted once the changes are made. If the administrator fails to stop the service, attempted modifications will have no effect even though the command reports "done." Available options are:

`-ns isns {`  
`-info`  
`-set {-server {<IPv4Addr>[:<port>]|<IPv6Addr>|<\[IPv6Addr\]:port>}}`  
`-esiport <port>`  
`-clear {-server|-esiport|-all}}`  

To display name service options, type `-ns`.

`-ns isns`  
Selects the naming service to be configured.

**Note:** iSNS is the only naming service supported.

`-ns isns -info`  
Displays status information for the iSNS service such as state, iSNS
server address, entity, source, Entity Status Inquiry (ESI) information, and timeout value.

**-ns isns -set -server** {<IPv4Addr>[:<port>]|<IPv6Addr>|<IPv6Addr>:<port>}

Set the IPv4 address or the IPv6 address and, optionally, the port of the iSNS server. IPv6 addresses need to be enclosed in square brackets if a port is specified; the brackets do not signify optional content. The default port is 3205.

**-ns isns -set -esiport** <port>

Sets the Entity Status Inquiry port of iSNS client. This port is a TCP port that by default, when iSCSI service is started up, is dynamically bound at the iSNS client. If this option is not entered at all, the application will assign a port when the service is started.

**-clear** { -server| -esiport| -all}

Clear current settings for iSNS server, for the ESI port, or for both the iSNS server and ESI port.

When a -clear option is issued, the system returns the message "done," however, settings are not cleared until iSCSI service is restarted.

**iSCSI SERVICE OPTIONS**

**-service** { -start| -stop| -status}

Starts, stops, or displays the status of the iSCSI service.

**SNAP OPTIONS**

Performs snap management functions. Available options are:

**-snap** {

- **-list** [ -target <target_alias_name> -lun <lun_number>]

- **-info** <snap_name>

- **-create** -target <target_alias_name> -lun <lun_number> [-data <app_data>]

- **-modify** <snap_name> -data <app_data>

- **-delete** {<snap_name>| -target <target_alias_name> -lun <lun_number>} [-Force]

- **-restore** <snap_name>

- **-promote** <snap_name> -initiator <initiator_name>

- **-Demote** <snap_name> [-Force] }

To display snap management options, type -snap.

**-snap -list** [ -target <target_alias_name> -lun <lun_number>]

Lists all snapshots on the Data Mover or all snapshots of the specified iSCSI LUN (regardless of which application created the snapshots).
The server Commands

-snap -info <snap_name>
Displays information about the specified snapshot, including any application data (if the data is UTF-8 compliant).

-snap -create -target <target_alias_name> -lun <lun_number>[ -data <app_data>]
Creates a snapshot of the specified iSCSI LUN. The snapshot is crash consistent, after a power loss, and not necessarily consistent from the point of view of the host applications. If the -data option is included, the provided comment text (limited to 4096 bytes) is associated with the snapshot as application data and saved on persistent storage provided by the Data Mover. If it contains a space character, the string must be enclosed in double quotation marks. To clear existing data, provide a null string ("").

Note: Although the snapshot supports 4096 bytes, the command line interface might limit the number of characters you can type.

-snap -modify <snap_name> -data <app_data>
Replaces the application data (text or binary data) associated with the snapshot with the provided comment text (limited to 4096 bytes) or a null string. If it contains a space character, the string must be enclosed in double quotation marks. To clear existing data, provide a null string ("").

Note: Although the snapshot supports 4096 bytes, the command line interface might limit the number of characters you can type.

-snap -delete {<snap_name>| -target <target_alias_name> -lun <lun_number>} [-Force]
Deletes either the specified snapshot or all snapshots of the specified iSCSI LUN. The snapshot to delete cannot be promoted. To delete a snapshot created by an application other than the Control Station CLI, include -Force.

-snap -restore <snap_name>
Restores the content of a production iSCSI LUN from a snapshot of that LUN on the same target. The snapshot used for the LUN restoration is preserved, but all newer snapshots of the LUN are deleted. The snapshot must be the same size as the production LUN (that is, the restore fails if the production LUN was extended after the snapshot was taken). The snapshot (or a more recent snapshot of the same LUN) cannot be promoted.
The server Commands

**-snap -promote <snap_name> -initiator <initiator_name>**

Promotes the specified snapshot so it can be used as an iSCSI LUN. The new LUN has the same data content as the snapshot. A LUN mask is created to allow access to the LUN by the specified iSCSI initiator.

**-snap -Demote <snap_name> [-Force]**

Demotes the specified snapshot LUN. To demote a snapshot created by another application (not the Control Station CLI) or a snapshot that is in use, include -Force. The demoted LUN and its contents (including data written to the LUN while promoted) are irretrievable. The read-only snapshot is preserved.

**SEE ALSO** Configuring iSCSI Targets on VNX for File, Installing iSCSI Host Components on VNX for File, Using VNX Replicator, and nas_replicate.

**EXAMPLE #1**

To create a target with an alias of t1, and a portal group of 100, and a network portal of np=172.24.102.242, type:

```
$ server_iscsi server_2 -target -alias t1 -create 100:np=172.24.102.242
```

**EXAMPLE #2**

To create a target with an alias of t1, and a portal group of 100, and a network portal of np=172.24.102.242 [2002:c8c::24:172:65], type:

```
$ server_iscsi server_2 -target -alias t1 -create 100:np=172.24.102.242,[2002:c8c::24:172:65]
```

**EXAMPLE #3**

To create a target with an alias of t1, and a portal group of 100, with a port of 3261, and a network portal of np=172.24.102.242:3261,[2002:c8c::24:172:65]:3261, type:

```
```

**EXAMPLE #4**

To rename a target alias, type:

```
$ server_iscsi server_2 -target -rename t1 t2
```

**EXAMPLE #5**

To bind target portal groups as specified by the alias t1 to an existing target, 200:np=172.24.102.243, type:

```
$ server_iscsi server_2 -target -bind t1 200:np=172.24.102.243
```
The server Commands

**EXAMPLE #6**  To bind target portal groups as specified by the alias `t1` to an existing target with a port of 63, `200:np=2002:c8c::24:172:63`, type:

```
$ server_iscsi server_2 -target -bind t1
200:np=2002:c8c::24:172:63
```

`server_2 : done`

**EXAMPLE #7**  To configure server_2 to use the iSNS server at IP address `200:c8c::24:172:63` with the default port, type:

```
$ server_iscsi server_2 -ns isns -set -server 2002:c8c::24:172:63
```

`server_2 : done`

**EXAMPLE #8**  To list all targets, type:

```
$ server_iscsi server_2 -target -list
```

```
server_2 :
alias_name  portal_groups  iscsi_name
--- ------ ----------------- -----------------
t1           100,200        iqn.1992-05.com.emc:apm000438070430000-1
```

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>alias_name</td>
<td>Alias for the iSCSI target's qualified name.</td>
</tr>
<tr>
<td>portal_groups</td>
<td>Portal groups used by the target.</td>
</tr>
<tr>
<td>iscsi_name</td>
<td>iSCSI qualified name of the target.</td>
</tr>
</tbody>
</table>

**EXAMPLE #9**  To create a new standard iSCSI logical unit number (LUN) of 0 and assign it to an existing target, `t1`, with a specified size of `102400` MB on file system, `ufs1`, type:

```
$ server_iscsi server_2 -lun -number 0 -create t1 -size 102400 -fs ufs1
```

`server_2 : done`

**EXAMPLE #10**  To create a new virtually provisioned iSCSI logical unit number (LUN) of 1 and assign it to an existing target, `t1`, with a specified size of `10` GB on file system, `ufs1`, type:

```
$ server_iscsi server_2 -lun -number 1 -create t1 -size 10G -fs ufs1 -vp yes
```

`server_2 : done`

**EXAMPLE #11**  To create a new, protected, read-only iSCSI LUN of 2 and assign it to an existing target `t1`, with a specified size of `10` GB on file system, `ufs1`, type:

```
$ server_iscsi server_2 -lun -number 2 -create t1 -size 10G -fs ufs1 -readonly yes
```
EXAMPLE #12  To extend LUN 0 on target, t1, by 4096 MB, type:

```
$ server_iscsi server_2 -lun -extend 0 -target t1 -size 4096M
```

EXAMPLE #13  To display a list of all LUNs on a Data Mover, type:

```
$ server_iscsi server_2 -lun -list
```

<table>
<thead>
<tr>
<th>lun</th>
<th>size(MB)</th>
<th>filesystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>106496</td>
<td>ufs1 ( id=22 )</td>
</tr>
<tr>
<td>1</td>
<td>10240</td>
<td>ufs1 ( id=22 )</td>
</tr>
<tr>
<td>2</td>
<td>10240</td>
<td>ufs1 ( id=22 )</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>Alias for the iSCSI target's qualified name.</td>
</tr>
<tr>
<td>lun</td>
<td>ID for the LUN.</td>
</tr>
<tr>
<td>size (MB)</td>
<td>Size of the LUN in MBs.</td>
</tr>
<tr>
<td>filesystem</td>
<td>Name of the file system containing the LUN.</td>
</tr>
</tbody>
</table>

EXAMPLE #14  To display information for LUN 0, on target t1, type:

```
$ server_iscsi server_2 -lun -info 0 -target t1
```

Logical Unit 0 on target t1:
(Production) fsid=22 size=106496MB alloc=0MB dense

path=/ufs1/fs22_T1_LUN0_APM00043807043_0000/fs22_T1_LUN0_APM00043807043_0000
(extended)

replication=none
max_extension_size=84494MB

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>fsid</td>
<td>ID of the file system containing the LUN.</td>
</tr>
<tr>
<td>size</td>
<td>Reserved size of the LUN in MBs.</td>
</tr>
<tr>
<td>alloc</td>
<td>Actual space allocated for the LUN.</td>
</tr>
<tr>
<td>dense</td>
<td>Storage method used by the LUN.</td>
</tr>
</tbody>
</table>
# The server Commands

## EXAMPLE #15
To display statistical information for all LUNs on a target **t1**, type:

```
$ server_iscsi server_2 -lun -stat -all -target t1
```

```
server_2 :  
 Logical Unit 0 on target t1:  
 (Production) fsid=22 size=106496MB alloc=218MB dense  
 path=/ufs1/fs22_T1_LUN0_APM00043807043_0000/fs22_T1_LUN0_APM00043807043_0000  (extended)  
 replication=none  
 max_extension_size=84441MB  
 Statistics:  
 Total SCSI CDBs: 6837  
 Read: 2381 Failed: 0 Blks read: 147769  
 Write: 4314 Failed: 0 Blks written: 449833
```

```
Logical Unit 1 on target t1:  
 (Production) fsid=22 size=10240MB alloc=53MB virtually provisioned  
 path=/ufs1/fs22_T1_LUN1_APM00043807043_0000/fs22_T1_LUN1_APM00043807043_0000  
 replication=none  
 max_extension_size=2086911MB  
 Statistics:  
 Total SCSI CDBs: 2272  
 Read: 1179 Failed: 0 Blks read: 111831  
 Write: 973 Failed: 0 Blks written: 110617
```

```
Logical Unit 2 on target t1:  
 (Production) fsid=22 size=10240MB alloc=0MB dense  
 path=/ufs1/fs22_T1_LUN2_APM00043807043_0000/fs22_T1_LUN2_APM00043807043_0000  
 replication=available destination  
 max_extension_size=84441MB  
 Statistics:  
 Total SCSI CDBs: 0  
 Read: 0 Failed: 0 Blks read: 0  
 Write: 0 Failed: 0 Blks written: 0
```

## EXAMPLE #16
To set a LUN mask on target **t1**, for initiator **iqn.1991-05.com.microsoft:winserver1.nasdocs.emc.com**, with the access rights to LUNs 0 and 1, type:
The server Commands

$ server_iscsi server_2 -mask -set t1 -initiator iqn.1991-05.com.microsoft:winserver1.nasdocs.emc.com -grant 0,1

server_2 : done

EXAMPLE #17  To display a list of LUN masks, type:

$ server_iscsi server_2 -mask -list

server_2 :

<p>| target: t1 |</p>
<table>
<thead>
<tr>
<th>Grant LUNs</th>
</tr>
</thead>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>Alias name of the target.</td>
</tr>
<tr>
<td>initiator_name</td>
<td>Qualified name of the iSCSI initiator on the host.</td>
</tr>
<tr>
<td>grant LUNs</td>
<td>Number of the LUNs to which the initiator is granted access.</td>
</tr>
</tbody>
</table>

EXAMPLE #18  To clear the LUN mask for the initiator, iqn.1991-05.com.microsoft:winserver1.nasdocs.emc.com, type:

$ server_iscsi server_2 -mask -clear t1 -initiator iqn.1991-05.com.microsoft:winserver1.nasdocs.emc.com

server_2 : done

EXAMPLE #19  To configure an iSNS client on the Data Mover for iSNS server with IP address of 172.24.102.66, type:

$ server_iscsi server_2 -ns isns -set -server 172.24.102.66

server_2 : done

Note: Before modifying the -ns server or port, iSCSI service must be stopped, then restarted once the changes are made. If the administrator fails to stop the service, attempted modifications will have no effect until the service is started even though the command reports "done."

EXAMPLE #20  To start the iSCSI service on the Data Mover, type:

$ server_iscsi server_2 -service -start

server_2 : done

EXAMPLE #21  To display the iSNS client service status, type:

$ server_iscsi server_2 -ns isns -info

server_2 :

iSNS Client Service Status
--------------------------
The server Commands

**service** : started
**server** : 172.24.102.66:3205/tcp
**state** : SYNC
**entity** : isns:00000002
**source** : iqn.1992-05.com.emc:apm000438070430000-1
**ESI support** : TRUE
**ESI port** : 52427
**ESI Time** : 1131484090 (Tue Nov 8 21:08:10 2005)
**timeout** : 3000
**trace** : FALSE
**PG support** : TRUE

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>service</strong></td>
<td>Whether the iSNS service is started or stopped.</td>
</tr>
<tr>
<td><strong>server</strong></td>
<td>IP address of the iSNS server and port.</td>
</tr>
<tr>
<td><strong>state</strong></td>
<td>State of the iSNS server. Possible states are:</td>
</tr>
<tr>
<td></td>
<td>- SYNC — The iSNS client is OK.</td>
</tr>
<tr>
<td></td>
<td>- VIRGIN — The iSNS client is not in use—there are no targets or portals configured.</td>
</tr>
<tr>
<td></td>
<td>- INDOUBT — The iSNS client cannot currently contact the iSNS server and is retrying.</td>
</tr>
<tr>
<td></td>
<td>- NOTSYNC — The iSNS client has experienced multiple failures attempting to contact the iSNS server and is retrying.</td>
</tr>
<tr>
<td></td>
<td>- DEAD — The iSNS client cannot contact the iSNS server. Still retrying at a lower frequency.</td>
</tr>
<tr>
<td><strong>entity</strong></td>
<td>Name of the network entity that serves as a container for iSCSI targets and portal objects on the Data Mover.</td>
</tr>
<tr>
<td><strong>source</strong></td>
<td>Name of the iSCSI target used as the source attribute in iSNS requests that require source identification.</td>
</tr>
<tr>
<td><strong>ESI support</strong></td>
<td>Whether Entity Status Inquiry (ESI) is supported as the means for refreshing the entity registration.</td>
</tr>
<tr>
<td><strong>ESI port</strong></td>
<td>TCP port that the iSNS client service listens to for incoming ESI messages.</td>
</tr>
<tr>
<td><strong>ESI Time</strong></td>
<td>Last time the ESI was received from the iSNS server or UNKNOWN if never.</td>
</tr>
<tr>
<td><strong>timeout</strong></td>
<td>Network timeout value (in milliseconds) for communication with the iSNS server.</td>
</tr>
<tr>
<td><strong>trace</strong></td>
<td>Whether extensive tracing output of iSNS messages, both incoming and outgoing, is enabled (TRUE) or disabled (FALSE).</td>
</tr>
<tr>
<td><strong>PG support</strong></td>
<td>Whether portal groups are supported; if not, all portals are implicitly placed in one portal group.</td>
</tr>
</tbody>
</table>

**EXAMPLE #22**  To display status information for the target t1, type:

```
$ server_iscsi server_2 -target -info t1
```

**server_2 :**

Target Alias Name: t1
Target iSCSI Name: iqn.1992-05.com.emc:apm000438070430000-1
Serial Number: 1
Portal Group Tag: 100
  Portal: 172.24.102.242:3260
Portal Group Tag: 200
  Portal: 172.24.102.243:3260
Logical Units:

0 : (Production) fsid=22 size=106496MB alloc=0MB dense
  path=/ufs1/fs22_T1_LUN0_APM00043807043_0000/fs22_T1_LUN0_APM00043807043_0000
1 : (Production) fsid=22 size=10240MB alloc=0MB virtually_provisioned
  path=/ufs1/fs22_T1_LUN1_APM00043807043_0000/fs22_T1_LUN1_APM00043807043_0000
2 : (Production) fsid=22 size=10240MB alloc=0MB dense
  path=/ufs1/fs22_T1_LUN2_APM00043807043_0000/fs22_T1_LUN2_APM00043807043_0000

Connected Initiators:

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Alias Name</td>
<td>Alias for the iSCSI target's qualified name.</td>
</tr>
<tr>
<td>Target iSCSI Name</td>
<td>iSCSI qualified name of the target.</td>
</tr>
<tr>
<td>Serial Number</td>
<td>Unique, VNX-wide serial number for the target.</td>
</tr>
<tr>
<td>Portal Group Tag</td>
<td>Portal groups used by the target.</td>
</tr>
<tr>
<td>Portal</td>
<td>IP address and port of the portals in the portal group.</td>
</tr>
<tr>
<td>Logical Units</td>
<td>Logical units (LUNs) assigned to the target.</td>
</tr>
<tr>
<td>fsid</td>
<td>ID the file system containing the LUN.</td>
</tr>
<tr>
<td>size</td>
<td>Reserved size of the LUN in MBs.</td>
</tr>
<tr>
<td>alloc</td>
<td>Amount of data currently stored in the LUN.</td>
</tr>
<tr>
<td>dense</td>
<td>Storage method used by the LUN.</td>
</tr>
<tr>
<td>path</td>
<td>File system path to the LUN.</td>
</tr>
<tr>
<td>Connected Initiators</td>
<td>Qualified name of the initiators logged in to the target.</td>
</tr>
</tbody>
</table>

**EXAMPLE #23**  To display statistics for the target t1, and the current sessions on the target, type:

```bash
$ server_iscsi server_2 -target -stat t1
```

**server_2:**
Target Alias Name: t1
Target iSCSI Name: iqn.1992-05.com.emc:apm000438070430000-1

[Login/Loginout Stats]
Login Accepted: 1
Login Failed: 0
  Redirect: 0
  Authorize: 0
  Authenticate: 0
  Negotiate: 0
  Other: 0
Logout Normal: 0
Logout Other: 0
Last Login Failed: N/A

[Session Stats]
Command PDUs: 5886
  Read Command PDUs: 3604
Write Command PDUs: 2256
Response PDUs: 5886
Data-In PDUs: 1936
Data-Out PDUs: 0
R2Ts: 0
Tx Data Octets: 132134422
Rx Data Octets: 132573696
Digest Errors: 0
Connection Timeout Errors: 0

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Alias Name</td>
<td>Alias for the iSCSI target's qualified name.</td>
</tr>
<tr>
<td>Target iSCSI Name</td>
<td>iSCSI qualified name of the target.</td>
</tr>
<tr>
<td>Login Accepts</td>
<td>Number of accepted login requests.</td>
</tr>
<tr>
<td>Login Failed</td>
<td>Number of rejected login requests.</td>
</tr>
<tr>
<td>Redirect</td>
<td>Number of logins that are redirected.</td>
</tr>
<tr>
<td>Authorize</td>
<td>Number of logins that failed due to authorization failure.</td>
</tr>
<tr>
<td>Authenticate</td>
<td>Number of login failures due to failed authentication.</td>
</tr>
<tr>
<td>Negotiate</td>
<td>Number of login failures due to errors in iSCSI parameter negotiation.</td>
</tr>
<tr>
<td>Other</td>
<td>Number of login failures due to other causes.</td>
</tr>
<tr>
<td>Logout Normal</td>
<td>Number of logouts requesting session closing.</td>
</tr>
<tr>
<td>Logout Other</td>
<td>Number of other logouts.</td>
</tr>
<tr>
<td>Initiator</td>
<td>ID of initiator logged into target.</td>
</tr>
<tr>
<td>TSIH</td>
<td>iSCSI session handle.</td>
</tr>
<tr>
<td>ISID</td>
<td>iSCSI session ID.</td>
</tr>
<tr>
<td>Command PDUs</td>
<td>Number of SCSI command protocol data units (PDUs) received by target.</td>
</tr>
<tr>
<td>Read Command PDUs</td>
<td>Number of read command PDUs received by target.</td>
</tr>
<tr>
<td>Write Command PDUs</td>
<td>Number of write command PDUs received by target.</td>
</tr>
<tr>
<td>Response PDUs</td>
<td>Number of iSCSI response PDUs sent by target.</td>
</tr>
<tr>
<td>Data-In PDUs</td>
<td>Number of data-in PDUs.</td>
</tr>
<tr>
<td>Data-Out PDUs</td>
<td>Number of data-out PDUs.</td>
</tr>
<tr>
<td>R2Ts</td>
<td>Number of request to transfers issued during the session.</td>
</tr>
<tr>
<td>Tx Data Octets</td>
<td>Amount of transmitted data (in bytes).</td>
</tr>
<tr>
<td>Rx Data Octets</td>
<td>Amount of received data (in bytes).</td>
</tr>
<tr>
<td>Digest Errors</td>
<td>Number or errors in data or header digests or both.</td>
</tr>
</tbody>
</table>
| Connection Timeout Errors | Number of errors caused by the connection between target and initiator timing out.

**EXAMPLE #24**

To unbind the network portal **172.24.102.243:3260** on a target **t1**, type:

```
$ server_iscsi server_2 -target -unbind t1

np=172.24.102.243:3260
```

server_2 : done
EXAMPLE #25  To display the status of the iSCSI service, iSCSI session statistics and statistics about login/logout attempts, type:

$$\texttt{server_iscsi server_2 -service -status}$$

```
server_2:
**** ISCSI Service ****
Status: Running
Iscsi is configured to listen on ports:
TCP Port: 3260
Header Digests are preferred
Data Digests are not preferred
[Session Stats]
Session Failures: 0
Last Session Failed: N/A
[Login/Logout Stats]
Login Accepted: 0
Login Failed: 0
    Redirect: 0
    Authorize: 0
    Authenticate: 0
    Negotiate: 0
    Other: 0
Logout Normal: 0
Logout Other: 0
Last Login Failed: N/A
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the iSCSI service on the Data Mover.</td>
</tr>
<tr>
<td>TCP Port</td>
<td>Listening port used by the iSCSI service.</td>
</tr>
<tr>
<td>Header Digests</td>
<td>Whether header digests are enabled or disabled.</td>
</tr>
<tr>
<td>Data Digests</td>
<td>Whether data digests are enabled or disabled.</td>
</tr>
<tr>
<td>Session Failures</td>
<td>Number of session failures.</td>
</tr>
<tr>
<td>Last Session Failed</td>
<td>ID of the last failed session.</td>
</tr>
<tr>
<td>Login Accepts</td>
<td>Number of accepted login requests.</td>
</tr>
<tr>
<td>Login Failed</td>
<td>Number of rejected login requests.</td>
</tr>
<tr>
<td>Redirect</td>
<td>Number of logins that are redirected.</td>
</tr>
<tr>
<td>Authorize</td>
<td>Number of logins that failed due to authorization failure.</td>
</tr>
<tr>
<td>Authenticate</td>
<td>Number of login failures due to failed authentication.</td>
</tr>
<tr>
<td>Negotiate</td>
<td>Number of login failures due to errors in iSCSI parameter negotiation.</td>
</tr>
<tr>
<td>Other</td>
<td>Number of login failures due to other causes.</td>
</tr>
<tr>
<td>Logout Normal</td>
<td>Number of logouts requesting session closing.</td>
</tr>
<tr>
<td>Logout Other</td>
<td>Number of other logouts.</td>
</tr>
<tr>
<td>Last Failure Type</td>
<td>Type of last login failure.</td>
</tr>
</tbody>
</table>
The server Commands

**EXAMPLE #26**
To stop iSCSI service, type:

```
$ server_iscsi server_2 -service -stop
```

server_2 : done

**EXAMPLE #27**
To delete an iSCSI LUN 1 on the target t1, type:

```
$ server_iscsi server_2 -lun -delete 1 -target t1
```

server_2 : done

**EXAMPLE #28**
To clear iSNS client service configuration, type:

```
$ server_iscsi server_2 -ns isns -clear -server
```

server_2 : done

When a `-clear` is issued, the system returns the message “done”, however, settings are not cleared until iSNS service is restarted.

**EXAMPLE #29**
To delete the target t1, type:

```
$ server_iscsi server_2 -target -delete t1
```

server_2 : done

**EXAMPLE #30**
To change LUN 24 (a read-only replication destination LUN) on target t1 to a read/write LUN, type:

```
$ server_iscsi server_2 -lun -modify 24 -target t1 -readonly no
```

server_2 : done

**EXAMPLE #31**
To change LUN 11 (a read/write LUN) on target t1 to a read-only LUN, type:

```
$ server_iscsi server_2 -lun -modify 11 -target t1 -readonly yes
```

server_2 : done

**EXAMPLE #32**
To list the snapshots of LUN 22 on target t2 on server_2, type:

```
$ server_iscsi server_2 -snap -list -target t2 -lun 22
```

<table>
<thead>
<tr>
<th>Snap Name</th>
<th>Lun</th>
<th>Target</th>
<th>Create Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>fs14_T11_LUN22_FCNBD054507021_003C.ckpt006 24</td>
<td>24</td>
<td>t2</td>
<td>Thu Mar 06 14:33:09 EST 2008</td>
</tr>
<tr>
<td>fs14_T11_LUN22_FCNBD054507021_003C.ckpt005 24</td>
<td>24</td>
<td>t2</td>
<td>Thu Mar 06 14:29:22 EST 2008</td>
</tr>
</tbody>
</table>
The server Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snap Name</td>
<td>Name of the snapshot.</td>
</tr>
<tr>
<td>Lun Number</td>
<td>Number of the LUN used to create the snapshot.</td>
</tr>
<tr>
<td>Target</td>
<td>Alias of the target to which the LUN belongs.</td>
</tr>
<tr>
<td>Create Time</td>
<td>Date and time that the snapshot was created.</td>
</tr>
</tbody>
</table>

EXAMPLE #33  To view information about the specified snapshot, type:

```
$ server_iscsi server_2 -snap-info
```

server_2:
Snap Name = fs14_T11_LUN18_000172900116_0004.ckpt008
Lun Number = 18
Target = t2
Promoted Number =
Promoted Initiator =
Application Label = CelerraCS
LUN Size MB = 10240
Create Time = Tue Mar 04 10:14:06 EST 2008
Application Data = windows_app_data

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snap Name</td>
<td>Name of the snapshot.</td>
</tr>
<tr>
<td>Lun Number</td>
<td>Number of the LUN used to create the snapshot.</td>
</tr>
<tr>
<td>Target</td>
<td>Alias of the target to which the LUN belongs.</td>
</tr>
<tr>
<td>Create Time</td>
<td>Date and time that the snapshot was created.</td>
</tr>
<tr>
<td>Application Data</td>
<td>Comment text or other data associated with the snapshot.</td>
</tr>
</tbody>
</table>

EXAMPLE #34  To view information about the specified promoted snapshot, type:

```
$ server_iscsi server_2 -snap -info
```

server_2:
Snap Name = fs14_T8_LUN24_000183500235_0006.ckpt461
Lun Number = 24
Target = t2
Create Time = Wed Mar 05 11:30:52 EST 2008
Application Data = windows_app_data
Promoted Number = 132
Application Label = CelerraCS
LUN size MB = 256
The server Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Promoted Number</td>
<td>LUN number of the promoted snapshot.</td>
</tr>
<tr>
<td>Promoted Initiator</td>
<td>iSCSI initiator granted access to the promoted LUN.</td>
</tr>
<tr>
<td>Application Label</td>
<td>Label identifying the creator of the snapshot.</td>
</tr>
<tr>
<td>LUN size MB</td>
<td>Size of the promoted LUN in megabytes.</td>
</tr>
</tbody>
</table>

**EXAMPLE #35**
To create a snapshot of LUN 24 on target t1 on server_2 and attach the specified comment text, type:

```bash
$ server_iscsi server_2 -snap -create -target t1 -lun 24 -data "Q3 analysis snap Target t1 LUN 24"
```

server_2 :
Info 26306936878: OK.id=fs23_t1_LUN24_HK190807300021_0000.ckpt000

**EXAMPLE #36**
To modify the comment text associated with the specified snapshot on server_2, type:

```bash
$ server_iscsi server_2 -snap -modify fs14_T8_LUN24_000183500235_0006.ckpt461 -data "Q4 analysis snap Target t1 LUN 24"
```

server_2 :
OK

**EXAMPLE #37**
To delete the specified snapshot on server_2, type:

```bash
$ server_iscsi server_2 -snap -delete fs14_T8_LUN24_000183500235_0006.ckpt461
```

server_2 :
OK

**EXAMPLE #38**
To delete all snapshots of LUN 24 on target t1 on server_2, type:

```bash
$ server_iscsi server_2 -snap -delete -target t1 -lun 24
```

server_2 :
OK

**EXAMPLE #39**
To restore a production LUN using the specified snapshot on server_2, type:

```bash
$ server_iscsi server_2 -snap -restore fs14_T8_LUN24_000183500235_0006.ckpt461
```

server_2 :
OK
**EXAMPLE #40**  
To promote the specified snapshot and grant access to the specified iSCSI initiator, type:

```
$ server_iscsi server_2 -snap -promote  
fs14_T8_LUN24_000183500235_0006.ckpt461 -initiator  
```

server_2 :
OK

**EXAMPLE #41**  
To demote the specified snapshot regardless of the application that created it, type:

```
$ server_iscsi server_2 -snap -Demote
fs14_T8_LUN24_000183500235_0006.ckpt461 -Force
```

server_2 :
OK
server_kerberos

Manages the Kerberos configuration within the specified Data Movers.

SYNOPSIS

server_kerberos {<movername>|ALL}

- add realm=<realm_name>, kdc=<fqdn_kdc_name>[:<port>]
  [,kdc=<fqdn_kdc_name>[:<port>]...]
  [,kpasswd=<fqdn_kpasswd_server_name>]
  [,kadmin=<kadmin_server>]
  [,domain=<domain_name>][,defaultrealm]
- delete realm=<realm_name>
- keytab
- ccache [-flush]
- list
- kadmin [<kadmin_options>]

DESCRIPTION

server_kerberos adds, deletes, lists the realms within the Kerberos configuration of a Data Mover, and manages the Data Movers service principals and keys.

server_kerberos displays the key table content, and specifies a kadmin server.

OPTIONS

- add realm=<realm_name>, kdc=<fqdn_kdc_name>
  Adds the specified realm to the Kerberos configuration on the specified Data Mover. The <realm_name> is the fully qualified domain name of the Kerberos realm to be added to the key distribution center (KDC) configuration. The <fqdn_kdc_name> is the fully qualified domain name of the KDC for the specified realm.

  Note: The -add option is only relevant if you are using a UNIX/Linux Kerberos KDC.

  [:<port>]
  Specifies a port that the KDC listens on.

  [,kdc=<fqdn_kdc_name>[:<port>]...]
  Specifies additional KDCs with ports that KDCs listen on.

  [,kpasswd=<fqdn_kpasswd_server_name>]
  Specifies a password server for the KDC. The <fqdn_kpasswd_server_name> must be a fully qualified domain name for the server.
[.kadmin=<kadmin_server>]
Specifies the kadmin server.

[.domain=<domain_name>]
The <domain_name> is the full name of the DNS domain for the realm.

[.defaultrealm]
Indicates that the default realm is to be used.

 deleting realm=<realm_name>
Deletes the specified realm from the Kerberos configuration for the specified Data Mover.

Note: The -delete option is only relevant if you are using a UNIX/Linux Kerberos KDC.

-keytab
Displays the principal names for the keys stored in the keytab file.

-ccache
Displays the entries in the Data Movers Kerberos credential cache.

Note: The -ccache option can also be used to provide EMC Customer Support with information for troubleshooting user access problems.

[-flush]
Flushes the Kerberos credential cache removing all entries. Credential cache entries are automatically flushed when they expire or during a Data Mover reboot.

Once the cache is flushed, Kerberos obtains new credentials when needed. The repopulation of credentials may take place immediately, over several hours, or be put off indefinitely if no Kerberos activity occurs.

-list
Displays a listing of all configured realms on a specified Data Mover or on all Data Movers.

-kadmin [ <kadmin_options>]
Invokes the kadmin tool with the following specified options:

[ -r <realm>]
Specifies a realm as the default database realm.
The server Commands

[-p <principal>] Specifies the principal for authentication. Otherwise, kadmin will append "/admin" to the primary principal name of the default cache, the value of the USER environment variable, or the username as obtained with getpwuid, in order of preference.

[-q <query>] Runs kadmin in non-interactive mode. This passes the query directly to kadmin, which performs the query, then exits.

[-w <password>] Uses a specified password instead of prompting for a password.

[-s <admin_server> [:<port>]] Specifies the kadmin server with its associated port.

Note: The kadmin tool is only relevant if you are using a UNIX/Linux Kerberos KDC. You must be root to execute the -kadmin option.

SEE ALSO Configuring NFS on VNX, server_checkup, and server_nfs.

OUTPUT Dates appearing in output are in UTC format.

EXAMPLE #1 To add a realm to the Kerberos configuration of a Data Mover, type:

$ server_kerberos server_2 -add realm=nasdocs.emc.com,kdc=winserver1.nasdocs.emc.com,dom

ain=nasdocs.emc.com

server_2 : done

EXAMPLE #2 To list the keytabs, type:

$ server_kerberos server_2 -keytab

server_2 :
Dumping keytab file

keytab file major version = 0, minor version 0

-- Entry number 1 --
principal: DM102-CGE0$@NASDOCS.EMC.COM
realm: NASDOCS.EMC.COM
encryption type: rc4-hmac-md5
principal type 1, key version: 332
  key length: 16, key: b1c199a6ac11cd529df172e270326d5e
  key flags: (0x0), Dynamic Key, Not Cached
  key cache hits: 0

-- Entry number 2 --
principal: DM102-CGE0$@NASDOCS.EMC.COM
realm: NASDOCS.EMC.COM
encryption type: des-cbc-md5
principal type 1, key version: 332
  key length: 8, key: ced9a23183619267
  key flags:(0x0), Dynamic Key, Not Cached
  key cache hits: 0

-- Entry number 3 --
principal: DM102-CGE0@$NASDOCS.EMC.COM
realm: NASDOCS.EMC.COM
encryption type: des-cbc-crc
principal type 1, key version: 332
  key length: 8, key: ced9a23183619267
  key flags:(0x0), Dynamic Key, Not Cached
  key cache hits: 0

-- Entry number 4 --
principal: host/dm102-cge0$@NASDOCS.EMC.COM
realm: NASDOCS.EMC.COM
encryption type: rc4-hmac-md5
principal type 1, key version: 332
  key length: 16, key: b1c199a6ac11cd529df172e270326d5e
  key flags:(0x0), Dynamic Key, Not Cached
  key cache hits: 0

<... removed ...>

-- Entry number 30 --
principal: cifs/dm102-cge0.nasdocs.emc.com@NASDOCS.EMC.COM
realm: NASDOCS.EMC.COM
encryption type: des-cbc-crc
principal type 1, key version: 333
  key length: 8, key: d95e1940b910ec61
  key flags:(0x0), Dynamic Key, Not Cached
  key cache hits: 0

End of keytab entries. 30 entries found.

Note: This is a partial listing due to the length of the output.

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>principal type</td>
<td>Type of the principal as defined in the GSS-API. Reference to RFC 2743.</td>
</tr>
<tr>
<td>key version</td>
<td>Every time a key is regenerated its version changes.</td>
</tr>
</tbody>
</table>
EXAMPLE #3 To list all the realms on a Data Mover, type:

```
$ server_kerberos server_2 -list
```

```
server_2 :
Kerberos common attributes section:
  Supported TGS encryption types:  rc4-hmac-md5 des-cbc-md5 des-cbc-crc
  Supported TKT encryption types:  rc4-hmac-md5 des-cbc-md5 des-cbc-crc
  Use DNS locator:                yes

End of Kerberos common attributes.

Kerberos realm configuration:

```
realm name:           NASDOCS.EMC.COM
  kdc:              winserver1.nasdocs.emc.com
  admin server:     winserver1.nasdocs.emc.com
  kpasswd server:   winserver1.nasdocs.emc.com
  default domain:   nasdocs.emc.com
```

End of Kerberos realm configuration.

Kerberos domain_realm section:
```
  DNS domain = Kerberos realm
  .nasdocs.emc.com = NASDOCS.EMC.COM
```

End of Krb5.conf domain_realm section.

EXAMPLE #4 To specify a kadmin server, type:

```
$ server_kerberos server_2 -add
  realm=eng.nasdocs.emc.com,kdc=winserver1.nasdocs.emc.com
  ,kadmin=172.24.102.67
```

```
server_2 : done
```

**Note:** You must be root to execute the `-kadmin` option. Replace `$` with `#` as the root login is a requirement.

EXAMPLE #5 To delete a realm on a Data Mover, type:

```
$ server_kerberos server_2 -delete
  realm=eng.nasdocs.emc.com
```

```
server_2 : done
```

EXAMPLE #6 To display the credential cache on a Data Mover, type:

```
$ server_kerberos server_2 -ccache
```

```
server_2 :
  Dumping credential cache

Names:
  Client:  DM102-CGE0@$NASDOCS.EMC.COM
```
Service: WINSERVER1.NASDOCS.EMC.COM
Target: HOST/WINSERVER1.NASDOCS.EMC.COM@NASDOCS.EMC.COM

Times:
  Auth: 09/12/2005 07:15:04 GMT
  Start: 09/12/2005 07:15:04 GMT
  End: 09/12/2005 17:15:04 GMT

Flags: PRE_AUTH, OK_AS_DELEGATE

Encryption Types:
  Key: rc4-hmac-md5
  Ticket: rc4-hmac-md5

Names:
  Client: DM102-CGE0$@NASDOCS.EMC.COM
  Service: winserver1.nasdocs.emc.com
  Target: ldap/winserver1.nasdocs.emc.com@NASDOCS.EMC.COM

Times:
  Auth: 09/12/2005 07:15:04 GMT
  Start: 09/12/2005 07:15:04 GMT
  End: 09/12/2005 17:15:04 GMT

Flags: PRE_AUTH, OK_AS_DELEGATE

Encryption Types:
  Key: rc4-hmac-md5
  Ticket: rc4-hmac-md5

Names:
  Client: DM102-CGE0$@NASDOCS.EMC.COM
  Service: NASDOCS.EMC.COM
  Target: krbtgt/NASDOCS.EMC.COM@NASDOCS.EMC.COM

Times:
  Auth: 09/12/2005 07:15:04 GMT
  Start: 09/12/2005 07:15:04 GMT
  End: 09/12/2005 17:15:04 GMT

Flags: INITIAL, PRE_AUTH

Encryption Types:
  Key: rc4-hmac-md5
  Ticket: rc4-hmac-md5

End of credential cache entries.

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>client</td>
<td>Client name and its realm.</td>
</tr>
<tr>
<td>service</td>
<td>Domain controller and its realm.</td>
</tr>
<tr>
<td>target</td>
<td>Target name and its realm.</td>
</tr>
<tr>
<td>auth</td>
<td>Time of the initial authentication for the named principal.</td>
</tr>
<tr>
<td>start</td>
<td>Time after which the ticket is valid.</td>
</tr>
<tr>
<td>end</td>
<td>Time after which the ticket will not be honored (its expiration time).</td>
</tr>
</tbody>
</table>
The server Commands

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>flags</td>
<td>Options used or requested when the ticket was issued.</td>
</tr>
<tr>
<td>key</td>
<td>Key encryption type.</td>
</tr>
<tr>
<td>ticket</td>
<td>Ticket encryption type.</td>
</tr>
</tbody>
</table>
EXAMPLE #7  To flush the credential cache on a Data Mover, type:

$ server_kerberos server_2 -ccache flush

server_2 :
Purging credential cache.
Credential cache flushed.
server_ldap

Manages the LDAP-based directory client configuration and LDAP over SSL for the specified Data Movers.

SYNOPSIS

server_ldap {<movername>|ALL}
  -set [-p] -domain <FQDN> | -basedn <attribute_name>=<attribute_value>[,...]
  [-servers {<IPv4_addr>[:<port>] | <IPv6_addr>|<\[IPv6_addr\]:port>}[,...]}
  [-profile <profile>]
  [-nisdomain <NIS_domain>]
  [-binddn <bind_DN> | {-<kerberos -kaccount <account_name> [-realm <realm_name>]}]
  [-ssl_enabled {y|n}]
  [-sslpersona {none |<persona_name>}}
  [-ssl_cipher {default |<cipher_list}>]
  -clear
  -info [-verbose]
  -service { -start | -stop | -status}
  -lookup { -user <username>
    | -group <groupname>
    | -uid <uid>
    | -gid <gid>
    | -hostname <hostname>
    | -netgroup <groupname>}

DESCRIPTION

server_ldap configures, starts, stops, deletes, and displays the status of the LDAP-based directory client configuration, and queries the LDAP-based directory server.

OPTIONS

- set [-p] - domain <FQDN> | - basedn <attribute_name>=<attribute_value>[,...]
  Specifies the LDAP-based directory client domain for the specified Data Mover and starts the service. The -p option requests a prompt for the password. A password is required in conjunction with a bind distinguished name in order to specify the use of simple authentication. The -basedn option specifies the Distinguished Name (DN) of the directory base, an x509 formatted name that uniquely identifies the directory base. For example: ou=abc,o=def,c=ghi. If a base distinguished name contains space characters, enclose the entire string within double quotation marks and enclose the name with a backslash and double quotation mark. For example, ""\"cn=abc,cn=def ghi,dc=com\""."
It is recommended configuring an LDAP-based directory client by using the `-basedn` option instead of the `-domain` option. The DN provides the root position for:

- Searching for iPlanet profiles
- Defining default search containers for users, groups, hosts, and netgroups according to RFC 2307. An iPlanet profile and OpenLDAP or Active Directory with SFU or IdMU ldap.conf file are only required for customized setups.

**Note:** In the case in which the DN of the directory base contains dots and the client is configured using the domain name, the default containers may not be set up correctly. For example, if the name is `dc=my.company,dc=com` and it is specified as domain name `my.company.com`, VNX incorrectly defines the default containers as `dc=my,dc=company,dc=com`.

```
[-servers {<IPv4_addr>[:<port>] | <IPv6_addr> | 
\[IPv6_addr\]:port}>[, ...]]
```

Specifies the IP addresses of the LDAP-based directory client servers. `<IPv4_addr>` or `<IPv6_addr>` indicates the IP address of the LDAP-based directory servers. IPv6 addresses need to be enclosed in square brackets if a port is specified; the brackets do not signify optional content. The `<port>` option specifies the LDAP-based directory server TCP port number. If the port is not specified, the default port is 389 for LDAP and 636 for SSL-based LDAP. It is recommended that at least two LDAP servers are defined, so that DART can switch to the second server in case the first cannot be reached.

**Note:** IP addresses of the LDAP-based directory servers do not have to be included every time with the `server_ldap` command once you have indicated the configuration server, and if configuring the same LDAP-based directory service.

```
[-profile <profile>]
```

Specifies the profile name or the profile distinguished name which provides the iPlanet client with configuration information about the directory service. For example, both the following values are allowed: `-profile vnx_profile` and `-profile cn=vnx_profile,ou=admin,dc=mycompany,dc=com.`
The server Commands

Note: It is recommended that unique profile names be used in the Directory Information Tree (DIT). The specified profile is searched for by scanning the entire tree and if it is present in multiple locations, the first available profile is used unless the profile distinguished name is specified.

[-nisdomain <NIS_domain>]
Specifies the NIS domain of which the Data Mover is a member since an LDAP-based directory domain can host more than one NIS domain.

[-binddn <bind_DN> | {-kerberos -kaccount <account_name> [-realm <realm_name>]}]
Specifies the distinguished name (DN) or Kerberos account of the identity used to bind to the service. Active Directory with SFU or IdMU requires an authentication method that uses simple authentication, SSL, or Kerberos.

Simple authentication requires that a DN be specified along with a password. For SSL-based client authentication to succeed, the Data Mover certificate Subject must match the distinguished name for an existing user (account) at the directory server.

Note: To configure a LDAP-based directory service for authentication, -binddn is not required if the -sslpersona option is specified. In this case, SSL-based client authentication will be used.

The Kerberos account name must be the CIFS server computer name known by the KDC. The account name must terminate with a $ symbol.

By default, the Data Mover assumes that the realm is the same as the LDAP domain provided in the -domain or -basedn options. But a different realm name can be specified if necessary.

[-sslenabled [y | n]]
Enables (y) or disables (n) SSL. SSL is disabled by default.

[-sslpersona {none | <persona_name>}] Specifies the key and certificate of the directory server. If a persona has been previously configured, none disables the user of a client key and certificate. The -sslpersona option without the -binddn option indicates that the user wants to authenticate using the client (persona) certificate. To authenticate using the client certificate, the LDAP server must be configured to always
request (or require) the persona certificate during the SSL transaction, or the authentication will fail. If authentication using the client certificate is not desired, then the -binddn option must be used. The configuration rules are explained in Table 1 on page 575.

Note: The -sslpersona option does not automatically enable SSL, but configures the specified value. The value remains persistent and is used whenever SSL is enabled.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Configuration rules</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td><strong>Data Mover configuration</strong></td>
</tr>
<tr>
<td>SSL enabled on Data Mover, LDAP server should accept SSL, anonymous authentication is used.</td>
<td>server_ldap -sslenabled y</td>
</tr>
<tr>
<td>SSL enabled, password-based authentication is used.</td>
<td>server_ldap -p -binddn cn=foo -sslenabled y</td>
</tr>
<tr>
<td>SSL enabled, SSL certificate authentication is used, LDAP server should be configured to request client certificate.</td>
<td>server_ldap -sslenabled y -sslpersona default (use server_certificate to verify that the certificate for the Data Mover’s default persona exists)</td>
</tr>
</tbody>
</table>

Note: The user should refer to the LDAP server documentation for information about configuring the server to request the client certificate.

[-sslcipher [default | <cipher_list>]]
Specifies default or the cipher list.

Note: The -sslcipher option does not automatically enable SSL, but configures the specified value. The value remains persistent and is used whenever SSL is enabled.

-clear
Deletes the LDAP-based directory client configuration for the specified Data Mover and stops the service.

-info
Displays the service status and the static and dynamic configuration.

[-verbose]
Adds troubleshooting information to the output.

-service (-start | -stop | -status)
The -start option enables the LDAP-based directory client service.
The server Commands

The LDAP-based directory client service is also restarted when the VNX is rebooted. The `-stop` option disables the LDAP-based directory client service, and the `-status` option displays the status of the LDAP-based directory service.

-lookup
  `{user=<username> | group=<groupname> | uid=<uid> | gid=<gid> |` 
  `hostbyname=<hostname> | netgroup=<groupname>`

Provides lookup information about the specified resource for troubleshooting purposes.

SEE ALSO Configuring VNX Naming Services.

EXAMPLE #1 To configure the use of an LDAP-based directory by a Data Mover, type:

```
$ server_ldap server_4 -set -domain nasdocs.emc.com
-servers 172.24.102.62
```

server_4 : done

EXAMPLE #2 To configure the use of an LDAP-based directory by a Data Mover using the Distinguished Name of the server at IPv4 address 172.24.102.62 with the default port, type:

```
$ server_ldap server_2 -set -basedn
dc=nasdocs,dc=emc,dc=com -servers 172.24.102.62
```

server_2 : done

EXAMPLE #3 To configure the use of an LDAP-based directory by a Data Mover using the Distinguished Name of the server at IPv6 address 2002:c8c::24:172:63 with the default port, type:

```
$ server_ldap server_2 -set -basedn
dc=nasdocs,dc=emc,dc=com -servers 2002:c8c::24:172:63
```

server_2 : done

EXAMPLE #4 To configure the use of an LDAP-based directory by a Data Mover and specify the use of the client profile, type:

```
$ server_ldap server_4 -set -domain nasdocs.emc.com
-servers 172.24.102.62 -profile celerra_profile
-nisdomain nasdocs -sslenabled y
```

server_4 : done

EXAMPLE #5 To configure the use of an LDAP-based directory by a Data Mover and specify the use of the client profile using its distinguished name, type:
EXEMPLE #6
To specify the NIS domain to which the Data Mover is a member, type:

```
$ server_ldap server_2 -set -domain nasdocs.emc.com -servers 172.24.102.62 -nisdomain nasdocs
```

server_2 : done

EXEMPLE #7
To configure the use of simple authentication by specifying a bind Distinguished Name (DN) and password, type:

```
$ server_ldap server_2 -set -p -domain nasdocs.emc.com -servers 172.24.102.10 -binddn "cn=admin,cn=users,dc=nasdocs,dc=emc"
```

server_2 : Enter Password:********
done

EXEMPLE #8
To configure the use of an LDAP-based directory by a Data Mover using SSL, type:

```
$ server_ldap server_4 -set -basedn dc=nasdocs,dc=emc,dc=com -servers 172.24.102.62 -sslenabled y
```

server_4 : done

EXEMPLE #9
To configure the use of an LDAP-based directory by a Data Mover using SSL and user key and certificate, type:

```
$ server_ldap server_4 -set -basedn dc=nasdocs,dc=emc,dc=com -servers 172.24.102.62 -sslenabled y -sslpersona default
```

server_4 : done

EXEMPLE #10
To configure the use of an LDAP-based directory by a Data Mover using SSL and using specified ciphers, type:

```
$ server_ldap server_4 -set -basedn dc=nasdocs,dc=emc,dc=com -servers 172.24.102.62 -sslenabled y -sslcipher "RC4-MD5,RC4-SHA"
```

server_4 : done

EXEMPLE #11
To display information about the LDAP-based directory configuration on a Data Mover, type:

```
$ server_ldap server_4 -info
```

server_4 :
LDAP domain: nasdocs.emc.com
EXAMPLE #12  To configure the use of Kerberos authentication by specifying a Kerberos account, type:

\$ server_ldap server_2 -set -basedn
dc=nasdocs,dc=emc,dc=com -servers 172.24.102.62 -kerberos
-kaccount cifs_compname$

server_2 : done

EXAMPLE #13  To display detailed information about the LDAP-based directory configuration on a Data Mover, type:

\$ server_ldap server_2 -info -verbose

server_ldap server_2 -i -v
server_2 :
LDAP domain: devldapdom1.lcsc
  base DN: dc=devldapdom1,dc=lcsc
  State: Configured - Connected
NIS domain: devldapdom1.lcsc
  Configuration file - TTL: 1200 seconds
  Next configuration update in 1107 seconds
LDAP configuration servers:
  Server 10.64.220.148 port 389 : Active, connected
    SSL not enabled, Persona: none specified, Cipher Suite List: none specified
  Server 10.64.220.114 port 389 : Spare, disconnected
    SSL not enabled, Persona: none specified, Cipher Suite List: none specified
Domain naming contexts:
  dc=devldapdom1,dc=lcsc
Domain supported LDAP controls:
  1.3.6.1.4.1.4203.1.9.1.1
  2.16.840.1.113730.3.4.18
  2.16.840.1.113730.3.4.2
  1.3.6.1.4.1.4203.1.10.1
  1.2.840.113556.1.4.319
  1.2.826.0.1.334810.2.3
  1.2.826.0.1.3344810.2.3
  1.3.6.1.1.13.2
  1.3.6.1.1.13.1
  1.3.6.1.1.12
Domain supported authentication mechanisms:
Default search base: dc=devldapdom1,dc=lcsc
Domain default search Scope: single-level
passwd base DN:
  ou=people,dc=devldapdom1,dc=lcsc - search scope single-level
  passwd object class: posixAccount
passwd attributes: cn, uid, uidNumber, gidNumber, userPassword,
loginShell, gecos, description

  group base DN:
    ou=group,dc=devldapdom1,dc=lcsc - search scope single-level
  group object class: posixGroup
  group attributes: cn, gidNumber, userPassword, memberUid, description

hosts base DN:
  ou=hosts,dc=devldapdom1,dc=lcsc - search scope single-level
  host object class: ipHost
  host attributes: cn, ipAddressNumber, description

netgroup base DN:
  ou=netgroup,dc=devldapdom1,dc=lcsc - search scope single-level
  netgroup object class: nisNetgroup
  host attributes: cn, nisNetgroupTriple, memberNisNetgroup description

**EXAMPLE #14**  
To display lookup information about the user **nasadmin**, type:

```
$ server_ldap server_4 -lookup -user nasadmin
```

server_4 :
user: nasadmin, uid: 1, gid: 201, gecos: nasadmin, home dir: /home/nasadmin,
shell: /bin/csh

**EXAMPLE #15**  
To display the status of the LDAP-based directory service, type:

```
$ server_ldap server_4 -service -status
```

server_4 :
LDAP service active

**EXAMPLE #16**  
To stop the LDAP-based directory service, type:

```
$ server_ldap server_4 -service -stop
```

server_4 : done

**EXAMPLE #17**  
To delete the LDAP configuration for the specified Data Mover and
stop the service, type:

```
$ server_ldap server_4 -clear
```

server_4 : done
server_log

Displays the log generated by the specified Data Mover.

SYNOPSIS

server_log <movername>
[-a][-f][-n][-s][-v][-t]

DESCRIPTION

server_log reads and displays the log generated by the Data Mover. Information in the log file is read from oldest to newest. To view that most recent log activity, add `tail` to the end of your command line.

OPTIONS

No arguments
Displays the contents of the log added since the last reboot.

-a
Displays the complete log.

-f
Monitors the growth of the log by entering into an endless loop, pausing, reading the log being generated. The output is updated every second. To exit, press `Ctrl-C` together.

-n
Displays the log without the time stamp.

-s
Displays the time in `yyyy-mm-dd` format when each command in the log was executed.

-v | -t
Displays the log files in verbose form or terse form.

EXAMPLE #1

To monitor the growth of the current log, type:

$ server_log server_2 -f

NAS LOG for slot 2:
-------------------
0 keys=0 h=0 nc=0

1200229390: SVFS: 6: Merge Start FsVol:118 event:0x0
1200229390: CFS: 6: Resuming fs 24
1200229390: SVFS: 6: 118:D113118_736:Merge hdr=82944 prev=99328 id=113 chunk=0 s
tableEntry=7
1200229390: UFS: 6: Volume name:Sh122113
1200229390: UFS: 6: starting gid map file processing.
The server Commands

1200229390: UFS: 6: gid map file processing is completed.
1200229390: DPSVC: 6: DpRequest::done() BEGIN reqType:DpRequest_VersionInt_SchSr
cReq Caller:DpRequest_Caller_Scheduler status:0
1200229390: DPSVC: 6: SchedulerSrc=118_APM00062400708_0000_253_APM00062400708_00
00, curState=active, input=refreshDone
1200229390: DPSVC: 6: DpVersion::getTotalBlocksVolume enter
1200229390: DPSVC: 6: DpVersion::getTotalBlocksVolume found newV 118.ckpt003, bl
ocks 17534
1200229390: DPSVC: 6: DpVersion::getTotalBlocksVolume 0 blocks for vnumber 1038
1200229390: DPSVC: 6: DpVersion::getTotalBlocksVolume found oldV 118.ckpt004
1200229390: DPSVC: 6: DpVersion::getTotalBlocksVolume exit
1200229390: DPSVC: 6: SchedulerSrc=118_APM00062400708_0000_253_APM00062400708_00
00, newState=active
1200229390: SVFS: 6: D113118_736: After Merge err:4 full:0 mD:0
e changePrevChunk
1200229510: DPSVC: 6: refreshSnap: cur=1200229510, dl=1200229520, kbytes=0, setu
p=0, rate=1000
1200229510: DPSVC: 6: SchedulerSrc=199_APM00062400708_0000_258_APM00062400708_00
00, curState=active, input=refresh
1200229510: DPSVC: 6: DpRequest::execute() BEGIN reqType:DpRequest_VersionInt_SchS
rcReq Caller:DpRequest_Caller_Scheduler reqMode:0
1200229510: DPSVC: 6: DpRequest::execute() END reqType:DpRequest_VersionInt_SchS
rcReq Caller:DpRequest_Caller_Scheduler status:0 reqMode:0
1200229510: DPSVC: 6: SchedulerSrc=199_APM00062400708_0000_258_APM00062400708_00
00, newState=active

---More---

Note: This is a partial listing due to the length of the output.

EXAMPLE #2

To display the current log, type:

$ server_log server_2

NAS LOG for slot 2:
-------------------
0 keys=0 h=0 nc=0

2008-01-13 08:03:10: SVFS: 6: Merge Start FsVol:118 event:0x0
_503
2008-01-13 08:03:10: CFS: 6: Resuming fs 24
2008-01-13 08:03:10: SVFS: 6: D113118_736: Merge hdr:82944 prev=99328 id=113
chunk=0 stableEntry=7
2008-01-13 08:03:10: UFS: 6: Volume name:Sh122113
2008-01-13 08:03:10: UFS: 6: gid map file processing is completed.
2008-01-13 08:03:10: DPSVC: 6: DpRequest::done() BEGIN reqType:DpRequest_Version
Int_SchSrcRefresh reqCaller:DpRequest_Caller_Scheduler status:0
400708_0000, curState=active, input=refreshDone
2008-01-13 08:03:10: DPSVC: 6: DpVersion::getTotalBlocksVolume enter
2008-01-13 08:03:10: DPSVC: 6: DpVersion::getTotalBlocksVolume found newV 118.ckpt003, blocks 17534
2008-01-13 08:03:10: DPSVC: 6: DpVersion::getTotalBlocksVolume 0 blocks for vnumber 1038 totalB 0
2008-01-13 08:03:10: DPSVC: 6: DpVersion::getTotalBlocksVolume found oldV 118.ckpt004
2008-01-13 08:03:10: DPSVC: 6: DpVersion::getTotalBlocksVolume exit
400708_0000, newState=active
2008-01-13 08:03:10: SVFS: 6: D113118_736: After Merge err:4 full:0 mD:0
2008-01-13 08:05:10: DPSVC: 6: refreshSnap: cur=1200229510, dl=1200229520, kbytes=0, setup=0, rate=1000
400708_0000, curState=active, input=refresh
2008-01-13 08:05:10: DPSVC: 6: DpRequest::execute() BEGIN reqType:DpRequest_Version
Int_SchSrcRefresh reqCaller:DpRequest_Caller_Scheduler reqMode:0
--More--

Note: This is a partial listing due to the length of the output.

EXAMPLE #3  To display the log file without the time stamp, type:
$ server_log server_2 -n

NAS LOG for slot 2:
-------------------
0 keys=0 h=0 nc=0

SVFS: 6: Merge Start FsVol:118 event:0x0
SVFS: 6: D113118_736: hDr:82944 currInd:6, Destpmdv:D114118_503
CFS: 6: Resuming fs 24
SVFS: 6: 118:D113118_736:Merge hDr:82944 prev=99328 id=113 chunk=0 stableEntry=7
UFS: 6: Volume name:Sh122113
UFS: 6: starting gid map file processing.
UFS: 6: gid map file processing is completed.
DPSVC: 6: DpRequest::done() BEGIN reqType:DpRequest_Version
Int_SchSrcRefresh reqCaller:DpRequest_Caller_Scheduler status:0
DPSVC: 6: SchedulerSrc=118_APM00062400708_0000_253_APM00062400708_0000, curState=active, input=refreshDone
The server Commands

DPSVC: 6: DpVersion::getTotalBlocksVolume enter
DPSVC: 6: DpVersion::getTotalBlocksVolume found newV 118.ckpt003, blocks 17534
DPSVC: 6: DpVersion::getTotalBlocksVolume 0 blocks for vnumber 1038 totalB 0
DPSVC: 6: DpVersion::getTotalBlocksVolume found oldV 118.ckpt004
DPSVC: 6: DpVersion::getTotalBlocksVolume exit
DPSVC: 6: DpVersion::getTotalBytes 0 blocks 0 bytes
DPSVC: 6: SchedulerSrc=118_APM00062400708_0000_253_APM00062400708_0000, newState =active
SVFS: 6: D113118_736: After Merge err:4 full:0 mD:0
SVFS: 6: D113118_737: Chunk:0 hdrAdd:50176 ==> prevChunk:82944 before changePrev Chunk
SVFS: 6: D113118_737: Ch:0 hdr:50176 : prevCh:99328 after changePrev
DPSVC: 6: refreshSnap: cur=1200229510, dl=1200229520, kbytes=0, setup=0, rate=10 00
DPSVC: 6: SchedulerSrc=199_APM00062400708_0000_258_APM00062400708_0000, newState =active, input=refresh
DPSVC: 6: DpRequest::execute() BEGIN reqType:DpRequest_VersionInt_SchSrcRefresh reqCaller:DpRequest_Caller_Scheduler reqMode:0
DPSVC: 6: DpRequest::execute() END reqType:DpRequest_VersionInt_SchSrcRefresh reqCaller:DpRequest_Caller_Scheduler status:0 reqMode:0
DPSVC: 6: SchedulerSrc=199_APM00062400708_0000_258_APM00062400708_0000, newState =active
VBB: 6: VBB session list empty
CFS: 6: fs 0x78 type = dhfs being unmounted. Waiting for quiesce ...
CFS: 6: fs 0x78 type = dhfs unmounted
--More--

Note: This is a partial listing due to the length of the output.

EXAMPLE #4 To display all the current logs available, type:

$ server_log server_2 -a

NAS LOG for slot 2:
--------------------
1200152950: DPSVC: 6: refreshSnap: cur=1200152950, dl=1200152960, kbytes=0, setup=0, rate=666
1200152950: DPSVC: 6: SchedulerSrc=199_APM00062400708_0000_258_APM00062400708_0000, newState=active, input=refresh
1200152950: DPSVC: 6: DpRequest::execute() BEGIN reqType:DpRequest_VersionInt_SchSrcRefresh reqCaller:DpRequest_Caller_Scheduler reqMode:0
1200152950: DPSVC: 6: DpRequest::execute() END reqType:DpRequest_VersionInt_SchSrcRefresh reqCaller:DpRequest_Caller_Scheduler status:0 reqMode:0
1200152950: DPSVC: 6: SchedulerSrc=199_APM00062400708_0000_258_APM00062400708_0000, newState=active
1200152950: VBB: 6: VBB session list empty
1200152950: CFS: 6: fs 0x78 type = dhfs being unmounted. Waiting for quiesce ...
The server Commands

1200152950: CFS: 6: fs 0x78 type = dhfs unmounted
1200152950: SVFS: 6: pause() requested on fsid:78
1200152950: SVFS: 6: pause done on fsid:78
1200152950: SVFS: 6: Cascaded Delete...
1200152950: SVFS: 6: D120199_1131: createBlockMap PBM root=0 keys=0 h=0 nc=0
1200152950: SVFS: 6: Merge Start FsVol:199 event:0x0
1200152950: CFS: 6: Resuming fs 78
1200152950: SVFS: 6: 199:D120199_1130:Merge hdr=99328 prev=82944 id=120 chunk=0 stableEntry=7
1200152950: UFS: 6: Volume name:Sh217120
1200152950: SVFS: 6: D120199_1130: After Merge err:4 full:0 md:0
1200152950: D113118_736: hdr:82944 currInd:6, Destpmdv:D114118_503
1200229390: Resuming fs 24
1200229390: 26042826752: Merge Start FsVol:118 event:0x0
1200229390: 26040008704: Resuming fs 24
1200229390: 26042433536: Volume name:Sh122113
1200229390: 26042433536: starting gid map file processing.
1200229390: 26042433536: gid map file processing is completed.
1200229390: 26045513728: DpRequest::done() BEGIN reqType:DpRequest_VersionInt_SchSr cRefresh reqCaller:DpRequest_Caller_Scheduler status:0
1200229390: DPSVC: 6: SchedulerSrc=118_APM00062400708_0000_253_APM00062400708_0000, curState=active, input=refreshDone

Note: This is a partial listing due to the length of the output.

---More---

EXAMPLE #5  To display the current log in terse form, type:

```
$ server_log server_2 -t
```

NAS LOG for slot 2:
-------------------
0 keys=0 h=0 nc=0

```
1200229390: 26043285504: 122: Allocating chunk:3 Add:50176 Chunks:24
1200229390: 26042826752: Merge Start FsVol:118 event:0x0
1200229390: 26042826752: D113118_736: hdr:82944 currInd:6, Destpmdv:D114118_503
1200229390: 26040008704: Resuming fs 24
1200229390: 26042433536: Volume name:Sh122113
1200229390: 26042433536: starting gid map file processing.
1200229390: 26042433536: gid map file processing is completed.
1200229390: 26045513728: DpRequest::done() BEGIN reqType:DpRequest_VersionInt_SchSr cRefresh reqCaller:DpRequest_Caller_Scheduler status:0
1200229390: DPSVC: 6: SchedulerSrc=118_APM00062400708_0000_253_APM00062400708_0000, curState=active, input=refreshDone
1200229390: 26045513728: DpVersion::getTotalBlocksVolume enter
```

blocks 17534
EXAMPLE #6
To display the current log in verbose form, type:

```
$ server_log server_2 -v
```

DART Work Partition Layout found @ LBA 0x43000 (134MB boundary)

slot 2) About to dump log @ LBA 0xc7800

NAS LOG for slot 2:
---------------------

About to print log from LBA c8825 to c97ff
0 keys=0 h=0 nc=0

```
logged time  = 2008-01-13 08:03:10
id           = 26043285504
severity     = INFO
component    = DART
facility     = VRPL
baseid       = 0
type         = STATUS
argument name = arg0
argument value = 122: Allocating chunk:3 Add:50176 Chunks:24

argument type = string (8)
b brief description = 122: Allocating chunk:3 Add:50176 Chunks:24
full description   = No additional information is available.
```
The recommended action is No recommended action is available. Use the text from the error message's brief description to search the Knowledgebase on Powerlink. After logging in to Powerlink, go to Support > Knowledgebase Search > Support Solutions Search.

logged time = 2008-01-13 08:03:10
id = 26042826752
severity = INFO
component = DART
facility = SVFS
baseid = 0
type = STATUS
argument name = arg0
argument value = Merge Start FsVol:118 event:0x0
argument type = string (8)
brief description = string (8)
full description = No additional information is available.

Note: This is a partial listing due to the length of the output.
server_mount

Mounts file systems and manages mount options for the specified Data Movers.

SYNOPSIS

server_mount {<movername>|ALL}
  [-all]
  | [-Force][-option <options>] <fs_name> [<mount_point>]
<options>:
  [ro|rw][primary=<movername>]
  [ceppcifs]
  [ceppnfs]
  [nonotify][nooplock]
  [notifyonaccess][notifyonwrite]
  [accesspolicy={NT|UNIX|SECURE|NATIVE|MIXED|MIXED_COMPAT}]
  [nolock|wlock|rwlock]
  [cvfsname=<newname>]
  [noscan]
  [noprefetch]
  [uncached]
  [cifssyncwrite]
  [triggerlevel=<value>]
  [ntcredential]
  [renamepolicy={CIFS|FULL|NO}]
  [cifsnanoroundup]
  [nfsv4delegation={NONE|READ|RW}]

DESCRIPTION

server_mount attaches a file system to the specified <mount_point> with the specified options, and displays a listing of mounted file systems. server_umount unmounts the file system.

The ALL option executes the command for all Data Movers.

Note: The primary=<movername> option is not used.

OPTIONS

No arguments
Displays a listing of all mounted, and temporarily unmounted file systems.

-all
Mounts all file systems in the mount table.

-Force -option rw <fs_name> <mount_point>
Forces a mount of a file system copy (created using fs_timefinder) as read-write. By default, all file system copies are mounted as read-only.
<fs_name> <mount_point>
Mounts a file system to the specified <mount_point>. When a file system is initially mounted, the <mount_point> is required; however, remounting a file system after a temporary unmount does not require the use of a <mount_point>.

[-option <options>]
Specifies the following comma-separated options:

[ro|rw]
Specifies the mount as read-write (default), or read-only which is the default for checkpoints and TimeFinder/FS.

Note: MPFS clients do not acknowledge file systems that are mounted read-only and allow their clients to write to the file system.

[ceppcifs]
It enables the CEPA events for CIFS on a file system. This option is enabled by default.

[ceppnfs]
It enables the CEPA events for NFS on a file system.

Note: If ceppnfs is used without the ceppcifs option, the CEPA events for CIFS are disabled. To enable CEPA events for NFS and CIFS on a file system, make sure that you add both these options in the command.

[accesspolicy-={NT UNIX SECURE NATIVE MIXED MIXED_COMPAT}]
Indicates the access control policy as defined in the table.

Note: When accessed from a Windows client, ACLs are only checked if the CIFS user authentication method is set to the recommended default, NT. This is set using the -add security option in the server_cifs command.

<table>
<thead>
<tr>
<th>Access policy</th>
<th>CIFS clients</th>
<th>NFS clients</th>
</tr>
</thead>
<tbody>
<tr>
<td>NATIVE (default)</td>
<td>ACL is checked.</td>
<td>UNIX rights are checked.</td>
</tr>
<tr>
<td>UNIX</td>
<td>ACL and UNIX rights are checked.</td>
<td>UNIX rights are checked.</td>
</tr>
<tr>
<td>NT</td>
<td>ACL is checked.</td>
<td>ACL and UNIX rights are checked.</td>
</tr>
<tr>
<td>Access policy</td>
<td>CIFS clients</td>
<td>NFS clients</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SECURE</td>
<td>ACL and UNIX rights are checked.</td>
<td>ACL and UNIX rights are checked.</td>
</tr>
<tr>
<td>MIXED</td>
<td>ACL is checked. If there is not an ACL, one is created based on the UNIX mode bits. Access is also determined by the ACL. NFSv4 clients can manage the ACL. An ACL modification rebuilds the UNIX mode bits but the UNIX rights are not checked.</td>
<td>ACL is checked. If there is not an ACL, one is created based on the UNIX mode bits. Access is also determined by the ACL. NFSv4 clients can manage the ACL. A modification to the UNIX mode bits rebuilds the ACL permissions but the UNIX rights are not checked.</td>
</tr>
<tr>
<td>MIXED_COMPAT</td>
<td>If the permissions of a file or directory were last set or changed by a CIFS client, the ACL is checked and the UNIX rights are rebuilt but are not checked. If the permissions of a file or directory were last set or changed by an NFS client, the UNIX rights are checked and the ACL is rebuilt but is not checked.</td>
<td>If the permissions of a file or directory were last set or changed by an NFS client, the UNIX rights are checked and the ACL is rebuilt but is not checked. If the permissions of a file or directory were last set or changed by a CIFS client, the ACL is checked and the UNIX rights are rebuilt but are not checked.</td>
</tr>
</tbody>
</table>

**Note:** The MIXED policy translates the UNIX ownership mode bits into three ACEs: Owner, Group, and Everyone, which can result in different permissions for the Group ACE and the Everyone ACE. The MIXED_COMPAT policy does not translate a UNIX Group into a Group ACE. The Everyone ACE is generated from the UNIX Group.

```
cvfsname=<newname>
```

Changes the default name of the checkpoint in each of the .ckpt directories. The default name is the timestamp of when the checkpoint was taken.

```
noprefetch
```

Turns prefetch processing off. When on (default), performs read ahead processing for file systems.

**CAUTION**

Turning the prefetch option to off may affect performance.
The server Commands

For CIFS clients only

When mounting a file system, if the default options are not manually entered, the options are active but not displayed in the listing of mounted file systems. Available options are:

[nonotify]
Turns notify off. When on (default), the notify option informs the client of changes made to the directory file structure.

[nooplock]
Turns opportunistic locks (oplocks) off. When oplocks are on (default), they reduce network traffic by enabling clients to cache the file and make changes locally. To turn Windows oplocks off, unmount the file system, then remount with nooplock.

[notifyonaccess]
Provides a notification when a file system is accessed. By default, notifyonaccess is disabled.

[notifyonwrite]
Provides a notification of write access to a file system. By default, the notifyonwrite option is disabled.

[noscan]
Disables the Virus Checker protocol for a file system. The Virus Checker protocol is enabled using server_setup and managed by server_viruschk.

[uncached]
Allows well-formed writes (that is, multiple of a disk block and disk block aligned) to be sent directly to the disk without being cached on the server.

[cifssyncwrite]
Performs an immediate synchronous write on disk independently of CIFS write protocol option. This can impact write performance.

[triggerlevel=<value>]
Specifies the deepest directory level at which notification occurs. The default is 512. The value -1 disables the notification feature.

[ntcredential]
Enables the VNX to take full account of a user’s Windows group memberships when checking an ACL for access through NFS. When a UNIX user initiates a full request for a file system object, the UNIX UID is mapped to the Windows SID, then merges the user’s UNIX and Windows groups together to generate a Windows NT Credential. This applies to NT, SECURE, MIXED, and MIXED_COMPAT access-checking policies.
The server Commands

(renamepolicy={CIFS | FULL | NO})
Enables or disables control if any file or directory is opened on the current directory or any subdirectory, before the current directory is renamed. CIFS (default) stops the renaming of CIFS directories when in use by CIFS clients. FULL denies permission for the renaming of CIFS and NFS directories when in use by CIFS or NFS clients. NO automatically performs the directory rename without checking if a CIFS or NFS client is opening the directory.

Note: The renamepolicy is not supported by NFSv4.

cifsnanoroundup
Rounds up to the next second any date set by a CIFS client.

For NFS clients only

[nolock | wlock | rwlock]
Indicates the impact of locking behavior on NFSv2 and NFSv3 clients against NFSv4 and CIFS file locking. In NFSv2 and NFSv3, locking rules are cooperative, so a client is not prevented from accessing a file locked by another client if it does not use the lock procedure. NFSv2 and NFSv3 locks as advisory. An advisory lock does not affect read and write access to the file, but informs other users that the file is already in use.

Note: NFSv4 and CIFS clients have mandatory locking schemes and do not require a locking policy.

<table>
<thead>
<tr>
<th>Locking policy</th>
<th>NFS clients</th>
</tr>
</thead>
<tbody>
<tr>
<td>nolock</td>
<td>This (default) can open and write to a file when it is locked by CIFS or NFSv4 clients.</td>
</tr>
<tr>
<td>wlock</td>
<td>This can read but cannot write data to a file locked by CIFS or NFSv4 clients.</td>
</tr>
<tr>
<td>rwlock</td>
<td>This (recommended) cannot read or write data to files locked by CIFS or NFSv4 clients.</td>
</tr>
</tbody>
</table>

(nfsv4delegation=(NONE | READ | RW))
Indicates that specific actions on a file are delegated to the NFSv4 client. NONE indicates that no file delegation is granted. READ indicates only read delegation is granted. RW (default) indicates write delegation is granted.

SEE ALSO Managing Volumes and File Systems with VNX Automatic Volume Management, Managing Volumes and File Systems for VNX Manually,
The server Commands

Configuring NFS on VNX, Configuring and Managing CIFS on VNX, Using VNX SnapSure, nas_fs, server_checkup, server_export, server_mountpoint, server_nfs, server_setup, server_umount, and server_viruschk.

**EXAMPLE #1** To display all mounted file systems on server_2, type:

```bash
$ server_mount server_2
```

```
server_2 : root_fs_2 on /uxfs,perm,rw
root_fs_common on /.etc_common uxfs,perm,ro
ufs2 on /ufs2 uxfs,perm,rw
```

**EXAMPLE #2** To mount all file systems temporarily umounted from the mount table of server_2, type:

```bash
$ server_mount server_2 -all
```

```
server_2 : done
```

**EXAMPLE #3** To mount ufs1, on mount point /ufs1, and enable CEPP for both CIFS and NFS, type:

```bash
$ server_mount server_2 -o ceppcifs,ceppnfs ufs1 /ufs1
```

```
server_2 : done
```

**EXAMPLE #4** To mount ufs1, on mount point /ufs1, with nonotify, nolock, and cifssyncwrite turned on, type:

```bash
$ server_mount server_2 -option nonotify,nolock,cifssyncwrite ufs1 /ufs1
```

```
server_2 : done
```

**EXAMPLE #5** To mount ufs1, on mount point /ufs1, with the access policy set to NATIVE, and nooplock turned on, type:

```bash
$ server_mount server_2 -option accesspolicy=NATIVE,nooplock ufs1 /ufs1
```

```
server_2 : done
```
EXAMPLE #6  To mount ufs1, on mount point /ufs1, with noscan and noprefetch set to on, type:

$ server_mount server_2 -option noscan,noprefetch ufs1 /ufs1  

server_2 : done

EXAMPLE #7  To mount ufs1, on mount point /ufs1, with notifyonaccess, notifyonwrite set to on, type:

$ server_mount server_2 -option notifyonaccess,notifyonwrite ufs1 /ufs1  

server_2 : done

EXAMPLE #8  To mount a copy of a file system ufs1_snap1 on mount point /ufs1_snap1 with read-write access, type:

$ server_mount server_2 -Force -option rw ufs1_snap1 /ufs1_snap1  

server_2 : done

EXAMPLE #9  To mount ufs1, on mount point /ufs1, with uncached writes turned on, type:

$ server_mount server_2 -option uncached ufs1 /ufs1  

server_2 : done

EXAMPLE #10  To mount ufs1, on mount point /ufs1, with the trigger level of notification change set to 256, type:

$ server_mount server_2 -option triggerlevel=256 ufs1 /ufs1  

server_2 : done

EXAMPLE #11  To mount ufs1, on mount point /ufs1, change the default name of the checkpoint in the .ckpt directory, and specify a mount point, type:

$ server_mount server_2 -option cvfsname=test ufs1 /ufs1  

server_2 : done

EXAMPLE #12  To mount ufs1, on mount point /ufs1, with the access policy set to MIXED, type:

$ server_mount server_2 -option accesspolicy=MIXED ufs1 /ufs1  

server_2 : done

EXAMPLE #13  To mount ufs1, on mount point /ufs1, with the access policy set to MIXED_COMPAT, type:

$ server_mount server_2 -option accesspolicy=MIXED_COMPAT ufs1 /ufs1
EXAMPLE #14  To mount ufs1, as a part of the nested file system nmfs1, type:

$ server_mount server_2 ufs1 /nmfs1/ufs1

EXAMPLE #15  To mount ufs1, specifying that no file is granted to the NFSv4 client, type:

$ server_mount server_2 ufs1 nfsv4delegation=NONE
**server_mountpoint**

Manages mount points for the specified Data Movers.

**SYNOPSIS**

```
server_mountpoint {<movername>|ALL}
  -list
  | {-create|-delete|-exist} <pathname>
```

**DESCRIPTION**

`server_mountpoint` creates, deletes, lists, or queries a mount point for the specified Data Mover or all Data Movers.

The **ALL** option executes the command for all Data Movers.

**OPTIONS**

- **-list**
  Lists all mount points for the specified Data Movers.

- **-create <pathname>**
  Creates a mount point. A `<pathname>` must begin with a slash (`/`).

- **-delete <pathname>**
  Deletes a mount point.

- **-exist <pathname>**
  Displays whether or not a mount point exists.

**SEE ALSO**


**EXAMPLE #1**

To create a mount point on *server_2*, type:

```
$ server_mountpoint server_2 -create /ufs1
```

```
s: done
```

**EXAMPLE #2**

To list all mount points on a *server_2*, type:

```
$ server_mountpoint server_2 -list
```

```
s: .etc_common
ufs1
ufs1_ckpt1
ufs2
ufs3
```
EXAMPLE #3  To verify that the mount point /ufs1, exists on all Data Movers, type:

   $ server_mountpoint ALL -exist /ufs1

server_2 : /ufs1 : exists
server_3 : /ufs1 : does not exist

EXAMPLE #4  To delete the mount point /ufs1, on server_2, type:

   $ server_mountpoint server_2 -delete /ufs1

server_2 : done
**server_mpfs**

Sets up and configures MPFS protocol.

**SYNOPSIS**

```
server_mpfs {<movername>|ALL}
    -set <var>=<value>
    -add <number_of_threads>
    -delete <number_of_threads>
    -Stats
    -Default [<var>]
    -mountstatus
```

**DESCRIPTION**

`server_mpfs` sets up the MPFS protocol. The configuration values entered with this command are saved into a configuration file on the Data Mover. MPFS is not supported on the NS series.

`server_setup` provides information to start and stop MPFS for a Data Mover.

The **ALL** option executes the command for all Data Movers.

**OPTIONS**

No arguments

Lists the current MPFS configuration.

- **-set <var>=<value>**
  Sets the specified value for the specified variable. Currently, the only valid `<var>` is **threads**.

If this command is executed before the `server_setup -P mpfs -o start` command is issued, the system sets the number of threads that will be started with the `server_setup -o start` command, thereby overriding the default number of threads. If this command is executed after MPFS service is started, threads are to be added and removed dynamically.

- **-add <number_of_threads>**
  Increases the previously specified number of MPFS threads (default=16) by `<number_of_threads>` for the specified Data Movers.

- **-delete <number_of_threads>**
  Decreases the number of threads by the `<number_of_threads>` indicated for the specified Data Movers.

- **-Stats**
  Displays the current MPFS server statistics.
**-mountstatus**
Displays the mountability of file systems for MPFS.

Certain file systems cannot be mounted using MPFS, therefore before attempting to mount a file system on an MPFS client, compatibility should be determined. File systems that are not supported are running quotas, have checkpoints, or are using TimeFinder/FS.

**-Default** [<var>]
Without a <var> entry, resets all variables to their factory-default values. Currently the only valid <var> is threads.

If a <var> is specified, only the specified value is reset to its factory-default value.

---

**Note:** Variable names are case-sensitive.

**SEE ALSO**
Using VNX Multi-Path File System, server_setup, and server_mt.

**EXAMPLE #1**
To set a value for a specified MPFS variable, type:

```
$ server_mpfs server_2 -set threads=32
server_2 :done
```

**EXAMPLE #2**
To display the MPFS stats for server_2, type:

```
$ server_mpfs server_2 -Stats
```

---

**EXAMPLE #3**
To reset all variables back to their factory default value, type:

```
$ server_mpfs server_2 -Default
server_2 :done
```

**EXAMPLE #4**
To check the mount status of a Data Mover, type:

```
$ server_mpfs server_2 -mountstatus
```

---

<table>
<thead>
<tr>
<th>fs</th>
<th>mpfs compatible?</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>no</td>
<td>not a ufs file system</td>
<td></td>
</tr>
<tr>
<td>testing_renaming</td>
<td>no</td>
<td>volume structure not FMP compatible</td>
</tr>
<tr>
<td>server2_fs1_ckpt</td>
<td>no</td>
<td>volume structure not FMP compatible</td>
</tr>
</tbody>
</table>
The server Commands

Example #5
To add 16 threads to server_2, type:

```
$ server_mpfs server_2 -add 16
```

server_2 : done

Example #6
To delete 16 threads from server_2, type:

```
$ server_mpfs server_2 -delete 16
```

server_2 : done
**server_mt**

Manages the magnetic tape drive for the specified Data Mover.

**SYNOPSIS**

```
server_mt <movername>
[-f <device>][-N] <command> [<count>]
```

**DESCRIPTION**

`server_mt` sends commands to a magnetic tape drive located on the specified Data Mover.

By default, `server_mt` performs the requested operation once. Specify `<count>` to perform multiple operations.

**OPTIONS**

Operations for the command are as follows:

- `eof`, `weof`
  Write `<count>` EOF marks at the current position on the tape.

- `fsf`
  Forward space over `<count>` EOF marks. The tape is positioned on the first block of the file.

- `bsf`
  Back space over `<count>` EOF marks. The tape is positioned on the beginning-of-tape side of the EOF mark.

- `asf`
  Absolute space to `<count>` file number. This is equivalent to a rewind followed by a `fsf <count>`.

For the following commands, `<count>` is ignored:

- `eom`
  Space to the end of recorded media on the tape. This is useful for appending files onto previously written tapes.

- `rewind`
  Rewind the tape.

- `offline, rewoffl`
  Rewind the tape and, if appropriate, take the drive unit offline by unloading the tape.

- `status`
  Print status information about the tape unit.
The server Commands

**erase**
Erase the entire tape.

- **-f** <device>
  Specifies the magnetic tape device to which the command is applied, and the <device> which is the name of the tape drive.

- **-N**
  Indicates no rewind at the end of the tape.

The `server_devconfig <movername> -list -scsi -nondisks` command displays a listing of devices.

**EXAMPLE**
To send the **rewind** command to **tape1** (magnetic tape drive) on a Data Mover, type:

```
$ server_mt server_2 -f tape1 rewind
```

```
server_2: done
```
server_name

Manages the name for the specified Data Movers.

You must delete all user-defined interconnects configured for a Data Mover before you can rename it using this command. After you rename the Data Mover, you must re-create the source and peer interconnects with the new Data Mover name and then restart any associated replication sessions.

SYNOPSIS

server_name {<movername>|ALL} [<new_name>]

DESCRIPTION

server_name displays and renames the current movername for the specified Data Mover.

The ALL option executes the command for all Data Movers.

OPTIONS

No arguments
Displays movernames for all Data Movers.

<new_name>
Changes the current movername to the newly specified movername. The movername is case-sensitive and supports the following characters: a through z, A through Z, 0 through 9, _ (underscore), - (hyphen), though names may not start with a hyphen.

The ALL option is not available when renaming Data Movers.

SEE ALSO

nas_rp.

EXAMPLE #1

To change the current movername for a Data Mover, type:

$ server_name server_2 my_srv2

server_2 : my_srv2

EXAMPLE #2

To view all movernames, type:

$ server_name ALL

server_2 : my_srv2
server_3 : my_srv3
server_4 : my_srv4
server_5 : my_srv5
server_netstat

Displays the network statistics for the specified Data Mover.

SYNOPSIS

server_netstat {<movername>}
[-a][-i][-r][-s][-p {tcp|udp|icmp|ip}]

DESCRIPTION

server_netstat displays network statistics for the specified Data Mover.

OPTIONS

No arguments
Displays the active TCP connections.

-a
Displays the state of all the IP, ICMP, TCP, and UDP sockets.

-i
Displays a summary of the state of all physical interfaces including
the total amount of data transferred both ways and the number of
errors during this process as well as the MTU size and MAC address.

-r
Displays the routing table.

-s
Displays the statistics of each protocol.

-p {tcp|udp|icmp|ip}
Limits the display of the sockets, or the statistics of each protocol to
the specified protocol. tcp is default.

SEE ALSO

server_ifconfig and server_route.

EXAMPLE #1
To display all active IP, ICMP, TCP, and UDP sockets, type:

$ server_netstat server_2

Proto Local Address                  Foreign Address       (state)
******************************************************************
tcp   172.24.102.237.2483            172.24.102.17.445     ESTABLISHED

tcp   172.24.102.237.netbios-ssn     172.24.103.63.3962    ESTABLISHED

EXAMPLE #2
To display the states of the IP, ICMP, TCP, and UDP sockets, type:

$ server_netstat server_2 -a

Proto Local Address                  Foreign Address       (state)
******************************************************************
tcp   172.24.102.237.2483            172.24.102.17.445     ESTABLISHED
## EXAMPLE #3

To display a summary of the state of all physical interfaces, type:

```bash
$ server_netstat server_2 -i
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Mtu</th>
<th>Ibytes</th>
<th>Ierror</th>
<th>Obytes</th>
<th>Oerror</th>
<th>PhysAddr</th>
</tr>
</thead>
<tbody>
<tr>
<td>fxp0</td>
<td>1500</td>
<td>758568220</td>
<td>0</td>
<td>534867239</td>
<td>0</td>
<td>8:0:1b:43:49:9a</td>
</tr>
<tr>
<td>cge0</td>
<td>9000</td>
<td>18014329</td>
<td>0</td>
<td>7195540</td>
<td>0</td>
<td>8:0:1b:42:46:3</td>
</tr>
<tr>
<td>cge1</td>
<td>9000</td>
<td>306495706</td>
<td>0</td>
<td>9984</td>
<td>0</td>
<td>8:0:1b:42:46:4</td>
</tr>
</tbody>
</table>
The server Commands

EXAMPLE #4  To display routing table statistics, type:

```
$ server_netstat server_2 -r
```

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Mask</th>
<th>Type</th>
<th>Proto</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0</td>
<td>172.24.102.254</td>
<td>255.255.255.0</td>
<td>DIRECT</td>
<td>RIP</td>
<td>cge0</td>
</tr>
<tr>
<td>128.221.253.0</td>
<td>128.221.253.2</td>
<td>255.255.255.0</td>
<td>DIRECT</td>
<td>RIP</td>
<td>fxp0</td>
</tr>
<tr>
<td>172.24.102.0</td>
<td>172.24.102.237</td>
<td>255.255.255.0</td>
<td>DIRECT</td>
<td>RIP</td>
<td>cge0</td>
</tr>
<tr>
<td>128.221.252.0</td>
<td>128.221.252.2</td>
<td>255.255.255.0</td>
<td>DIRECT</td>
<td>RIP</td>
<td>fxp0</td>
</tr>
</tbody>
</table>

EXAMPLE #5  To display the statistics of each protocol, type:

```
$ server_netstat server_2 -s
```

ip:

***
2315636 total packets received
0 bad header checksums
0 with unknown protocol
4 fragments received
0 fragments dropped (dup or out of space)
0 fragments dropped after timeout
4 packets reassembled
2 packets forwarded
13046 packets not forwardable
13046 no routes
2302596 packets delivered
2267772 total packets sent
3 packets fragmented
0 packets not fragmentable
6 fragments created

icmp:

*****
162 calls to icmp_error
Output histogram:
  echo reply: 1079145
destination unreachable: 90
echo: 1996
Input histogram:
  echo reply: 1993
destination unreachable: 162
routing redirect: 0
echo: 1079145
time exceeded: 0
address mask request: 0
1081300 messages received
1081231 messages sent

tcp:
****
437648 packets sent
2 data packets retransmitted
0 resets
434138 packets received
212 connection requests
19 connections lingered

udp:
****
0 incomplete headers
27048 bad ports
760361 input packets delivered
744999 packets sent

**EXAMPLE #6** To display TCP protocol statistics, type:

$ server_netstat server_2 -s -p tcp

tcp:
****
437690 packets sent
2 data packets retransmitted
0 resets
434195 packets received
212 connection requests
19 connections lingered
server_nfs

Manages the NFS service, including secure NFS and NVSv4, for the specified Data Movers.

SYNOPSIS

server_nfs {<movername>|ALL}
   -secnfs
     [-service {<start>|<stop>}
     | -user {<list>|<info> {principal=<user_principal>|handle=<handle_number>}}
     | -release {principal=<user_principal>|handle=<handle_number>}}
     | -principal {<create> <service@server>|<delete> <service@server>}
     | -mapper
       {<info>|<set> {<source> {auto|nis|file} [path=<file_path>]}|<passwddb {nis|file}}>}
     | -v4
       [-service {<start>|<stop>}
       | -client {<list>}
       | -info {index=<index>|hostname=<host>|ipaddress=<ipaddr>}
       | -release {index=<index>|hostname=<host>|ipaddress=<ipaddr>}}
       | -pnfs -service {<start>|<stop>}
       | -stats [-zero] [-rpc | -nfs]}

DESCRIPTION

server_nfs manages the NFS service, including secure NFS and NVSv4.

For secure NFS, server_nfs displays the status of the secure NFS service, stops and starts Kerberos authentication for secure NFS, and manages users authenticated through secure NFS. It also manages secure NFS security principals, releases the security context of a particular user, and manages the secure NFS user mapping service. Secure NFS supports UNIX-based, Linux-based, and Windows-based Kerberos KDCs.

For NVSv4, server_nfs stops and starts the NVSv4 service, lists NVSv4 clients, and displays information about those clients.

Note: NVSv2 and v3 services are not started or stopped by this command.

The ALL option executes the command for all Data Movers.

OPTIONS

-secnfs
Displays the status of the secure NFS service.
**The server Commands**

- **-service** **{-start|-stop}**
  Enables or disables the secure NFS service on the Data Mover.

- **-user -list**
  Lists all the NFS users authenticated by secure NFS.

- **-user -info** **{principal=<user_principal>|handle=<handle_number>}**
  Displays the attributes of an authenticated user as specified by the <user_principal> or <handle_number>. The <user_principal> is the Kerberos representation of a user. It is the username followed by the Kerberos realm name. The <handle_number> is the identifier of a secure NFS user authentication context and is displayed using **-user -list**.

- **-user -release** **{principal=<user_principal>|handle=<handle_number>}}**
  Releases the authentication context of the specified user.

- **-principal** **{-create <service@server>|-delete <service@server>}}**
  Creates or deletes a secure NFS service instance.

- **-mapper -info**
  Displays the secure NFS mapping service configurations for the local file.

- **-mapper -set** **{-source {auto|nis|file [path=<file_path>]}**
  Sets the type of mapping provider and the file path if it is a local file (the default file is gsscred_db). These options are only relevant when using a Solaris UNIX Kerberos KDC. The default and recommended value is **-source auto**.

- **-mapper -set -passwddb {nis|file}**
  Sets the location of the password database.

- **-mapper -mapping -list** **[name=<user_name>|uid=<UID>]**
  Lists one or more user-mapping records as specified by <user_name> or <UID>. This option is only relevant if you are using a local mapping file and a Solaris UNIX Kerberos KDC.

- **-mapper -mapping -create** **{name=<user_name>|uid=<UID>}}**
  Creates a new mapping record. This option is only relevant if you are using a local mapping file and a Solaris UNIX Kerberos KDC.
The server Commands

-mapper mapping delete [name=<user_name> | uid=<UID>]
Deletes the mapping record from the database. This option is only relevant if you are using a local mapping file and a Solaris UNIX Kerberos KDC.

-v4
Displays the status of the NFSv4 service.

-service {start | stop}
Enables or disables the NFSv4 service on the Data Mover.

To start the NFSv4 service, the /nas/server/slot_<x>/config file, where <x> is the slot number of the Data Mover, must be configured so that the hivers argument is set to 4. If the hivers is not set to 4, append the nfs start line and ensure that the hivers value is set to 4. Restart the Data Mover to apply the change. Other versions of NFS can continue to run even if NFSv4 is disabled.

-client list
Lists all NFSv4 client systems that have established state with the VNX NFSv4 server.

-client info [index | hostname=<host> | ipaddress=<ipaddr>]
Displays the attributes of the client (clientID) as specified by the index, <host>, or <ipaddr>. The index value is obtained from the client list.

-client release [index | hostname=<host> | ipaddress=<ipaddr>]
Releases the state of the client (clientID) as specified by the index, <host>, or <ipaddr>. The index value is obtained from the client list.

-pnfs service {start | stop}
The pNFS is part of the NFS version 4.1 that allows clients to access storage devices directly and in parallel.

The pNFS service starts automatically once the NFSv4 starts. The pNFS service is stopped temporarily by using the -stop argument. The service can be restarted using the -start argument.

-stats [-zero]
Displays all statistics or resets to zero all NFS and RPC statistics.
The server Commands

[-rpc | -nfs]
Displays RPC statistics or displays NFS statistics.

[-zero] [-rpc]
Resets to zero all RPC statistics.

[-zero] [-nfs]
Resets to zero all NFS statistics.

SEE ALSO
Configuring NFS on VNX and server_kerberos.

EXAMPLE #1
To display the status of the secure NFS service, type:

$ server_nfs server_2 -secnfs

server_2 :
RPCSEC_GSS server stats

Credential count: 2
  principal: nfs@dm112-cge0.nasdocs.emc.com
  principal: nfs@dm112-cge0

  Total number of user contexts: 1
  Current context handle: 3

EXAMPLE #2
To enable secure NFS service on server_2, type:

$ server_nfs server_2 -secnfs -service -start

server_2 : done

EXAMPLE #3
To disable secure NFS service on server_2, type:

$ server_nfs server_2 -secnfs -service -stop

server_2 : done

EXAMPLE #4
To display all secure NFS service instances, type:

$ server_nfs server_2 -secnfs -user -list

server_2 :
RPCSEC_GSS server stats

Credential count: 2
  principal: nfs@dm112-cge0.nasdocs.emc.com
  principal: nfs@dm112-cge0

  Total number of user contexts: 1
  Current context handle: 3

PARTIAL user contexts:
  Total PARTIAL user contexts: 0

USED user contexts:
principal=nfsuser1@NASDOCS.EMC.COM, service=nfs@dm112-cge0.nasdocs.emc.com, handle=3, validity=35914s
Total USED user contexts: 1

EXAMPLE #5  To display the attributes of an authenticated server as specified by the handle, type:

$ server_nfs server_2 -secnfs -user -info handle=3

EXAMPLE #6  To release the authentication context of the user specified by the handle, type:

$ server_nfs server_2 -secnfs -user -release handle=3

EXAMPLE #7  To create a secure NFS service instance, type:

$ server_nfs server_2 -secnfs -principal -create nfs1@dm112-cge0.nasdocs.emc.com

EXAMPLE #8  To delete a secure NFS service instance, type:

$ server_nfs server_2 -secnfs -principal -delete nfs1@dm112-cge0.nasdocs.emc.com

EXAMPLE #9  To set the mapping provider for the file, type:

$ server_nfs server_2 -secnfs -mapper -set -source file

EXAMPLE #10 To set the location of the password database, type:

$ server_nfs server_2 -secnfs -mapper -set -passwddb file

EXAMPLE #11 To display the secure NFS mapping service configurations for the local file, type:

$ server_nfs server_2 -secnfs -mapper -info
The server Commands

```
gsscred db = File /etc/gsscred.db
gsscred db version = Dart_v1
passwd db = File
```

EXAMPLE #12  To create a new mapping record, type:

```
$ server_nfs server_2 -secnfs -mapper -mapping -create
name=nfsuser1
```

server_2 : done

EXAMPLE #13  To display a list of the mapping records, type:

```
$ server_nfs server_2 -secnfs -mapper -mapping -list
```

server_2 :
0401000B06092A864886F712010202000001A7365636E667573657231407374617465732E65632E636F6D
1000      nfsuser1, kerberos_v5

EXAMPLE #14  To delete a mapping record, type:

```
$ server_nfs server_2 -secnfs -mapper -mapping -delete
name=nfsuser1
```

server_2 : done

EXAMPLE #15  To enable the NFSv4 service on server_2, type:

```
$ server_nfs server_2 -v4 -service -start
```

server_2 : done

EXAMPLE #16  To start the NFSv4 service, type:

```
$ server_nfs {<movername>|ALL} -v4 -service -start
```

server_2 : done

Note: The pNFS service starts automatically once the NFSv4 starts. To start the NFSv4 service, ensure that the /nas/server/slot_<x>/config file has the hivers set to 4, where <x> is the slot number, for example slot_2, slot_3, and so on. The nfs entry in the above file should look like:

```
nfs config hivers=4.
```

EXAMPLE #17  To stop the pNFS service temporarily, type:

```
$ server_nfs {<movername>|ALL} -pnfs -service –stop
```

server_2 : done

Note: When the pNFS service is stopped, all granted layouts are recalled and requests for new layouts are denied.

EXAMPLE #18  To restart the pNFS service while the NFSv4 service is running, type:

```
$ server_nfs {<movername>|ALL} -pnfs -service –start
```
server_2 : done

**EXAMPLE #19**  To display the status of the NFSv4 service and the pNFS service, type:

```
$ server_nfs server_2 -v4
```

```
server_2 :
-------------- nfsv4 server status ---------------
  *   Service Started   *
  *   pNFS service Started   *
  *   (yet operating)   *

-------- NFSv4 Clients --------
Confired Clients : 1
UnConfired Clients : 0
Number of users : 0
Number of lock owners : 0
Longest List : 0
Shortest List : 0
Greatest depth to date : 0
Average List length : 0.00
Domain Name : Not Defined

-------- NFSv4 State --------
Opens : 4
Locks : 0
Delegations : 4
Layouts : 0
Free : 524280

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confirmed Clients</td>
<td>Active client (ready to work).</td>
</tr>
<tr>
<td>UnConfirmed Clients</td>
<td>Client in the process to establishing context.</td>
</tr>
<tr>
<td>Number of users</td>
<td>To be removed in non-debug images.</td>
</tr>
<tr>
<td>Longest List</td>
<td>To be removed in non-debug images.</td>
</tr>
<tr>
<td>Shortest List</td>
<td>To be removed in non-debug images.</td>
</tr>
<tr>
<td>Greatest depth to date</td>
<td>To be removed in non-debug images.</td>
</tr>
<tr>
<td>Average List length</td>
<td>To be removed in non-debug images.</td>
</tr>
<tr>
<td>Opens</td>
<td>Number of open files.</td>
</tr>
<tr>
<td>Locks</td>
<td>Number of locks being held.</td>
</tr>
<tr>
<td>Delegations</td>
<td>Number of granted delegations.</td>
</tr>
<tr>
<td>Free</td>
<td>To be removed in non-debug images.</td>
</tr>
</tbody>
</table>

**EXAMPLE #20**  To display all NFSv4 clients, type:
Command Line Interface Reference for File 7.0
### The server Commands

<table>
<thead>
<tr>
<th>proc</th>
<th>ncalls</th>
<th>%totcalls</th>
<th>ms/call</th>
<th>failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>v3null</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3getattr</td>
<td>233843</td>
<td>78.8</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3setattr</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3lookup</td>
<td>37130</td>
<td>12.5</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3access</td>
<td>11828</td>
<td>4.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3readlink</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3read</td>
<td>120</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3write</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3create</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3mkdir</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3symlink</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3mknod</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3remove</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3rmdir</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3rename</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3link</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3readdir</td>
<td>4005</td>
<td>1.3</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3readdirplus</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3fsstat</td>
<td>4878</td>
<td>1.6</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3fsinfo</td>
<td>4878</td>
<td>1.6</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3pathconf</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v3commit</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
</tbody>
</table>

Server nfs (v4):

<table>
<thead>
<tr>
<th>proc</th>
<th>ncalls</th>
<th>%totcalls</th>
<th>ms/call</th>
<th>failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>v4null</td>
<td>4</td>
<td>0.0</td>
<td>3.0</td>
<td>0</td>
</tr>
<tr>
<td>v4compound</td>
<td>33645</td>
<td>48.8</td>
<td>0.1</td>
<td>0</td>
</tr>
<tr>
<td>v4reserved</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4access</td>
<td>217</td>
<td>0.3</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4close</td>
<td>44</td>
<td>0.1</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4commit</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4create</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4delegPrg</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4delegRet</td>
<td>30</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4getAttr</td>
<td>858</td>
<td>1.2</td>
<td>0.1</td>
<td>0</td>
</tr>
<tr>
<td>v4getFh</td>
<td>220</td>
<td>0.3</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4link</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4lock</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4lockT</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4lockU</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4lookup</td>
<td>171</td>
<td>0.2</td>
<td>0.0</td>
<td>37</td>
</tr>
<tr>
<td>v4lookupp</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4nVerify</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4open</td>
<td>48</td>
<td>0.1</td>
<td>8.2</td>
<td>37</td>
</tr>
<tr>
<td>v4openAttr</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4open_Conf</td>
<td>5</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4open DG</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4putFh</td>
<td>1305</td>
<td>1.9</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4putpubFh</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>v4putrootFh</td>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
</tbody>
</table>
The server Commands

```
v4read          1               0.0             0.0             0
v4readDir       21              0.0             0.6             0
v4readLink      0               0.0             0.0             0
v4remove        30              0.0             2.9             0
v4rename        2               0.0             0.0             0
v4renew         32335           46.9            0.0             2
v4restoreFh     0               0.0             0.0             0
v4saveFh        2               0.0             0.0             0
v4secInfo       0               0.0             0.0             0
v4setAttr       39              0.1             0.7             0
v4setClntid     2               0.0             0.0             0
v4clntid_Conf   2               0.0             0.0             0
v4verify        0               0.0             0.0             0
v4write         24              0.0             5.7             0
v4rel_Lockown   0               0.0             0.0             0
v4backChanCtl   0               0.0             0.0             0
v4bindConn      0               0.0             0.0             0
v4exchangeId    0               0.0             0.0             0
v4createSess    0               0.0             0.0             0
v4destroySess   0               0.0             0.0             0
v4setStateid    0               0.0             0.0             0
v4getDirDeleg   0               0.0             0.0             0
v4getDevInfo    0               0.0             0.0             0
v4getDevList    0               0.0             0.0             0
v4layoutCmmt    0               0.0             0.0             0
v4layoutGet     0               0.0             0.0             0
v4layoutRet     0               0.0             0.0             0
v4secinfoNoName 0               0.0             0.0             0
v4sequence      0               0.0             0.0             0
v4setSsv        0               0.0             0.0             0
v4testStateid   0               0.0             0.0             0
v4wantDeleg     0               0.0             0.0             0
v4destroyCld    0               0.0             0.0             0
v4reclaimCmpl   0               0.0             0.0             0
v4illegal       0               0.0             0.0             0
```

Server lookupcache:
```
nHit            nFind           nNegadd         nChecked
39459           46408           21              39459
```

Server rpc:
```
nCalls          nBadRpcData     nDuplicates     nResends        nBadAuths
822126          0               0               0               0
```

Where:
```
<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>nCalls</td>
<td>Number of calls for each NFS operation.</td>
</tr>
<tr>
<td>%totCalls</td>
<td>Percentage of calls per operation out of total NFS calls received.</td>
</tr>
<tr>
<td>ms/call</td>
<td>Average time taken for the NFS operations.</td>
</tr>
</tbody>
</table>
```
EXAMPLE #25  To display RPC statistics, type:

```
$ server_nfs server_2 -stats -rpc
server_2 :  
Server rpc:
ncalls          nBadRpcData     nDuplicates     nResends        nBadAuths
822155          0               0               0               0
```

EXAMPLE #26  To reset statistics counters, type:

```
$ server_nfs {<movername>|ALL} -stats -zero
server_2 : done
server_3 : done
server_4 : done
server_5 : done
```
server_nis

Manages the Network Information Service (NIS) configuration for the specified Data Movers.

SYNOPSIS

server_nis {<movername>|ALL}
   [ -delete | <domainname> {<ip_addr>,...}] | {<hostname>|<ip_addr>} [,{<hostname>|<ip_addr>}...]
   | -status

DESCRIPTION

server_nis provides connectivity to the NIS lookup servers.

The ALL option executes the command for all Data Movers.

OPTIONS

No arguments

Displays the NIS configuration.

-delete

Deletes all the NIS lookup servers.

Creates a list of up to 10 IP addresses to be used as NIS lookup servers for the specified <domainname>. Up to 10 NIS servers in a single NIS domain are supported for each Data Mover. A Data Mover can support only one NIS domain.

Each time an NIS domain is configured, the previous configuration is overwritten.

To provide high availability, you must configure your environment with multiple NIS servers. If you configure only a single NIS server, a problem with this NIS server could result in data unavailability.

Queries the NIS lookup servers using either a <hostname> or IP address. Multiple hostnames or IP addresses can be queried. Prior to performing a query, the NIS server must be activated and operational.

-status

Displays the status of the NIS lookup server even when the NIS server is not operational.

SEE ALSO

Configuring VNX Naming Services and server_dns.
EXAMPLE #1  To provide connectivity to the NIS lookup server for the specified domain, type:

```
$ server_nis server_2 nasdocs 172.24.102.30
```

server_2 : done

EXAMPLE #2  To query NIS lookup servers using both a hostname and IP address, type:

```
$ server_nis server_2 test40,172.24.102.36,test44
```

server_2 :
test40 = 172.24.102.30
test46 = 172.24.102.36
test44 = 172.24.102.34

EXAMPLE #3  To display the NIS configuration, type:

```
$ server_nis server_2
```

server_2 : yp domain=nasdocs server=172.24.102.30

EXAMPLE #4  To display the status of the NIS lookup servers, type:

```
$ server_nis server_2 -status
```

server_2 :
NIS default domain: nasdocs
NIS server 172.24.102.30

If NIS was not started, the output of this command will appear as:

```
$ server_nis server_2 -status
```

server_2 :
NIS not started

EXAMPLE #5  To delete all of the NIS lookup servers for a Data Mover, type:

```
$ server_nis server_2 -delete
```

server_2 : done
**server_param**

Manages parameter information for the specified Data Movers.

**SYNOPSIS**

`server_param {<movername>|ALL} -info`  
| -facility {<facility_name>|-all} {-list|-info {<param_name> [-verbose]|-all}  
| -facility <facility_name> -modify <param_name> -value <new_value>

**DESCRIPTION**

`server_param` modifies and displays information on parameters for the VNX. The **ALL** option executes the command for all Data Movers.

**OPTIONS**

- **-info**
  Displays the facilities that can be modified.

- **-facility {<facility_name>|-all} -list**
  Lists all of the parameters with their default, current, and configured values for the specified facility or facilities.

- **-facility {<facility_name>|-all} -info {<param_name> [-verbose]|-all}**
  Displays information on the specified parameter or parameters for the specified facility or facilities. The `-verbose` option displays a more detailed description, if one is available, for the specified `<param_name>`.

- **-facility <facility_name> -modify <param_name> -value <new_value>**
  Modifies the value of the specified `<param_name>` and changes the value to `<new_value>`. The `<facility_name>` is the facility for `<param_name>`.

**SEE ALSO**

*Parameters Guide for VNX for File.*

**EXAMPLE #1**

To view all facilities that can be modified on `server_2`, type:

```
$ server_param server_2 -info
```

<table>
<thead>
<tr>
<th>facility</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLRCompliance</td>
<td>File LevelRetention</td>
</tr>
<tr>
<td>NDMP</td>
<td>Network Data Management Protocol</td>
</tr>
<tr>
<td>PAX</td>
<td>Data Backup/Restore facility for NDMP/server_archive</td>
</tr>
<tr>
<td>RCP</td>
<td>Replication Control Protocol</td>
</tr>
</tbody>
</table>
The server Commands

VRPL          Volume replication control
ana           Adaptec Network Adapter
arp           Address Resolution Protocol
cfs           Common File System
cifs          Common internet FS (Microsoft Network File Protocol)
config        Misc. configuration parameters for CIFS
cvfs          Celerra Virtual File System
dedupe        File-level Data Deduplication
dns           Domain Name Service
ds            Domain Controller service for CIFS
fctach        Agilent Fibre Channel Controller
file          Overall file system parameters
filesystem     File system
ftpd          File Transfer Protocol Daemon
http          Hypertext Transfer Protocol
ip            Internet protocol
iscsi         Internet Scsi Protocol
ldap          Lightweight Directory Access Protocol
lockd         Network Lock Manager
mount         NFS Mount Protocol
nbs           Network Block Service Protocol
nfs           Network File System
nfsv4         NFS version 4 protocol
quota         File system quota management
replication    Data replication facility
shadow        Cross Protocol naming support
ssl           SSL security network protocol
statd         Host status demon
streamio      Streaming tape I/O support
svtl          Software-Based Tape Library
tcp           Transmission Control Protocol
tftp          Trivial File Transfer Protocol Daemon
trunk         Network trunking support
vbb           Volume Based Backup
vdevice       Virtual IP Device Parameters
viruschk      Virus checking service

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>facility</td>
<td>Facility for the parameter.</td>
</tr>
<tr>
<td>description</td>
<td>Description of the facility.</td>
</tr>
</tbody>
</table>

EXAMPLE #2  To view the PAX parameters that can be modified, type:

```bash
$ server_param server_2 -facility PAX -list
```
The server Commands

writeToArch                          PAX          1          1
paxReadBuff                          PAX         64         64
writeToTape                          PAX          1          1
filter.numDirFilter                  PAX          5          5
paxWriteBuff                         PAX         64         64
filter.numFileFilter                 PAX          5          5
filter.dialect                       PAX         ''         ''
nFTSThreads                          PAX          8          8
paxStatBuff                          PAX        128        128
readWriteBlockSizeInKB               PAX         64         64
nRestore                             PAX          8          8
filter.caseSensitive                 PAX          1          1
scanOnRestore                        PAX          1          1
noFileStreams                        PAX          0          0
allowVLCREstoreToUFS                 PAX          0          0

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>param_name</td>
<td>Name of the parameters with the specified facility that can be modified.</td>
</tr>
<tr>
<td>facility</td>
<td>Facility for the parameters.</td>
</tr>
<tr>
<td>default</td>
<td>Default value for the parameter.</td>
</tr>
<tr>
<td>current</td>
<td>Current value used by the Data Mover.</td>
</tr>
<tr>
<td>configured</td>
<td>Value set by the user. If some user action is pending (such as a Data Mover reboot), it might not have taken effect.</td>
</tr>
</tbody>
</table>

If the values for current and configured differ, refer to the user_action field of the -info option.

EXAMPLE #3 To view information on the nThread parameter, type:

$ server_param server_2 -facility PAX -info nThread

server_2 :
name                      = nThread
facility_name             = PAX
default_value             = 64
current_value             = 64
configured_value          =
user_action               = none
change_effective          = immediate
range                     = (1,128)
description               = Number of worker threads per backup session

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>facility_name</td>
<td>Facility for the parameter.</td>
</tr>
<tr>
<td>default_value</td>
<td>Default value set for the parameter.</td>
</tr>
<tr>
<td>current_value</td>
<td>Value set on the Data Mover.</td>
</tr>
</tbody>
</table>
The server Commands

### EXAMPLE #4
To modify the configured nThread parameter, type:

```
$ server_param server_2 -facility PAX -modify nThread -value 32
```

server_2 : done

### EXAMPLE #5
To modify the configured cipher parameter, type:

```
$ server_param server_2 -facility PAX -modify cipher -value foobar
```

server_2 : done

Warning 17716815750: server_2 : You must reboot server_2 for paxReadBuff changes to take effect.

To verify the configured cipher parameter, restart the Data Mover and type:

```
$ server_param server_2 -facility ssl -info cipher
```

<table>
<thead>
<tr>
<th>name</th>
<th>cipher</th>
</tr>
</thead>
<tbody>
<tr>
<td>facility_name</td>
<td>ssl</td>
</tr>
<tr>
<td>default_value</td>
<td>ALL:!ADH:!SSLv2:@STRENGTH</td>
</tr>
<tr>
<td>current_value</td>
<td>ALL:!ADH:!SSLv2:@STRENGTH</td>
</tr>
<tr>
<td>configured_value</td>
<td>foobar</td>
</tr>
<tr>
<td>user_action</td>
<td>reboot DataMover</td>
</tr>
<tr>
<td>change_effective</td>
<td>reboot DataMover</td>
</tr>
<tr>
<td>range</td>
<td>*</td>
</tr>
<tr>
<td>description</td>
<td>Keyword specifying the default supported SSL cipher suites (e.g: ALL:!LOW:@STRENGTH)</td>
</tr>
</tbody>
</table>

Note: If the current_value and configured_value parameters differ and if the user_action and change_effective parameters display the text reboot DataMover, restart the Data Mover. After restarting the Data Mover, if the current_value and configured_value parameters continue to differ, it indicates that the Data Mover encountered an error after it was restarted. Check the server_log output to view the error reported.

To view the server_log command output file, type:

```
$ server_log server_2 | grep param
```

... 2009-08-25 12:20:59: ADMIN: 3: Command failed: param ssl cipher=foobar ...
EXAMPLE #6  To view the values of the NDMP port ranges on the Data Mover server_2, type:

```
$ server_param server_2 -facility NDMP -info portRange
```

```
server_2 :
name                    = portRange
facility_name           = NDMP
default_value           = 1024-65535
current_value           = 1024-65535
configured_value        =
user_action             = none
change_effective        = immediate
range                   = 1024-65535
description             = Port range for NDMP data connection listening
```

EXAMPLE #7  To set the values of the NDMP port ranges on the Data Mover server_2, type:

```
$ server_param server_2 -facility NDMP -modify portRange
   -value 50000-50100
```

```
server_2 : done
```

EXAMPLE #8  To display the parameters for the SSL facility, type:

```
$ server_param server_2 -facility ssl -info -all
```

```
server_2 :
name             = trace
facility_name    = ssl
default_value    = 0x00000000
current_value    = 0x00000000
configured_value =
user_action      = none
change_effective = immediate
range            = (0x00000000,0xffffffff)
description      = Define SSL traces displayed in the server log

name             = timeout
facility_name    = ssl
default_value    = 5
current_value    = 5
configured_value =
user_action      = reboot DataMover
change_effective = reboot DataMover
range            = (1,120)
description      = Timeout (in seconds) used to receive SSL packets from network during SSL handshake

name             = protocol
facility_name    = ssl
```
The server Commands

default_value = 0
current_value = 0
configured_value =
user_action = reboot DataMover
change_effective = reboot DataMover
range = (0,2)
description = Set the default ssl protocol. Possible values are: 0=all ssl/tls protocol are allowed, 1=only sslev3 is allowed, 2=only sslev1 is allowed

name = threads
facility_name = ssl
default_value = 10
current_value = 10
configured_value =
user_action = reboot DataMover
change_effective = reboot DataMover
range = (4,30)
description = Number of SSL threads

name = cipher
facility_name = ssl
default_value = 'ALL:!ADH:!SSLv2:@STRENGTH'
current_value = 'ALL:!ADH:!SSLv2:@STRENGTH'
configured_value =
user_action = none
change_effective = reboot DataMover
range = '*'
description = Keyword specifying the default supported SSL cipher suites (e.g: ALL:!LOW:@STRENGTH)

EXAMPLE #9 To display the default SSL parameters on server_2, type:

$ server_param server_2 -facility ssl -list

server_2 :
param_name            facility    default    current   configured
trace                                ssl 0x00000000 0x00000000
timeout                             ssl          5          5
protocol                            ssl          0          0
threads                             ssl         10         10
cipher                             ssl 'ALL:!ADH:!SSLv2:@STRENGTH'

EXAMPLE #10 To modify the SSL dedicated threads to 20, type:

$ server_param server_2 -facility ssl -modify threads -value 20

server_2 : done

Warning 17716815750: server_2 : You must reboot server_2 for threads changes to take effect.
EXAMPLE #11  To modify the default cipher suite to all (except low-security algorithms and MD5), type:

```
$ server_param server_2 -facility ssl -modify cipher
   -value 'ALL:!LOW:!MD5:@STRENGTH'
```

server_2 : done

Warning 17716815750: server_2 : You must reboot server_2 for cipher changes to take effect.

EXAMPLE #12  To display the default ftpd parameters, type:

```
# server_param server_2 -facility ftpd -list
```

server_2 :
param_name facility default current configured
shortpathdir ftpd 0 0
defaultdir ftpd / /
wildcharsInDir ftpd 0 0
bounceAttackChk ftpd 1 1

EXAMPLE #13  To display the parameters for the ftpd facility, type:

```
$ server_param server_2 -facility ftpd -info -all
```

server_2 :
name = shortpathdir
facility_name = ftpd
default_value = 0
current_value = 0
configured_value =
user_action = none
change_effective = immediate
range = (0,1)
description = Enable return file name instead of full pathname in DIR command

name = defaultdir
facility_name = ftpd
default_value = /
current_value = /
configured_value =
user_action = none
change_effective = immediate
range = *
description = Sets the default working directory for FTP

name = wildcharsInDir
facility_name = ftpd
default_value = 0
current_value = 0
configured_value =
user_action = none
change_effective = immediate
range            = (0,1)
description      = Enable wild characters for directory names
name             = bounceAttackChk
facility_name    = ftpd
default_value    = 1
current_value    = 1
configured_value = user_action      = none
change_effective = immediate
range            = (0,1)
description      = Enable bounce attack check

detailed_description
Enable (1) or disable (0) return file name instead of full pathname in the commands dir or ls. if wild char are used this parameter is inefficient.

EXAMPLE #14 To display the detailed description of the shortpathdir parameter for the ftpd facility, type:

```bash
$ server_param server_2 -facility ftpd -info shortpathdir -verbose
```

server_2 :
name             = shortpathdir
facility_name    = ftpd
default_value    = 0
current_value    = 0
configured_value = user_action      = none
change_effective = immediate
range            = (0,1)
description      = Enable return file name instead of full pathname in DIR command
detailed_description
Enable (1) or disable (0) return file name instead of full pathname in the commands dir or ls. if wild char are used this parameter is inefficient.
The server Commands

server_pax

Displays and resets backup and restore statistics and file system information for a backup session already in progress.

SYNOPSIS

server_pax {<movername>|ALL}
  -stats [-reset | -verbose]

DESCRIPTION

server_pax displays and resets backup and restore statistics and file system information for a backup session already in progress.

OPTIONS

-stats

Displays backup and restore statistics for a backup or restore session in progress.

  [-reset | -verbose]

The -reset option resets the statistics for a backup or restore session already in progress. The -verbose option displays backup performance statistics and file size distribution data for a backup or restore session in progress.

SEE ALSO


EXAMPLE #1

To reset the statistics for a backup session already in progress, type:

$ server_pax server_2 -stats -reset

server_2 : done

EXAMPLE #2

To view the verbose statistics for an active NDMP backup session on server_2, type:

$ server_pax server_2 -stat -verbose

******************************** SUMMARY PAX STATS ************************

---- NASS STATS ----
nass00 is not doing backup
nass01 is not doing backup
nass02 is not doing backup
nass03 is not doing backup

---- NASA STATS ----
** nasa thid 0 (RESTORE) **
Backup root directory: /16m_ok_1_0
Total bytes processed: 12140605440
Total file processed: 738
throughput: 68 MB/sec
average file size: 16065KB
Total nasa wait nass count: 0
Total nasa wait nass time: 0 msec
Total time since last reset: 170 sec
Tape device name: /dev/c80t0l0

- 0 size file processed: 13
- 1 -- 8KB size file processed: 1
- 8KB+1 -- 16KB size file processed: 0
- 16KB+1 -- 32KB size file processed: 0
- 32KB+1 -- 64KB size file processed: 0
- 64KB+1 -- 1MB size file processed: 0
- 1MB+1 -- 32MB size file processed: 724
- 32MB+1 -- 1GB size file processed: 0
- 1G more size file processed: 0

fs /16m_ok_1_0 size is: 120855445504 Bytes
Estimated time remain is 1524 sec

nasa01 is not doing backup/restore
nasa02 is not doing backup/restore
nasa03 is not doing backup/restore

---- NASW STATS ----
nasw00 RESTORE (in progress)
Session Total Time: 00:02:50 (h:min:sec)
Session Idle Time: 00:00:56 (h:min:sec)
KB Tranferred: 11858820 Block Size: 61440 (60 KB)
Average Transfer Rate: 68 MB/Sec 239 GB/Hour
Average Burst Transfer: 101 MB/Sec 357 GB/Hour
__Point-in-Time__ (over the last 10 seconds):
Rate=69 MB/Sec Burst=96 MB/Sec Idle=283 msec/sec
Get Pool: 17 buffers Put Pool: 29 buffers
Compression Page not available
ReadC=0.00 WriteC=0.00 Read=0 KB Written=0 KB

nasw01 BACKUP (terminated)
nasw02 BACKUP (terminated)
nasw03 BACKUP (terminated)

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NASS STATS</strong></td>
<td>Thread responsible for traversing the file system and providing metadata for each directory or file or both.</td>
</tr>
<tr>
<td>Total file processed</td>
<td>Total number of files or directories or both for which metadata was processed.</td>
</tr>
<tr>
<td>Total NASS wait NASA count</td>
<td>The number of times NASS waited for NASA.</td>
</tr>
<tr>
<td>Total NASS wait NASA time</td>
<td>Amount of time NASS waited for NASA.</td>
</tr>
<tr>
<td>Total time since last reset</td>
<td>Time since the last reset; a reset occurs automatically when a backup completes.</td>
</tr>
<tr>
<td>fts_build time</td>
<td>Time spent building the file system or directory tree.</td>
</tr>
<tr>
<td>getstatpool</td>
<td>If the value is consistently 0, then NASA may be slowing down the backup.</td>
</tr>
<tr>
<td>putstatpool</td>
<td>If the value is consistently 0, then NASS may be slowing down the backup.</td>
</tr>
<tr>
<td><strong>NASA STATS</strong></td>
<td>Thread responsible for writing file header information, reading file data, and writing to the buffer.</td>
</tr>
<tr>
<td>Backup root directory</td>
<td>Directory being backed up.</td>
</tr>
<tr>
<td>Value</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Total bytes processed</td>
<td>Bytes backed up since the last reset or start of the current backup.</td>
</tr>
<tr>
<td>Total file processed</td>
<td>Number of files backed up since the start or reset of the current backup.</td>
</tr>
<tr>
<td>Throughput</td>
<td>How fast NASA processed data.</td>
</tr>
<tr>
<td>Average file size</td>
<td>Average file size for the current backup.</td>
</tr>
<tr>
<td>Total nasa wait nass count time</td>
<td>Number of times NASA waited for NASS.</td>
</tr>
<tr>
<td>Total nasa wait nass time</td>
<td>Amount of time NASA waited for NASS.</td>
</tr>
<tr>
<td>Total time since last reset</td>
<td>Amount of time since the backup statistics were reset; a reset occurs automatically when a backup completes.</td>
</tr>
<tr>
<td>Tape device name</td>
<td>Target device for the backup data.</td>
</tr>
<tr>
<td>File size statistics</td>
<td>Statistics on the size of files backed up since the start or reset of the current backup.</td>
</tr>
<tr>
<td><strong>NASW STATS</strong></td>
<td>Thread responsible for getting data from the buffer pool, writing it to tape or sending it to a remote Data Mover.</td>
</tr>
<tr>
<td>Session total time</td>
<td>Total time of the current session.</td>
</tr>
<tr>
<td>Session idle time</td>
<td>Idle time for the current session.</td>
</tr>
<tr>
<td>KB transferred</td>
<td>Total KB transferred.</td>
</tr>
<tr>
<td>Average transfer rate</td>
<td>Average transfer rate for the current session's data per second and per hour.</td>
</tr>
<tr>
<td>Average burst transfer</td>
<td>Burst transfer rate in MB/s and GB/s.</td>
</tr>
<tr>
<td>Write block counters (List/Direct)</td>
<td>Scatter/gather write count.</td>
</tr>
<tr>
<td><em>Point-in-time</em> (over the last 10 seconds)</td>
<td>Information on data processed during a 10 second interval.</td>
</tr>
<tr>
<td>Rate</td>
<td>Transfer rate in MB/s.</td>
</tr>
<tr>
<td>Burst</td>
<td>Burst transfer rate in MB/s.</td>
</tr>
<tr>
<td>Idle</td>
<td>Amount of time NASW was idle in msec.</td>
</tr>
<tr>
<td>Get pool</td>
<td>Number of buffers in get pool; if value is consistently 0, then NASA and NASS may be slowing down the backup.</td>
</tr>
<tr>
<td>Put pool</td>
<td>Number of buffers in put pool; if value is consistently 0, then the tape may be slowing down the backup.</td>
</tr>
<tr>
<td>Compression rate retrieved</td>
<td>Compression rate.</td>
</tr>
<tr>
<td>ReadC</td>
<td>Read compression rate at the tape device.</td>
</tr>
<tr>
<td>WriteC</td>
<td>Write compression rate at the tape device.</td>
</tr>
<tr>
<td>Read</td>
<td>Amount of data read in KB.</td>
</tr>
<tr>
<td>Written</td>
<td>Amount of data written in KB.</td>
</tr>
</tbody>
</table>
EXAMPLE #3  To view the verbose statistics for an active NDMP restore session on server_2, type:

```
$ server_pax server_2 -stats -verbose
```

```
server_2 :
*************** SUMMARY PAX STATS ****************

---- NASS STATS ----
nass00 is not doing backup
nass01 is not doing backup
nass02 is not doing backup
nass03 is not doing backup

---- NASA STATS ----
** nasa thid 0 (non-DAR RESTORE) **
The first five entries of restore name list are:
original name: /filt, destination name /ufsdbbr/r_filter_pax
Total bytes processed: 172326912
Total file processed: 42
throughput: 7 MB/sec
average file size: 4006KB
Total nasa wait nass count: 0
Total nasa wait nass time: 0 msec
Total time since last reset: 21 sec
Tape device name: c0t0l1
dir or 0 size file processed: 17
  1 -- 8KB size file processed: 6
  8KB+1 -- 16KB size file processed: 18
16KB+1 -- 32KB size file processed: 0
32KB+1 -- 64KB size file processed: 0
64KB+1 -- 1MB size file processed: 1
1MB+1 -- 32MB size file processed: 0
32MB+1 -- 1GB size file processed: 0
  1G more size file processed: 0

nasa01 is not doing backup/restore
nasa02 is not doing backup/restore
nasa03 is not doing backup/restore

---- NASW STATS ----
nasw00 RESTORE (in progress)
Session Total Time: 00:00:21 (h:min:sec)
Session Idle Time: 00:00:00 (h:min:sec)
KB Tranferred: 168384  Block Size: 32768 (32 KB)
Average Transfer Rate: 7 MB/Sec  27 GB/Hour
Average Burst Transfer: 7 MB/Sec  27 GB/Hour
__Point-in-Time__ (over the last 10 seconds):
Rate=6 MB/Sec  Burst=7 MB/Sec  Idle=0 msec/sec
Get Pool: 61 buffers  Put Pool: 0 buffers

nasw01 No session found
nasw02 No session found
nasw03 No session found
```
**server_ping**

Checks the network connectivity for the specified Data Movers.

**SYNOPSIS**

```
server_ping {<movername>|ALL} [-send | -interface <interface>] {<hostname>|<ip_addr}]
```

**DESCRIPTION**

`server_ping` sends ICMP ECHO_REQUEST messages to a remote host and receives ICMP ECHO_REPLY messages back that verify and determine network connectivity.

As the transmission time is stamped into the ECHO_REQUEST message, `server_ping` displays the transmission delay between the Data Mover and the remote host.

The **ALL** option executes the command for all Data Movers.

**OPTIONS**

`<hostname>|<ip_addr>`

Provides a one-shot test to determine if the specified `<hostname>` or `<ip_addr>` is accessible. The IP address may be used if NIS or DNS or both are not operational.

`-send {<hostname>|<ip_addr>}`

Repeats sending/receiving ECHO_REQUEST messages to a specified `<hostname>` or `<ip_addr>` for a remote host. The IP address may be used if NIS or DNS or both are not operational.

Press Ctrl+C to quit.

When invoking `server_ping` with a hostname, the `/etc/hosts` file is checked first, then the NIS and the DNS servers (if operational on the Data Mover) are checked. If the name does not exist in either case, an error message is displayed.

`-interface {<interface> <hostname>|<ip_addr}>`

Uses the address of the specified interface name as the source address. If a route to the destination exists with a matching source interface, the requests will also be transmitted out the specified interface. Otherwise, the routing table will be searched for the most specific route to the destination and the interface associated with this route will be used to transmit the request.

**SEE ALSO**

`Configuring and Managing Networking on VNX`, `server_dns`, `server_ifconfig`, and `server_nis`. 
**EXAMPLE #1**  To display connectivity from a Data Mover to the outside world, type:

```
$ server_ping server_2 172.24.102.2
server_2 : 172.24.102.2 is alive, time= 0 ms
```

or

```
$ server_ping server_2 172.24.102.5
server_2 :
Error 6: server_2 : No such device or address
no answer from 172.24.102.5
```

**EXAMPLE #2**  To display connectivity for a Data Mover to the outside world while sending continuous ECHO_REQUEST messages, type:

```
$ server_ping server_2 -send 172.24.102.2
server_2 :
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 3 ms
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 0 ms
172.24.102.2 is alive, time= 0 ms
```

**EXAMPLE #3**  To display connectivity from a Data Mover to the outside world using the specified interface, type:

```
$ server_ping server_2 -interface cge0 172.24.102.2
server_2 : 172.24.102.2 is alive, time= 0 ms
```
server_ping6

Checks the IPv6 network connectivity for the specified Data Movers.

SYNOPSIS

server_ping6 {<movername>|ALL} [-send | -interface <interface>] {<hostname>|<ip6_addr>}

DESCRIPTION

server_ping6 sends ICMPv6_ECHO_REQUEST messages to a remote host and receives ICMPv6_ECHO_REPLIES messages back that verify and determine network connectivity.

As the transmission time is stamped into the ECHO_REQUEST message, server_ping6 displays the transmission delay between the Data Mover and the remote host.

The ALL option executes the command for all Data Movers.

OPTIONS

<hostname>|<ip6_addr>
Provides a one-shot test to determine if the specified <hostname> or <ip6_addr> is accessible. The IP address may be used if NIS or DNS or both are not operational.

-send {<hostname>|<ip6_addr>}
Repeats sending/receiving ECHO_REQUEST messages to a specified <hostname> or <ip6_addr> for a remote host. The IP address may be used if NIS or DNS or both are not operational. Press Ctrl C to quit.

When invoking server_ping6 with a hostname, the /etc/hosts file is checked first, then the NIS and the DNS servers (if operational on the Data Mover) are checked. If the name does not exist in either case, an error message is displayed.

Note: server_ping6 requests sent to a link-local or multicast destination address require the specification of a '%<intf>' scope identifier.

-interface {<interface> <hostname>|<ip6_addr>}
Provides a one-shot test to determine if the specified <hostname> or <ip6_addr> is accessible through the specified <interface>. The IP address may be used if NIS or DNS or both are not operational.
EXAMPLE #1  To verify Data Mover connectivity from cge0_int1 on server_2 to a client at 3ffe:0000:3c4d:0015:0435:0200:0300:00aa, type:

$$\textit{server\_ping6} \textit{server\_2} -\textit{interface} \textit{cge0\_int1}$$

$$3ffe:0000:3c4d:0015:0435:0200:0300:00aa$$

server_2 : 3ffe:0000:3c4d:0015:0435:0200:0300:00aa is alive, time= 0 ms

or

$$\textit{server\_ping6} \textit{server\_2} -\textit{interface} \textit{cge0\_int1}$$

$$3ffe:0000:3c4d:0015:0435:0200:0300:00aa$$

server_2 : 
Error 6: server_2 : No such device or address
no answer from client

EXAMPLE #2  To ping link-local address fe80::260:16ff:fe0c:205%cge0_0000_ll, type:

$$\textit{server\_ping6} \textit{server\_2}$$

$$fe80::260:16ff:fe0c:205%cge0_0000\_ll$$

server_2 : fe80::260:16ff:fe0c:205%cge0_0000_ll is alive, time= 0 ms

or

$$\textit{server\_ping6} \textit{server\_2}$$

$$fe80::260:16ff:fe0c:205%cge0_0000\_ll$$

server_2 : 
Error 6: server_2 : No such device or address
no answer from client

EXAMPLE #3  To ping multicast address ff02::1%cge0_0000_ll, type:

$$\textit{server\_ping6} \textit{server\_2} \textit{ff02::1%cge0_0000\_ll}$$

server_2 : ff02::1%cge0_0000_ll is alive, time= 0 ms

or

$$\textit{server\_ping6} \textit{server\_2} \textit{ff02::1%cge0_0000\_ll}$$

server_2 : 
Error 6: server_2 : No such device or address
no answer from client
server_rip

Manages the Routing Information Protocol (RIP) configuration for the specified Data Movers.

SYNOPSIS

server_rip {<movername>|ALL}
  status
  | {{ripin|noripin}{<interface_name>[,...]}}

DESCRIPTION

server_rip disables and enables RIP processing for the specified interfaces on a specified Data Mover. server_rip displays the status of the RIP configuration for the specified Data Mover. By default, all the interfaces process RIP messages. server_setup provides information to start and stop RIP for a Data Mover.

The ALL option executes the command for all Data Movers.

OPTIONS

status
Displays the status of the routed daemon, RIP processing, and the interfaces with RIP processing disabled.

ripin|noripin {<interface_name>[,...]}
ripin enables RIP processing and noripin disables RIP processing for the specified interfaces.

SEE ALSO

server_setup.

EXAMPLE #1
To disable an interface for RIP processing, type:

$ server_rip server_2 noripin cge0

EXAMPLE #2
To display the status of RIP processing, type:

$ server_rip server_2 status

EXAMPLE #3
To enable RIP processing for a specified interface, type:

$ server_rip server_2 ripin cge0
**server_route**

Manages the routing table for the specified Data Movers.

**SYNOPSIS**

```
server_route {<movername>|ALL} 
   -flush
   -DeleteAll
   -list
   {-add|delete} default <gateway>
   {-add|delete} [host|net] <dest> <gateway> [netmask]
```

**DESCRIPTION**

`server_route` creates, displays, and deletes routing entries for the specified Data Movers.

The `ALL` option executes the command for all Data Movers.

**OPTIONS**

- **-flush**
  Temporarily removes all manually configured routing entries for as long as the Data Mover remains up. System generated entries are not removed. A reboot of the Data Mover re-establishes the routing entries that were temporarily removed.

- **-DeleteAll**
  Permanently removes all the routing entries.

- **-list**
  Lists all routing entries including the subnet mask for each route.

To display the full routing table for a Data Mover, type:

```
$ server_netstat <movername> -r
```

- **{-add|delete} default <gateway>**
  Adds or deletes a default gateway for all unspecified destinations. The Data Mover passes all packets not qualifying against a specific route entry to the specified gateway. The keyword `default` implies a network destination address of 0.0.0.0. A default route uses a fixed mask of 0.0.0.0.

- **{-add|delete} net <dest> <gateway> [netmask]**
  Adds or deletes a routing entry to a network host as specified by the optional keyword `net` (default) to the specified destination and gateway address. The `<dest>` argument represents the network for the routing entry, while the `<gateway>` argument indicates the network gateway to which packets should be addressed.
The <netmask> is an optional field for network addresses. In case a netmask is not specified, a netmask will be assigned based on the class of the address and addresses assigned on the system. A <netmask> of 0.0.0.0 or 255.255.255.255 is invalid in the case of net routes. The default for the route is net.

\{-add|\-delete\} host <dest> <gateway>

Adds or deletes a routing entry to a particular host as specified by the optional keyword host (net is the default) to the specified destination and gateway address. The <dest> argument represents the specific host for the routing entry, while the <gateway> argument indicates the network gateway to which packets should be addressed.

By default, a netmask of 255.255.255.255 is assigned to host routes.

**SEE ALSO** Configuring and Managing Networking on VNX, server_netstat, and server_ifconfig.

**EXAMPLE #1** To list the routing table for server_2, type:

```
$ server_route server_2 -list
```

server_2 :
net 128.221.253.0 128.221.253.2 255.255.255.0 e131
net 128.221.252.0 128.221.252.2 255.255.255.0 e130
net 172.24.102.0 172.24.102.238 255.255.255.0 cge0
host 127.0.0.1 127.0.0.1 255.255.255.255 loop

Where: The output displayed is as follows:

<routing type> <desination> <gateway> <mask> <network interface>

**EXAMPLE #2** To create a host routing entry, type:

```
$ server_route server_2 -add host 10.10.1.2 172.24.102.77
```

server_2 : done

**EXAMPLE #3** To create a default gateway for all unspecified destinations, type:

```
$ server_route server_2 -add default 172.24.102.254
```

server_2 : done

**EXAMPLE #4** To add a network routing entry with a netmask, type:

```
$ server_route server_2 -add net 192.162.0.0 172.24.102.238 255.255.0.0
```

server_2 : done
EXAMPLE #5  To delete a routing entry, type:

```
$ server_route server_2 -delete host 10.10.1.2
  172.24.102.77
server_2 : done
```

EXAMPLE #6  To flush the routing table for a Data Mover, type:

```
$ server_route server_2 -flush
server_2 : done
```

EXAMPLE #7  To delete all routing entries, type:

```
$ server_route server_2 -DeleteAll
server_2 : done
```
server_security

Manages the security policy settings for the specified Data Movers. The VNX provides support for Group Policy Objects (GPOs) by retrieving and storing a copy of the GPO settings for each CIFS server joined to a Windows domain.

SYNOPSIS

server_security {<movername>|ALL} {{-update|-info} -policy gpo [server=<server_name>| domain=<domain_name>] | {-add|-delete|-modify} -policy chap -name <client_name> | -info -policy chap [-name <client_name>]

DESCRIPTION

server_security queries or updates the security policy settings on the Data Movers. The VNX stores the GPO settings in a GPO cache on the Data Mover. When you start the CIFS service on a Data Mover, the VNX reads the settings stored in the GPO cache, and then retrieves the most recent GPO settings from the Windows domain controller.

The ALL option executes the command for all Data Movers.

OPTIONS

-update -policy gpo
Forces an update of the specified GPO settings.

-info -policy gpo
Queries and displays the GPO settings.

[-server=<server_name>|domain=<domain_name>]
Limits operations to the specified server or domain. The server refers to the configured CIFS servers on the Data Movers. If no server or domain has been specified, the operations are performed to all configured servers.

-add -policy chap -name <client_name>
Adds a new Challenge Handshake Authentication Protocol (CHAP) security for the specified <client_name>.

-delete -policy chap -name <client_name>
Deletes a CHAP security for the specified <client_name>.

-modify -policy chap -name <client_name>
Modifies a CHAP security setting for the specified <client_name>.

-info -policy chap [-name <client_name>]
Displays current CHAP security settings for the Data Mover, or the specified <client_name>. 
SEE ALSO  Configuring iSCSI Targets on VNX for File, server_iscsi, and server_checkup.

EXAMPLE #1  To query the GPO settings for all configured CIFS servers, type:

```
$ server_security server_2 -info -policy gpo
```

server_2 :
Server compname: dm102-cge0
Server NetBIOS: DM102-CGE0
Domain: nasdocs.emc.com
Kerberos Max Clock Skew (minutes): 5
LAN Manager Auth Level: Not defined
Digitally sign client communications (always): Not defined
Digitally sign client communications (if server agrees): Not defined
Digitally sign server communications (always): Not defined
Digitally sign server communications (if client agrees): Not defined
Send unencrypted password to connect to third-party SMB servers: Not defined
Disable machine account password changes: Not defined
Maximum machine account password age: Not defined
Audit account logon events: Not defined
Audit account management: Not defined
Audit directory service access: Not defined
Audit logon events: Not defined
Audit object access: Not defined
Audit policy change: Not defined
Audit privilege use: Not defined
Audit process tracking: Not defined
Audit system events: Not defined
Back up files and directories: Not defined
Restore files and directories: Not defined
Bypass traverse checking: Not defined
Generate security audits: Not defined
Manage auditing and security log: Not defined
Access this computer from the network: Not defined
Deny access to this computer from the network: Not defined
Take ownership of files or other objects: Not defined
EMC Virus Checking: Not defined
Maximum security log size: Not defined
Restrict guest access to security log: Not defined
Retention period for security log: Not defined
Retention method for security log: Not defined
Maximum system log size: Not defined
Restrict guest access to system log: Not defined
Retention period for system log: Not defined
Retention method for system log: Not defined
Maximum application log size: Not defined
Restrict guest access to application log: Not defined
Retention period for application log: Not defined
Retention method for application log: Not defined
Disable background refresh of Group Policy: Not defined
Group Policy Refresh interval (minutes): 90
Refresh interval offset (minutes): Not defined
EXAMPLE #2 To add a new CHAP security for client1, type:

$ server_security server_2 -add -policy chap -name client1

server_2 : Enter Secret:**** done

EXAMPLE #3 To display CHAP information for client1, type:

$ server_security server_2 -info -policy chap -name client1

server_2 :
chapdb name=client1 pass=********

EXAMPLE #4 To update the GPO settings for the CIFS server, type:

$ server_security server_2 -update -policy gpo
server=dm32-cge0

server_2 : done

EXAMPLE #5 To modify a password for client1, type:

$ server_security server_2 -modify -policy chap -name client1

server_2 : Enter New Secret:**** done

EXAMPLE #6 To delete CHAP security for client1, type:

$ server_security server_2 -delete -policy chap -name client1

server_2 : done
server_setup

Manages the type and protocol component for the specified Data Movers.

SYNOPSIS

gserver_setup {<movername>|ALL}
   -type {nas|standby}
   -Load <image_name>
   -Protocol {cifs|mpfs|viruschk|rip|cdms}
      -option {start[=<n>]|stop[,comment=<message>]|delete}

DESCRIPTION

server_setup sets up the type and protocol component for the specified Data Mover.

The ALL option executes the command for all Data Movers.

OPTIONS

-type {nas|standby}
Reboots the Data Mover as the type specified. When setting up a standby relationship between a primary Data Mover and a standby, make sure that the standby Data Mover has the same disk configuration as the primary.

-Load <image_name>
Loads an image onto a Data Mover. The image may be an emergency patch or debug image provided by EMC. Do not use this option unless directed by EMC customer service.

-Protocol {cifs|mpfs|viruschk|rip|cdms}
Selects the protocol configuration to be managed. Multiple protocol configurations can be enabled for a Data Mover.

Note: NFS and RIP processing are enabled by default. NFS cannot be disabled.

-option
Specifies the following options for the specified protocol:

start
Starts the specified protocol configuration. To verify whether the protocol configuration has started, use the respective server command for the protocol.

[=<n>]
Indicates the number of threads for users of CIFS, virus checker, and CDMS. The recommended number of CIFS
threads for each user is based on the amount of memory on the Data Mover. If there is less than 1 GB memory, the default thread number is 96, but if the memory is greater than 1 GB, the default thread number is 256.

---

**Note:** The `=[<n>]` option is not available for RIP protocol configurations.

---

**stop**

Stops the protocol configuration.

```
[.comment=<message>]
```

Allows a message enclosed with quotes to be entered. The message is delivered directly to CIFS clients.

---

**Note:** The `[.comment=<message>]` option is not available for RIP or virus checker protocol configurations.

---

**delete**

Deletes the protocol configuration.

---

**SEE ALSO**  
`Configuring Standbys on VNX`, `server_cdms`, `server_checkup`, `server_mpfs`, `server_rip`, `server_ssh`, and `server_viruschk`.

---

**EXAMPLE #1**  
To set up a Data Mover as a standby, type:

```
$ server_setup server_3 -type standby
```

server_3: reboot in progress 0.0.0.0.0.0.0.0.1.3.3.3.3.3.4. done
done

**EXAMPLE #2**  
To set up a Data Mover as nas, type:

```
$ server_setup server_3 -type nas
```

server_3: reboot in progress 0.0.0.0.0.0.0.0.1.3.3.3.3.3.4. done
done

---

**EXAMPLE #3**  
To start the CIFS protocol for a Data Mover, type:

```
$ server_setup server_2 -Protocol cifs -option start
```

server_2: done

**EXAMPLE #4**  
To start the virus checker protocol for a Data Mover, type:

```
$ server_setup server_2 -Protocol viruschk -option start
```

server_2: done
EXAMPLE #5  To stop the CIFS protocol for a Data Mover, type:

$ server_setup server_2 -Protocol cifs -option stop,comment="CIFS is going down"

server_2 : done

EXAMPLE #6  To delete the CIFS protocol configuration for a Data Mover, type:

$ server_setup server_2 -Protocol cifs -option delete

server_2 : done

Warning 4037: server_2 : CIFS configurations for server_2 and VDM(s) currently loaded on it

EXAMPLE #7  To load a new image onto a Data Mover, type:

$ server_setup server_2 -Load nas.exe

server_2: will load : nas.exe
server_snmpd

Manages the Simple Network Management Protocol (SNMP) configuration values for the specified Data Movers.

SYNOPSIS
server_snmpd {<movername>|ALL}
   -info
   -service {
      -status
      -stop
      -start
   }
   -modify [-location <sys_location>]
            [-contact <sys_contact>]
            [-community {-clear|<community>}}]
   -user {
      -list
      -create <name> -authpw -privpw
      -delete <name>
      -modify <name> -authpw -privpw
   }

DESCRIPTION
server_snmpd controls SNMP server agent and SNMP configuration for the specified Data Mover. The service can be started/stopped depending on user preferences; the location and contact can be configured. SNMPv1 community strings can be created and cleared. SNMPv3 users can be created or deleted, and modified. The ALL option executes the command for all Data Movers.

OPTIONS
   -info
   Displays information about the SNMP server agent and configuration such as location, contact, and community.

   -service {-status | -stop | -start}
   Starts, stops, or displays the current SNMP service.

   -modify
   Modifies the configuration of the SNMP service.

      -location <sys_location>
      Modifies the location value.

      -contact <sys_contact>
      Modifies the syscontact value.
The server Commands

-community {-clear|<community>}
Modifies the community name being used by the server SNMPD agent and enables the SNMPv1/SNMPv2c access to the agent. The -clear option allows the administrator to disable the community.

-user {-list}
Lists the SNMPv3 users on the Data Mover.

-create <name> -authpw -privpw
Creates an SNMPv3 user on the Data Mover.

-delete <name>
Deletes an SNMPv3 user on the Data Mover.

-modify <name> -authpw -privpw
Modifies the SNMPv3 passwords of a user on the Data Mover. Both the passwords should be changed together.

EXAMPLE #1
To view the SNMP service properties of the Data Mover server_2, type:

```
$ server_snmpd server_2 -info
```

server_2:
enabled    = yes
location   = RTP, NC
contact    = John Doe
community  = public

EXAMPLE #2
To view the SNMP service properties of all Data Movers, type:

```
$ server_snmpd ALL -info
```

server_2:
enabled    = yes
location   = RTP, NC
contact    = John Doe
community  = public

server_3:
enabled    = no
location   = Bangalore, India
contact    = Pandu
community  = testcommunity

EXAMPLE #3
To start the SNMP service on the Data Mover server_2, type:

```
$ server_snmpd server_2 -service -start
```

OK

EXAMPLE #4
To stop the SNMP service on the Data Mover server_3, type:
The server Commands

```
server_3:
OK

EXAMPLE #5 To view the SNMP service status for all Data Movers, type:

server_2:
SNMP Running
server_3:
SNMP Stopped

EXAMPLE #6 To modify the location, contact, and community values on Data Mover server_2, type:

server_2:
OK

EXAMPLE #7 To clear the community value on Data Mover server_2, type:

server_2:
OK

EXAMPLE #8 To display the list of SNMPv3 users on all Data Movers, type:

server_2:
user
gsmith
smith

server_3:
user
clowe

EXAMPLE #9 To create a new user John, on Data Mover server_2, type:

server_2:
OK

EXAMPLE #10 To delete the user John, on Data Mover server_2, type:

```
EXAMPLE #11  To modify the passwords of the user John, on Data Mover server_2, type:

$ server_snmpd server_2 -user -modify John -authpw -privpw

Enter the authentication password:*********
Confirm the authentication password:*********
Enter the privacy password:*********
Confirm the privacy password:*********
server_2:
OK
server_ssh

Manages and configures the SSH server on the specified Data Mover.

SYNOPSIS

server_ssh serverX
   -info
   | -start
   | -stop
   | -modify {
      -banner <banner file> |
      -port <port number> |
      -threads <number> |
      -defaultdir <path> |
      -restricthomdir {enable|disable} |
      -application <application_name> |
      -cipher <keyword> |
      -compression {yes|delayed|no} |
      -mac <keyword> |
      -maxauthtries <number> |
      -passwordauthentication {enable|disable} |
      -permitemptypassword {enable|disable} |
      -pubkeyauthentication {enable|disable} |
      -timeout <seconds> }
   | -append {
      -allowhosts <ip>[,<ip>...] |
      -allowgroups <group>[,<group>...] |
      -allowusers <user>[,<user>...] |
      -denyhosts <ip>[,<ip>...] |
      -denygroups <group>[,<group>...] |
      -denyusers <user>[,<user>...] }}
   | -remove {
      -allowhosts <ip>[,<ip>...] |
      -allowgroups <group>[,<group>...] |
      -allowusers <user>[,<user>...] |
      -denyhosts <ip>[,<ip>...] |
      -denygroups <group>[,<group>...] |
      -denyusers <user>[,<user>...] }|
   | -generate [-type {rsa|dsa} [-keysize <bits>]]
   | -delete [-type {rsa|dsa}]

OPTIONS

   -info
   Displays the current configuration and state of the SSH server.

   -start
   Starts the SSH server. Allocates the pools of SSH threads and
   applications threads, and listens to incoming client requests from the
   network.
If the SSH server is already active, an error is returned.

**-stop**  
Stops the SSH server. Stops listening to incoming client requests, and releases all the SSH threads.

If the SSH server is inactive, an error is returned.

**-modify**  
Modifies some configuration parameters of the SSH server. The arguments are:

[-**banner** <banner file>]
The file name specified should contain an absolute path on the Data Mover. The file contains an unformatted text, which is sent to the SSH clients during the authentication step of the connection. It is a welcome message. The file name is saved in the SSH configuration file on the Data Mover. If the file name is "" or an empty string, then the current file name is removed from the configuration. If the file does not exist or is unreadable, an error message is returned.

[-**port** <number>]
Changes the TCP lists port of the SSH server. The server must be stopped before executing this command. The SSH server listens on only one TCP port. This port number is saved in the SSH configuration file on the Data Mover. If this port is invalid, or already used by another active service of the Data Mover, an error message is returned.

[-**threads** <number>]
Allocates the number of threads dedicated to the SSH server. If the server is started, the threads are restarted by this action, and all the established SSH connections are disconnected. Ten default threads are started when the server starts. The supported values are from 4 to 128 threads.

[-**defaultdir** <path>]
Specifies the default directory of an authenticated user when the home directory is undefined or inaccessible. If the given path does not exist or is inaccessible, then an error message is returned. The default value is / or forward slash.
The server Commands

[-restricthomedir {enable | disable}]
Restricts authenticated users to their home directory tree, to prevent them from accessing other user’s directories. If their home directory is undefined or unreachable, the login is denied. This feature is enabled, by default, for security reasons.

[-application <application_name>]
Enables or disables the specified SSH application. The names of applications are case insensitive. If an SSH client tries to execute a disabled application, the Data Mover returns an error. The applications supported by the SSH server in VNX version 7.0 are SFTP and SCP. The default applications are disabled. To disable an enabled application, use the same option with the name no<application>. For instance, use the -application nosftp option to disable SFTP.

[-cipher <keyword>]
Specifies the symmetric ciphers allowed to encrypt an SSH connection. The openSSH documentation or http://www.openssh.com provides more information on the supported ciphers algorithms. The default option is undefined, which means all these ciphers are allowed.

-compression {yes | delayed | no}
Specifies whether the compression of data inside an SSH connection is allowed, disallowed, or delayed.

   The argument must be:
   • yes: always compressed, if possible
   • delayed: compressed after the user has authenticated successfully, if possible
   • no: compression allowed

The default is delayed.

-mac <keyword>
Specifies the available Message Authentication Code or MAC algorithms to guarantee the integrity of the SSH packets on the network. The default value is undefined, which means all these algorithms are allowed.

-maxauthtries <number>
Specifies the maximum number of authentication attempts permitted per SSH connection. The supported values are between 1 and 99. The default value is 6.
-passwordauthentication {enable | disable}
Specifies whether password authentication is allowed during the establishment of an SSH connection. The parameters passwordAuthentication and pubkeyAuthentication cannot be disabled simultaneously. The default value is enabled.

-permitemptypassword {enable | disable}
When password authentication is allowed, it specifies whether the server allows login to accounts with empty password strings. The supported values are: disabled or not permitted, and enabled or permitted. The default value is disabled.

-pubkeyauthentication {enable | disable}
Specifies whether public key authentication is allowed during the establishment of an SSH connection. The options passwordAuthentication and pubkeyAuthentication cannot be disabled simultaneously.

The default value is enabled.

-timeout <seconds>
Specifies the time out in seconds before a network disconnection in case of inactivity. The default value is 900 seconds or 15 minutes. The range is from 10 to 3600 seconds.

-append
Appends to the current configuration new allowed or denied host/group/user. The arguments are:

-allowhosts <ip>[,<ip>...]  
Defines new client IPs or client subnets that are allowed to connect through SSH to the Data Mover. IP is the IPv4 or IPv6 address or subnet. In case of subnets, the CIDR notation must be used (for instance: 192.168.0.0./16 or 2001:DB8::/48). Network names are ignored.

If specified, SSH connections are allowed only for clients if their IP matches one of the defined allowed hosts, that is, any client host not declared on that list receives systematically an access denied error. If a subnet is defined on the allowed hosts list, then the client IP must belong to this subnet to be allowed to connect to the Data Mover.

If no allowed host is defined, then all client hosts are allowed to connect by default.
The list of the allowed host is saved in the SSH configuration file on the Data Mover. Duplicate items of the allowed host are prohibited. The Data Mover can save up to 256 different allowed hosts.

**-allowgroups <group>[,<group>...]**
Defines new groups of users that are allowed to connect through SSH to the Data Mover. The group should be the name of the allowed group and numerical group ID are ignored.

If specified, SSH connections are allowed only for users whose primary or supplementary group list matches one of the defined allowed groups, which means any user not a member of one of these groups receives an access denied error.

If no allowed group is defined, then all users of the groups are allowed to connect by default.

The list of allowed groups is saved in the SSH configuration file on the Data Mover.

Duplicate names of an allowed group are prohibited.

The Data Mover can save up to 256 different allowed groups.

**-allowusers <user>[,<user>...]**
Defines new users that are allowed to connect through an SSH server to the Data Mover. User ID should be the name of a user; numerical user IDs are ignored. If the user is a CIFS user, the format should be user@domain or domain\user.

If specified, SSH connections are allowed only for user names that match one of the patterns, which means any user not listed as an allowed user receives an access denied error.

If no allowed user is defined, then all users are allowed to connect by default.

The list of the allowed users is saved in the SSH configuration file on the Data Mover.

Duplicate names of allowed users are prohibited.

The Data Mover can save up to 256 different allowed users.

**-denyhosts <ip>[,<ip>...]**
Defines new client IPs or client subnets that are disallowed to connect through the SSH server to the Data Mover. IP is the IPv4 or IPv6 address or subnet (CIDR notation) of the client host. Network names are ignored.
If specified, SSH connections are disallowed for clients whose IP address matches one of the defined denied host or IP address belongs to a defined subnet, which means any client host declared in that list receives systematically an access denied error.

If no denied host is defined, then all client hosts are allowed to connect. This is the default.

The list of the denied host is saved in the SSH configuration file on the Data Mover.

Duplicate items of denied host are prohibited.

The Data Mover can save up to 256 different denied hosts.

**-denygroups** `<group>,<group>...`

Defines new groups of users that are disallowed to connect through SSH to the Data Mover. ‘group’ should be the name of the group and numerical group ID are ignored.

If specified, SSH connections are disallowed for users whose primary group or supplementary group list matches one of the defined “denied group”, i.e. any user member of one of these groups receives systematically an “Access denied” error.

If no “denied group” is defined, then all users of all groups are allowed to connect. This is the default.

The list of the denied groups is saved in the SSH configuration file on the Data Mover.

Duplicate names of “denied group” are prohibited.

The Data Mover can save up to 256 different denied groups.

**-denyusers** `<user>`

Defines a new user that is disallowed to connect through SSH to the Data Mover. “user” should be the name of the user and numerical user ID are ignored.

If the user is a CIFS user, his format should be user@domain or domain\user.

If specified, SSH connections are disallowed for user names that match one of the patterns, i.e. any user listed as a denied user receives systematically an access denied error.

If no denied user is defined, then all users are allowed to connect. This is the default.
The list of the denied users is saved in the SSH configuration file on the DART.

Duplicate names of denied users are prohibited.

The Data Mover can save up to 256 different denied users.

**-remove**

Removes from the current configuration an allowed/denied group/user. The arguments are:

- **-allowhosts** <ip>[,<ip>...]
  Removes the specified IPs or subnets from the allowed hosts list.

- **-allowgroups** <group>[,<group>...]
  Removes the specified groups from the allowed groups list.

- **-allowusers** <user>[,<user>...]
  Removes the specified users from the allowed users list.

- **-denyhosts** <ip>[,<ip>...]
  Removes the specified IPs or subnets from the denied hosts list.

- **-denygroups** <group>[,<group>...]
  Removes the specified groups from the denied groups list.

- **-denyusers** <user>[,<user>...]
  Removes the specified users from the denied users list.

**-generate**

Generates new public and private keys and saves it as the host key of the Data Mover. This operation must be the first step during the setup procedure of SSH. It could take some minutes to complete.

By default, two host keys are generated: a 2048-bit RSA key and a 1024-bit DSA key.

If the host keys of the Data Mover do not exist when the SSH server is started the first time, the keys are automatically generated with the default parameters.

The host keys are stored in the /.etc directory of the Data Mover, in an inaccessible directory.

The options are:

- **-type** {rsa | dsa} Specifies the type of key to generate. The two valid types are: rsa and dsa. If not specified, both keys types are generated.
The server Commands

-keysize <bits>: Specifies the number of bits of the host key. Default is 2048 bits for RSA keys and 1024 bits for DSA keys.

Valid range for RSA keys is between 2048 and 8192 bits.

The length of a DSA key must be 1024 bits exactly, as specified by FIPS 186-2.

-delete: Deletes all the generated host keys of the Data Mover. This command is useful if the administrator needs to generate new host keys.

The options are:

-type {rsa | dsa}: This argument specifies the type of key to delete. The two valid types are: rsa and dsa. If not specified, both key types are deleted.

GENERAL NOTES

◆ The allow or deny directives are processed in the following order: denyhost, allowhost, denyusers, allowusers, denygroups and finally allowgroups.

◆ The allow or deny directives can specify multiple items separated by a , or comma without spaces.

◆ It is recommended to encapsulate IPv6 address in [ ] or square brackets.

◆ The host keys are generated automatically when the SSH server is started for the very first time if no host key exists. In that case, both keys are generated with their default size (RSA and DSA keys each).

◆ VDM are not supported by the server_ssh command.

EXAMPLE #1

To display the current configuration, type:

```bash
$ server_ssh server_2 -info
```

```
server_2 : done

SERVICE CONFIGURATION

<table>
<thead>
<tr>
<th>Port</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>running</td>
</tr>
<tr>
<td>Thread count</td>
<td>4</td>
</tr>
<tr>
<td>Banner</td>
<td>/server2fs1/banner.txt</td>
</tr>
<tr>
<td>Default home directory</td>
<td>/</td>
</tr>
<tr>
<td>Restrict home directory</td>
<td>disabled</td>
</tr>
<tr>
<td>Application</td>
<td>sftp, scp</td>
</tr>
<tr>
<td>Cipher</td>
<td>&lt;all&gt;</td>
</tr>
<tr>
<td>Compression</td>
<td>delayed</td>
</tr>
</tbody>
</table>
```
The server Commands

Mac : <all>
Max authentication tries : 5
Password authentication : enabled
Permit empty password : disabled
Public key authentication : enabled
Timeout : 30 seconds

EXAMPLE #2 To display the current configuration including allow and deny directives, type:

$ server_ssh server_3 -info

server_3 : done

SERVICE CONFIGURATION
Port : 22
State : running
Thread count : 16
Banner : /fs40/banner.txt
Default home directory : /
Restrict home directory : disabled
Application : sftp,scp
Cipher : <all>
Compression : no
Mac : <all>
Max authentication tries : 10
Password authentication : enabled
Permit empty password : disabled
Public key authentication : enabled
Timeout : 900 seconds

ACCESS CONTROL
Allowed IPs : <all>
Denied IPs : 192.168.34.23,192.168.34.10
Allowed users : jack, tony, angela, root, sheila, Monica
Denied users : <none>
Allowed groups : <all>
Denied groups : <none>

EXAMPLE #3 To start the SSH server, type:

$ server_ssh server_3 -start

server_3 : done

EXAMPLE #4 To stop the SSH server, type:

$ server_ssh server_3 -stop

server_3 : done

EXAMPLE #5 To update the number of parameter threads, type:

$ server_ssh server_3 -modify -threads 16

server_3 : done
The server Commands

EXAMPLE #6  To remove an allowed group, type:

server_3 : done

$ server_ssh server_3 -remove -allowgroups guests

EXAMPLE #7  To modify the banner file, type:

server_2 : done

$ server_ssh server_2 -modify -banner /server2fs1/banner.txt

Note: The file /server2fs1/banner.txt must exist for the Data Mover point of view.

EXAMPLE #8  To modify the timeout to five minutes, type:

server_2 : done

$ server_ssh server_2 -modify -timeout 300

EXAMPLE #9  To add a new allowed user defined in NIS or LDAP, type:

server_2 : done

$ server_ssh server_2 -append -allowusers john

EXAMPLE #10  To add a new allowed user defined in the dom10 Windows domain, type:

server_2 : done

$ server_ssh server_2 -append -allowusers dom10\cindy

EXAMPLE #11  To add a new allowed group of users, type:

server_2 : done

$ server_ssh server_2 -append -allowgroups admin

EXAMPLE #12  To add a new allowed client IP, type:

server_2 : done

$ server_ssh server_2 -append -allowhosts 110.171.1.10

EXAMPLE #13  To add new allowed client hosts using their subnet, type:

server_2 : done

$ server_ssh server_2 -append -allowhosts 110.121.0.0/16

EXAMPLE #14  To add a new denied user, type:

server_2 : done

$ server_ssh server_2 -append -denyusers john

EXAMPLE #15  To add a new denied group of users, type:

$ server_ssh server_2 -append -denygroups guest
**EXAMPLE #16**
To add a new denied client IP, type:

```
$ server_ssh server_2 -append -denyhosts 110.171.1.54
```

**Note:** This operation may take a long time. The SSH server must be stopped as the Data Mover does not have existing host keys.

**EXAMPLE #17**
To generate a new host key for the Data Mover, type:

```
$ server_ssh server_2 -generate
```

**ERROR CASE #1**
To change the banner file to a non-existing file, type:

```
$ server_ssh server_2 -modify -banner foo
```

**ERROR CASE #2**
To enable an unknown application on top of SSH, type:

```
$ server_ssh server_2 -modify -application foo
```

**ERROR CASE #3**
To change the number of SSHD threads to an unauthorized value, type:

```
$ server_ssh server_2 -modify -threads 256
```

**ERROR CASE #4**
To change the port of the SSH server to an already used port, type:

```
$ server_ssh server_2 -modify -port 445
```

**ERROR CASE #5**
To delete the existing host keys of the Data Mover, type:

```
$ server_ssh server_2 -delete
```

**Note:** The SSH server must be stopped.
The server Commands

Note: 445 is used by the CIFS server.

**ERROR CASE #5** To regenerate the Data Mover host keys while the SSH server is active, type:

$ server_ssh server_2 -generate

server_2 :
Error 13163823112: server_2 : The SSH server must be stopped before executing this command.

**ERROR CASE #6** To generate the Data Mover host key, and specify an invalid key type, type:

$ server_ssh server_2 -generate -type foo

server_2 :
Error 13163823109: server_2 : Invalid SSH configuration: Bad KEYTYPE value attribute.

**ERROR CASE #7** To generate the Data Mover host key, and specify an RSA key with an invalid size, type:

$ server_ssh server_2 -generate -type RSA -keysize 23

server_2 :
Error 13163823110: server_2 : Invalid configuration value for the SSH server: Bad keysize value specified.

**ERROR CASE #8** To generate the Data Mover host key as a DSA key when a key of this type already exists, type:

$ server_ssh server_2 -generate -type dsa

server_2 :
Error 13163823123: server_2 : The command failed as the DSA host key is already defined.
server_standby

Manages the standby and RDF relationships for the specified Data Movers.

SYNOPSIS

server_standby {<movername>|ALL}
-activate {mover|rdf}
-restore {mover|rdf}
-delete {mover[=<movername>]|rdf}
-verify {mover}
-create {mover=<source_movername> [-policy <policy_type>]|rdf[=<slot_number>]}

DESCRIPTION

server_standby manages the standby and RDF relationship for the specified Data Mover.

The ALL option executes the command for all Data Movers.

CAUTION

The ALL option should not be used when creating, restoring, or activating standby or RDF Data Movers.

OPTIONS

-activate {mover|rdf}
Activates the standby Data Mover or the RDF Data Mover for the primary Data Mover. Before activating the standby, the reboot caused by creating the standby relationship must be complete.

-delete {mover[=<movername>]|rdf}
Deletes relationship between the standby and primary Data Mover or deletes the RDF Data Mover for the primary Data Mover. If you have more than one standby for a primary, you can specify which standby is to be deleted by entering a <movername>. server_setup provides information to change the type defined for a Data Mover.

-restore {mover|rdf}
Restores the standby Data Mover or the RDF Data Mover for the primary Data Mover once it has been activated.

-verify {mover}
Verifies the standby status of the Data Mover after a hardware replacement. If the Data Mover hardware has been replaced, the reboot cycle may not be displayed.
-create {mover=<source_movername> [-policy <policy_type>]}
Designates the <source_movername> as the standby. The following rules apply when selecting a standby Data Mover:

- The primary Data Mover cannot be a standby Data Mover.
- A standby cannot be created for a standby.
- No file systems can be mounted on the standby.
- The standby must have the same disk configuration as the primary.
- One standby Data Mover can be created for multiple Data Movers. To do this, execute server_standby -create for each primary Data Mover.

-policy <policy_type>
When a fault is detected on the primary Data Mover, the following -policy options apply:

  auto
  Initiates immediate activation of the standby Data Mover.

  retry
  Attempts to recover the primary Data Mover first, then if recovery fails, initiates activation of the standby.

  manual (default)
  Reboots the primary Data Mover. No action on the standby is initiated.

-create rdf[=<slot_number>]]
Designates a remote RDF standby Data Mover as indicated by its slot number.

SEE ALSO
Configuring Standbys on VNX and server_setup.

EXAMPLE #1
To create a standby relationship between server_2 (primary) and server_3 (standby), type:

```bash
$ server_standby server_2 -create mover=server_3
server_2 : server_3 is rebooting as standby
```

Note: Before any other actions can take place, the reboot must be complete.
## The server Commands

### EXAMPLE #2
To activate the server_3 (standby) to take over for server_2 (primary), type:

```bash
$ server_standby server_2 -activate mover
```

```
server_2 :
server_2 : going offline
server_3 : going active
replace in progress ...done
failover activity complete
commit in progress (not interruptible)...done

server_2 : renamed as server_2.faulted.server_3
server_3 : renamed as server_2
```

### EXAMPLE #3
To restore server_3 as the standby Data Mover and server_2.faulted.server_3 as the primary, type:

```bash
$ server_standby server_2 -restore mover
```

```
server_2 :
server_2 : going standby
server_2.faulted.server_3 : going active
replace in progress ...done
failover activity complete
commit in progress (not interruptible)...done

server_2 : renamed as server_3
server_2.faulted.server_3 : renamed as server_2
```

### EXAMPLE #4
To verify readiness of the standby Data Mover, type:

```bash
$ server_standby server_2 -verify mover
```

```
server_2 : ok
```

### EXAMPLE #5
To delete the standby relationship for server_2, type:

```bash
$ server_standby server_2 -delete mover
```

```
server_2 : done
```

### EXAMPLE #6
To create a standby relationship for three Data Movers, type:

```bash
$ server_standby server_2 -create mover=server_3
```

```
server_2 : server_3 is rebooting as standby
```

```bash
$ server_standby server_4 -create mover=server_3
```

```
server_4 : done
```

```bash
$ server_standby server_5 -create mover=server_3
```

```
server_5 : done
```

---

**Note:** Before any other actions can take place, the reboot must be complete.
server_stats

Displays sets of statistics that are running on the specified Data Mover.

SYNOPSIS

server_stats <movername>
   -list
   | -info [-all|<statpath_name>[,...]]
   | -service { -start [-port <port_number>] |
   | -stop |
   | -delete |
   | -status }
   | -monitor -action {status|enable|disable}
   |
   [{{ -monitor {statpath_name|statgroup_name}[,...]
   | -monitor {statpath_name|statgroup_name}
   [-sort <field_name>]
   [-order {asc|desc}]
   [-lines <lines_of_output>] }...
   [-count <count>]
   [-interval <seconds>]
   [-terminationsummary {no|yes|only}]
   [-format {text [-titles {never|once|repeat}]|csv}]
   [-type {rate|diff|accu}]
   [-file <output_filepath> [-overwrite]]
   | -resolve
   ]

DESCRIPTION

server_stats provides real-time statistics for the specified Data Mover. Statistics are displayed in a time-series fashion at the end of each interval. The statistics are organized as a hierarchy of families that replaces the summary and table collections. Previously defined summary and table collections have been converted to system-defined statistics groups that can be used as arguments with the new -monitor option:

<table>
<thead>
<tr>
<th>System-defined statistics group name</th>
<th>User-defined statistics group name</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic-std</td>
<td>-summary basic</td>
</tr>
<tr>
<td>caches-std</td>
<td>-summary caches</td>
</tr>
<tr>
<td>cifs-std</td>
<td>-summary cifs</td>
</tr>
<tr>
<td>nfs-std</td>
<td>-summary nfs</td>
</tr>
<tr>
<td>cifsOps-std</td>
<td>-table cifs</td>
</tr>
<tr>
<td>diskVolumes-std</td>
<td>-table dvol</td>
</tr>
</tbody>
</table>

server_stats 665
The server Commands

<table>
<thead>
<tr>
<th>System-defined statistics group name</th>
<th>User-defined statistics group name</th>
</tr>
</thead>
<tbody>
<tr>
<td>metaVolumes-std</td>
<td>-table fsvol</td>
</tr>
<tr>
<td>netDevices-std</td>
<td>-table net</td>
</tr>
<tr>
<td>nfsOps-std</td>
<td>-table nfs</td>
</tr>
</tbody>
</table>

Note: The -summary and -table options are deprecated and will not be supported in future releases.

IP Resolution
The nfs.client and cifs.client stats are indexed by client IP address. Customers requested that client name be shown instead.

To implement this, IP address have to be resolved to client name. The server_stats command now does this resolution by default when a users requests these stats.

NIS, DNS, or a local hosts file must be setup for resolution to work.

UID Resolution
The nfs.user stat is indexed by user id (UID). To maintain consistency, UIDs need to be resolved to User Names.

The server_stats command now does this resolution by default when a users requests this stat.

The NIS Service or a local passwd file must have user information for resolution to work.

GID Resolution
The nfs.group stat is indexed by group id (GID). To maintain consistency, GIDs need to be resolved to Group Names.

The server_stats command now does this resolution by default when a users requests this stat. This is a support stat that requires the -vis support argument.

The NIS Service or a local group file must have group information for resolution to work.

In addition, server_stats manages the Statistics Monitoring service (statmonService) running on Data Movers including the ability to disable and enable statistics.
The new statistics are:

cifs.user

Provides cifs read and write statistics by call and bytes correlated to cifs users. It displays the same data that cifs.client does but correlated to user as opposed to IP address.

The statistics contain the following information:

- totalCalls
- readCalls
- writeCalls
- suspectCalls
- totalBytes
- readBytes
- writeBytes
- avgTime
- server
- client

The default sort field is totalCalls.

nfs.user

Provides nfs read and write statistics by call and bytes correlated to nfs users. It displays the same data that nfs.client does but correlated to user as opposed to IP address.

The statistics will contain the following information:

- totalCalls
- readCalls
- writeCalls
- suspectCalls
- totalBytes
- readBytes
- writeBytes
- avgTime

The default sort field is totalCalls.
The server Commands

**store.volume**

Provides Disk Volume read and write statistics by blocks and bytes correlated to File System and Disk Volume. It displays top File Systems per disk volume. To list filesystems for a specific disk volume (for e.g. a volume named d133), run the server_stats command as: $ server_stats server_2 -m store.volume.d133

The statistics contain the following information:

- totalBlocks
- readBlocks
- writeBlocks
- totalBytes
- readBytes
- writeBytes

The default sort field is totalBlocks.

**OPTIONS**

**No arguments**
Displays a basic summary of statistics for the specified Data Mover as defined by the basic-std Statistics Group.

- **list**
  Displays all defined statistics starting with the statgroup names followed by statpaths and their types.

- **info**
  Displays the statgroup and statpath information.

- **service**
  Specifies whether to start, stop, delete, or query the status of the statmonService. The statmonService runs on the Data Mover and listens or the server_stats requests.

  - **-start**
    Starts the statmonService on the Data Mover. If the -port argument is specified, it is used by statmonManager service. These settings are persistent and execute as part of the Data Mover’s boot-up configurations.

  - **-stop**
    Shuts down the statmonService on the specified Data Mover.
The server Commands

[-delete]
Deletes the statmonService persistent configurations so it does not execute as part of the Data Mover’s boot-up settings. If -delete is executed while the statmonService is running, the service stops and its configuration is deleted.

[-status]
Checks the status of the statmonService on the specified Data Mover.

-monitorn [-action]
Enables, disables, or queries the state of the stats collection.

-monitorn [statpath_name | statgroup_name]
Takes a comma-separated list of statpath and statgroup names. In cases where stats are available for multiple elements, the user can specify an element name or use 'ALL-ELEMENTS' to refer to all elements at once.

Any duplicate statpath or statgroup names is consolidated and reported once. The options below are only applicable to Set and Correlated Set statpath names:

[-sort <field_name>]
Sorts a Set based on <field_name> values.

[-order {asc | desc}]
Sorted Sets can be ordered in ascending or descending order.

[-lines <lines_of_output>]
Sets output can be limited to display the Set statistics <lines_of_output> lines.

Set statistics is not sorted if none of the above three options are specified. If some of these options are specified with a Set statpath name, the omitted options default as follows:

- sort defaults to the element name
- lines defaults to all
- order default value depends on the –sort field. If the Set is being sorted on a numeric field, -order defaults to descending order; otherwise, it defaults to ascending order. For sorting based on the Set element name, -order always defaults to ascending order.

Each Correlated Set are defined with a default sort field and order. If some of these options are specified with a Set statpath name, the omitted options default as follows:
-sort default field is unique for each Correlated Set
-lines always defaults to 20 for all Correlated Sets
-order default value depends on the -sort field. If the Correlated Set is being sorted on a numeric field, -order defaults to descending order; otherwise, it defaults to ascending order. Correlated Sets cannot be sorted on non numeric fields, including the Correlated Set element ID.

[-count <count>]
Specifies the number of reports that are generated. Reports are produced after each set interval of time. The <count> must be a positive integer. By default, reports are generated indefinitely till the user presses Ctrl-C.

[-interval <seconds>]
Specifies the number of seconds between samples. The <seconds> must be an integer, 1 through 300 or higher. The default is 15.

[-terminationsummary {no | yes | only}]
Enables, disables, or manages the termination summary that appears after the <count> lines have completed or a Ctrl-C interrupt is executed. All items, both active and inactive, are reported. The summary reports average, minimum, and maximum values for numeric fields collected over the complete run. The no and yes choices control the summary message. The only option suppresses the reporting of the time series and reports only the summary. The default value is yes.

[-format {text | csv}]
Sets the format for printing statistics. The text (default) format prints statistics separated by a sequence of blank characters when rendered in a fixed-sized font. The time stamp that appears on the front of each snapshot is local time in the form hh:mm:ss.

The csv format supports the display of elements in dynamic Set statistic. A dynamic collection is a Set statistic where new elements are reported in any sample. In order to support this dynamic behavior for the csv format and keep the csv output consistent between the title line and the sample data lines, the title line is reprinted with samples where new elements are detected.

To avoid seeing multiple title lines in the csv output, use the –file option that saves the output to a specified file instead of outputting to the standard output.
The advantage of using the –file option is that server_stats prints samples to the file and does not print the title line except once at the end of the session when the title line is complete. In this case, the title line appears once in the output at the top of the file.

The csv format produces output in a comma-separated list and renders the local time in a yyyy/mm/dd hh:mm:ss format.

The csv output and the -file option:

• Without the -file option: The output to stdout may show multiple title lines where new elements were detected.

• With the -file option: The output is printed to the specified file so at the end of the session, the final titles line is printed once at the top of the file. The result is a cleaner output than multiple titles lines where new elements were detected.

[-titles {never | once | repeat}]
Controls the generation of titles. The never option indicates that titles are not printed, the once option indicates that they are only printed at the top of the report, and the repeat option indicates that they are repeated for each time-series snapshot. The default setting is once.

The -titles option cannot be combined with the -format csv option since titles could be dynamically updated based on the dynamic nature of some statistics. The -format option provides more information.

[-type {rate | diff | accu}]
Specifies the display type of value for statistics with monotonically increasing values. The display type applies to statistics that increase monotonically, for example, network in-bound bytes. Other statistics that represent a point-in-time value, for example, current CIFS connections, are not affected by this option. The rate value displays the rate of change since the previous sample, the diff value displays the change in value since the previous sample, and the accu value displays the change in value since the initial sample. The default display type is rate.

[-file <output_filepath> [-overwrite]]
Allows the users to specify a filename to save the output of session instead of printing to standard output (stdout). This is necessary to provide a cleaner csv output. Since the titles line continues to change each time a new element is detected, it is printed a number of times to stdout. However, with the -file option, the titles line appears once at the top of the output.
The server Commands

In case <output_filepath> exists, the results will be appended to it. The -overwrite argument can be used to change this behavior so that an existing file is deleted before the results are written to it.

The argument of the -file option is a valid file name and does not have a default argument.

-noresolve
The -noresolve option is a new server_stats option that prevents any id resolution of any kind. This includes: FS IDs, Volume IDs, MetaVolume IDs, Quota Tree IDs, IP Addresses, UIDs, and GIDs.

SEE ALSO Managing Statistics for VNX.

EXAMPLE #1 To display the basic-std group by indicating the rate of change since the previous sample, type:

```bash
$ server_stats server_2 -monitor basic-std -interval 5 -count 5 -type rate
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>CPU</th>
<th>Network</th>
<th>Network</th>
<th>dVol</th>
<th>dVol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Util%</td>
<td>In KiB/s</td>
<td>Out KiB/s</td>
<td>Read KiB/s</td>
<td>Write KiB/s</td>
</tr>
<tr>
<td>02:53:09</td>
<td>33</td>
<td>88618</td>
<td>729</td>
<td>9988</td>
<td>89730</td>
</tr>
<tr>
<td>02:53:11</td>
<td>61</td>
<td>142057</td>
<td>1087</td>
<td>18632</td>
<td>167076</td>
</tr>
<tr>
<td>02:53:13</td>
<td>29</td>
<td>98110</td>
<td>762</td>
<td>8976</td>
<td>78599</td>
</tr>
<tr>
<td>02:53:15</td>
<td>46</td>
<td>109783</td>
<td>836</td>
<td>14288</td>
<td>125717</td>
</tr>
<tr>
<td>02:53:17</td>
<td>34</td>
<td>111847</td>
<td>886</td>
<td>10652</td>
<td>95719</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>server_2</th>
<th>CPU</th>
<th>Network</th>
<th>Network</th>
<th>dVol</th>
<th>dVol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>Util%</td>
<td>In KiB/s</td>
<td>Out KiB/s</td>
<td>Read KiB/s</td>
<td>Write KiB/s</td>
</tr>
<tr>
<td>Minimum</td>
<td>29</td>
<td>88618</td>
<td>729</td>
<td>8976</td>
<td>78599</td>
</tr>
<tr>
<td>Average</td>
<td>41</td>
<td>110083</td>
<td>860</td>
<td>12507</td>
<td>111368</td>
</tr>
<tr>
<td>Maximum</td>
<td>61</td>
<td>142057</td>
<td>1087</td>
<td>18632</td>
<td>167076</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Time the poll was taken.</td>
</tr>
<tr>
<td>CPU Util</td>
<td>Disk utilized in percentage in this interval.</td>
</tr>
<tr>
<td>Network In KiB/s</td>
<td>Network kibibytes received over all network interfaces.</td>
</tr>
<tr>
<td>Network Out KiB/s</td>
<td>Network kibibytes sent over all network interfaces.</td>
</tr>
<tr>
<td>dVol Read KiB/s</td>
<td>Storage kibibytes received from all server-storage interfaces.</td>
</tr>
<tr>
<td>dVol Write KiB/s</td>
<td>Storage kibibytes sent to all server-storage interfaces.</td>
</tr>
</tbody>
</table>

EXAMPLE #2 To display the basic-std group by indicating the change in value since the previous sample, type:
### The server Commands

```
$ server_stats server_2 -monitor basic-std -interval 5 -count 5 -type diff
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>CPU</th>
<th>Network</th>
<th>Network</th>
<th>dVol</th>
<th>dVol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp</td>
<td>Util</td>
<td>In KiB</td>
<td>Out KiB</td>
<td>Read</td>
<td>Write</td>
</tr>
<tr>
<td>%</td>
<td></td>
<td>diff</td>
<td>diff</td>
<td>diff</td>
<td>KiB diff</td>
</tr>
<tr>
<td>02:53:29</td>
<td>46</td>
<td>267660</td>
<td>2136</td>
<td>26128</td>
<td>232654</td>
</tr>
<tr>
<td>02:53:31</td>
<td>38</td>
<td>200668</td>
<td>1543</td>
<td>23144</td>
<td>211182</td>
</tr>
<tr>
<td>02:53:33</td>
<td>46</td>
<td>226761</td>
<td>1749</td>
<td>26488</td>
<td>230558</td>
</tr>
<tr>
<td>02:53:35</td>
<td>48</td>
<td>246921</td>
<td>1876</td>
<td>28720</td>
<td>255957</td>
</tr>
<tr>
<td>02:53:37</td>
<td>40</td>
<td>212353</td>
<td>1673</td>
<td>23016</td>
<td>210573</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>server_2</th>
<th>CPU</th>
<th>Network</th>
<th>Network</th>
<th>dVol</th>
<th>dVol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Util</td>
<td>In KiB</td>
<td>Out KiB</td>
<td>Read</td>
<td>Write</td>
</tr>
<tr>
<td>%</td>
<td></td>
<td>diff</td>
<td>diff</td>
<td>diff</td>
<td>KiB diff</td>
</tr>
<tr>
<td>Minimum</td>
<td>38</td>
<td>200668</td>
<td>1543</td>
<td>23016</td>
<td>210573</td>
</tr>
<tr>
<td>Average</td>
<td>44</td>
<td>230873</td>
<td>1795</td>
<td>25499</td>
<td>228185</td>
</tr>
<tr>
<td>Maximum</td>
<td>48</td>
<td>267660</td>
<td>2136</td>
<td>28720</td>
<td>255957</td>
</tr>
</tbody>
</table>

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Time the poll was taken.</td>
</tr>
<tr>
<td>CPU Util %</td>
<td>Disk utilized in percentage in this interval.</td>
</tr>
<tr>
<td>Network In KiB diff</td>
<td>Network kibibytes received over all network interfaces per differential value.</td>
</tr>
<tr>
<td>Network Out KiB diff</td>
<td>Network kibibytes sent over all network interfaces per differential value.</td>
</tr>
<tr>
<td>dVol Read KiB diff</td>
<td>Storage kibibytes received from all server-storage interfaces per differential value.</td>
</tr>
<tr>
<td>dVol Write KiB diff</td>
<td>Storage kibibytes sent to all server-storage interfaces per differential value.</td>
</tr>
</tbody>
</table>

**EXAMPLE #3**

To display the basic-std group by indicating the change in value since the first sample, type:

```
$ server_stats server_2 -monitor basic-std -interval 5 -count 5 -type accu
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>CPU</th>
<th>Network</th>
<th>Network</th>
<th>dVol</th>
<th>Read KiB</th>
<th>Write KiB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp</td>
<td>Util</td>
<td>In KiB</td>
<td>Out KiB</td>
<td></td>
<td>Read KiB</td>
<td>Write KiB</td>
</tr>
<tr>
<td>%</td>
<td></td>
<td>diff</td>
<td>diff</td>
<td>diff</td>
<td>KiB</td>
<td>KiB</td>
</tr>
<tr>
<td>02:53:48</td>
<td>42</td>
<td>236257</td>
<td>1880</td>
<td>25504</td>
<td>224832</td>
<td></td>
</tr>
<tr>
<td>02:53:50</td>
<td>54</td>
<td>505640</td>
<td>3983</td>
<td>55760</td>
<td>500538</td>
<td></td>
</tr>
<tr>
<td>02:53:52</td>
<td>29</td>
<td>686282</td>
<td>5377</td>
<td>74096</td>
<td>662494</td>
<td></td>
</tr>
<tr>
<td>02:53:54</td>
<td>46</td>
<td>922765</td>
<td>7183</td>
<td>101704</td>
<td>908813</td>
<td></td>
</tr>
<tr>
<td>02:53:56</td>
<td>41</td>
<td>1125518</td>
<td>8777</td>
<td>126640</td>
<td>1134362</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>server_2</th>
<th>CPU</th>
<th>Network</th>
<th>Network</th>
<th>dVol</th>
<th>Read KiB</th>
<th>Write KiB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Util</td>
<td>In KiB</td>
<td>Out KiB</td>
<td></td>
<td>Read KiB</td>
<td>Write KiB</td>
</tr>
<tr>
<td>%</td>
<td></td>
<td>diff</td>
<td>diff</td>
<td>diff</td>
<td>KiB</td>
<td>KiB</td>
</tr>
<tr>
<td>Minimum</td>
<td>29</td>
<td>236257</td>
<td>1880</td>
<td>25504</td>
<td>224832</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>42</td>
<td>695293</td>
<td>5440</td>
<td>76741</td>
<td>686208</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>54</td>
<td>1125518</td>
<td>8777</td>
<td>126640</td>
<td>1134362</td>
<td></td>
</tr>
</tbody>
</table>
The server Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Time the poll was taken.</td>
</tr>
<tr>
<td>CPU Util</td>
<td>Disk utilized in percentage in this interval.</td>
</tr>
<tr>
<td>Network In KiB</td>
<td>Network kibibytes received over all network interfaces per accumulated value.</td>
</tr>
<tr>
<td>Network Out KiB</td>
<td>Network kibibytes sent over all network interfaces per accumulated value.</td>
</tr>
<tr>
<td>dVol Read KiB</td>
<td>Storage kibibytes received from all server-storage interfaces per accumulated value.</td>
</tr>
<tr>
<td>dVol Write KiB</td>
<td>Storage kibibytes sent to all server-storage interfaces per accumulated value.</td>
</tr>
</tbody>
</table>

**EXAMPLE #4** To display a list of statistics group names followed by statpaths and their types, type:

```
$ server_stats server_3 -list
```

<table>
<thead>
<tr>
<th>Type</th>
<th>Stat Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Correlated Set</td>
<td>cifs.user</td>
</tr>
<tr>
<td>Counter</td>
<td>cifs.user.ALL-ELEMENTS.totalCalls</td>
</tr>
<tr>
<td>Counter</td>
<td>cifs.user.ALL-ELEMENTS.readCalls</td>
</tr>
<tr>
<td>Counter</td>
<td>cifs.user.ALL-ELEMENTS.writeCalls</td>
</tr>
<tr>
<td>Fact</td>
<td>cifs.user.ALL-ELEMENTS.suspectCalls</td>
</tr>
<tr>
<td>Counter</td>
<td>cifs.user.ALL-ELEMENTS.totalBytes</td>
</tr>
<tr>
<td>Counter</td>
<td>cifs.user.ALL-ELEMENTS.readBytes</td>
</tr>
<tr>
<td>Counter</td>
<td>cifs.user.ALL-ELEMENTS.writeBytes</td>
</tr>
<tr>
<td>Fact</td>
<td>cifs.user.ALL-ELEMENTS.avgTime</td>
</tr>
<tr>
<td>Fact</td>
<td>cifs.user.ALL-ELEMENTS.server</td>
</tr>
<tr>
<td>Fact</td>
<td>cifs.user.ALL-ELEMENTS.client</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Correlated Set</td>
<td>nfs.user</td>
</tr>
<tr>
<td>Counter</td>
<td>nfs.user.ALL-ELEMENTS.totalCalls</td>
</tr>
<tr>
<td>Counter</td>
<td>nfs.user.ALL-ELEMENTS.readCalls</td>
</tr>
<tr>
<td>Counter</td>
<td>nfs.user.ALL-ELEMENTS.writeCalls</td>
</tr>
<tr>
<td>Fact</td>
<td>nfs.user.ALL-ELEMENTS.suspectCalls</td>
</tr>
<tr>
<td>Counter</td>
<td>nfs.user.ALL-ELEMENTS.totalBytes</td>
</tr>
<tr>
<td>Counter</td>
<td>nfs.user.ALL-ELEMENTS.readBytes</td>
</tr>
<tr>
<td>Counter</td>
<td>nfs.user.ALL-ELEMENTS.writeBytes</td>
</tr>
<tr>
<td>Fact</td>
<td>nfs.user.ALL-ELEMENTS.avgTime</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Set</td>
<td>store.volume</td>
</tr>
<tr>
<td>Correlated Set</td>
<td>store.volume.ALL-ELEMENTS.fileSystem</td>
</tr>
<tr>
<td>Counter</td>
<td>store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.totalBlocks</td>
</tr>
<tr>
<td>Counter</td>
<td>store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.readBlocks</td>
</tr>
<tr>
<td>Counter</td>
<td>store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.writeBlocks</td>
</tr>
<tr>
<td>Counter</td>
<td>store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.totalBytes</td>
</tr>
<tr>
<td>Counter</td>
<td>store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.readBytes</td>
</tr>
<tr>
<td>Counter</td>
<td>store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.writeBytes</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

**EXAMPLE #5** To display the statgroup information, type:
$ server_stats server_2 -info statsb

server_2 :
name = statsB
description = My group # 2
type = Group - user-defined
member_stats = nfs.basic,cifs.basic,iscsi.basic
member_elements =
member_of = statsA

EXAMPLE #6 To display information of statistics group names followed by statpaths, type:

$ server_stats server_2 -info

server_2 :
name = statsA
description = My group # 1
type = Group - user-defined
member_stats = statsB
member_elements =
member_of =

... name = cifs
description = The CIFS-protocol service
type = Family
member_stats =
member_elements =
member_of =

EXAMPLE #7 To start the statmonService, type:

$ server_stats server_2 -service -start -port 7777
statmonService started on port: 7777.

EXAMPLE #8 To stop the statmonService, type:

$ server_stats server_2 -service -stop
server_2: done.

EXAMPLE #9 To delete the statmonService configurations from the boot-up settings of the Data Mover, type:

$ server_stats server_2 -service -delete
server_2: done.

EXAMPLE #10 To query the status of the statmonService, type:

$ server_stats server_2 -service -status
server_2: The statmonService has started.
Interface=INTERNAL
Port=7777
The statmonService is listing for incoming network connections
Max Connections: 32, Current: 0

**EXAMPLE #11**  To enable stats collection, type:

```
$ server_stats server_2 -monitor -action enable
```

server_2: done.

**EXAMPLE #12**  To query the state of the stats collection, type:

```
$ server_stats server_2 -monitor -action status
```

server_2: Statistics are enabled.

**EXAMPLE #13**  To display five iterations of the cifs-std statistics group with a three second interval, type:

```
$ server_stats server_2 -monitor cifs-std -i 5-c 5
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS Avg</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS Avg</th>
<th>CIFS</th>
<th>CIFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Total Ops/s</td>
<td>Read Ops/s</td>
<td>Read KiB/s</td>
<td>Read Size</td>
<td>Write Ops/s</td>
<td>Write KiB/s</td>
<td>Write Size</td>
<td>Share Connections</td>
<td>Open Files</td>
</tr>
<tr>
<td>02:54:31</td>
<td>2133</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>1947</td>
<td>110600</td>
<td>57</td>
<td>96</td>
<td>587</td>
</tr>
<tr>
<td>02:54:34</td>
<td>1895</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>1737</td>
<td>99057</td>
<td>57</td>
<td>96</td>
<td>631</td>
</tr>
<tr>
<td>02:54:37</td>
<td>2327</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>2104</td>
<td>119556</td>
<td>57</td>
<td>96</td>
<td>649</td>
</tr>
<tr>
<td>02:54:40</td>
<td>2109</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>1864</td>
<td>106081</td>
<td>57</td>
<td>96</td>
<td>653</td>
</tr>
<tr>
<td>02:54:43</td>
<td>2439</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>2172</td>
<td>123578</td>
<td>57</td>
<td>96</td>
<td>639</td>
</tr>
</tbody>
</table>

**EXAMPLE #14**  To display five iterations of the nfs-std statistics group with a three second interval, type:

```
$ server_stats server_2 -monitor nfs-std -i 5-c 5
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS Avg</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>Total Ops/s</td>
<td>Read Ops/s</td>
<td>Read KiB/s</td>
<td>Read Size</td>
<td>Write Ops/s</td>
<td>Write KiB/s</td>
<td>Write Size</td>
</tr>
<tr>
<td>Minimum</td>
<td>1895</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>1737</td>
<td>99057</td>
<td>57</td>
</tr>
<tr>
<td>Average</td>
<td>2180</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>1965</td>
<td>111775</td>
<td>57</td>
</tr>
<tr>
<td>Maximum</td>
<td>2439</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>2172</td>
<td>123578</td>
<td>57</td>
</tr>
</tbody>
</table>

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Time the poll was taken.</td>
</tr>
<tr>
<td>CIFS Total Ops/s</td>
<td>Total operations per second.</td>
</tr>
<tr>
<td>CIFS Read Ops/s</td>
<td>CIFS read operations per second in the interval.</td>
</tr>
<tr>
<td>CIFS Read KiB/s</td>
<td>CIFS read data response in kibibytes per second.</td>
</tr>
<tr>
<td>CIFS Avg Size Read KiB</td>
<td>Average read data response.</td>
</tr>
<tr>
<td>CIFS Write Ops/s</td>
<td>CIFS write operations per second.</td>
</tr>
<tr>
<td>CIFS Write KiB/s</td>
<td>CIFS write data response in kibibytes per second.</td>
</tr>
<tr>
<td>CIFS Avg Size Write KiB</td>
<td>Average write data size.</td>
</tr>
<tr>
<td>CIFS Share Connections</td>
<td>Number of CIFS protocol connections.</td>
</tr>
<tr>
<td>CIFS Open Files</td>
<td>Number of open CIFS files.</td>
</tr>
</tbody>
</table>
The server Commands

server_2 Total NFS NFS NFS Avg NFS NFS NFS Avg NFS
Timestamp NFS Read Read Read Size Write Write Write Size Active
Ops/s Ops/s KiB/s Bytes Ops/s KiB/s Bytes Threads
13:44:53 20650 4121 67506 16774 2214 29737 13754 648
13:44:54 11663 2318 37140 16407 1238 17307 14316 648
13:44:55 8678 1790 30761 17597 945 12511 13557 648
13:44:56 17655 3543 56382 16296 1967 27077 14096 648
13:44:57 20302 4033 63822 16205 2271 31469 14189 648

Summary

Minimum 8678 1790 30761 16205 945 12511 13557 648
Average 15790 3161 51122 16656 1727 23620 13982 648
Maximum 20650 4121 67506 17597 2271 31469 14316 648

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Time the poll was taken.</td>
</tr>
<tr>
<td>Total NFS Ops/s</td>
<td>Total number of operations per second.</td>
</tr>
<tr>
<td>NFS Read Ops/s</td>
<td>NFS read operations per second in the interval.</td>
</tr>
<tr>
<td>NFS Read KiB/s</td>
<td>NFS read data response in kibibytes per second.</td>
</tr>
<tr>
<td>NFS Avg Read Size Bytes</td>
<td>Average read data response.</td>
</tr>
<tr>
<td>NFS Write Ops/s</td>
<td>NFS write operations per second.</td>
</tr>
<tr>
<td>NFS Write KiB/s</td>
<td>NFS write data response in kibibytes per second.</td>
</tr>
<tr>
<td>NFS Avg Write Size Bytes</td>
<td>Average write data size.</td>
</tr>
<tr>
<td>NFS Active Threads</td>
<td>Number of NFS active threads.</td>
</tr>
</tbody>
</table>

**EXAMPLE #15**  
To display five iterations of the caches-std statistics group with a three second interval, type:

```
$ server_stats server_2 -monitor caches-std -i 3 -c 5
```

<table>
<thead>
<tr>
<th>server_2 DNLC</th>
<th>OF Cache</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Hit</td>
<td>Hit</td>
</tr>
<tr>
<td>Ratio %</td>
<td>Ratio %</td>
<td>Hit %</td>
</tr>
<tr>
<td>02:55:26</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>02:55:29</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>02:55:32</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>02:55:35</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>02:55:38</td>
<td>-</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>server_2 DNLC</th>
<th>OF Cache</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>Hit</td>
<td>Hit</td>
</tr>
<tr>
<td>Ratio %</td>
<td>Ratio %</td>
<td>Hit %</td>
</tr>
<tr>
<td>Minimum</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>Maximum</td>
<td>-</td>
<td>100</td>
</tr>
</tbody>
</table>

server_stats
### Example #16

To display the netDevices-std statistics group with a three second interval, type:

```
$ server_stats server_2 -monitor netDevices-std -i 3 -c 3
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>device</th>
<th>Network</th>
<th>Network</th>
<th>Network</th>
<th>Network</th>
<th>Network</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Pkts/s</td>
<td>Errors/s</td>
<td>KiB/s</td>
<td>Pkts/s</td>
<td>Errors/s</td>
<td>KiB/s</td>
</tr>
<tr>
<td>Timestamp</td>
<td></td>
<td>In</td>
<td>In</td>
<td></td>
<td>Out</td>
<td></td>
<td>Out</td>
</tr>
<tr>
<td>02:55:52</td>
<td>mge0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>mge1</td>
<td>17</td>
<td>23</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cge0</td>
<td>3593</td>
<td>0</td>
<td>26566</td>
<td>2289</td>
<td>0</td>
<td>203</td>
</tr>
<tr>
<td></td>
<td>cge1</td>
<td>6912</td>
<td>0</td>
<td>50206</td>
<td>4444</td>
<td>0</td>
<td>378</td>
</tr>
<tr>
<td></td>
<td>cge2</td>
<td>3637</td>
<td>0</td>
<td>25570</td>
<td>2342</td>
<td>0</td>
<td>209</td>
</tr>
<tr>
<td>02:55:55</td>
<td>mge0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>mge1</td>
<td>7</td>
<td>0</td>
<td>9</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>cge0</td>
<td>3444</td>
<td>0</td>
<td>24744</td>
<td>2252</td>
<td>0</td>
<td>204</td>
</tr>
<tr>
<td></td>
<td>cge1</td>
<td>7415</td>
<td>0</td>
<td>53354</td>
<td>4721</td>
<td>0</td>
<td>400</td>
</tr>
<tr>
<td></td>
<td>cge2</td>
<td>3913</td>
<td>0</td>
<td>27796</td>
<td>2502</td>
<td>0</td>
<td>222</td>
</tr>
<tr>
<td>02:55:58</td>
<td>mge0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>mge1</td>
<td>32</td>
<td>0</td>
<td>39</td>
<td>19</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>cge0</td>
<td>4029</td>
<td>0</td>
<td>29334</td>
<td>2594</td>
<td>0</td>
<td>230</td>
</tr>
<tr>
<td></td>
<td>cge1</td>
<td>7461</td>
<td>0</td>
<td>54030</td>
<td>4791</td>
<td>0</td>
<td>406</td>
</tr>
<tr>
<td></td>
<td>cge2</td>
<td>3902</td>
<td>0</td>
<td>27319</td>
<td>2505</td>
<td>0</td>
<td>223</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>server_2</th>
<th>device</th>
<th>Network</th>
<th>Network</th>
<th>Network</th>
<th>Network</th>
<th>Network</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Pkts/s</td>
<td>Errors/s</td>
<td>KiB/s</td>
<td>Pkts/s</td>
<td>Errors/s</td>
<td>KiB/s</td>
</tr>
<tr>
<td>Summary</td>
<td></td>
<td>In</td>
<td>In</td>
<td></td>
<td>Out</td>
<td></td>
<td>Out</td>
</tr>
<tr>
<td>Minimum</td>
<td>mge0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>mge1</td>
<td>7</td>
<td>0</td>
<td>9</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>cge0</td>
<td>3444</td>
<td>0</td>
<td>24744</td>
<td>2252</td>
<td>0</td>
<td>203</td>
</tr>
<tr>
<td></td>
<td>cge1</td>
<td>6912</td>
<td>0</td>
<td>50206</td>
<td>4444</td>
<td>0</td>
<td>378</td>
</tr>
<tr>
<td></td>
<td>cge2</td>
<td>3637</td>
<td>0</td>
<td>25570</td>
<td>2342</td>
<td>0</td>
<td>209</td>
</tr>
<tr>
<td></td>
<td>cge3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Average</td>
<td>mge0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>mge1</td>
<td>19</td>
<td>0</td>
<td>24</td>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>cge0</td>
<td>3689</td>
<td>0</td>
<td>26882</td>
<td>2378</td>
<td>0</td>
<td>213</td>
</tr>
<tr>
<td></td>
<td>cge1</td>
<td>7263</td>
<td>0</td>
<td>52530</td>
<td>4652</td>
<td>0</td>
<td>395</td>
</tr>
<tr>
<td></td>
<td>cge2</td>
<td>3817</td>
<td>0</td>
<td>26895</td>
<td>2450</td>
<td>0</td>
<td>218</td>
</tr>
<tr>
<td></td>
<td>cge3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Maximum</td>
<td>mge0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>mge1</td>
<td>32</td>
<td>0</td>
<td>39</td>
<td>19</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>cge0</td>
<td>4029</td>
<td>0</td>
<td>29334</td>
<td>2594</td>
<td>0</td>
<td>230</td>
</tr>
<tr>
<td></td>
<td>cge1</td>
<td>7461</td>
<td>0</td>
<td>54030</td>
<td>4791</td>
<td>0</td>
<td>406</td>
</tr>
<tr>
<td></td>
<td>cge2</td>
<td>3913</td>
<td>0</td>
<td>27796</td>
<td>2505</td>
<td>0</td>
<td>223</td>
</tr>
</tbody>
</table>
The server Commands

EXAMPLE #17 To display the netDevices-std statistics group without the summary and with a three second interval, type:

```
$ server_stats server_2 -monitor netDevices-std -i 3 -c 3 -terminationsummary no
```

<table>
<thead>
<tr>
<th>server_2 device</th>
<th>Network In Pkts/s</th>
<th>Network In Errors/s</th>
<th>Network In KiB/s</th>
<th>Network Out Pkts/s</th>
<th>Network Out Errors/s</th>
<th>Network Out KiB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>mge0</td>
<td>16</td>
<td>0</td>
<td>1</td>
<td>19</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>mge1</td>
<td>43</td>
<td>0</td>
<td>60</td>
<td>24</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>cge0</td>
<td>3960</td>
<td>0</td>
<td>29053</td>
<td>2547</td>
<td>0</td>
<td>226</td>
</tr>
<tr>
<td>cge1</td>
<td>6709</td>
<td>0</td>
<td>48414</td>
<td>4296</td>
<td>0</td>
<td>366</td>
</tr>
<tr>
<td>cge2</td>
<td>4829</td>
<td>0</td>
<td>33996</td>
<td>3125</td>
<td>0</td>
<td>281</td>
</tr>
</tbody>
</table>

EXAMPLE #18 To display the cifsOps-std statistics with a five second interval, type:

```
$ server_stats server_2 -monitor cifsops-std -i 5 -c 3
```

<table>
<thead>
<tr>
<th>server_2 device</th>
<th>SMB Operation</th>
<th>Op Calls/s</th>
<th>Min uSec</th>
<th>Max uSec</th>
<th>Avg uSec/call</th>
</tr>
</thead>
<tbody>
<tr>
<td>mge0</td>
<td>SMB1_Close</td>
<td>89</td>
<td>45</td>
<td>406775</td>
<td>10273</td>
</tr>
<tr>
<td>mge0</td>
<td>SMB1_WriteX</td>
<td>1837</td>
<td>30</td>
<td>1618776</td>
<td>144030</td>
</tr>
<tr>
<td>mge0</td>
<td>SMB1_CreateNTX</td>
<td>84</td>
<td>51</td>
<td>458090</td>
<td>379</td>
</tr>
</tbody>
</table>

Value Definition

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Time the poll was taken.</td>
</tr>
<tr>
<td>Device</td>
<td>Name of the network device.</td>
</tr>
<tr>
<td>Network In Pkts/s</td>
<td>Network packets received per second.</td>
</tr>
<tr>
<td>Network In Errors/s</td>
<td>Network input errors encountered per second.</td>
</tr>
<tr>
<td>Network In KiB/s</td>
<td>Network kibibytes received per second.</td>
</tr>
<tr>
<td>Network Out Pkts/s</td>
<td>Network packets sent per second.</td>
</tr>
<tr>
<td>Network Out Errors/s</td>
<td>Network output errors encountered per second.</td>
</tr>
<tr>
<td>Network Out KiB/s</td>
<td>Network kibibytes sent per second.</td>
</tr>
</tbody>
</table>
### The server Commands

<table>
<thead>
<tr>
<th>Operation</th>
<th>Calls/s</th>
<th>uSec</th>
<th>uSec</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMB1_Mkdir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Rmdir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Open</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Create</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Close</td>
<td>89</td>
<td>45</td>
<td>406775</td>
<td>10057</td>
</tr>
<tr>
<td>SMB1_Flush</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Unlink</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Rename</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_GetAttr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_SetAttr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Read</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Write</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Lock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Unlock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_CreateTmp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_MkNew</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_ChkPath</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1.Exit</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Lseek</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_LockRead</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_WriteUnlock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_ReadBlockRaw</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_WriteBlockRaw</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_SetAttrExp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_GetAttrExp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_LockingX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Trans</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_TransSec</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Copy</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Move</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Echo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_WriteClose</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_OpenX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_ReadX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_WriteX</td>
<td>1837</td>
<td>30</td>
<td>1618776</td>
<td>121976</td>
</tr>
<tr>
<td>SMB1_CloseTreeDisco</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Trans2Prim</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Trans2Secd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_FindClose2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_FindNotifyClose</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_TreeConnect</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_TreeDisco</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_NEGProt</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_SessSetupX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_UserLogoffX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>
### The server Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>SMB1_Count</th>
<th>SMB1_Sec</th>
<th>SMB1_Total</th>
<th>SMB1_Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMB1_TreeConnectX</td>
<td>0</td>
<td>82</td>
<td>499</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_DiskAttr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Search</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_FindFirst</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_FindUnique</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_FindClose</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_TransNT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_TransNTSecd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_CreateNTX</td>
<td>84</td>
<td>51</td>
<td>458090</td>
<td>379</td>
</tr>
<tr>
<td>SMB1_CancelNT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_SendMessage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_BeginMessage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_EndMessage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_MessageText</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Negotiate</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_SessionSetup</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Logoff</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_TreeConnect</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_TreeDisConnect</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Create</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Close</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Flush</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Read</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Write</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Lock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Ioctl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Cancel</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Echo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_QueryDirectory</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_ChangeNotify</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_QueryInfo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_SetInfo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_OplockBreak</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Mkdir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Rmdir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Open</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Create</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Close</td>
<td>105</td>
<td>45</td>
<td>406775</td>
<td>11664</td>
</tr>
<tr>
<td>SMB1_Flush</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Unlink</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Rename</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_GetAttr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_SetAttr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Read</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Write</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Lock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Unlock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_CreateTmp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_MkNew</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_ChkPath</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Exit</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Lseek</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Command</td>
<td>Calls</td>
<td>Creator Calls</td>
<td>Creator Calls</td>
<td>Creator Calls</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>SMB1_LockRead</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_WriteUnlock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_ReadBlockRaw</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_WriteBlockRaw</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_SetAttrExp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_GetAttrExp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_LockingX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Trans</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_TransSec</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Copy</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Move</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Echo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_WriteClose</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_OpenX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_ReadX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_WriteX</td>
<td>1941</td>
<td>30</td>
<td>1618776</td>
<td>133062</td>
</tr>
<tr>
<td>SMB1_CloseTreeDisco</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Trans2Prim</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Trans2Secd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_FindClose2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_FindNotifyClose</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_TreeConnect</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_TreeDisco</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_NegProt</td>
<td>0</td>
<td>44</td>
<td>85</td>
<td>85</td>
</tr>
<tr>
<td>SMB1_SessSetupX</td>
<td>0</td>
<td>1088</td>
<td>12058</td>
<td>12058</td>
</tr>
<tr>
<td>SMB1_UserLogoffX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_TreeConnectX</td>
<td>0</td>
<td>82</td>
<td>499</td>
<td>499</td>
</tr>
<tr>
<td>SMB1_DiskAttr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Search</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_FindFirst</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_FindUnique</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_FindClose</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_TransNT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_TransNTSecd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_CreateNTX</td>
<td>104</td>
<td>51</td>
<td>458090</td>
<td>1335</td>
</tr>
<tr>
<td>SMB1_CancelNT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_SendMessage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_BeginMessage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_EndMessage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_Negotiate</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_SessionSetup</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_Logoff</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_TreeConnect</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_TreeDisConnect</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_Create</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_Close</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_Hash</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_Trans</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_Write</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_Lock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB2_IOCTL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The server Commands
### The server Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>SMB2_Cancel</th>
<th>SMB2_Echo</th>
<th>SMB2_QueryDirectory</th>
<th>SMB2_ChangeNotify</th>
<th>SMB2_QueryInfo</th>
<th>SMB2_SetInfo</th>
<th>SMB2_OplockBreak</th>
<th>Maximum</th>
<th>SMB1_Mkdir</th>
<th>SMB1_Rmdir</th>
<th>SMB1_Open</th>
<th>SMB1_Create</th>
<th>SMB1_OpenX</th>
<th>SMB1_WriteX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Maximum</td>
<td>122</td>
<td>45</td>
<td>406775</td>
<td>14663</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Flush</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Unlink</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Rename</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_GetAttr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_SetAttr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Read</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Write</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Lock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Unlock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_CreateTmp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Mkdir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Rmdir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Open</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Create</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_OpenX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_WriteX</td>
<td>2119</td>
<td>30</td>
<td>1618776</td>
<td>144030</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_CloseTreeDisco</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Trans2Prim</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_Trans2Secd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_FindClose2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_FindNotifyClose</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_TreeConnect</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_TreeDisco</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_NegProt</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_SessSetupX</td>
<td>0</td>
<td>1088</td>
<td>12058</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB1_UserLogoffX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The server Commands

<table>
<thead>
<tr>
<th>SMB Operation</th>
<th>Op Calls/s</th>
<th>Min uSec</th>
<th>Max uSec</th>
<th>Avg uSec/Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMB1_TreeConnectX</td>
<td>0</td>
<td>82</td>
<td>499</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_DiskAttr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_Search</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_FindFirst</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_FindUnique</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_FindClose</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_TransNT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_TransNTSecd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_CreateNTX</td>
<td>126</td>
<td>51</td>
<td>458090</td>
<td>1826</td>
</tr>
<tr>
<td>SMB1_CancelNT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_SendMessage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_BeginMessage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_EndMessage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB1_MessageText</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Negotiate</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_SessionSetup</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Logoff</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Punch</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Create</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Close</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Flush</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Read</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Write</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Lock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Ioctl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Cancel</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_Echo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_QueryDirectory</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_ChangeNotify</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_QueryInfo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_SetInfo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMB2_OplockBreak</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Time the poll was taken.</td>
</tr>
<tr>
<td>SMB Operation</td>
<td>Name of the SMB operation.</td>
</tr>
<tr>
<td>Op Calls/s</td>
<td>Number of calls to this SMB operation per second.</td>
</tr>
<tr>
<td>Min uSec</td>
<td>Minimum time in microseconds per call.</td>
</tr>
<tr>
<td>Max uSec</td>
<td>Maximum time in microseconds per call.</td>
</tr>
<tr>
<td>Avg uSec/Call</td>
<td>Average time in microseconds consumed per call.</td>
</tr>
</tbody>
</table>

EXAMPLE #19

To display the cifsOps-std statistics group without the summary and with a five second interval, type:

```
$ server_stats server_2 -m cifsops-std -i 5-c 3 -te no
```
The server Commands

EXAMPLE #20  To display the nfsOps-std statistics group without the summary and with a five second interval, type:

```
$ server_stats server_2 -m nfsops-std -i 5 -c 3 -te no
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>NFS Op</th>
<th>NFS Op Calls/s</th>
<th>NFS Op Errors/s</th>
<th>NFS Op uSec/Call</th>
<th>NFS Op %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Op</td>
<td>Op</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calls/s</td>
<td>Errors/s</td>
<td>uSec/call</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03:18:21</td>
<td>v3Read</td>
<td>23442                0</td>
<td>63846</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>v3Write</td>
<td>23372               0</td>
<td>99156</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>03:18:24</td>
<td>v3Read</td>
<td>23260               0</td>
<td>65756</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>v3Write</td>
<td>23243               0</td>
<td>101135</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>03:18:27</td>
<td>v3Read</td>
<td>23385               0</td>
<td>66808</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>v3Write</td>
<td>23323               0</td>
<td>102201</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Time the poll was taken.</td>
</tr>
<tr>
<td>NFS Op</td>
<td>Name of the NFS operation.</td>
</tr>
<tr>
<td>NFS Op Calls/s</td>
<td>Number of calls to this NFS operation per second.</td>
</tr>
<tr>
<td>NFS Op Errors/s</td>
<td>Number of times the NFS operation failed per second.</td>
</tr>
<tr>
<td>NFS Op uSec/Call</td>
<td>Average time in microseconds consumed per call.</td>
</tr>
<tr>
<td>NFS Op %</td>
<td>Percent of total NFS calls attributed to this operation.</td>
</tr>
</tbody>
</table>

EXAMPLE #21  To display the diskVolumes-std statistics group without the summary and with a five second interval, type:

```
$ server_stats server_2 -m diskVolumes-std -i 5 -c 3 -te no
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>dVol</th>
<th>Queue Depth</th>
<th>Read Ops/s</th>
<th>Read KiB/s</th>
<th>Read Avg Size</th>
<th>Write Ops/s</th>
<th>Write KiB/s</th>
<th>Write Avg Size</th>
<th>Util %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02:58:09</td>
<td>NBS1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>8192</td>
<td>1</td>
<td>7</td>
<td>6827</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>root_ldisk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>461</td>
<td>490</td>
<td>1090</td>
</tr>
<tr>
<td></td>
<td>d7</td>
<td>0</td>
<td>113</td>
<td>904</td>
<td>8192</td>
<td>530</td>
<td>19619</td>
<td>37881</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>d11</td>
<td>0</td>
<td>249</td>
<td>1995</td>
<td>8192</td>
<td>431</td>
<td>11640</td>
<td>27634</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>d8</td>
<td>0</td>
<td>68</td>
<td>547</td>
<td>8192</td>
<td>372</td>
<td>11472</td>
<td>31607</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>d12</td>
<td>33</td>
<td>424</td>
<td>3389</td>
<td>8192</td>
<td>609</td>
<td>20045</td>
<td>33705</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>d9</td>
<td>0</td>
<td>36</td>
<td>291</td>
<td>8192</td>
<td>592</td>
<td>20339</td>
<td>35161</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>d13</td>
<td>0</td>
<td>333</td>
<td>2664</td>
<td>8192</td>
<td>347</td>
<td>11925</td>
<td>35158</td>
<td>93</td>
</tr>
</tbody>
</table>

server_stats
### The server Commands

<table>
<thead>
<tr>
<th>dVol</th>
<th>Queue Depth</th>
<th>Read Ops/s</th>
<th>Read KiB/s</th>
<th>Avg Read Size Bytes</th>
<th>Write Ops/s</th>
<th>Write KiB/s</th>
<th>Avg Write Size Bytes</th>
<th>Util %</th>
</tr>
</thead>
<tbody>
<tr>
<td>d10</td>
<td>0</td>
<td>24</td>
<td>189</td>
<td>8192</td>
<td>385</td>
<td>31896</td>
<td>11896</td>
<td>63</td>
</tr>
<tr>
<td>d14</td>
<td>36</td>
<td>573</td>
<td>4581</td>
<td>8192</td>
<td>454</td>
<td>20173</td>
<td>45468</td>
<td>100</td>
</tr>
</tbody>
</table>

**EXAMPLE #22**

To display the metaVolumes-std statistics group without the summary and with a five second interval, type:

```
$ server_stats server_2 -m metaVolumes-std -i 5 -c 3 -te no
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>MetaVol</th>
<th>Read Requests/s</th>
<th>Read KiB/s</th>
<th>Read Avg Size Ops/s</th>
<th>Read Requests/s</th>
<th>Read KiB/s</th>
<th>Write Avg Write Size Ops/s</th>
<th>Write Avg Write KiB/s</th>
<th>Write Avg Write Size KiB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Time the poll was taken.</td>
</tr>
<tr>
<td>dVol</td>
<td>Name of the disk volume.</td>
</tr>
<tr>
<td>Queue Depth</td>
<td>Queue depth of the disk volume.</td>
</tr>
<tr>
<td>Read Ops/s</td>
<td>Number of read operations per second.</td>
</tr>
<tr>
<td>Read KiB/s</td>
<td>Kibibytes read per second.</td>
</tr>
<tr>
<td>Avg Read Size Bytes</td>
<td>Average size in bytes of read requests per second.</td>
</tr>
<tr>
<td>Write Ops/s</td>
<td>Number of write operations per second.</td>
</tr>
<tr>
<td>Write KiB/s</td>
<td>Kibibytes written per second.</td>
</tr>
<tr>
<td>Avg Write Size Bytes</td>
<td>Average size in bytes for write requests per second.</td>
</tr>
<tr>
<td>Util %</td>
<td>Disk utilized in percentage.</td>
</tr>
</tbody>
</table>

**EXAMPLE #22**

To display the metaVolumes-std statistics group without the summary and with a five second interval, type:

```
$ server_stats server_2 -m metaVolumes-std -i 5 -c 3 -te no
```
The server Commands

| ufs_3  | 0   | 0   | -  | 0   | 11  | 2155 | 200580 | 11 |
| ufs_2  | 347 | 2776| 8192| 347 | 140 | 23544 | 172208 | 140|
| ufs_0  | 315 | 2517| 8192| 315 | 148 | 21427 | 147916 | 148|
| ufs_1  | 654 | 5229| 8192| 654 | 313 | 45512 | 148895 | 313|
| root_fs_3 | 1  | 11  | 8192| 1   | 0   | 0     | -      | 0  |

02:58:40  SNBS5  0  0  -  0  3  37  12743  3
SNBS1  0  0  -  0  3  1  512  3
ufs_4  0  0  -  0  159  1257  8089  157
ufs_5  0  0  -  0  160  1273  8158  159
ufs_3  0  0  -  0  2  511  224695  2
ufs_2  396 | 3166| 8192| 396 | 195 | 27326 | 143200 | 195|
ufs_0  431 | 3446| 8192| 431 | 187 | 29574 | 162161 | 187|
ufs_1  408 | 3262| 8192| 408 | 159 | 27782 | 178784 | 159|
root_fs_3  1  5  8192  1  0  0  -  0  0  2  511  224695  2

02:58:43  SNBS5  0  0  -  0  1  5  5461  1
SNBS6  0  0  -  0  1  3  4608  1
ufs_4  0  0  -  0  146  1159  8136  145
ufs_5  0  0  -  0  148  1183  8174  148
ufs_3  0  0  -  0  8  1965  262144  8
ufs_2  522 | 4174| 8192| 522 | 219 | 35546 | 166238 | 219|
ufs_0  492 | 3933| 8192| 492 | 222 | 33356 | 153886 | 222|
ufs_1  467 | 3736| 8192| 467 | 188 | 31955 | 173819 | 188|

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaVol</td>
<td>Name of the meta volume associated with the file system.</td>
</tr>
<tr>
<td>Read Request/s</td>
<td>Number of read requests per second to this volume.</td>
</tr>
<tr>
<td>Read KiB/s</td>
<td>Kibibytes read per second.</td>
</tr>
<tr>
<td>Avg Read Size Bytes</td>
<td>Average size for read requests to this volume.</td>
</tr>
<tr>
<td>Read Ops/s</td>
<td>Number of read operations per second.</td>
</tr>
<tr>
<td>Write Requests/s</td>
<td>Number of write requests per second.</td>
</tr>
<tr>
<td>Write KiB/s</td>
<td>Number of kibibytes written per second to this volume.</td>
</tr>
<tr>
<td>Avg Write Size Bytes</td>
<td>Average size in bytes for write requests.</td>
</tr>
<tr>
<td>Write Ops/s</td>
<td>Number of write operations per second.</td>
</tr>
</tbody>
</table>

EXAMPLE #23  To display the nfsOps-std statistics group sorted by the percentage of all the NFS operations for the five second interval, type:

```
$ server_stats server_2 -monitor nfsOps-std -sort opPct -i 5 -c 3 -te
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>NFS Op</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS Op %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Calls/s</td>
<td>Errors/s</td>
<td>uSec/call</td>
<td>Op</td>
<td>Op</td>
</tr>
<tr>
<td>03:18:57</td>
<td>v3Read</td>
<td>23263</td>
<td>0</td>
<td>81632</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>v3Write</td>
<td>23352</td>
<td>0</td>
<td>116645</td>
<td>50</td>
</tr>
<tr>
<td>03:19:00</td>
<td>v3Read</td>
<td>23431</td>
<td>0</td>
<td>82443</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>v3Write</td>
<td>23345</td>
<td>0</td>
<td>118088</td>
<td>50</td>
</tr>
<tr>
<td>03:19:03</td>
<td>v3Read</td>
<td>23176</td>
<td>0</td>
<td>84759</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>v3Write</td>
<td>23326</td>
<td>0</td>
<td>119608</td>
<td>50</td>
</tr>
</tbody>
</table>
EXAMPLE #24 To display the nfsOps-std statistics group sorted by the average time in microseconds used for the five second interval, in ascending order, type:

```
$ server_stats server_2 -m nfsops-std -sort avgTime -order asc -i 5 -c 3 -te no
```

```
server_2    NFS Op       NFS       NFS         NFS    NFS Op %
Timestamp                 Op        Op          Op
Calls/s  Errors/s   uSec/call
04:05:27   v3Write       605         0     8022318         100
          v3Create        2         0    25304786           0
04:05:30   v3Create        8         0    7722823           1
          v3Write       579         0    8435543           99
04:05:33   v3Create        41         0    1468883           7
          v3Write       567         0    8690860           93
```

EXAMPLE #25 To display the nfsOps-std statistics group sorted by the average time in microseconds used for the five second interval, in descending order, and including the three counts of data output, type:

```
$ server_stats server_2 -m nfsops-std -sort avgTime -order desc -lines 3 -i 5 -c 3 -te no
```

```
server_2    NFS Op      NFS         NFS         NFS    NFS Op %
Timestamp                 Op          Op          Op
Calls/s    Errors/s   uSec/call
04:09:39   v3Create        1           0    31657550           0
          v3Write      610           0     6223366         100
04:09:44    v3Write      607           0     6275942          98
          v3Create       11           0     3978054           2
04:09:49    v3Write      574           0     6691264          93
          v3Create       42           0     1073819           7
```

EXAMPLE # 26 To display the information about minInterval and defaultInterval, type:

```
$ server_param server_2 -f statmon -info minInterval
```

```
server_2 :  
name = minInterval
facility_name = statmon
default_value = 5
current_value = 5
configured_value =
user_action = none
change_effective = immediate
range = (1,300)
description = Minimum Interval time used by the statmon session
```

```
$ server_param server_2 -f statmon -info defaultInterval
```

```
server_2:  
name = defaultInterval
```
facility_name = statmon
default_value = 15
current_value = 10
configured_value = 10
user_action = none
change_effective = immediate
range = (1,300)
description = Default Interval time used by statmon session

**EXAMPLE #27**  To change the value of minInterval and defaultInterval, type:

```
$ server_param server_2 -f statmon -modify minInterval -value 5
```

server_2: done.

```
$ server_param server_2 -f statmon -modify defaultInterval -value 10
```

server_2: done.

**EXAMPLE #28**  To verify that server_stats displays an error message if the value specified to modify for minInterval and defaultInterval variables is not in range, type:

```
$ server_param server_2 -facility statmon -modify minInterval -value 1000
```

server_2:
Error 4418: server_2: 1000 is not in range (1,300)

```
$ server_param server_2 -facility statmon -modify defaultInterval -value 400
```

server_2:
Error 4418: server_3: 400 is not in range (1,300)

**EXAMPLE #29**  To check whether server_stats displays an error message when a specified interval is smaller than the minInterval specified by server_param, type:

```
$ server_param server_2 -facility statmon -modify minInterval -value 5
```

server_2: done

```
$ server_stats server_2 -m -i 1 nfs.basic
ERROR (13421969453): The requested interval (1) is less than the minInterval value (5).
```

**EXAMPLE #30**  To verify that server_stats displays an error message while it requests the stats data by using a defaultInterval value that is less than the minInterval value, type:

```
$ server_param server_2 -facility statmon -modify minInterval -value 5
```

server_2: done
The server Commands

```
$ server_param server_2 -facility statmon -modify
defaultInterval -value 1
server_2: done

$ server_stats server_2 -m nfs.basic
ERROR (13421969452): The defaultInterval parameter value (1) is less than the
minInterval parameter value (5).
```

**EXAMPLE #31** To display the Correlated Set list, type:

```
$ server_stats server_3 -l
```

```
server_3 :
  Type     Stat Name

... Correlated Set   cifs.user
    Counter    cifs.user.ALL-ELEMENTS.totalCalls
    Counter    cifs.user.ALL-ELEMENTS.readCalls
    Counter    cifs.user.ALL-ELEMENTS.writeCalls
    Fact       cifs.user.ALL-ELEMENTS.suspectCalls
    Counter    cifs.user.ALL-ELEMENTS.totalBytes
    Counter    cifs.user.ALL-ELEMENTS.readBytes
    Counter    cifs.user.ALL-ELEMENTS.writeBytes
    Fact       cifs.user.ALL-ELEMENTS.avgTime
    Fact       cifs.user.ALL-ELEMENTS.server
    Fact       cifs.user.ALL-ELEMENTS.client

... Correlated Set   nfs.user
    Counter    nfs.user.ALL-ELEMENTS.totalCalls
    Counter    nfs.user.ALL-ELEMENTS.readCalls
    Counter    nfs.user.ALL-ELEMENTS.writeCalls
    Fact       nfs.user.ALL-ELEMENTS.suspectCalls
    Counter    nfs.user.ALL-ELEMENTS.totalBytes
    Counter    nfs.user.ALL-ELEMENTS.readBytes
    Counter    nfs.user.ALL-ELEMENTS.writeBytes
    Fact       nfs.user.ALL-ELEMENTS.avgTime

... Set               store.volume
  Correlated Set     store.volume.ALL-ELEMENTS.fileSystem
    Counter    store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.totalBlocks
    Counter    store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.readBlocks
    Counter    store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.writeBlocks
    Counter    store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.totalBytes
    Counter    store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.readBytes
    Counter    store.volume.ALL-ELEMENTS.fileSystem.ALL-ELEMENTS.writeBytes

```

**EXAMPLE #31** To display cifs.client information with IP resolution, type:

```
$ server_stats server_2 -i 2 -m cifs.client -l 10
```
### The server Commands

#### EXAMPLE #32
To display nfs.client information with IP resolution, type:

```
$ server_stats server_2 -i 2 -m nfs.client
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>Client</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS</th>
<th>NFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Total Opr/s</td>
<td>Read Op/s</td>
<td>Write Op/s</td>
<td>Suspicious Ops</td>
<td>Total KiB/s</td>
<td>Read KiB/s</td>
<td>Write KiB/s</td>
<td>uSec/call</td>
<td>Avg KiB/s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:48:09</td>
<td>83</td>
<td>0</td>
<td>83</td>
<td>0</td>
<td>42044</td>
<td>0</td>
<td>42044</td>
<td>18213</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id=10.103.11.106</td>
<td>70</td>
<td>0</td>
<td>66</td>
<td>0</td>
<td>28448</td>
<td>0</td>
<td>28448</td>
<td>3182</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id=10.103.11.104</td>
<td>52</td>
<td>0</td>
<td>52</td>
<td>0</td>
<td>26659</td>
<td>0</td>
<td>26659</td>
<td>39984</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:48:11</td>
<td>97</td>
<td>0</td>
<td>97</td>
<td>0</td>
<td>49868</td>
<td>0</td>
<td>49868</td>
<td>13244</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p25.perf1.com</td>
<td>87</td>
<td>0</td>
<td>82</td>
<td>0</td>
<td>35815</td>
<td>0</td>
<td>35815</td>
<td>2549</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:48:13</td>
<td>116</td>
<td>0</td>
<td>116</td>
<td>0</td>
<td>59576</td>
<td>0</td>
<td>59576</td>
<td>10201</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p23.perf1.com</td>
<td>99</td>
<td>0</td>
<td>91</td>
<td>0</td>
<td>38273</td>
<td>0</td>
<td>38273</td>
<td>1306</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p24.perf1.com</td>
<td>51</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>26224</td>
<td>0</td>
<td>26224</td>
<td>11014</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:48:15</td>
<td>85</td>
<td>0</td>
<td>85</td>
<td>0</td>
<td>43591</td>
<td>0</td>
<td>43591</td>
<td>17385</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p25.perf1.com</td>
<td>70</td>
<td>0</td>
<td>62</td>
<td>0</td>
<td>27396</td>
<td>0</td>
<td>27396</td>
<td>551</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p24.perf1.com</td>
<td>45</td>
<td>0</td>
<td>45</td>
<td>0</td>
<td>23214</td>
<td>0</td>
<td>23214</td>
<td>14504</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### EXAMPLE #33
To monitor cifs.user information, type:

```
$ server_stats server_2 -i 1 -m cifs.user
```

<table>
<thead>
<tr>
<th>server_2</th>
<th>User name</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
<th>CIFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Total Opr/s</td>
<td>Read Opr/s</td>
<td>Write Opr/s</td>
<td>Suspicious Oprs</td>
<td>Total KiB/s</td>
<td>KIB/s</td>
<td>Write KiB/s</td>
<td>uSec/call</td>
<td>Avg KIB/s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:27:31</td>
<td>1175</td>
<td>0</td>
<td>1047</td>
<td>0</td>
<td>59786</td>
<td>0</td>
<td>59786</td>
<td>18279</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PERFL\src00000</td>
<td>1177</td>
<td>0</td>
<td>1071</td>
<td>0</td>
<td>60848</td>
<td>0</td>
<td>60848</td>
<td>19075</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:27:33</td>
<td>1176</td>
<td>0</td>
<td>1113</td>
<td>0</td>
<td>63689</td>
<td>0</td>
<td>63689</td>
<td>51307</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PERFL\src00000</td>
<td>1112</td>
<td>0</td>
<td>1042</td>
<td>0</td>
<td>58750</td>
<td>0</td>
<td>58750</td>
<td>18479</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:27:35</td>
<td>1086</td>
<td>0</td>
<td>995</td>
<td>0</td>
<td>56694</td>
<td>0</td>
<td>56694</td>
<td>18157</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PERFL\src00000</td>
<td>1086</td>
<td>0</td>
<td>995</td>
<td>0</td>
<td>56694</td>
<td>0</td>
<td>56694</td>
<td>18157</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Summary**

- **Minimum**: PERFL\src00000
- **Average**: PERFL\src00000
- **Maximum**: PERFL\src00000
### EXAMPLE #34
To monitor nfs.user information, type:

```shell
$ server_stats server_2 -i 1 -m nfs.user
```

### EXAMPLE #35
To monitor store.volume information, type:

```shell
$ server_stats server_2 -i 1 -m store.volume
```

### EXAMPLE #36
To monitor NFS user, type:

```shell
$ server_stats server_2 -i 1 -m nfs.user
```

---

**The server Commands**

**EXAMPLE #34**

<table>
<thead>
<tr>
<th>Time</th>
<th>IP Address</th>
<th>Total</th>
<th>Read</th>
<th>Write</th>
<th>Suspicious</th>
<th>Total</th>
<th>Read</th>
<th>Write</th>
<th>uSec/call</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00045</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00033</td>
<td>67</td>
<td>0</td>
<td>26</td>
<td>0</td>
<td>160</td>
<td>0</td>
<td>160</td>
<td>208</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00024</td>
<td>66</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>144</td>
<td>0</td>
<td>144</td>
<td>148</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00010</td>
<td>64</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>160</td>
<td>0</td>
<td>160</td>
<td>93</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00016</td>
<td>60</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>104</td>
<td>0</td>
<td>104</td>
<td>166</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00034</td>
<td>60</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>128</td>
<td>0</td>
<td>128</td>
<td>195</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00035</td>
<td>59</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>104</td>
<td>0</td>
<td>104</td>
<td>278</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00005</td>
<td>58</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>96</td>
<td>0</td>
<td>96</td>
<td>136</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00018</td>
<td>58</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>96</td>
<td>0</td>
<td>96</td>
<td>306</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00048</td>
<td>58</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>104</td>
<td>0</td>
<td>104</td>
<td>210</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00008</td>
<td>57</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>96</td>
<td>0</td>
<td>96</td>
<td>264</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00023</td>
<td>57</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>88</td>
<td>0</td>
<td>88</td>
<td>582</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00044</td>
<td>57</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>112</td>
<td>0</td>
<td>112</td>
<td>61</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00011</td>
<td>56</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>104</td>
<td>0</td>
<td>104</td>
<td>110</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00007</td>
<td>56</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>88</td>
<td>0</td>
<td>88</td>
<td>84</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00020</td>
<td>56</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>96</td>
<td>0</td>
<td>96</td>
<td>101</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00032</td>
<td>56</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>88</td>
<td>0</td>
<td>88</td>
<td>146</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00018</td>
<td>56</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>104</td>
<td>0</td>
<td>104</td>
<td>118</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00039</td>
<td>56</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>96</td>
<td>0</td>
<td>96</td>
<td>131</td>
</tr>
<tr>
<td>09:24:05</td>
<td>PERF1.com\src00041</td>
<td>56</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>88</td>
<td>0</td>
<td>88</td>
<td>234</td>
</tr>
</tbody>
</table>

---

**To monitor NFS user, type:**

```shell
server_stats server_2 -i 1 -m nfs.user
```

---

**To monitor store.volume information, type:**

```shell
server_stats server_2 -i 1 -m store.volume
```

---

**To monitor NFS user, type:**

```shell
server_stats server_2 -i 1 -m nfs.user
```
<table>
<thead>
<tr>
<th>id</th>
<th>2</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>4369</th>
</tr>
</thead>
<tbody>
<tr>
<td>id=558</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>286</td>
</tr>
<tr>
<td>id=563</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2231</td>
</tr>
<tr>
<td>id=569</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>228</td>
</tr>
<tr>
<td>id=585</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>247</td>
</tr>
<tr>
<td>id=588</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>282</td>
</tr>
<tr>
<td>id=589</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25418</td>
</tr>
<tr>
<td>id=591</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>214</td>
</tr>
<tr>
<td>id=595</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3700</td>
</tr>
<tr>
<td>id=551</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8535</td>
</tr>
<tr>
<td>id=556</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>238</td>
</tr>
<tr>
<td>id=557</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>312</td>
</tr>
<tr>
<td>id=564</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12921</td>
</tr>
<tr>
<td>id=582</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5748</td>
</tr>
</tbody>
</table>

The server Commands
server_sysconfig

Manages the hardware configuration for the specified or all Data Movers.

SYNOPSIS

server_sysconfig {<movername>|ALL}
   {-Platform
    | -pci [<device> [-option <options>]]
    | -virtual -delete [-Force] <device>
    | -virtual [-info <device>]
    | -virtual -name <device> -create trk
      -option "device=<device,..> [protocol=lacp][lb=<mac|ip|tcp>]"
    | -virtual -name <device> -create fsn
      -option primary=<device> device=<device,..>}

DESCRIPTION

server_sysconfig displays and modifies the hardware configuration of the Data Movers.

To use Ethernet channels and link aggregations, a complimentary configuration on the switch is required. The switch must support:

- IEEE standard Ethernet, Fast Ethernet, or Gigabit Ethernet
- IEEE 802.3ad Link Aggregation protocol

server_sysconfig implements a virtual device which combines one or more devices on the same Data Mover into a single, logical device addressable as a unit. Fail-safe networks, Ethernet channels, and link aggregations are virtual devices. Both Ethernet channels and link aggregations are trunking devices.

The ALL option executes the command for all Data Movers.

OPTIONS

-Platform
Displays the hardware configuration of the Data Mover, including processor type and speed, the bus speed in MHz, the main memory in MB, and the motherboard type.

-pci [<device>]
Displays the attributes of the specified network adapter card or for all devices.

-pci <device> -option <options>
Sets the attributes of the specified network adapter card installed in the Data Mover. Options must be comma-separated and double quoted.
Ethernet

**speed=\{10 \mid 100 \mid 1000 \mid auto\}**
Sets the speed for the port. *auto* (default) turns on autonegotiation; setting a fixed speed disables autonegotiation.

**duplex=\{full \mid half \mid auto\}**
Sets the duplex to *full*, *half*, or *auto*. The *auto* (default) turns autonegotiation on; setting a fixed duplex disables autonegotiation.

---

**CAUTION**

The speed and duplex settings on both sides of the physical connection must be the same. Mismatched speed and duplex settings can cause errors and impact network performance. For example, if the duplex is set to half on one end and full on the other, there might be network errors and performance issues.

---

**Gigabit Ethernet Fiber**

For Gigabit Ethernet Fiber connections, the speed is automatically set to 1000, and since it must remain at that setting, no speed setting is required.

**linkneg=\{enable \mid disable\}**
Disables autonegotiation on the network adapter card if it is not supported by the network Gigabit switch. The default is *enable*.

**rxflowctl=\{enable \mid disable\}**
Enables the ability to accept and process pause frames. The default is *disable*.

**txflowctl=\{enable \mid disable\}**
Enables pause frames to be transmitted. The default is *disable*.

---

**Gigabit Ethernet Copper**

**speed=\{10 \mid 100 \mid 1000 \mid auto\}**
Sets the speed for the port. The *auto* (default) turns on autonegotiation; setting a fixed speed disables autonegotiation.

**duplex=\{full \mid half \mid auto\}**
Sets the duplex to *full*, *half*, or *auto*. The *auto* (default) turns autonegotiation on; setting a fixed duplex disables autonegotiation.
CAUTION

The speed and duplex settings on both sides of the physical connection must be the same. Mismatched speed and duplex settings can cause errors and impact network performance. For example, if the duplex is set to half on one end and full on the other, there might be network errors and performance issues.

rxflowctl={enable|disable}
Enables the ability to accept and process pause frames. The default is disable.

txflowctl={enable|disable}
Enables pause frames to be transmitted. The default is disable.

-virtual -delete [-Force] <device>
Deletes an Ethernet channel, link aggregation, or fail-safe network device. Valid entries for a <device> are trk or fsn. The -Force option is required if an IP address has been assigned.

CAUTION

All of the IP interfaces configured over a virtual device are deleted when the -Force option is used. As a result, any applications on the VNX that use these interfaces might get disrupted. Among these applications are CIFS and VNX Replicator. If all of the interfaces used by a specific CIFS server get deleted by the server_sysconfig -Force command, the server does not communicate with clients any further, and Windows users lose access to their data.

-virtual
Information for all devices is displayed.

-virtual -info <device>
Displays information for the specified Ethernet channel, link aggregation, or fail-safe network device.

-virtual -name <device> -create trk -option
device=<device,...>
Creates a virtual device for trk (Ethernet channel or link aggregation). An Ethernet channel can combine two, four, or eight (must be a multiple of two) ports into a single virtual device. A maximum of eight devices for Fast Ethernet is allowed:

◆ For Fast Ethernet trunking, use ana0 through ana7.
◆ For Gigabit Ethernet trunking, use `ace0` and `ace1`.
◆ For Gigabit Ethernet using the NS series, use `cge0` through `cge5`.

The `-option` string defines one or more devices for the Ethernet channel or link aggregation. The `<device,...>` specifies the device or devices separated by commas. The `-option` string must have the options separated with a space, and must be enclosed with double quotes. If not, the command appears to execute, but does not pick up any options not contained within double quotes.

A gigabit trunk can be created on more than two gigabit devices. The VNX has only two gigabit devices (`ace0, ace1`), so two ports are created. The NS series has a six gigabit port, so a four port gigabit trunk can be created.

For link aggregation there can be one or more number of ports, up to a maximum of 12.

**CAUTION**

The virtual device must be created before the IP interfaces can be configured for a trunk network device.

```
[protocol=lacp]
Initializes link aggregation control protocol for Ethernet devices which allows a non-exponential number of devices. For example, one, three, or five entries.

[lb=mac|ip|tcp]
Specifies the statistical load balancing method that applies to the aggregator/trunk. The `mac` option indicates that the output link on the trunk/aggregator is chosen based on source and destination MAC addresses. The `ip` (default) option indicates that the output link on the trunk/aggregator is chosen based on source and destination IP addresses. This is useful in the situation that the Data Mover is connected to a router or to a layer 3 switch. The `tcp` option indicates that the output link on the trunk/aggregator is chosen based on source and destination IP addresses as well as source and destination TCP ports.
```

```
-virtual -name <device> -create fsn -option primary=<device> device=<device,...>
Creates a virtual device for fsn (fail-safe network). The fail-safe network device is a high-availability feature that provides switch-level redundancy. Configuring a failover channel is optional. Specifies the device name of the virtual device.
```
The -option string indicates a primary device for the fail-safe network device and a second device to be defined as the standby in case the primary device fails. The <device,...> specifies the device or devices separated by commas. The -option string must be defined in sequence, separated with a space, and enclosed with double quotes. If not, the command appears to execute, but does not pick up any options not contained within double quotes.

**CAUTION**

The virtual device must be created before the IP interfaces can be configured for the fail-safe network device.

**SEE ALSO**

Configuring and Managing Networking on VNX, Configuring and Managing Network High Availability on VNX, server_dbms, and server_ifconfig.

**INTERFACE OUTPUTS**

The network interface cards available are dependent on the type of system used.

For the VNX, the following NICs are available: loop, ace, ana, cge, el30, el31, fpa, and fa2. Note that loop, el30, and el31 are for internal use only.

For the NS series, the following NICs are available: loop, cge, el30, el31, and fge.

VDMs are included in both the CNS and NS series.

**EXAMPLE #1**

For the NS series, to view the system configuration for a Data Mover, type:

```bash
$ server_sysconfig server_2 -Platform
```

server_2 :
Processor = Intel Pentium 4
Processor speed (MHz) = 3100
Total main memory (MB) = 4023
Mother board = Barracuda XP
Bus speed (MHz) = 533
Bios Version = 3.30
Post Version = Rev. 02.14

For the CNS series, to view the system configuration for a Data Mover, type:

```bash
$ server_sysconfig server_2 -Platform
```

server_2 :
Processor = Intel Pentium 4
Processor speed (MHz) = 1600
Total main memory (MB) = 3967
Mother board = CMB-400
Bus speed (MHz) = 400
Bios Version = No Ver Info
Post Version = No Ver Info

**EXAMPLE #2**  For the NS series, to view the installed PCI configuration for a Data Mover, type:

```
$ server_sysconfig server_2 -pci
```

server_2 : PCI DEVICES:

On Board:
- Agilent Fibre Channel Controller
  - 0: fcp-0  IRQ: 22 addr: 50060160006004f0
  - 0: fcp-1  IRQ: 21 addr: 50060161006004f0
  - 0: fcp-2  IRQ: 18 addr: 50060162006004f0
  - 0: fcp-3  IRQ: 20 addr: 50060163006004f0

- Broadcom Gigabit Ethernet Controller
  - 0: fge0  IRQ: 24
    linkneg=enable txflowctl=disable rxflowctl=disable
  - 0: fge1  IRQ: 23
    linkneg=enable txflowctl=disable rxflowctl=disable
  - 0: cge0  IRQ: 24
    speed=auto duplex=auto txflowctl=disable rxflowctl=disable
  - 0: cge1  IRQ: 23
    speed=auto duplex=auto txflowctl=disable rxflowctl=disable
  - 0: cge2  IRQ: 26
    speed=auto duplex=auto txflowctl=disable rxflowctl=disable
  - 0: cge3  IRQ: 25
    speed=auto duplex=auto txflowctl=disable rxflowctl=disable
  - 0: cge4  IRQ: 28
    speed=auto duplex=auto txflowctl=disable rxflowctl=disable
  - 0: cge5  IRQ: 27
    speed=auto duplex=auto txflowctl=disable rxflowctl=disable

For the CNS series, to view the installed PCI configuration for a Data Mover, type:
$ server_sysconfig server_2 -pci

server_2 : PCI DEVICES:

Slot: 1
  Emulex LP9000 Fibre Channel Controller
    0: fcp-0 IRQ: 23 addr: 10000000c92b5a10
    1: fcp-1 IRQ: 24 addr: 10000000c92b5a11

Slot: 2
  Emulex LP9000 Fibre Channel Controller
    0: fcp-2 IRQ: 22 addr: 10000000c92b514e

Slot: 4
  Intel 10/100/1K Ethernet Controller
    0: cge0 IRQ: 18
      speed=auto duplex=auto txflowctl=disable
    1: cge1 IRQ: 19
      speed=auto duplex=auto txflowctl=disable
    2: cge2 IRQ: 20
      speed=auto duplex=auto txflowctl=disable
    3: cge3 IRQ: 21
      speed=auto duplex=auto txflowctl=disable

Slot: 5
  Alteon Tigon-2 Gigabit Ethernet Controller
    0: ace0 IRQ: 25
      linkneg=enable txflowctl=disable
    0: ace0 IRQ: 25
      linkneg=enable txflowctl=disable

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>On Board</td>
<td>The names of each PCI card installed.</td>
</tr>
<tr>
<td>0</td>
<td>Port number inside the slot the card is plugged into. If the card put in</td>
</tr>
<tr>
<td>IRQ</td>
<td>the slot has 4 ports, the first port is marked as 0, second port as 1,</td>
</tr>
<tr>
<td></td>
<td>third port as 2, and fourth port as 3.</td>
</tr>
<tr>
<td>speed</td>
<td>Speed configured. Possible values are: auto, 10, 100, 1000.</td>
</tr>
<tr>
<td>duplex</td>
<td>Duplex setting configured. Possible values are: auto, half, full.</td>
</tr>
<tr>
<td>txflowctl</td>
<td>Transmit MAC flow control. Possible values are: disable, enable.</td>
</tr>
<tr>
<td>rxflowctl</td>
<td>Receive MAC flow control. Possible values are: disable, enable.</td>
</tr>
</tbody>
</table>
EXAMPLE #3  To set the 100 Mbits speed and full duplex setting for cge0 interface, type:

```
$ server_sysconfig server_2 -pci cge0 -option
   speed=100,duplex=full
```

server_2 : done

EXAMPLE #4  To display the hardware configuration for network device, cge0, type:

```
$ server_sysconfig server_2 -pci cge0
```

server_2 :
On Board:
Broadcom Gigabit Ethernet Controller
  0: cge0  IRQ: 24
  speed=100 duplex=full txflowctl=disable rxflowctl=disable

EXAMPLE #5  To create an Ethernet channel as a virtual device, type:

```
$ server_sysconfig server_2 -virtual -name trk0_ec
    -create trk -option "device=cge2,cge3"
```

server_2 : done

EXAMPLE #6  To display all virtual devices on server_2, type:

```
$ server_sysconfig server_2 -virtual
```

server_2 :
Virtual devices:
trk0_ec    devices=cge2 cge3
fsn    failsafe nic devices :
trk    trunking devices : trk0_ec

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Devices</td>
<td>All the configured virtual devices (trunking or fail safe) on the server.</td>
</tr>
<tr>
<td>devices</td>
<td>Lists the virtual or physical device names that in the &lt;virtual device&gt;.</td>
</tr>
<tr>
<td>failsafe nic devices</td>
<td>Name of FSN virtual devices configured on the server.</td>
</tr>
<tr>
<td>trunking devices</td>
<td>Name of trunking virtual devices configured on the server.</td>
</tr>
</tbody>
</table>

EXAMPLE #7  To view information for the Ethernet channel, trk0_ec, type:

```
$ server_sysconfig server_2 -virtual -info trk0_ec
```

server_2 :
*** Trunk trk0_ec: Link is Up ***
*** Trunk trk0_ec: Statistical Load Balancing is IP ***
Device    Link Duplex    Speed
----------------------------------------
cge2      Up    Full        1000 Mbs

cge3      Up    Full        1000 Mbs
The server Commands

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Devices</td>
<td>Name of devices participating in this trunk.</td>
</tr>
<tr>
<td>Link</td>
<td>Status of the link on the specified device (can be up or down).</td>
</tr>
<tr>
<td>Duplex</td>
<td>Value of negotiated duplex. Values can be full or half.</td>
</tr>
<tr>
<td>Speed</td>
<td>Value of negotiated speed.</td>
</tr>
</tbody>
</table>

**EXAMPLE #8** To create a fail-safe network device as a virtual device, type:

```
$ server_sysconfig server_2 -virtual -name fsn0 -create
fsn -option "primary=cge2 device=cge2,cge3"
```

server_2 : done

**EXAMPLE #9** To view the fail-safe network device, type:

```
$ server_sysconfig server_2 -virtual -info fsn0
```

server_2 :

*** FSN fsn0: Link is Up ***
active=cge2 primary=cge2 standby=cge3

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Name of the active device in the FSN.</td>
</tr>
<tr>
<td>primary</td>
<td>Primary device in the FSN.</td>
</tr>
<tr>
<td>standby</td>
<td>Standby device in the FSN.</td>
</tr>
</tbody>
</table>

**EXAMPLE #10** To create an aggregated link using the LACP protocol with load balancing method set to **mac**, type:

```
$ server_sysconfig server_2 -virtual -name trk0_la
-create trk -option "device=cge2,cge3 protocol=lacp
lb=mac"
```

server_2 : done

**EXAMPLE #11** To delete an Ethernet channel, **trk0_ec**, type:

```
$ server_sysconfig server_2 -virtual -delete -Force trk0_ec
```

server_2 : done
server_sysstat

Displays the operating system statistics for the specified Data Movers.

SYNOPSIS

server_sysstat {<movername>|ALL} [-blockmap]

DESCRIPTION

server_sysstat displays the operating system statistics and the status of blockmap memory for the specified Data Movers. The ALL option executes the command for all Data Movers.

OPTIONS

No arguments
Displays all operating system statistics.

[-blockmap]
Displays the status of blockmap memory in the Data Mover. Blockmaps are paged memory used by SnapSure to locate each block of data comprising the checkpoint. Twenty-five percent of physical RAM on the Data Mover is reserved for blockmap memory.

EXAMPLE #1

To display all operating system statistics, type:

$ server_sysstat server_2

server_2 :
threads runnable = 6
threads blocked = 1399
threads I/J/Z = 2
memory free (kB) = 2689924
cpu idle_% = 99

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>threads runnable</td>
<td>Threads running as well as the ones waiting for a cpu to become available.</td>
</tr>
<tr>
<td>threads blocked</td>
<td>Threads waiting on either a condition variable or a Sthread Mutex.</td>
</tr>
<tr>
<td>threads I/J/Z</td>
<td>Threads in the following state: IDLE, JOINED, and ZOMBIE.</td>
</tr>
<tr>
<td>memory free (kB)</td>
<td>The amount of free memory from the RAM that is available to Data Mover.</td>
</tr>
<tr>
<td>cpu idle_%</td>
<td>Idle time of the system averaged across all the processors.</td>
</tr>
</tbody>
</table>

EXAMPLE #2

To display current blockmap memory status, type:

$ server_sysstat server_2 -blockmap

server_2 :
total paged in = 0
total paged out = 1
page in rate = 0
The server Commands

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>page out rate</td>
<td>= 0</td>
</tr>
<tr>
<td>block map memory quota</td>
<td>= 1048576 (KB)</td>
</tr>
<tr>
<td>block map memory consumed</td>
<td>= 624 (KB)</td>
</tr>
</tbody>
</table>

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>total paged in</td>
<td>Total number of blockmap pages paged in since the system booted.</td>
</tr>
<tr>
<td>total paged out</td>
<td>Total number of blockmap pages paged out since the system booted.</td>
</tr>
<tr>
<td>page in rate</td>
<td>Number of blockmap pages paged in per second (over last 180 seconds).</td>
</tr>
<tr>
<td>page out rate</td>
<td>Number of blockmap pages paged out per second (over last 180 seconds).</td>
</tr>
<tr>
<td>block map memory quota</td>
<td>Current value of the blockmap memory quota.</td>
</tr>
<tr>
<td>block map memory consumed</td>
<td>Amount of memory consumed for blockmaps.</td>
</tr>
</tbody>
</table>
server_tftp

Manages the Trivial File Transfer Protocol (TFTP) for the specified Data Movers.

SYNOPSIS

server_tftp {<movername>|ALL}
    -service { -status | -stop | -start | -stats }
    | -info
    | -set [-path <pathname>] [ -readaccess { all | none } ] [ -writeaccess { all | none } ]
    | -clear

DESCRIPTION

server_tftp provides a service to read and write files to and from a remote server, displays the status of the service, and the number of attempted transfers.

The ALL option executes the command for all Data Movers.

OPTIONS

-service { -status | -stop | -start | -stats }
Displays the status of the TFTP service, stops, and starts the service on the specified Data Mover, or displays TFTP statistics.

-info
Displays the number of attempted TFTP transfers and the status information for ongoing transfers.

-set [-path <pathname>]
Sets the directory <pathname> for the TFTP service to use for file transfers. If the -set option has been executed, re-executing the -set option with a different <pathname> overwrites the original <pathname>.

    [ -readaccess { all | none } ]
Sets read access for all users or no users (default).

    [ -writeaccess { all | none } ]
Sets write access for all users or no users (default).

-clear
Clears the path/readaccess/writeaccess settings the TFTP service for the specified Data Mover.

SEE ALSO

Using TFTP on VNX.
EXAMPLE #1  To start TFTP service for server_2, type:

$ server_tftp server_2 -service -start

server_2 : done

EXAMPLE #2  To display the status of the TFTP service for server_2, type:

$ server_tftp server_2 -service -status

server_2 :
Tftp Running

EXAMPLE #3  To modify a path on server_2 for TFTP service with read access for all, and write access for nobody, type:

$ server_tftp server_2 -set -path /ufs1 -readaccess all -writeaccess none

server_2 : done

EXAMPLE #4  To display TFTP information for server_2, type:

$ server_tftp server_2 -info

server_2 :
path="/ufs1/" readaccess=all writeaccess=none

EXAMPLE #5  To display statistics for server_2, type:

$ server_tftp server_2 -service -stats

server_2 :
Attempted Transfers:28
Successful Transfers:27
createdthrds:28
deletedthrds:28
timedoutthrds:0
TotalBinds:28
TotalUnbinds:28
BindFailures:0
InvalidAttempts:0
AttemptedReadTransfers:19
SuccessfulReadTransfers:19
AttemptedWriteTransfers:9
SuccessfulWriteTransfers:8

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attempted Transfers</td>
<td>Total TFTP transfers attempted by that time.</td>
</tr>
<tr>
<td>Successful Transfers</td>
<td>Total number of successful TFTP transfers.</td>
</tr>
<tr>
<td>createdthrds</td>
<td>Total number of TFTP threads created (equal to total transfers).</td>
</tr>
<tr>
<td>deletedthrds</td>
<td>Total number of threads deleted (equal to total created threads).</td>
</tr>
<tr>
<td>timedoutthrds</td>
<td>Number of timed-out threads. For TFTP transfers, in case of any failures, each thread will time out and free itself.</td>
</tr>
</tbody>
</table>
The server Commands

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalBinds</td>
<td>Total number of binds.</td>
</tr>
<tr>
<td>TotalUnbinds</td>
<td>Total number of unbinds.</td>
</tr>
<tr>
<td>BindFailures</td>
<td>Number of bind failures. If the port we try to bind to is in use, the bind fails, and retries with a different port.</td>
</tr>
<tr>
<td>InvalidAttempts</td>
<td>Invalid TFTP transfer requests from clients such as trying to transfer a non-existent file.</td>
</tr>
<tr>
<td>AttemptedReadTransfers</td>
<td>Total TFTP read transfers initiated.</td>
</tr>
<tr>
<td>SuccessfulReadTransfers</td>
<td>Total TFTP read transfers successfully completed.</td>
</tr>
<tr>
<td>AttemptedWriteTransfers</td>
<td>Total TFTP write transfers initiated.</td>
</tr>
<tr>
<td>SuccessfulWriteTransfers</td>
<td>Total TFTP write transfers successfully completed.</td>
</tr>
</tbody>
</table>

**EXAMPLE #6**  
To stop TFTP service on server_2, type:

```
$ server_tftp server_2 -service -stop
```

server_2 : done

**EXAMPLE #7**  
To delete the settings for the TFTP service on server_2, type:

```
$ server_tftp server_2 -clear
```

server_2 : done
server_umount

Unmounts file systems.

SYNOPSIS

server_umount {<movername> | ALL}
[-perm | -temp | -Force] {-all | <fs_name> | <mount_point>}

DESCRIPTION

server_umount permanently or temporarily (default) unmounts a mounted file system by its <mount_point> or <fs_name>, or unmounts all file systems.

The ALL option executes the command for all Data Movers.

OPTIONS

-all | <fs_name> | <mount_point>
Unmounts all file systems, a specific file system, or a file system specified by its mount point.

Note: A mount point always begins with a slash (/).

[-perm]
Permanently deletes the mount entry from mount table.

Note: The export table is not affected by deletes made with server_umount. Export entries remain in the export table regardless of the mount status.

[-temp]
Indicates that the unmount does not persist across reboots. The mount entry reappears after a reboot. Default is -temp.

[-Force]
Forces the unmount of a production file system.

CAUTION

Unmounting a PFS for which automatic checkpoint scheduling (configured using the VNX Web Manager or VNX Native Manager) has been implemented is not recommended. Doing so will corrupt the schedule and cause checkpoint data to be lost.

SEE ALSO

Managing Volumes and File Systems with VNX Automatic Volume Management,
Managing Volumes and File Systems for VNX Manually, Using VNX SnapSure, `server_mount`, `server_mountpoint`, and `fs_ckpt`.

**EXAMPLE #1**
To permanently unmount a file system, type:

```
$ server_umount server_2 -perm /bin
```

server_2: done

**EXAMPLE #2**
To temporarily unmount a file system by specifying its mount point as `/bin`, type:

```
$ server_umount server_2 -temp /bin
```

server_2: done

**EXAMPLE #3**
To temporarily unmount a file system by specifying its file system name as `ufs1`, type:

```
$ server_umount server_2 -temp ufs1
```

server_2: done
server_uptime

Displays the length of time that a specified Data Mover has been running since the last reboot.

SYNOPSIS

server_uptime {<movername>|ALL}

DESCRIPTION

server_uptime displays the length of time that the specified Data Mover has been running since its last reboot.

The ALL option executes the command for all Data Movers.

OPTIONS

No arguments

Displays the length of time that a Data Mover has been up.

EXAMPLE #1

To display how long a Data Mover has been up, type:

$ server_uptime server_2
server_2 : up 28 days 0 hours 15 min 41 secs

EXAMPLE #2

To display how long all the Data Movers have been up, type:

$ server_uptime ALL
server_2 : up 28 days 0 hours 15 min 41 secs
server_3 : up 2 days 5 hours 11 min 31 secs
server_user

Manages user accounts for the specified Data Movers.

SYNOPSIS

server_user {<movername>|ALL}
   -list
   -add [-md5][-passwd] <name>
   -modify <name>
   -delete <name>
   -passwd [-disable|-lock|-unlock [-force]] <name>

DESCRIPTION

server_user creates, modifies, or deletes a user account for a specified Data Mover. The server_user adds or disables, locks or unlocks an optional password for a user account. Data Mover user accounts provide administrative access to certain VNX features and third-party applications.

The ALL option executes the command for all Data Movers.

This command must be executed from the /nas/sbin directory. su to root to execute this command.

OPTIONS

-list
Lists the user accounts.

-add [-md5][-passwd] <name>
Adds a new user account with the login <name>.

The -md5 option allows you to specify MD5 password encryption for the new account.

The -passwd option allows you to specify a password for the new user account. Passwords have a six character minimum and cannot be empty.

-modify <name>
Modifies a user account.

-delete <name>
Deletes a user account.

-passwd [-disable|-lock|-unlock [-force]] <name>
Creates, changes, removes, locks, and unlocks the password for a user. If a password lock is applied, the user account is available to root only. The password does not have to be supplied during the execution of the server_user command. To unlock a password, use the -force option.
SEE ALSO  Using FTP on VNX, and Configuring NDMP Backups to Disk on VNX.

EXAMPLE #1  To create a user account for user1, type:

```
# /nas/sbin/server_user server_2 -add user1
```

Creating new user user1
User ID: 100
Group ID: 101
Comment: 
Home directory: 
Shell: 

Note: Comment, Home directory and Shell are optional, all others are required.

EXAMPLE #2  To create a user account for NDMP connections, with MD5 password encryption and to configure the password, type:

```
# /nas/sbin/server_user server_2 -add -md5 -passwd user_name
```

Creating new user user_name
User ID: 101
Group ID: 100
Home directory: 
Changing password for user user_name
New passwd: 
Retype new passwd: 

EXAMPLE #3  To list the user accounts, type:

```
# /nas/sbin/server_user server_2 -list
```

server_2: 
APM000438070430000_APM000420008180000:LNEa7Fjh/43jQ:9000:9000:ftsQgHsc2oMrdysaOn WeLhN8vB::ndmp_md5 
user1::100:101::: 
user_name:WX72mBTFp/qV.::101:100:W9z7HIndimdaHs2anCL20EBfNd::ndmp_md5 

EXAMPLE #4  To modify account information for user1, type:

```
# /nas/sbin/server_user server_2 -modify user1
```

Modifying user account user1 
1  User ID (100) 
2  Group ID (101) 
3  Home directory () 
4  Comment () 
5  Shell () 

Please select a field to modify, "done" to apply your changes or "quit" to cancel: 2 
Group ID: 102 
Please select a field to modify, "done" to apply your changes or "quit" to cancel: quit
EXAMPLE #5  To lock an account password for ndmp, type:

```
# /nas/sbin/server_user server_2 -passwd -lock user_name
```

Changing password for user user_name
Locking password for user user_name

EXAMPLE #6  To disable the password for user1, type:

```
# /nas/sbin/server_user server_2 -passwd -disable user1
```

Changing password for user user1
Removing password for user user1

EXAMPLE #7  To unlock an account password for user1, type:

```
# /nas/sbin/server_user server_2 -passwd -unlock -force user1
```

Changing password for user user1
Unlocking password for user user1

EXAMPLE #8  To delete a user account for user1, type:

```
# /nas/sbin/server_user server_2 -delete user1
```
server_usermapper

Provides an interface to manage the Internal Usermapper service.

SYNOPSIS

server_usermapper  {<movername>|ALL}
     | -enable [primary=<ip>][config=<path>]
     | -disable
     | -remove -all
     | -Import {-user|-group} [-force] <pathname>
     | -Export {-user|-group} <pathname>

DESCRIPTION

server_usermapper enables, disables, deletes, and displays the status of an Internal Usermapper service running on a Data Mover. Only one primary Usermapper service is recommended for a VNX environment. In a single VNX environment, there should only be one primary instance of the Usermapper service. All the other Data Movers in that environment are clients of the primary or secondary service.

server_usermapper displays whether the Usermapper service is configured as a primary or secondary service. Additionally, it displays information such as the operational status of the service and, if the service is a secondary service, the IP address of the primary Usermapper service used by the secondary.

This command is relevant only for Internal Usermapper, which runs on the Data Mover. It cannot be used to manage External Usermapper, which runs on the Control Station or an external host.

The ALL option executes the command for all Data Movers.

OPTIONS

No arguments
Displays the state of the Internal Usermapper service.

-enable
Enables the Usermapper service. You do not need to issue this option if you are using the default Internal Usermapper configuration. In this case, primary Usermapper is automatically enabled when the NAS software is installed. You only need to issue this option if you are modifying a default Internal Usermapper configuration, or if you are upgrading from External to Internal Usermapper.

Use the -enable command with caution. It changes the relationship between the Data Mover and the Usermapper without confirming the change.
[primary=<ip_address>] [config=<pathname>]

The **primary** option designates the local Usermapper service as secondary by indicating which primary service it depends on. The primary Usermapper is identified by its network IP address.

The **config** option indicates that the Usermapper service should use an existing Usermapper configuration file to define UID/GID ranges. This option is only relevant if you are upgrading from External to Internal Usermapper.

---

**Note:** If there is no specific reason to use particular UID and GID ranges for your environments’ domains, EMC encourages you to use the automatic mapping method and let Internal Usermapper automatically assign new UIDs/GIDs based on the next available values. If you need to use an existing Usermapper configuration file, you must specify the **config** option during the upgrade procedure, that is, before Internal Usermapper has begun issuing default UIDs and GIDs.

---

**-disable**
Disables the Usermapper service.

**-remove -all**
Removes all entries from the Usermapper databases and destroys the database structure. The Usermapper service must be disabled before you can issue this option.

---

**CAUTION**

It is recommended that you consult with Customer Support before issuing the **-remove -all** option. This option deletes all Usermapper database entries and may result in users losing access to file systems. If you decide to use the **-remove -all** option, you should first back up your existing Usermapper database file and **usrmap.cfg** file (if one is in use).

---

**-Import** {-user | -group} [-force] <pathname>

Imports Usermapper database information from the file specified by <pathname>.

By default, only new entries are added to the Usermapper database. If an entry in the imported file does not match a similar entry in the existing database, the entry in the imported file is ignored unless the **-force** option is selected. If the **-force** option is selected, the existing database is deleted and replaced with new entries.
CAUTION

It is recommended that you consult with Customer Support before issuing the -force option. This option overwrites the existing Usermapper database file and may result in users losing access to file systems. If you decide to use the -force option, you should first back up your existing Usermapper database file and usormap.cfg file (if one is in use).

-Export { -user | -group } <pathname>
Exports all the SID, user, and group information from the Usermapper database to the file specified by <pathname>. You can specify any filename but the name should include the suffix .passwd or .group depending on the file type.

The -Export option is relevant only for a primary Usermapper service.

SEE ALSO  Using nttxmap for CIFS User Mapping on VNX.

EXAMPLE #1  To verify the status of Internal Usermapper, type:

```bash
$ server_usermapper server_2
```

server_2 : Usrmapper service: Enabled
Service Class: Primary

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usrmapper service:</td>
<td>The operational status of the service.</td>
</tr>
<tr>
<td>Service Class:</td>
<td>If the service is a primary or secondary service.</td>
</tr>
<tr>
<td>Primary</td>
<td>The IP address of the primary Usermapper service used by a secondary service. The (c) against the IP address indicates that the primary Usermapper is available and has been connected.</td>
</tr>
</tbody>
</table>

EXAMPLE #2  To enable a secondary Usermapper service, type:

```bash
$ server_usermapper server_4 -enable
primary=172.24.102.238
```

server_4 : done

EXAMPLE #3  To verify the status of Internal Usermapper for the primary Usermapper, type:

```bash
$ server_usermapper server_2
```

server_2 : Usrmapper service: Enabled
Service Class: Primary
EXAMPLE #4  To verify the status of Internal Usermapper for the secondary Usermapper, type:

$ server_usermapper server_4

server_4 : Usrmapper service: Enabled
Service Class: Secondary
Primary = 172.24.102.238 (c)

EXAMPLE #5  To export user information from the Usermapper database, type:

$ server_usermapper server_2 -Export -user
/home/nasadmin/users_server_2.passwd

server_2 : done

EXAMPLE #6  To export group information from the Usermapper database, type:

$ server_usermapper server_2 -Export -group
/home/nasadmin/group_server_2.group

server_2 : done

EXAMPLE #7  To import the user file users_server_2.passwd for server_2, type:

$ server_usermapper server_2 -Import -user
/home/nasadmin/users_server_2.passwd

server_2 : done

EXAMPLE #8  To import the group file group_server_2.group for server_2, type:

$ server_usermapper server_2 -Import -group
/home/nasadmin/group_server_2.group

server_2 : done

EXAMPLE #9  To disable an Internal Usermapper service, type:

$ server_usermapper server_2 -disable

server_2 : done

EXAMPLE #10  To remove all entries from the Usermapper database, type:

$ server_usermapper server_2 -remove -all

server_2 : Warning: This operation will erase all user/group mappings. CIFS users may lose access.
Continue (Y/N):
done
**server_version**

Displays the software version running on the specified Data Movers.

**SYNOPSIS**

```
server_version {<movername>|ALL}
```

**DESCRIPTION**

`server_version` displays the version of the software for the specified Data Mover.

The `ALL` option executes the command for all Data Movers.

**OPTIONS**

No arguments

Displays the software version.

**SEE ALSO**

`nas_version`.

**EXAMPLE #1**

To display the software version for a Data Mover, type:

```
$ server_version server_2
```

```
server_2 : Product: EMC Celerra File Server Version: T5.5.80.0
```

**EXAMPLE #2**

To display the system software version for all Data Movers, type:

```
$ server_version ALL
```

```
server_2 : Product: EMC Celerra File Server Version: T5.5.80.0
server_3 : Product: EMC Celerra File Server Version: T5.5.80.0
server_4 : Product: EMC Celerra File Server Version: T5.5.80.0
server_5 : Product: EMC Celerra File Server Version: T5.5.80.0
```
**server_viruschk**  
Manages the virus checker configuration for the specified Data Movers.

**SYNOPSIS**

```
server_viruschk {<movername>|ALL} [-audit|-update] [-set accesstime={now|none|[<yy><mm><dd><hh>]<mm>[.<ss>]}}] [-fsscan [<fs_name> {-list -create [offline]|-delete}]]
```

**DESCRIPTION**

`server_viruschk` displays, audits, and updates the virus checker configuration file for the specified Data Mover.

The virus checker configuration file is named viruschecker.conf and is located on the Control Station in the /nas/sys directory. This is a template file and should not be edited directly. Copy the viruschecker.conf file to another directory for edits using a text editor, then once the changes have been saved, the file must be copied to a Data Mover using the `server_file` command.

To start and stop the virus checker for a Data Mover, `server_setup` provides information.

The **ALL** option executes the command for all Data Movers.

**OPTIONS**

- **No arguments**
  Displays the virus checker configuration.

- **-audit**
  Displays the status of virus checker, including the number of files checked, and their progress.

- **-update**
  Integrates the changes made to the virus configuration file without stopping the virus checker.

- **-set accesstime={now|none|[<yy><mm><dd><hh>]<mm>[.<ss>]}}**
  Sets the reference time on the virus checker configuration file to **now**, disables the time scan, or sets the access time in the specified format.

- **-fsscan**
  Displays the file system scan status for all file systems.
The server Commands

[<fs_name> {-list -create [offline] -delete}]]

The -list option displays the scan status for the specified file system. The -create option initializes a full scan on the file system <fs_name> and the offline options allow the file system scan on all offline files. By default, offline file systems are not included. The -delete option stops the scan.

SEE ALSO Using Common AntiVirus Agent, server_checkup, and server_setup.

EXAMPLE #1 To display the virus checker configuration, type:

$ server_viruschk server_2

server_2 :
10 threads started
1 Checker IP Address(es):
172.24.102.18 ONLINE at Mon Jan 31 18:35:43 2005 (GMT-00:00)
RPC program version: 3
CAVA release: 3.3.5, AV Engine: Network Associates
Last time signature updated: Thu Jan 27 19:38:35 2005 (GMT-00:00)
31 File Mask(s):
*.zip *.tar *.arj *.arc *.z
No File excluded
Share \DM112-CGE0\CHECK$
RPC request timeout=25000 milliseconds
RPC retry timeout=5000 milliseconds
High water mark=200
Low water mark=50
Scan all virus checkers every 60 seconds
When all virus checkers are offline:
Continue to work with Virus Checking and CIFS
Scan on read if access Time less than Thu Jan 27 19:38:35 2005 (GMT-00:00)
Panic handler registered for 65 chunks

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>threads started</td>
<td>The number of threads that have been started.</td>
</tr>
<tr>
<td>Checker IP Address(es)</td>
<td>The number of VC servers defined in /etc/viruschecker.conf.</td>
</tr>
<tr>
<td>version=2</td>
<td>CAVA uses RPC program version 2.</td>
</tr>
<tr>
<td>Share</td>
<td>The UNC name used by CAVA to access the Data Mover.</td>
</tr>
<tr>
<td>RPC request timeout=</td>
<td>Time out for the full CAVA request.</td>
</tr>
<tr>
<td>RPC retry timeout=</td>
<td>Time out for one unitary CAVA request.</td>
</tr>
<tr>
<td>High water mark=</td>
<td>A log event is generated when the number of files in the request queue becomes greater than 200.</td>
</tr>
<tr>
<td>Low water mark=</td>
<td>A log event is generated when the number of files in the request queue become less than 50.</td>
</tr>
</tbody>
</table>
EXAMPLE #2  
To display the status of the virus checker, type:

```
$ server_viruschk server_2 -audit
```

```
server_2:
Total Requests : 138
Requests in progress : 25

NO ANSWER from the Virus Checker Servers: 0
ERROR_SETUP : 0
FILE_NOT_FOUND : 0
ACCESS_DENIED : 0
FAIL : 0
TIMEOUT : 0
Total Infected Files : 875
Deleted Infected Files : 64
Renamed Infected Files : 0
Modified Infected Files : 811
min=70915 uS, max=1164891 uS, average=439708 uS

15 File(s) in the collector queue
10 File(s) processed by the AV threads
Read file '/.etc/viruschecker.audit' to display the list of pending requests
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Infected Files</td>
<td>The number of files found that contained viruses. This displays only if infected files are found and remains visible until the Data Mover is rebooted or the CAVA viruschecking service has been restarted.</td>
</tr>
<tr>
<td>Deleted Infected Files</td>
<td>The number of files that contained viruses that were deleted. This displays only if infected files are found and remains visible until the Data Mover is rebooted or the CAVA viruschecking service has been restarted.</td>
</tr>
<tr>
<td>Renamed Infected Files</td>
<td>The number of files that contained viruses that were renamed. This displays only if infected files are found and remains visible until the Data Mover is rebooted or the CAVA viruschecking service has been restarted.</td>
</tr>
<tr>
<td>Modified Infected Files</td>
<td>The number of files that contained viruses that were modified. This displays only if infected files are found and remains visible until the Data Mover is rebooted or the CAVA viruschecking service has been restarted.</td>
</tr>
</tbody>
</table>
The server Commands

**EXAMPLE #3**  To update the virus checker configuration file that is resident on the Data Mover, type:

```
$ server_viruschk server_2 -update
```

server_2 : done

**EXAMPLE #4**  To set the access time for the virus checker configuration file, type:

```
$ server_viruschk server_2 -set accesstime=now
```

server_2 : done

**EXAMPLE #5**  To start a scan on the file system, type:

```
$ server_viruschk server_2 -fsscan ufs1 -create
```

server_2 : done

**EXAMPLE #6**  To check the scan of a file system, type:

```
$ server_viruschk server_2 -fsscan ufs1 -list
```

server_2 :
FileSystem 24 mounted on /ufs1:
  8 dirs scanned and 22 files submitted to the scan engine
  firstFNN=0x0, lastFNN=0xe0f34b70, queueCount=0, burst=10

**EXAMPLE #7**  To check the scan status on all file systems, type:

```
$ server_viruschk server_2 -fsscan
```

server_2 :
FileSystem 24 mounted on /ufs1:
  8 dirs scanned and 22 files submitted to the scan engine
  firstFNN=0x0, lastFNN=0xe0f34b70, queueCount=0, burst=10
FileSystem 25 mounted on /ufs2:
  9 dirs scanned and 11 files submitted to the scan engine
  firstFNN=0x0, lastFNN=0xe0010b70, queueCount=0, burst=10

**EXAMPLE #8**  To stop a scan on a file system, type:

```
$ server_viruschk server_2 -fsscan ufs1 -delete
```

server_2 : done
server_vtlu

SYNOPSIS  Configures a virtual tape library unit (VTLU) on the specified Data Movers.

server_vtlu {<movername>|ALL}
  -service <service_options>
  -tlu <tlu_options>
  -storage <storage_options>
  -tape <tape_options>
  -drive <drive_options>
  -help

DESCRIPTION  server_vtlu creates, configures, and displays information on virtual tape library units on the specified Data Movers. At this time, server_vtlu supports only NDMP backups.

To get detailed options for VTLU service, TLU, storage, tape, and drive, type -help for the full usage, or type only -service, -tlu, -tape, and -drive, respectively, after typing the <movername>.

ALL executes the command for all Data Movers.

SERVICE OPTIONS  

SERVICE OPTIONS

 -service { -info | -set -chain <chain_id> }  The -info option lists the chain IDs for the VTLU service on the Data Mover. The -set -chain option specifies a <chain_id> for the VTLU device connection. The default chain ID is 0.

TLU OPTIONS

-tlu
  -list
  -info <tlu_id>
  -new [-robot { [-vendor <vendor>] [-product <product>]  [-revision <revision>]}]
  [-slots <num_slots>]
  [-impexp <num_impexp_slots>] [-drives <num_drives>]
  [-drive { [-vendor <vendor>] [-product <product>]  [-revision <revision>]}]
  -modify <tlu_id>
  { [-robot { [-vendor <vendor>] [-product <product>] [-revision <revision>]}]
  [-slots <num_slots>] [-impexp <num_impexp_slots>] [-drives <num_drives>]
  [-drive { [-vendor <vendor>] [-product <product>]  [-revision <revision>]}]
  -delete <tlu_id>}

-tlu -list  
Lists all VTLUs configured on the specified Data Movers.
-tlu -info <tlu_id>
Displays information for the specified VTLU.

-tlu -new [-robot {[-vendor <vendor>] [-product <product>] [-revision <revision>]}]
Creates a VTLU on the specified Data Mover with the specified robot's vendor, product, and revision information. If this information is not specified, the default values of EMC, svtlRobot, and 1.0 are used. These values have character limits of 8, 16, and 4, respectively.

[-slots <num_slots>]
Configures the number of slots in the VTLU. If no value is defined, then the default value of 32 is used.

[-impexp <num_impexp_slots>]
Configures the number of import/export slots in the VTLU. If no value is defined, the default value of 8 is used.

[-drives <num_drives>]
Configures the number of virtual tape drives in the VTLU. If no value is defined, then the default value of 4 is used.

[-drive {[-vendor <vendor>] [-product <product>] [-revision <revision>]}]
Specifies the vendor name, product name, and revision number of the virtual drive. If no values are defined, the default values of EMC, svtlDrive, and 1.0 are used. These values have character limits of 8, 16, and 4 respectively.

-tlu -modify <tlu_id> [-robot {[-vendor <vendor>] [-product <product>] [-revision <revision>]}]
Modifies the vendor name, product name, and revision number for the virtual robot.

[-slots <num_slots>]
Modifies the number of virtual slots in the VTLU.

[-impexp <num_impexp_slots>]
Modifies the number of virtual import/export slots in the VTLU.

[-drives <num_drives>]
Modifies the number of virtual tape drives in the VTLU.

[-drive {[-vendor <vendor>] [-product <product>] [-revision <revision>]}]
Modifies the vendor name, product name, and revision number for the VTLU drive.
-tlu -delete <tlu_id>
Deletes a configured VTLU. The storage associated with a VTLU must be deleted before the VTLU can be deleted.

STORAGE OPTIONS

-storage
{list [<tlu_id>]
| -new <fs_name> -tlu <tlu_id> [-tapesize <size>G][-tapes <num_tapes>]
| [-barcodeprefix <barcode_prefix>]
| [-destination {slot|impexp|vault}]
| -extend <fs_name> -tlu <tlu_id> [-tapesize <size>G][-tapes <num_tapes>]
| [-destination {slot|impexp|vault}]
| -import <fs_name> -tlu <tlu_id>
| [-destination {slot|impexp|vault}]
| -export <fs_name> -tlu <tlu_id>
| -delete {<fsname>|-barcodeprefix <barcode_prefix>} -tlu <tlu_id>}

-storage -list [<tlu_id>]
Lists storage information on the specified VTLU.

-storage -new <fs_name> -tlu <tlu_id>
Creates storage on an existing VTLU.

[-tapesize <size>G]
Specifies the size of the virtual tapes in the file system. If no value is defined, and the -tape <num_tapes> option is not defined, then the default value of 50 GB is used. If <num_tapes> is defined but the tape size is not, then the tape size is determined by the number of tapes and the file system size.

[-tapes <num_tapes>]
Specifies the number of virtual tapes in the file system. The default value is determined by the file system size and the value of -tapesize <size>.

[-barcodeprefix <barcode_prefix>]
Specifies the barcode prefix for the virtual tapes. The barcode prefix is the same across a file system. If no value is defined, then a four-character string is assigned automatically.

[-destination {slot|impexp|vault}]
Specifies the destination of the virtual tape as a slot, an import/export slot, or the vault. Slot is the default location.

-storage -extend <fs_name> -tlu <tlu_id>
Adds more tapes to a file system that has been extended.
The server Commands

[-tapesize <size>G]
Specifies the size of the virtual tape. If no value is defined, and the
-tapes <num_tapes> is not defined, then the default value of 50
GB is used. If the <num_tapes> is defined but <size> is not, then
the tape size is determined by the number of tapes and the file
system size.

[-tapes <num_tapes>]
Specifies the number of tapes to be added to a VTLU. The default
value is determined by the size of the file system and the tape
size.

[-destination {slot|impexp|vault}]
Specifies the destination of the virtual tape as a slot, an
import/export slot, or the vault. Slot is the default location.

-storage -import <fs_name> -tlu <tlu_id>
Imports a file system that was exported from another VTLU. The
configuration of a file system persists when it is imported.

[-destination {slot|impexp|vault}]
Specifies the destination of the imported storage as a slot, an
import/export slot, or the vault. Slot is the default location.

-storage -export <fs_name> -tlu <tlu_id>
Exports storage from a VTLU, which breaks the association between
the file system and the VTLU. Tapes must be ejected from the VTLU
and placed in the vault before you can export storage.

-storage -delete {<fs_name>|-barcodeprefix
<barcode_prefix>} -tlu <tlu_id>
Deletes the association between storage and the VTLU and cleans up
related metadata.

TAPE OPTIONS

-tape
{-list <tlu_id> [-storage <fs_name>]
| -info {<tape_barcode>|-all} -tlu <tlu_id>
| -insert <tape_barcode> -tlu <tlu_id>
| -eject <tape_barcode> -tlu <tlu_id>}
| -drive {-list <drive_id>
| | -info <drive_id> -tlu <tlu_id>
| | -unmount <drive_id> -tlu <tlu_id>}

-tape {-list <tlu_id> [-storage <fs_name>]}]
Lists the tapes configured on a VTLU, along with their associated file
systems.
-tape -info <tape_barcode|-all> -tlu <tlu_id>
Lists information on a particular tape, or on all tapes in a VTLU.

-tape -insert <tape_barcode> -tlu <tlu_id>
Moves a virtual tape from the vault to an import/export slot.

-tape -eject <tape_barcode> -tlu <tlu_id>
Ejects a tape from the import/export slot and places it in the vault.

-drive {-list <tlu_id>}
Lists the virtual tape drives associated with the specified VTLU.

-info <drive_id> -tlu <tlu_id> -id
Displays information on the specified tape drive.

-unmount <drive_id> -tlu <tlu_id>
Unmounts the tape from the drive and places it in its original source slot. If the source slot is unavailable, the tape is placed in another available slot.

**SEE ALSO**
Configuring NDMP Backups to Disk on VNX, Parameters Guide for VNX for File, and nas_fs.

**EXAMPLE #1**
To set the chain_ID for server_2, type:

$ server_vtlu server_2 -service -set -chain 1

server_2 : done

**EXAMPLE #2**
To list the chain_ID for the VTLU service on server_2, type:

$ server_vtlu server_2 -service -info

server_2 :
starting chain id = 1

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>starting chain id</td>
<td>Starting chain of the VTLU device name. The device name format is c&lt;x&gt;t&lt;y&gt;l&lt;z&gt;, where &lt;x&gt; is the starting chain. &lt;y&gt; and &lt;z&gt; are the target and LUN, respectively. The target and LUN values cannot be modified.</td>
</tr>
</tbody>
</table>

**EXAMPLE #3**
To create and configure a new VTLU on server_2, type:

$ server_vtlu server_2 -tlu -new -robot -vendor EMCCorp -product vtluRobot -revision 1.1a -slots 256 -impexp 64 -drives 2 -drive -vendor EMCCorp -product vtluDrive -revision 2.2a

server_2 :
id = 3
slots = 256
import/export slots = 64
robot vendor = EMCCorp
robot product           = vtluRobot
robot revision          = 1.1a
robot serial number     = P8gIgqs2k5
robot device name       = c1t010
drives                  = 2
drive vendor            = EMCCorp
drive product           = vtluDrive
drive revision          = 2.2a

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Unique VTLU identifier that is assigned automatically.</td>
</tr>
<tr>
<td>slots</td>
<td>Number of virtual slots in the VTLU.</td>
</tr>
<tr>
<td>import/export slots</td>
<td>Number of virtual import/export slots in the VTLU.</td>
</tr>
<tr>
<td>robot vendor</td>
<td>Vendor name of the virtual robot; maximum length is eight characters.</td>
</tr>
<tr>
<td>robot product</td>
<td>Product name of the virtual robot; maximum length is 16 characters.</td>
</tr>
<tr>
<td>robot revision</td>
<td>Revision number of the virtual robot; maximum length is four characters.</td>
</tr>
<tr>
<td>robot serial number</td>
<td>Serial number of the virtual robot that is assigned automatically.</td>
</tr>
<tr>
<td>robot device name</td>
<td>Device name of the virtual robot; only the first number, the starting chain, can be modified.</td>
</tr>
<tr>
<td>drives</td>
<td>Number of virtual drives in the VTLU.</td>
</tr>
<tr>
<td>drive vendor</td>
<td>Vendor name of the virtual drive; maximum length is eight characters.</td>
</tr>
<tr>
<td>drive product</td>
<td>Product name of the virtual drive; maximum length is 16 characters.</td>
</tr>
<tr>
<td>drive revision</td>
<td>Revision number of the virtual drive; maximum length is four characters.</td>
</tr>
</tbody>
</table>

**EXAMPLE #4**  To list all the VTLUs on server_2, type:

```
$ server_vtlu server_2 -tlu -list
```

server_2 :
<table>
<thead>
<tr>
<th>id</th>
<th>vendor</th>
<th>product</th>
<th>revision</th>
<th>serial_number</th>
<th>device_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>EMCCorp</td>
<td>vtluRobot</td>
<td>1.1a</td>
<td>P8gIgqs2k5</td>
<td>c1t010</td>
</tr>
</tbody>
</table>

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Unique VTLU identifier that is assigned automatically.</td>
</tr>
<tr>
<td>vendor</td>
<td>Vendor name of the virtual robot; maximum length is eight characters.</td>
</tr>
<tr>
<td>product</td>
<td>Product name of the virtual robot; maximum length is 16 characters.</td>
</tr>
<tr>
<td>revision</td>
<td>VTLU robot's revision number; maximum length is four characters.</td>
</tr>
<tr>
<td>serial_number</td>
<td>VTLU serial number that is assigned automatically.</td>
</tr>
<tr>
<td>device_name</td>
<td>The device name of the VTLU robot; only the first number, the starting chain, can be modified.</td>
</tr>
</tbody>
</table>

**EXAMPLE #5**  To display the information for the VTLU on the Data Mover identified by its ID, type:

```
$ server_vtlu server_2 -tlu -info 3
```

server_2 :
<table>
<thead>
<tr>
<th>id</th>
<th>= 3</th>
</tr>
</thead>
</table>
slots                   = 256
import/export slots    = 64
robot vendor           = EMCCorp
robot product          = vtluRobot
robot revision         = 1.1a
robot serial number    = P8gIgqs2k5
robot device name      = c1t010
drives                 = 2
drive vendor           = EMCCorp
drive product          = vtluDrive
drive revision         = 2.2a

**EXAMPLE #6** To modify vendor, product, and revision information for the robot and drive of VTLU 3 for server_2, type:

```
$ server_vtlu server_2 -tlu -modify 3 -robot -vendor EMC -product vRobot -revision 1.1b -drives 3 -drive -vendor EMC -product vDrive -revision 2.2b
```

server_2 : done

**EXAMPLE #7** To modify the number of virtual import/export slots and number of virtual slots of VTLU 1 for server_2, type:

```
$ server_vtlu server_2 -tlu -modify 3 -slots 8 -impexp 4
```

server_2 : done

**EXAMPLE #8** To add new storage for VTLU 3 on server_2, with 5 virtual tapes of 1 GB located in slots each with barcode prefix dstpre, using ufs1 file system, type:

```
$ server_vtlu server_2 -storage -new ufs1 -tlu 3 -tape size 1G -tapes 5 -barcodeprefix dstpre -destination slot
```

server_2 : done

**EXAMPLE #9** To extend VTLU 3 on server_2 by adding 2 virtual tapes of 1 GB and placing them in the import/export virtual slots, type:

```
$ server_vtlu server_2 -storage -extend ufs1 -tlu 3 -tapesize 1G -tapes 2 -destination impexp
```

server_2 : done

**EXAMPLE #10** To export storage from VTLU 3 stored on ufs1 located on server_2, type:

```
$ server_vtlu server_2 -storage -export ufs1 -tlu 3
```

server_2 : done

**EXAMPLE #11** To import the ufs2 file system to VTLU 3 and place the virtual tapes in the vault, type:

```
$ server_vtlu server_2 -storage -import ufs1 -tlu 3 -destination vault
```
The server Commands

server_2 : done

**EXAMPLE #12** To list the storage on VTLU 3, type:

```
$ server_vtlu server_2 -storage -list 3
```

tlu_id   filesystem        barcode_prefix
3        ufs1              dstpre

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>tlu_id</td>
<td>Unique VTLU identifier that is assigned automatically.</td>
</tr>
<tr>
<td>filesystem</td>
<td>Name of the file system associated with the VTLU.</td>
</tr>
<tr>
<td>barcode_prefix</td>
<td>Modifiable prefix assigned to virtual tapes that is constant across a file system.</td>
</tr>
</tbody>
</table>

**EXAMPLE #13** To list VTLU information on VTLU 3, type:

```
$ server_vtlu server_2 -tape -list 3
```

barcode       filesystem     capacity(GB)   location     source_slot
dstpre0001    ufs1           1              vault
dstpre0002    ufs1           1              vault
dstpre0003    ufs1           1              vault
dstpre0004    ufs1           1              vault
dstpre0005    ufs1           1              vault
dstpre0006    ufs1           1              vault
dstpre0000    ufs1           1              impexp:0

**Where:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>barcode</td>
<td>Virtual tape barcode, consisting of the modifiable barcode prefix and a four-digit number that is assigned automatically.</td>
</tr>
<tr>
<td>filesystem</td>
<td>Name of the file system.</td>
</tr>
<tr>
<td>capacity (GB)</td>
<td>Virtual tape capacity in GB.</td>
</tr>
<tr>
<td>location</td>
<td>Element type and element ID of the virtual tape; possible element types are slot, drive, import/export, robot, and vault.</td>
</tr>
<tr>
<td>source_slot</td>
<td>Slot ID of the tape's previous location.</td>
</tr>
</tbody>
</table>

**EXAMPLE #14** To insert the specified tape in a virtual import/export slot on VTLU 3, type:

```
$ server_vtlu server_2 -tape -insert dstpre0001 -tlu 3
```

server_2 : done

**EXAMPLE #15** To eject the specified tape from VTLU 3, type:

```
$ server_vtlu server_2 -tape -eject dstpre0001 -tlu 3
```

server_2 : done
EXAMPLE #16  To list the storage drive on VTLU 3, type:

```
$ server_vtlu server_2 -drive -list 3
```

```
sn

server_2 :
drive_id  device_name   serial_number  status    tape_barcode
0         c1t011        NXB2w4W000     empty
1         c1t012        3u0bx4W000     empty
2         c1t013        g0pgy4W000     empty
```

Where:

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>drive_id</td>
<td>Unique VTLU drive identifier that is assigned automatically.</td>
</tr>
<tr>
<td>device_name</td>
<td>The device name of the VTLU drive.</td>
</tr>
<tr>
<td>serial_number</td>
<td>The VTLU serial number that is automatically assigned.</td>
</tr>
<tr>
<td>status</td>
<td>Status of the virtual tape drive; possible values are empty, loaded, and in use.</td>
</tr>
<tr>
<td>tape_barcode</td>
<td>Barcode of the virtual tape if status is not empty.</td>
</tr>
</tbody>
</table>

EXAMPLE #17  To display information for drive 0 on VTLU 3, type:

```
$ server_vtlu server_2 -drive -info 0 -tlu 3
```

```
sn

server_2 :
id             = 0
device_name    = c1t011
serial_number  = NXB2w4W000
status         = empty
tape_barcode   =
```

EXAMPLE #18  To delete storage from VTLU 3, type:

```
$ server_vtlu server_2 -storage -delete ufs1 -tlu 3
```

```
sn

server_2 : done
```

EXAMPLE #19  To delete VTLU 3 from server_2, type:

```
$ server_vtlu server_2 -tlu -delete 3
```

```
sn

server_2 : done
```
This appendix contains information about command scripts that can be used to automate many VNX tasks.
Scripting guidelines

Table 2 on page 734 lists guidelines to keep in mind when creating VNX command scripts.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Action</th>
</tr>
</thead>
</table>
| Scheduling             | **NAS Database Backups:**  
The VNX backs up the NAS database that stores specific configuration information required for each Data Mover every hour, at one minute after the hour. During part of the backup, the database is locked, and some commands that rely on the database might not have access. It is recommended that command scripts avoid starting at one minute after the hour. Note that scripts with complex commands that run for an extended period may overlap the backup period.  
The duration of the backup may vary. Use the following Linux command to check the state of the backup process prior to executing scripts: `ps -ef | grep nasdb_backup`. If a lock condition occurs, wait a few minutes and retry. |
| Command sequencing      | Some commands must lock the database in order to execute. If multiple user-entered commands or scripts are active at the same time, some of these commands may lock the database and prevent other commands from executing. To avoid this, you should arrange commands whenever possible. |
| Sleep statements        | Some processes within a script can take time to complete. Use proper timing and adequate sleep statements to prevent timing-related issues. |
| Pipe and grep           | Piping script outputs through `grep` is a helpful tool to check the status of the script. Use periodic checks to `grep` for file or database locked messages, timeouts, resource unavailable warnings, and other failure or success messages, and use this information to check status, pause the script, or halt it. Detailed information about error messages can be found in the error messages guide. |
| Return code check       | All commands return a UNIX-style return code (for example: 0 for success or 1 for failure) or a text-based status code (for example, done) which can be used to help determine if the command completed or if there was an error or a conflict with the NAS database backup, or other commands being run. If a lock condition occurs, wait a few minutes and retry. If you create and run scripts, be sure to incorporate return code checks and verify for proper return codes from individual operations. |
The RECOVERABLE variable contains the following errors to retry on:
- Unable to acquire lock
- Resource temporarily unavailable
- Unable to connect to host
- Socket: All ports in use
- Database resource vanished
- Connection timed out
- NAS_DB locked object is stale

An example of what the RECOVERABLE variable looks like is as follows:

```
RECOVERABLE="unable to acquire lock|Resource temporarily unavailable|unable to connect to host|socket: All ports in use|database resource vanished|Connection timed out|NAS_DB locked object is stale".
```

The res variable contains the command output:

```
#!/bin/sh
########################################################
# File: nas_cmdrcvr.sh
# Created by: NAS Engineering
# Date: Thursday, May 25, 2006
# Version: 1.0
# Notes:
# 1) script will retry commands for specified period of time
#2) script will log messages to file only if there's available disk space
########################################################
NAS_DB=/nas
export NAS_DB
PATH=$PATH:$NAS_DB/bin:$NAS_DB/sbin:/usr/sbin:.
export PATH
RETRIES=60
SLEEPTIME=30
LOGDIR="/home/nasadmin"
LOGFILE="$0.log"
LOGLAST="$0.last"
DISKFULL=98
RECOVERABLE="Resource temporarily unavailable\\
unable to acquire lock\\
```
unable to connect to host
socket: All ports in use
database resource vanished
Connection timed out
NAS_DB locked object is stale

# function to log messages to a file
nas_log()
{
    DISKCHK=`df -k $LOGDIR|awk 'NR>1{print $5}'|sed 's/\%/\'/`
    # if there's enough free disk space, append to log
    if [ $DISKCHK -lt $DISKFULL ]; then
        TDSTAMP=`date +%Y-%m-%d' '%T`
        echo $TDSTAMP: $LOGMSG >> $LOGDIR/$LOGFILE
    fi
    # regardless of available space, always write last error
    echo $TDSTAMP: $LOGMSG > $LOGDIR/$LOGLAST
}

# function to execute (and potentially retry) commands
nas_cmd()
{
    # initialize variable(s)
    retry_count=0
    # loop until either successful of retry count exceeded
    while [ $retry_count -le $RETRIES ]; do
        # execute command and gather response
        RES=`$CMD 2>&1`
        # check if response means command is recoverable
        if [ `echo "$RES"|egrep -c "$RECOVERABLE"` -ne 0 ]; then
            # check retry count
            if [ $retry_count -ne $RETRIES ]; then
                # retry count has not been exceeded
                LOGMSG="Command ($CMD) failed with ($RES)...retrying in $SLEEPTIME s"
                nas_log
                sleep $SLEEPTIME
            else
                # retry count has been exceeded
            fi
        else
            # retry count has been exceeded
        fi
    done
}

# function to retry commands
nas_retry()
{
    # initialize variable(s)
    retry_count=0
    # loop until either successful or retry count exceeded
    while [ $retry_count -le $RETRIES ]; do
        # execute command and gather response
        RES=`$CMD 2>&1`
        # check if response means command is recoverable
        if [ `echo "$RES"|egrep -c "$RECOVERABLE"` -ne 0 ]; then
            # check retry count
            if [ $retry_count -ne $RETRIES ]; then
                # retry count has not been exceeded
                LOGMSG="Command ($CMD) failed with ($RES)...retrying in $SLEEPTIME s"
                nas_log
                sleep $SLEEPTIME
            else
                # retry count has been exceeded
            fi
        else
            # retry count has been exceeded
        fi
    done
}

# function to retry commands with a delay
nas_retry_delay()
{
    # initialize variable(s)
    retry_count=0
    # loop until either successful or retry count exceeded
    while [ $retry_count -le $RETRIES ]; do
        # execute command and gather response
        RES=`$CMD 2>&1`
        # check if response means command is recoverable
        if [ `echo "$RES"|egrep -c "$RECOVERABLE"` -ne 0 ]; then
            # check retry count
            if [ $retry_count -ne $RETRIES ]; then
                # retry count has not been exceeded
                LOGMSG="Command ($CMD) failed with ($RES)...retrying in $SLEEPTIME s"
                nas_log
                sleep $SLEEPTIME
            else
                # retry count has been exceeded
            fi
        else
            # retry count has been exceeded
        fi
    done
}

# function to retry commands with a delay
nas_retry_delay()
{
    # initialize variable(s)
    retry_count=0
    # loop until either successful or retry count exceeded
    while [ $retry_count -le $RETRIES ]; do
        # execute command and gather response
        RES=`$CMD 2>&1`
        # check if response means command is recoverable
        if [ `echo "$RES"|egrep -c "$RECOVERABLE"` -ne 0 ]; then
            # check retry count
            if [ $retry_count -ne $RETRIES ]; then
                # retry count has not been exceeded
                LOGMSG="Command ($CMD) failed with ($RES)...retrying in $SLEEPTIME s"
                nas_log
                sleep $SLEEPTIME
            else
                # retry count has been exceeded
            fi
        else
            # retry count has been exceeded
        fi
    done
}

# function to retry commands with a delay
nas_retry_delay()
{
    # initialize variable(s)
    retry_count=0
    # loop until either successful or retry count exceeded
    while [ $retry_count -le $RETRIES ]; do
        # execute command and gather response
        RES=`$CMD 2>&1`
        # check if response means command is recoverable
        if [ `echo "$RES"|egrep -c "$RECOVERABLE"` -ne 0 ]; then
            # check retry count
            if [ $retry_count -ne $RETRIES ]; then
                # retry count has not been exceeded
                LOGMSG="Command ($CMD) failed with ($RES)...retrying in $SLEEPTIME s"
                nas_log
                sleep $SLEEPTIME
            else
                # retry count has been exceeded
            fi
        else
            # retry count has been exceeded
        fi
    done
}
EMC has partially changed the layout or format of VNX internal databases. This change can impact the use of awk or grep utilities when used in scripts that assume specific positions of fields in databases.

To enable searching of the NAS database, VNX has developed a new query subsystem that appears as a hidden option on some of the nas_commands. This query subsystem enables you to specify the information you are interested in, allows you to format the output, and is independent of the database format.

CAUTION

Do not use grep and awk to scan the database files. Database positions may change and substrings may return false matches for database objects.

Following is an example of a query to view unused disks:

```
nas_disk -query:inuse==n -format:'%s
' -Fields:Id
```
To filter root disks, refer to the *List all non-root disks that are not in use.*

**Examples**

Use the following commands to view the tags (fields) that you can query:

```
nas_disk –query:tags
nas_fs –query:tags
nas_volume –query:tags
nas_slice –query:tags
```

Table 3 on page 738 contains a list of examples to help you get started. Note that these commands can be run on the Control Station CLI, so the hardcoded values can be replaced with shell script variables.

<table>
<thead>
<tr>
<th>Task</th>
<th>Example</th>
</tr>
</thead>
</table>
| Query the ID of a named file system                | `nas_fs -query:Name==RLL_fs10 -format:'%s
' -Fields:Id`                |
| Query the ID of a named file system without the new line | `nas_fs -query:Name==RLL_fs10 -format:'%s' -Fields:Id`                   |
| Query the name of a file system ID that corresponds to a particular ID | `nas_fs -query:id==20 -format:'%s
' -Fields:Name`                     |
| List of all server IDs                             | `nas_server -query:* -format:'%s
' -Fields:Id`                         |
| List of all server names                           | `nas_server -query:* -format:'%s
' -Fields:Name`                     |
| List of all the checkpoint file systems            | `nas_fs -query:type==ckpt -fields:name -format:'%s
'`                     |
| List type of file system with ID 20                | `nas_fs -query:id==20 -format:'%s
' -Fields:Type`                      |
| List the file systems that are in use              | `nas_fs -query:inuse==y -format:'%s
' -Fields:Name` or `nas_fs -query:inuse==y -format:'%s
' -Fields:Id` |
| Identify file system of which file system ID 28 is a backup | `nas_fs -query:id==28 -format:'%s' -Fields:BackupOf`                |
| List the name of the server with ID 2              | `nas_server -query:id==2 -format:'%s
' -fields:name`                   |
| View which volume file system is built on          | `nas_fs -query:Name==my_fs -format:%d -fields:VolumeID`                |
Table 3  Task and query examples (page 2 of 2)

<table>
<thead>
<tr>
<th>Task</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>View the block count of meta volume</td>
<td>nas_volume -query:Name==my_meta3 -format:%d -fields:Blocks</td>
</tr>
<tr>
<td>View the block size of meta volume</td>
<td>nas_volume -query:Name==JAH_meta3 -format:%d -fields:BlockSize</td>
</tr>
</tbody>
</table>
| Find which server IDs use fs123                 | nas_fs -query:name==fs123 -format:'%s
' -fields:ServerNumeric                                                                                                                      |
| List all non-root disks that are not in use     | nas_disk -query:inuse==n:IsRoot==False -format:"%s
" -fields:name                                                                                                                             |
| List unused volumes that contain “dc” in the volume name | nas_volume -query:inuse==n:IsRoot==False:name=dc -format:"%s
" -fields:name                                                                                                                             |
| List all available disks on a particular storage device (symm_id is a script/env variable) | nas_disk -query:inuse==n:SymmID==symm_id:IsRoot==False -format:"%s
" -fields:name                                                                                                                             |

**Query operators**  Use the operators in the table below when building your queries:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Having the pattern (sub-string match)</td>
</tr>
<tr>
<td>==</td>
<td>Exact string match</td>
</tr>
<tr>
<td>=-</td>
<td>Integer Minimum (not less than)</td>
</tr>
<tr>
<td>=+</td>
<td>Integer Maximum (not more than)</td>
</tr>
<tr>
<td>=*</td>
<td>Any</td>
</tr>
<tr>
<td>^=</td>
<td>NOT having the pattern</td>
</tr>
<tr>
<td>^==</td>
<td>NOT an exact match</td>
</tr>
<tr>
<td>^=^-</td>
<td>NOT Integer Minimum (is less than)</td>
</tr>
<tr>
<td>^=^+</td>
<td>NOT Integer Maximum (is more than)</td>
</tr>
<tr>
<td>^=^*</td>
<td>NOT Any (None)</td>
</tr>
</tbody>
</table>
This section contains the GNU General Public License (GPL). The GPL is the license for the Linux operating system. All EMC software, including the VNX software, is licensed by the EMC Software License included in the software kit.
GNU General Public License

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330
Boston, MA 02117-1307
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.
We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

Terms and conditions for copying, distribution, and modification

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification.”) Each licensee is addressed as “you.”

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does. This license states that:

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise
the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.
4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version,” you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.
NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
Symbols
/.etc/homedir 431
/etc/passwd 110
/nas/sbin 25
/nas/sys directory 719
/nasmcd/sbin 25

A
Access control levels
  creating 110
  deleting 110
  file systems 201, 330, 352, 374
  storage systems 166, 352
  volumes 373
Access list, iSCSI 548
Access procedures
  local 13
  remote 13
ACL database 229, 441
action, nas_event 192
Activating
  license keys 238
  time services 471
Active Directory database 438
addressing
  link-local 535
Aliases, using server_cifs 434
Antivirus 413
ARP table 396
Attached storage system, displaying 351
Auditing
  CIFS configuration 431
  AUTH_SYS 497

Authentication
  CIFS 430
  Data Mover 119
  HTTP 526
  kerberos 434
Automatic checkpoint scheduling 708
Automatic Data Mover reboot 470
Automatic File System Extension 199
Auto-migration processes 404
Automount map 113
  automount daemon 113
  conflicting list 113
  infile 113
  outfile 113
Automounting 113
Autonegotiation 695
Autonegotiation, for Gigabit Ethernet 695

B
Backing up file systems 382
bcpio 389
BCV 171, 175, 259
Block limit, quotas 273
Blockmap memory 703
Bus speed 698, 699

C
Case-sensitive, names and options 16
CDMS 404
VNX FileMover 58, 59, 60
  CIFS connections 64
  creating an HTTP connection 526
  HTTP configuration file 525
HTTP connections 66, 70
NFS connections 59, 61
NFS server 63
parameters 59
VNX Replicator 298
Chain number, SCSI 483, 484
CHAP security 640
Checking configuration parameters 413
Checkpoint 28
Checkpoint file system 30
creating 30
refreshing 30
Checkpoint scheduling 136
CIFS 430
ACL database 441
activating 643
adding NFS service 437
alias 434
auditing configuration 431
authentication 430
calendar name 433
collection 430
connections 64
creating a share 495
dialect 439
DNS 434
FQDN 437
homedir 431
local users 432
NetBIOS 431
network backups 59
Network neighborhood 432
passwords 437
security 438
security IDs 441
security, CIFS
   Kerberos 430
   statistics 442
   usermapper 430
CIFS client 59
CIFS shares 495
cifsnanoroundup 591
cifssyncwrite 590
VNX 35 1
VNX array properties 352
VNX for block backend support 257
VNX pools 203, 206, 209
VNX for block storage pools 209, 257
CLATA 171, 174, 259
Clone volumes 373
Clone, volume 374
CLSTD 171, 174, 259
CMATA 171, 174, 259
CMSTD 171, 174, 259
CNS series 699
Command set
   conventions 15
   synopsis 15
Command-line interface
   accessing the command-line interface 13
dual Control Stations 25
external Usermapper 714
failover 25
local access 13
NAS database 200
nas_commands 16
remote access 13
starting NFSv4 service 609
volume table 373
Conventions 15
Converting hostnames
   using DNS 493
Converting IP addresses
   using DNS 493
cpio 389
CPU, monitoring 469
Creating
   TimeFinder/FS copy 100
Current log 581
cvfsname 589
D
Daemons
   automount 113
Data Mover
activating the standby 662
authentication 119
current log 581
date and time 471
device configuration 483
disk space 491
halting 469
hardware configuration 483
inodes 491
internal Usermapper 714
listing the server table 329
mounting file systems 587
operating system statistics 703
pinging 632, 634
rebooting 469
security policy settings 640
standby
designating a policy 663
policy types 663
system software version 718
time zone 471
types
  nas 329
  rdf 329
  standby 329
  vdm 329
unmounting file systems 708, 719
user accounts 711
usermapper 714
Data Mover, GPO, using server_security 640
Data Mover, motherboard type 694
Data Mover, protocol 643
Data Movers
  standby 662
Dependency 413
Destination VNX Network Servers 166
Destination file, using server_file 508
Destination volume 375
Device configuration 483
  probe, save, display 483
Devices 483
Diagnostic messages
  server_archive 383
Dialect, using server_cifs 439
DIR3, MPD 226
Disk space 491

Disk table 170
deleting disks 170
Disk types
  BCV 171, 174, 259
  CLATA 171, 174, 259
  CLSTD 171, 174, 259
  CMATA 171, 174, 259
  CMSTD 171, 174, 259
  R1BCV 171, 174, 259
  R2BCV 171, 174, 259
  R2STD 171, 174, 259
  STD 171, 174, 259
Displaying
disk space 491
Displaying system parameters 620
Distinguished name 574
DNLC 678
DNS 434, 493
  activating 493
  lookup server 493
  server 493
Documentation, related 7
Domain, Windows NT 432
Dual Control Stations, cs_standby 25

E
ECHO_REPLY
  messages receive 632, 634
ECHO_REQUEST 632, 634
  messages send 632, 634
EMC Active Directory 438
emctar 390
Enabling
  licenses 238
  software 238
Enabling software 238
Error messages
  server_archive 395
Ethernet
  changing default values 695
  speed, duplex 695
Event configuration file 197
Event configuration files 192
Event store 192
Events, nas_event 184
F
facility, nas_event 192
Failover
activating a standby 662
Control Station 25
Fail-safe network device 696, 702
FarCopy environment 20
Fiber channel 483
File system
access control values 201, 330, 352, 374
aclchk 229
Auto extension 199
checking inodes 491
checkpoint 28, 30
connections 58
default type 205, 208
deleting 40, 200, 538, 539
extending 202
file-level retention 207
fsck 229
high watermark 208
information 200
maximum size 208
minimum volume size 374
mixed-access checking 201
mounting 587
nbpi 205, 208
percentage threshold 29, 30, 31
read-only 588
read-write 588
renaming 201
size 201
total capacity 491
types
ckpt 200
group 200
mgfs 200, 404
mirrords 200
nmfs 200
rawfs 200
uxfs 200
UNIX permissions 201
virtual provisioning 199
Windows permissions 201
File systems
mount points 595
File-Level Retention 91, 199, 205
FileMover
HTTP 526
FileMover connections 81
FQDN 64
Free disk space 491
FTP users 203
FTP, using server_file 508
Full sanity check 413
Gateway address 638
getreason 26
GID 110
GID, quotas 268, 270
Gigabit Ethernet 533
Gigabit Ethernet Copper 695
Gigabit Ethernet Fiber 695
Gigabit Ethernet, autonegotiation 695
Gigabit switch 695
GPO settings 641
GPO, group policy object 640
Group policy object 640
Group quotas 267
Group, file system 87
Halting a Data Mover 469
Health checkup 131
High watermark 208
Home directory, enabling 431
Hostname requirement
server_ping 632, 634
HTTP
authentication 526
client 68, 71
configuration file 525
connections 67
for VNX FileMover 525
protocol 529
requests 67, 71
threads 526
HTTP connections, VNX FileMover 66, 70
HTTPS
connections 70
I
I18N 332
ICMP
   ECHO_REPLY 632, 634
   ECHO_REQUEST 632, 634
Initiator, iSCSI 548
Inodes 491
Inodes, displaying 491
Internal Usermapper service 714
IP protocol, assigning 532
iPlanet
   client 572, 573
   domain 574
IP-to-MAC address 396
   setting 396
iSCSI
   access list 548
   configuration 543
   iSNS 543
   logical unit numbers (LUNs) 543
   LUN 546
   name service management 543
   portal group 547
   Qualified Name 544
   standard 544
   targets 543
   Unicode characters 544
   virtual provisioning 546
iSCSI replication
   nas_replicate 156, 199, 283, 298, 367, 543, 602, 620
iSCSI replication 298
   nas_replicate 543, 572
iSNS 543

K
Kadmind 566
KDC 564
Kerberos authentication 607
Kerberos configuration 564
   domain 564
   Kadmind 566
   KDC 564
   realm 564
Kerberos Domain Controller 438
Kerberos, authentication 434
Kerberos, user and data authentication 497
   keyword, nas_event 192

L
LACP, aggregation 697
Ldap 574
License key 238
License, software 238
Licensing
   nas_license 238
Link aggregation 696, 697
Link Aggregation protocol 694
Linked VNX Network Servers 118, 367
link-local address 535
Linux 25
Load balancing 697
Loading a debug/patch image 643
Local access 13
Log files 240, 580
LUN masks 312
LUNs, for Mirror View/S 166

M
MAC addresses 697
Magnetic tape drive 600
Managing protocols 643
Mask, iSCSI 547
Maximum Transmission Unit size 532
MD5 password encryption 711
MGFS 404
mgfs 404
Migrated file system 404
MirrorView/S 166, 244
MirrorView/Synchronous 166, 244
Mixed access-checking, for file systems 201
Mixed backend system 374
Mother board 698
Mount options 588
   read-only 588
   read-write 588
Mount points
   creating, listing, deleting 595
Mount protocol 405
mountd daemon 62
Mounting a file system 587
MPD 201, 225
Index

MPFS 597
  protocol 597
  threads 597, 598
MPFS client 598
MPFS clients 588
MTU 532
MTU size
  for Ethernet 533
  for Gigabit Ethernet 533
Multi-Protocol Directory 227
Multiprotocol directory 201

N
Name service management, iSNS 543
NAS 643
NAS database 200
nbpi 205, 208
NDMP
  connections 712
  creating user accounts 713
  passwords 713
NDMP backups 723
NearCopy environment 20
Nested mount file system 200
Nested mountpoints 200
NetBIOS name 434
NetBIOS name, alias 434
NetBIOS names 501
Netgroup 499
Netgroups 499
Network
  activate DNS client 493
  display statistics 603
  operating system statistics 703
  send ECHO_REQUEST messages 632, 634
Network adapter card 694
Network interface cards
  speed 694
Network interfaces
  creating an interface 532
  enabling and disabling 532
  IP protocol 532
  MTU size 532
Network interfaces, changing default values 695
Network Neighborhood 432
NFS connections 60
NFS connections, VNX FileMover 59, 61
NFS port 405
NFS protocol 61
NFS server, VNX FileMover 63
nfsd daemon 62
NFSv2 404, 607
NFSv3 404, 607
NFSv4 607, 609
NFSv4 clients 613
nfsv4delegation 591
NIS
  activating 618
  deleting 618
NIS client 618
NIS domain 574
NIS server requirement
  server_ping 632, 634
NMFS 209
nonotify 590
nooplock 590
noprefetch 589
noscan 590
Notifications, for mounted file systems 590
notifyonaccess 590
notifyonwrite 590
ntcredential 590
NTLM 434, 435

O
Offset default 340
Offset, Slice Volume 340
OpenLDAP
  client 573
OpenLDAP client 572
OpenLDAP domain 574
Operating system statistics 703
Oplock, default 590

P
Parameters 620, 621
  PAX 621
Password encryption 711
Password, for users 711
PAX 628
PAX parameters 621
Permissions, for UNIX and Windows 201
Command Line Interface Reference for File 7.0

PFS 28
Ping 632, 634
Policy type, for Standby
auto 663
manual 663
retry 663
Portal groups 547
Portmapper 405
POSIX compatibility 385
Power fail/crash recovery 470
Primary Control Station 25
Primary Data Mover
activating the standby 662
restoring the standby 662
Production file system 28
Production file system (PFS) 28
Protocol, setting up 643
Protocols
CDMS 643
CIFS 643
ICMP 603
IP 603, 697
MAC 697
managing 643
MPFS 597, 643
NFS 609
RPC 609
RIP 636
TCP 603, 697
TFTP 705
UDP 603
virus checker 719
VIRUSCHK 643

Q
Quota configuration 268
Quotas
edit 268, 270
group 267, 268, 270
proto user 268, 270
report 268, 270
user 267
Quotas, block limit 273

R
R1 20
R1 and R2 20
R1BCV 171, 175, 259
R1STD 171, 175, 259
R2 20
R2BCV 171, 175, 259
R2STD 171, 175, 259
RDF file systems 91
RDF relationship for remote VNX 118, 368
RDF relationships 662
Realm, for Kerberos 566
Rebooting a Data Mover 469
Remote access 13
Remote VNX
cel_fs 20
R1 and R2 20
SRDF 20
Remote VNX Network Server 118, 368
Remote disk 171
Remote file system 20
Remote NFS server, communicating with 404
Remotely-linked, VNX Network Server 20
Remotely-linked, file systems 20
Replication for iSCSI 298
Replication for remote VNX 118, 368
Replication sessions
displaying 300
listing 299
Restoring the standby Data Mover 662
RIP 636
RIP configuration 636
Root 498
root 110
Routing entries 637
Routing table 603
RPC bind 405
RPC statistics 610

S
Sanity check 413
SavVol 28
sbin 26
Scheduling 136
SCSI device table 484
listing 484
SCSI devices 178, 483
health check failures 484
Index

listing 484
Secondary file system 59, 61, 64, 66, 70
Secure NFS service 607
Security IDs 441
Security modes 439
Security policy settings 640
Security, CIFS 430
Security, for CIFS 438
Server tables, listing 329
Server types
nas, standby 643
server_log
examples 580
Slice entry
deleting 340
Slice information
displaying 340
Slice table
listing 340
Slice Volume
offset 340
Slice volumes
creating 340
SMB statistics 442
SnapSure 28
SnapSure checkpoint 136
SnapSure checkpoints 201
SNMP
agent 646
values 646
Software packages, nas_license 238
Software version, for Data Movers 718
Source file, using server_file 508
Source volume 375
SRDF link 118
SRDF relationship, initializing 285
SRDF-linked, VNX 118
Standalone CIFS server 435
Standby
activating 662
restoring 662
setting policy type 663
Standby Data Mover, activating 643
Statistics
CIFS 442
network 603
NFS 609
operating system 703
routing table 603
RPC 609
STD 171, 175, 259
Storage pools
creating 253
extending 253
system-defined 253
user-defined 253
Storage systems 166, 352
Stub files 60
Subnet 499
sv4cpio 390
sv4crc 390
Symmetrix backend support 209, 258, 333
Symmetrix pools 203
Symmetrix storage pools 209, 258, 333
Synchronizing time 471
Synopsis 15
syslog file 29
System events 184
System health checkup 131
System parameters 620
System version, displaying 371
systems 587

T
Tables
ARP 396
mounted file systems 587
NIS 618
routing 637
SCSI device table 484
slice 340
Tape backup 600
tar 390
TCP 405, 493, 533
TFTP 705
Threads
for CIFS, CDMS, or virus checker 643
HTTP 526
MPFS 597
Time services
using server_date 471
Time synchronization 471
Time zone 471
TimeFinder/FS 87, 98
TimeFinder/FS Far Copy 91
Timing sources 471
Trap, nas_event 192
Tree IDs 272
Tree quotas 270
triggerlevel 590
Trivial File Transfer Protocol (TFTP) 705
Trunked Ethernet channel 697
Trusted relationship 119

**U**
UDP 405, 493, 533
UDP sockets 603
UID 110
UID, quotas 268, 270
Umask 501
uncached 590
Unicode characters 544
UNIX permissions 201
UNIX/Linux Kerberos KDC 564
User accounts 711
User accounts for NDMP 713
User-defined storage pools 253
Usermapper 440, 714
Usermapper database 717
Usermapper, CIFS 430
usrmap.cfg file 715
ustar 390
UTC 443
UTF-8 438, 439
Utilities
  aclchk 228
  fsck 228

**V**
VDMs 329
displaying 330
loading 331
Virtual Data Movers 329
virtual device 697
Virtual devices 696
Virtual provisioning 199, 203, 207, 546
Virtual robot 724
Virtual tape library unit 723
Virtual tapes 729
Virus Checker
  noscan 590
Virus checker 719, 720
  auditing 719
  configuration file 719
  upgrading 719
viruschecker.conf 719
VLAN 533
VLTU 723
Volume table 373
Volumes
  access control levels 373
  cloning 373
  creating 373
  deleting 373
  displaying information 373
  extending 374
  getting info 373
  listing 373
  minimum size for file system 374
  renaming 373

**W**
Windows NT 4.0 406
Windows NT 4.0 domain 432
Windows NT Credential 590
Windows permissions 201
Windows popups 59
WINS server(s) 439