Sonexion CNG Installer and Administrator Guide

Contents

About *Sonexion CNG Installer and Administrator Guide*....4
CNG Introduction....5
CNG Hardware and Cabling....7
CNG Initial Configuration....10
Configure CNG Using the CLI....13
Specify Global Settings for Shares - CLI....14
Security Type Examples....16
Add Shared Resources - CLI....18
Specify Permissions and Restrictions by Group - CLI....20
Configure CNG Using the GUI....23
Specify Global Settings for Shares - GUI....25
Define Client Groups - GUI....28
Add Shared Resources - GUI....30
Specify Permissions and Restrictions by Group - GUI....32
Test Access to Shared Resource....35
Manage CNG After Initial Configuration....37
Manage CIFS/NFS Shares with CLI....38
Manage CIFS/NFS Shares with GUI....45
Set Shared Resource, Enable Sharing - GUI....45
Client Groups for a Shared Resource - GUI....47
Display Undefined Groups, Change Options....48
Delete Shares....49
Edit CIFS/NFS Configuration with CLI....50
Edit CIFS/NFS Configuration with GUI....56
Turn On or Off CIFS-NFS Shares....58
Monitor CNG Nodes with CLI....61
Monitor CNG Nodes Using Node Control Tab....66
CSSM Health Tab....68
CLI cscli cng Reference....73
`cng cifs`....76
`cng cifs idmap`....80
`cng clientgroup`....82
`cng interface`....84
`cng internaldb`....85

`cng join`....86
`cng nfs`....87
`cng node`....88
`cng share`....89
 `cng share cifs`....92
 `cng share nfs`....96
`cng show`....99

About *Sonexion CNG Installer and Administrator Guide*

This publication describes the process to install a CIFS/NFS Gateway (CNG) unit on a Sonexion system, using optional dedicated hardware server nodes with CIFS2 Export and NFS Export capabilities. The manual also includes a basic set of administrative tools, including a complete reference for CSCLI particular to the CNG.

This manual applies to Sonexion 2000 1.5 and 2.0, Sonexion 1600 1.5, and Sonexion 900 1.5.

This procedure is intended to be performed only by qualified Cray personnel. The procedure is not intended for use by customers.

Typographic Conventions

`Monospace`	A `Monospace` font indicates program code, reserved words or library functions, screen output, file names, path names, and other software constructs
`Monospaced Bold`	A **`bold monospace`** font indicates commands that must be entered on a command line.
`Oblique` or *`Italics`*	An *`oblique`* or *`italics`* font indicates user-supplied values for options in the syntax definitions
Proportional Bold	A **proportional bold** font indicates a user interface control, window name, or graphical user interface button or control.
`Alt-Ctrl-f`	`Monospaced` hypenated text typically indicates a keyboard combination

Record of Revision, publication HR5-6147

Publication Number	Date	Release	Comment
HR5-6147-0	May 2015	1.5	2000 only
HR5-6147-A	October 2015	2.0	2000, 1600, and 900

CNG Introduction

Sonexion provides support for the CIFS/NFS Gateway (CNG) feature, using optional dedicated hardware server nodes with CIFS2 Export and NFS Export capabilities.

The CNG feature provides parallel multi-client access through multiple CNG nodes, which allows the Lustre filesystem to be shared with enterprise clients (Windows, MAC, Linux) via NAS CIFS2 and NFS. The Clustered Trivial Database (CTDB) file sharing technology, which is part of Samba, provides shared access by multiple CNG nodes to Lustre files. The CNG feature provides:

- Full support for Sonexion clusters, software upgrades, bundles and logs, health monitoring, CNG ECN IP address changes, and installation
- Automatic configuration for supported Lustre Client Network (LCN) and Enterprise Client Network (ECN) interfaces
- Full CSCLI support of FRU inventory
- Manual FRU replacement of CNG chassis and nodes
- Fully-featured share configuration including security
- LCN: Full support for 40GbE and QDR/FDR IB
- ECN: Full support for 10GbE and 40GbE

CNG Chassis (Optional)

The CNG Chassis is an optional 2U four-node or two-node server that allows enterprise clients (such as Windows, Macintosh, or Solaris) to access the Lustre filesystem via the CIFS2 or NFS protocols.

The CNG chassis comes in two configurations, referred to as Configuration A and Configuration B. The chassis includes four or two CNG nodes, each with identical hardware configurations.

Table 1. CNG Chassis Configurations

	Configuration A	**Configuration B**
Onboard NIC Ports	2, used for Local Management Network (LMN)	2, used for Local Management Network (LMN)
Onboard ConnectX-3 Ports (supports IB QDR, IB FDR, and 40GbE)	1, used for Lustre Client Network (LCN).	1, used for Enterprise Client Network (ECN), supports 10GbE and 40GbE
PCIe Add-on Mellanox HCA Card	ConnectX-3 card with 1 port, used for Enterprise Client Network (ECN), supports10GbE and 40GbE	ConnectX-2 card with 1 port, used for Lustre Client Network (LCN), supports IB QDR and IB FDR

Configuration A is required if the LCN requires 10GbE or 40GbE support.

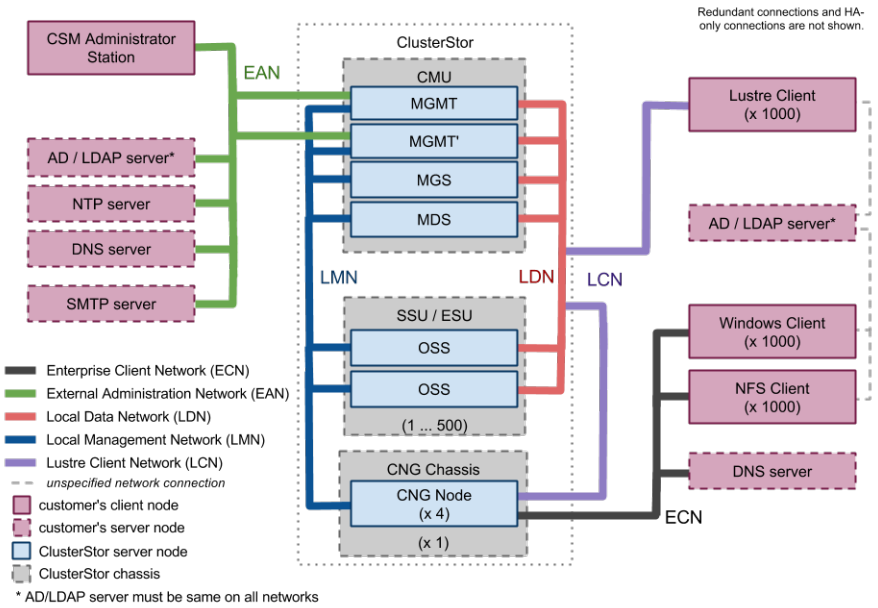
The CNG chassis supports:

- InfiniBand and 40GbE connections directly to the Lustre Client Network (LCN). At a customer site, these connections should be made to the customer's LCN, not to the Sonexion TOR network switches.
- 10GbE connections to the Enterprise Client Network (ECN) via a 10GbE switch that is part of the customer infrastructure.
- 1GbE connection to the Sonexion Local Management Network (LMN).

Network Architecture

The following diagram illustrates how CNG integrates with the various networks in the Sonexion solution.

Figure 1. Network Architecture with CNG



Sonexion implements two local in-rack networks and interacts with several external networks:

- The **Local Data Network (LDN)** provides a high-speed network that is used for data transport to and from the client machines and for connectivity between the MMU and all OSS nodes.
- The **Local Management Network (LMN)** provides network connectivity between the MGMT nodes and all manageable nodes and components in the Sonexion solution.
- The **Lustre Client Network (LCN)** is external to Sonexion and is used to provide data path connectivity between the Lustre clients and the Sonexion storage cluster.
- The **External Administration Network (EAN)** connects to the MGMT nodes to provide administrative access to the CSSM (CSSM).
- The **Enterprise Client Network (ECN)** connects Windows and/or NFS clients to the CNG nodes to provide access to files on the Lustre file system.

Each CNG node connects to the Sonexion LMN as do all other Sonexion server nodes. For access to the Lustre file system, the CNG nodes connect directly to the LCN to provide them with balanced access to the entire file system cluster. This is particularly important for Sonexion solutions with multiple racks. The Enterprise Client Network is unique to the CNG nodes and is seen only in Sonexion systems that include the CNG feature.

CNG Hardware and Cabling

Overview of CNG

Sonexion systems that include a CNG chassis support the CIFS/NFS Gateway feature . The CNG chassis is an optional 2U 4-node or 2-node server that allows enterprise clients (such as Windows, Mac, or Solaris) to access the Lustre filesystem via the CIFS2 or NFS protocols.

The CNG chassis comes in two configurations, referred to as Configuration A and Configuration B. The chassis includes four or two CNG nodes, each with identical hardware configurations.

Table 2. CNG Chassis Configurations

	Configuration A	**Configuration B**
Onboard NIC Ports	Two, used for Local Management Network (LMN)	Two, used for Local Management Network (LMN)
Onboard Connect X-3 Ports (supports IB QDR, IB FDR, and 40GbE)	One, used for Lustre Client Network (LCN).	One, used for Enterprise Client Network (ECN), supports 10GbE and 40GbE
PCIe Add-on Mellanox HCA Card	ConnectX-3 card with one port, used for Enterprise Client Network (ECN), supports10GbE and 40GbE.	ConnectX-2 card with 1 port, used for Lustre Client Network (LCN), supports IB QDR and IB FDR. NOTE: The Connect X-2 card only runs at QDR speed, but can be connected to an FDR network.

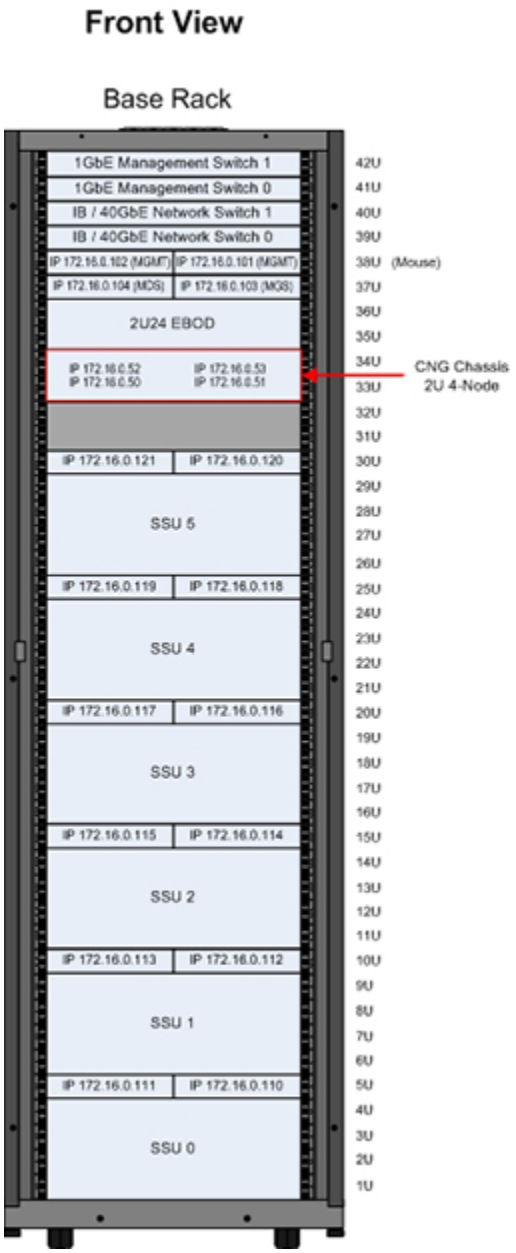
Configuration A is required if the LCN requires 10GbE or 40GbE support.

For release 1.5.0, the CNG chassis supports:

- InfiniBand and 40GbE connections directly to the Lustre Client Network (LCN). At a customer site, these connections must be made to the customer's LCN, not to the Sonexion TOR network switches.
- 10GbE or 40GbE connections to the Client Enterprise Network (ECN) via an Ethernet switch that is part of the customer infrastructure.
- 1GbE connection to the Sonexion Local Management Network (LMN).

The following figure depicts a simple one rack Sonexion 2000 system with a four-node CNG chassis installed.

Figure 2. Base Rack CNG Configuration Front View



Cabling Connections

The following diagrams depict the cabling connections for CNG Configuration A and Configuration B:

- Connections from the CNG chassis nodes to the management switches. These connections are made during manufacturing. The diagrams show 24-port management switches in use. For the most up-to-date information about these connections, see *Sonexion Field Installation Guide* for your system.
- Generic connections to the customer site's LCN. Each CNG node is connected to the site's high speed LCN core switch (InfiniBand or 40GbE).

- Generic connections to the customer site's ECN (10/40GbE). Each CNG node is connected to the network that hosts the site's enterprise clients. These are the clients that will be given access to the Lustre filesystem through the CNG feature.

Figure 3. CNG Node Cabling Configuration A

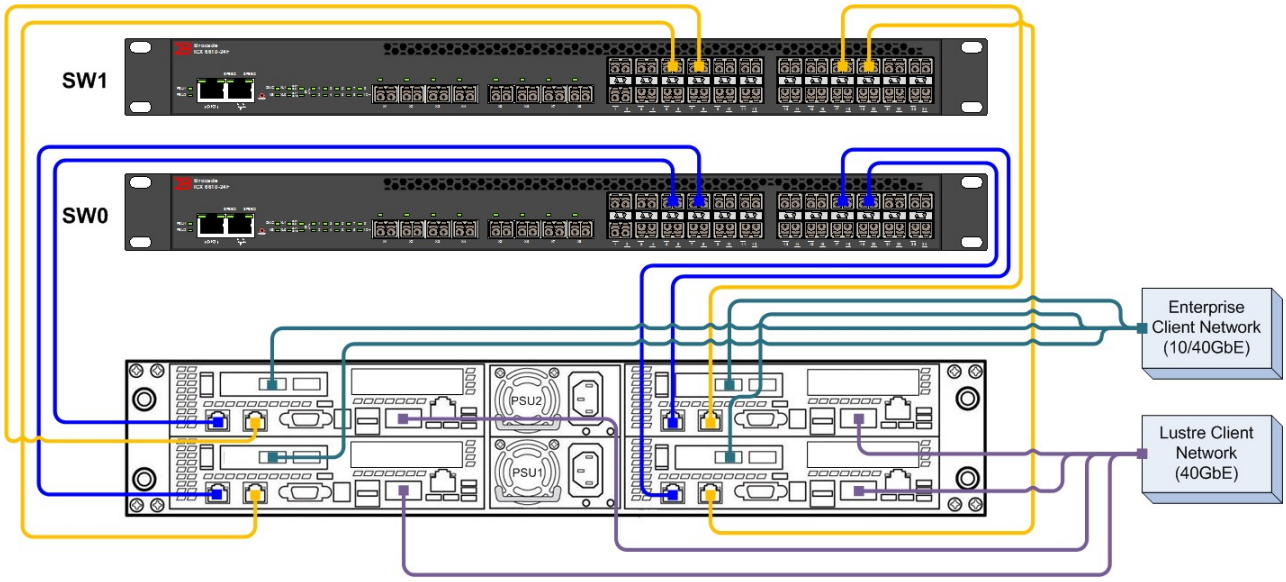
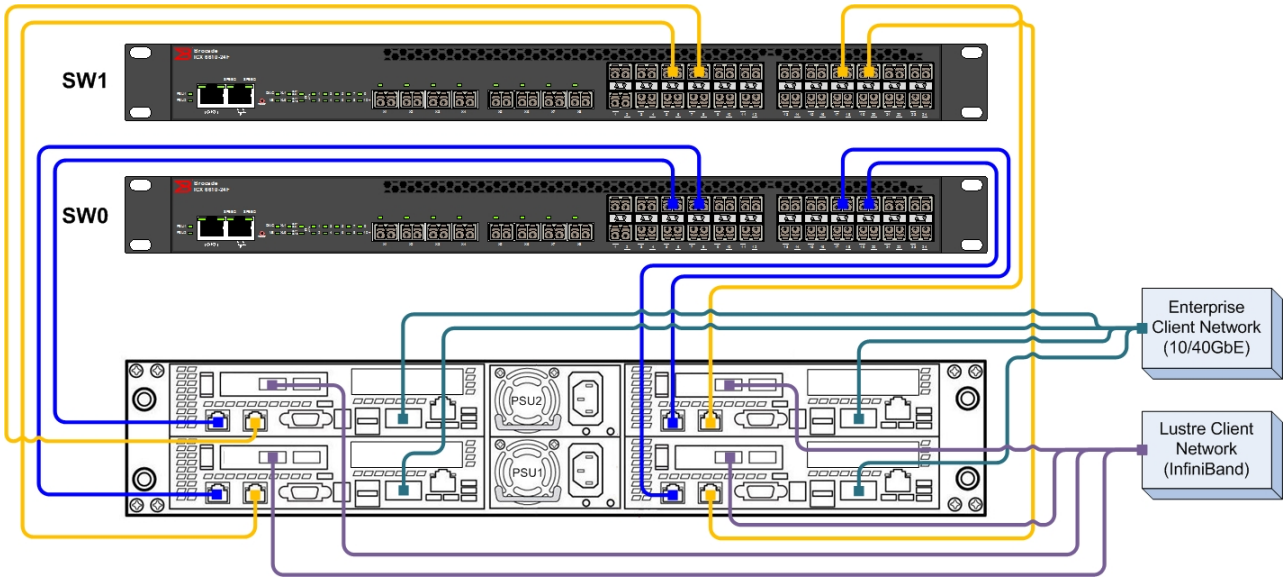


Figure 4. CNG Node Cabling for Configuration B



CNG Initial Configuration

The CIFS and NFS protocols may be configured at same time or they may be configured separately.

Sonexion 2000 1.5.0 supports one CNG chassis with either two or four CNG nodes.

Prerequisites

The following are required to configure CNG:

- CNG chassis installed in the Sonexion base rack.
- Sonexion 1.5.0 system with CNG hardware and software, installed and provisioned by Cray Manufacturing.
- The Customer Wizard must have been completed and the system must be set to Daily mode.
- Lustre user authentication must be configured. See "Lustre User Authentication", part of *Sonexion 2000 Field Installation Guide*
- Lustre must be running before shares can be created.

Optional Customer Site Prerequisites

Site administrators may find that the following additional system configuration options may help to achieve the best possible CNG performance and reliability:

- **Configure DNS with round-robin**. This specifically applies to the CNG nodes. Configuring DNS with round-robin can help prevent the customer site from experiencing a situation in which only one CNG node is serving all requests from enterprise clients to the Lustre filesystem. Also, since there is no HA functionality for CNG, if that one CNG node should fail, the site could experience a total loss of traffic flow between enterprise clients and the Lustre filesystem.
- **Load Balance the CNG Nodes**. To achieve a balanced distribution of load between the CNG nodes, site administrators may consider associating groups of enterprise clients with particular CNG nodes. This option can be used separately or in conjunction with the first option above.

Resource Sharing

You must configure CNG for resource sharing. You can configure the CIFS protocol only or the NFS protocol only, or both. Follow the appropriate instructions depending on your site's needs.

The configuration procedures can be performed using CSSM or the CSCLI. For more information see *Configure CNG Using the GUI* on page 23 or *Configure CNG Using the CLI* on page 13. In each case, configuring CNG consists of the following basic procedures:

- Enable CNG Nodes
- Specify the Global Configuration:
- Configure CNG Nodes
- Specify the Global Configuration Settings for Shares
- Define Client Groups

- Add Shared Resources
- Specify Share Permissions and Restrictions per Client Group

Overview of CNG User Security Models

CNG can be configured to use one of several models to authenticate enterprise users for access to the Lustre filesystem. These include:

- Guest Access
- Lightweight Directory Access Protocol (LDAP)
- Active Directory (AD)
- Network Information Service (NIS)

Each security model requires a slightly different approach when configuring CNG and Lustre upcall.

Guest Access

Guest access provides a simple method for file exchange. All network users are mapped to a special anonymous user (usually the nobody account will be used).

- For CIFS, share security must be configured. The guest-ok share option must be set to yes in order to allow guest users to access the share.
- For NFS, the squash share option must be set to all in order to map all remote users to the local nobody account.

LDAP

LDAP provides a safe method to authenticate remote users.

- Prerequisites:
  - Lustre LDAP support must be configured for lustre_users.
  - Samba LDAP schema must be configured and installed on the LDAP Server. For more information, see OpenLDAP documentation such as *OpenLDAP Configuration Quickstart*.
  - LDAP users must include sambaSamAccount objectclass.
  - LDAP groups must include sambaGroupMapping objectclass.
- For CIFS, LDAP configuration is done in 3 steps:
  - Configure CIFS globals settings
  - Apply CIFS globals settings
  - Join CIFS to the LDAP directory
- For NFS:
  - NFS will use the system's NSS base.
  - Squash needs to be unset using the squash-unset option. This will force the NFS daemon to stop mapping all remote users to the local nobody account.

AD

AD provides a safe method to authenticate remote users.

- Prerequisite: Lustre AD support must be configured for lustre_users.
- To use the ad idmap backend, the Active Directory Unix Attributes package must be configured and installed on the AD server. Otherwise, the rid idmap backend should be used.

  We recommend that the following user names not be used when setting up the Global CIFS/NFS configuration if AD is selected as the upcall type: nobody, root, and admin. For example, the following CSCLI command fails if AD is configured:

  ```
  cscli cng cifs globals --guest-account nobody
  ```

  A similar failure occurs when there is an attempt to configure this combination in the GUI.
- For CIFS / NFS:
  - CIFS/NFS uses Winbind for user lookup. So even when CNG is configured for NFS only, Samba and Winbind must still be configured.
  - When CNG is configured for NFS only, there is no need to enable CIFS exporting, even if Winbind and samba was configured for ADS.
- Configuration must be done in five steps:
  - Configure CIFS globals settings
  - Configure Winbind
  - Configure idmap backend (can use tdb2, rid, or ad)
  - Apply CIFS globals settings
  - Join CIFS to the AD directory

NIS

- NFS uses the system's NSS base. Samba uses NSS as a source for local users, but there is no way to use NIS or passwd/shadow for Samba because they use different hash methods for passwords.
- Lustre NIS support must be configured for lustre_users.
- For CIFS, since Samba can't directly use passwd/shadow/nis for user authentication, each user should be added to the Samba SAM Account database.
- For NFS:
  - NFS will use the system's NSS base.
  - Squash needs to be unset using the squash-unset option. This will force the NFS daemon to stop mapping all remote users to the local nobody account.

Configure CNG Using the CLI

About this task

Commands shown in this section are illustrative and may cover only a subset of the available command options. When a command appears it includes a link to the complete command reference entry, which appears in the section CLI Command Reference. The link will appear like this:

Command Reference: *cng cifs globals*

By default, all CNG nodes are enabled after the cluster has been installed during manufacture. The first steps in this procedure are provided in the event that the CNG nodes have been disabled. These steps explain how to check the node status and enable the CNG nodes if necessary.

Procedure

1. Check the CNG node status:

   ```
   [admin@n000]$ cscli cng show all
   ```

   Command Reference: *cng show all*

2. Check the command output for the node status. Here is command output showing the nodes disabled:

   ```
   CNG nodes:
   disabled: snx11000n[006-009]
   ```

3. If the nodes are disabled, enable them:

   ```
   [admin@n000]$ cscli cng node enable -a
   ```

   If only one node needs to be enabled you can run this command instead:

   ```
   [admin@n000]$ cscli cng node enable -n node_spec
   ```

   Command Reference: *cng node enable*

4. Apply your changes:

   ```
   [admin@n000]$ cscli cng apply -y
   ```

   ⚠ **CAUTION:** When using the `-y` or `--yes` parameters all configuration changes that you have made using cscli cng commands will be committed.

   Command Reference: *cng apply*

5. Log in to the primary MGMT node via SSH:

```
[Client]$ ssh -l admin primary_MGMT_node
```

6. Specify the ECN network configuration of the CNG nodes:

```
[admin@n000]$ cscli cng ecn -A IP_addr_CNG_node1 -A IP_addr_CNG_node2 -A
IP_addr_CNG_node3 -A IP_addr_CNG_node4 -D domain_name -N network -M
subnet_mask -G gateway
```

Command Reference: *cng ecn*

- Each CNG node requires a unique IP address on the ECN.
- *domain_name* is the fully qualified domain name that is used by all of the CNG nodes. A fully qualified domain name is critical for NFS statd upcall.
- *network* is the IP address for the ECN.
- *subnet_mask* is the ECN subnet mask and should be in prefix format (for example, `/XX`).
- *gateway* is the ECN gateway. The following example is for a system with four CNG nodes, with assigned ECN IP addresses of 10.106.52.151 through 10.106.52.154:

```
[admin@snx11000n000]$ cscli cng ecn -A 10.106.52.151 -A 10.106.52.152 -A
10.106.52.153 -A 10.106.52.154 -D cng100.xyus.xyratex.com -N 10.106.52.0 -
M 22 -G 10.106.52.1
```

7. Apply your changes:

```
[admin@n000]$ cscli cng apply -y
```

CAUTION: When using the `-y` or `--yes` parameters all configuration changes that you have made using cscli cng commands will be committed.

Specify Global Settings for Shares - CLI

Procedure

1. Enable CIFS sharing, NFS sharing, or both, as follows.

To enable CIFS only:

```
[admin@n000]$ cscli cng cifs enable -y
```

To enable NFS only:

```
[admin@n000]$ cscli cng nfs enable -y
```

To enable exporting of all protocols:

```
[admin@n000]$ cscli cng enable -y
```

Command Reference: *cng cifs enable*, *cng nfs enable*, *cng enable*

2. Configure CIFS global settings:

```
[admin@n000]$ cscli cng cifs globals -S "description" -N server_name -W
work_group --security security_type --guest-account account_name --realm
realm_name/--realm-unset
```

Command Reference: *cng cifs globals*

Notes:

- *description* is a short description of the server.
- *server_name* is the NetBIOS name of the Samba server.
- *security_type* is either `ads` (Active Directory), `user`, or `share`. See the *Official Samba HowTo and Reference Guide* for a detailed explanation of the security modes.
- *work_group* is the NetBIOS group to which the server belongs. If AD is configured, this is the AD domain.
- *realm_name* this is the AD realm name and should be set only if the ads security type is used. If security is other than ads, use the `--realm-unset` option instead.

You cannot define different parameters for different CNG nodes. See *Security Type Examples* on page 16 for examples of security type usage and configuration.

3. Apply your changes:

```
[admin@n000]$ cscli cng apply -y
```

CAUTION: When the `-y` or `--yes` parameters are used, all configuration changes that you have made using `cscli cng` commands will be committed.

Command Reference: *cng apply*

4. To define a client group:

```
[admin@n000]$ cscli cng clientgroup add -N name_of_client_group --network
network_spec
```

Notes:

- *name_of_client_group* is a unique name for the group of ECN clients that are included in the IP address range specified by *network_spec*.
- *network_spec* is an IP address range specified in CIDR format. Example:

```
[admin@snx11000n000]$ cscli cng clientgroup add -N testlab1 --network
10.0.0.0/24
```

5. Apply your changes:

```
[admin@n000]$ cscli cng apply -y
```

CAUTION: When the `-y` or `--yes` parameters are used, all configuration changes made using `cscli cng` commands are committed.

Command Reference: *cng apply*

6. You can define groups of specific enterprise clients by IP address range or hostname. Then, when specifying access permissions and restrictions for shares, you can use a defined client group to refine the sharing level. To define a client group:

```
[admin@n000]$ cscli cng clientgroup add -N name_of_client_group --network
network_spec
```

Notes:

- *name_of_client_group* is a unique name for the group of ECN clients that are included in the IP address range specified by *network_spec*.
- *network_spec* is an IP address range specified in CIDR format. For example:

```
[admin@snx11000n000]$ cscli cng clientgroup add -N testlab1 --network
10.0.0.0/24
```

7. Apply your changes:

```
[admin@n000]$ cscli cng apply -y
```

CAUTION: When the `-y` or `--yes` parameters are used, all configuration changes made using `cscli cng` commands are committed.

Security Type Examples

About this task

This topic shows typical ways to set the `cscli cng cifs globals --security` *`security_type`* option in *Specify Global Settings for Shares - CLI* on page 14.

Security Type share - Anonymous Access

Example for security type share, for anonymous access:

```
[admin@snx11000n000]$ cscli cng cifs globals --security share -N ctdb -W dc -S "test server"
```

Security Type `user` - LDAP

Example for security type user, for when LDAP is configured:

```
[admin@snx11000n000]$ cscli cng cifs globals --security user -N ctdb -W dc -S "test server"
```

IMPORTANT: When configuring for LDAP, using the user security type, the CNG nodes must also be joined to LDAP. First apply the CIFS global settings. Run the commands below, where password is the LDAP account password:

```
[admin@n000]$ cscli cng apply -y
[admin@n000]$ cscli cng join ldap -P password
```

Command Reference: *cng join ldap*, *cng apply*

Security Type `ads` - Active Directory (AD)

Example for security type `ads`, for when AD is configured:

```
[admin@snx11000n000]$ cscli cng cifs globals -S "cifs/nfs gateway" -N testshare -W
XYUS --security ads --realm xyus.xyratex.com
```

When configuring for AD, using the ads security type, you must also configure Winbind and join the CNG nodes to the AD domain. Perform the following steps:

Procedure

1. Configure Winbind to allow Windows NT domain users to appear and operate as UNIX users on a UNIX machine:

   ```
   [admin@n000]$ cscli cng cifs winbind --enum-users yes,no --enum-groups yes,no --
   use-default-domain yes,no --offline-logon true,false--cache-timeenum wb_ctime
   ```

   Command Reference: *cng cifs winbind*

   Notes:

   - `--enum_users` – On large installations using `winbindd`(8), it may be necessary to set this option to yes to suppress the enumeration of users through the `setpwent()`, `getpwent`() and `endpwent`() group of system calls.
   - `--enum_groups` – On large installations using `winbindd`(8), it may be necessary to set this option to yes suppress the enumeration of groups through the `setpwent()`, `getpwent`() and `endpwent`() group of system calls.
   - `--use-default-domain` – Set this parameter to yes if the `winbindd`(8) daemon should operate on users without domain component in their username.

2. Configure the idmap backend:

   ```
   [admin@n000]$ cscli cng cifs idmap add -d domain -b backend -r UID_range
   ```

   Command Reference: *cng cifs idmap add*

   Notes:

   - *domain* – name of the domain.
   - *backend* – the backend that handles domain authentication. Options are `ad`, `adex`, `autorid`, `hash`, `ldap`, `nss`, `rid`, `tdb`, `tdb2`.
   - *UID_range* – range of user IDs.

3. Apply the Winbind and **idmap** settings:

   ```
   [admin@n000]$ cscli cng apply -y
   ```

   Command Reference: *cng apply*

4. Join nodes to the AD domain:

   ```
   [admin@n000]$ cscli cng join ad -U ad_user -P password
   ```

   Command Reference: *cng join ad*

   Notes:

   - *ad_user* – is the name of an AD user that can add nodes to the AD domain.

- *password*– is the user's password. Example:

```
[admin@snx11000n000]$ cscli cng join ad -U neotest@xyus.xyratex.com -P N30test1
```

Add Shared Resources - CLI

About this task

Once CNG is enabled and the global configuration is complete, you can begin adding shared resources that will be available for access by enterprise clients.

Procedure

1. To add a new shared resource, run:

```
[admin@n000]$ cscli cng share add -s share_name -p path -f filesystem_name -d
"description" -U user -G group -M mode
```

Command Reference: *cng share add*

Notes:

- *share_name* – a name for the share
- *path* – relative path to the shared folder
- *filesystem_name* – name of the Sonexion filesystem
- *description* – a description of the share
- *user* – user ID for the share
- *group* – group ID for the share
- *mode* – Linux access mode, for example 0777 for full read, write, execute access.

If the folder name does not exist on the Lustre filesystem it will be created and the user, group, and mode will be set as specified in the command. If the folder already exists on the Lustre filesystem a share will be created for that existing folder and the user, group, and mode will be overridden with those specified in the command. For example, the following command creates a share named "testshare", on the "testfs" filesystem, with open permissions:

```
[admin@snx11000n000]$ cscli cng share add -s testshare -p testshare -f testfs -
d "Test share with pub access" -U nobody -G nobody -M 0777
```

2. To set CIFS share options, run:

```
[admin@n000]$ cscli cng share cifs set -s share_name
--writeable yes,no --browsable yes,no --guest-ok yes,no
--valid-users 'valid_users'
```

Command Reference: *cng share cifs set*

Notes:

- *share_name* – the name of the share
- `--writeable` – Select `yes` to let the user write to the share .

- `--browsable` – Select `yes` if the share should be viewable in the list of available shares and in the browse list.
- `--guest-ok` – Select `yes` if no password is required to access the shared resource.
- *valid_users* – Enter a specification that defines the users who can connect to the shared resource .

This command has numerous other options for this command, listed in the command reference entry. Example:

```
[admin@snx11000n000]$ cscli cng share cifs set -s $sharename
      --writeable yes --browsable yes --guest-ok no
      --valid-users '@"Domain Users"'
```

3. Repeat Steps 1 to 2 for any additional shares you want to add at this time.

4. Enable exporting of CIFS, NFS, or both.
   - CIFS only:

     ```
     [admin@n000]$ cscli cng cifs enable
     ```

   - NFS only:

     ```
     [admin@n000]$ cscli cng nfs enable
     ```

   - Enable exporting of all protocols:

     ```
     [admin@n000]$ cscli cng enable -y
     ```

     Command Reference: *cng cifs enable*, *cng nfs enable*, *cng enable*

5. Apply your changes:

   ```
   [admin@n000]$ cscli cng apply -y
   ```

   CAUTION: When the `-y` or `--yes` parameters are used, all configuration changes that you have made using `cscli cng` commands are committed.

   Command Reference: *cng apply*

6. If desired, you may check the CNG configuration settings:

   ```
   [admin@n000]$ cscli cng show all
   ```

   Command Reference: *cng show all*

   The command output will be similar to that shown below:

   ```
   [admin@snx11000n000]$ cscli cng show all
   Export protocols:
           CIFS      : disabled
           NFS       : disabled
   ECN Configuration:
           DNS name  : ctdb.xyua
           Addresses : ['10.106.52.241', '10.106.52.242']
           Network   : 10.106.52.0
   ```

```
        Netmask   : 24
        Gateway   : 10.106.52.1
CNG nodes:
        enabled: snx11000n[006-0007]
ECN Interface Configuration:
        Active Interfaces: eth40
CIFS Global Configuration:
        workgroup     : ctdb
        netbios name  : ctdb
        server string : test server
        security      : user
        realm         : xyus.xyratex.com
        guest account : nobody
        log level     : 1
        Winbind:
                winbind separator          : default
                winbind enum users         : yes
                winbind enum groups        : yes
                winbind use default domain : yes
                winbind offline logon      : default
                winbind cache time         : default
        Windows acl:
                nt acl support       : default
                acl compatibility    : default
                map acl inherit      : default
                map hidden           : default
                map system           : default
                map archive          : default
                map readonly         : default
                store dos attributes : default
                dos filemode         : default
Shares:
        testshare:
                uid        : nobody
                gid        : nobody
                mode       : 777
                path       : testshare
                filesystem : testfs
                enabled protocols: cifs, nfs
                description: Test share with pub access
                cifs options:
                        writeable yes
                        guest_ok yes
                        browsable yes
                networks:
                        nfs * writeable: true, squash: all, async: true
```

Specify Permissions and Restrictions by Group - CLI

About this task

Once a share has been added, you can specify per-client group permissions and restrictions. You can also specify multiple client groups for one share, thereby creating different access levels for different client groups.

The shared resource must have previously been added. See *Add Shared Resources - CLI* on page 18. Follow the procedure below to add per-client group permissions.

Procedure

1. Specify the desired client group and permissions for the shared resource:

```
[admin@n000]$ cscli cng share cifs clientgroup set -f filesystem_name -s
share_name -p path -N client_group --writeable yes,no --denied yes,no
```

   Command Reference: *cng share cifs clientgroup set*

   Notes:

   - *filesystem_name* – Name of the Sonexion filesystem
   - *share_name* – Name of the shared resource to which the client group will be added
   - *path* – Relative path to the shared folder
   - *client_group* – Name of the client group to add to the share
   - `--writeable` – Specify **yes** if the client group is allowed to write to the shared resource; specify **no** if the client group is only allowed read access.
   - `--denied` – Specify **yes** if the client group is not allowed to access the shared resource; otherwise, specify **no**.

   For example:

```
[admin@snx11000n000]$ cscli cng share cifs clientgroup set -f testfs -s
testshare -p testshare -N testlab1 --writeable yes --denied no
```

2. Apply your changes:

```
[admin@n000]$ cscli cng apply -y
```

   CAUTION: When the `-y` or `--yes` parameters are used, all configuration changes that you have made using `cscli cng` commands are committed.

   Command Reference: *cng apply*

3. To specify per-client group share permissions for NFS, the shared resource must have previously been added. To add per-client group permissions, first specify the desired client group and permissions for the shared resource:

```
[admin@n000]$ cscli cng share nfs clientgroup set -f filesystem_name -s
share_name -p path -N client_group --writeable yes,no --async yes,no --
subtree-check yes/no --squash root,no_root,all --squash-unset --anonuid
anonymous_user_id --anonuid-unset --anongid anonymous_group_id --anongid-unset
```

   Command Reference: *cng share nfs clientgroup set*

   Notes:

   - *filesystem_name* – Name of the Sonexion filesystem
   - *share_name* – Name of the shared resource to which the client group will be added
   - *path* – Relative path to the shared folder
   - *client_group* – Name of the client group to add to the share

- `--writeable` – Specify **yes** if the client group is allowed to write to the shared resource; specify **no** if the client group is only allowed read access.
- `--squash` – Specify `root` to map request from UID/GID 0 to the anonymous UID/GID; specify `no_root` to turn off root squashing, specify all to map all UIDs/GIDs to the anonymous user. Use `--squash-unset` to unset the squash option and use default settings.
- `--anonuid` – Explicitly set the UID of the anonymous account. Use `anonuid-unset` to unset this option.
- `--anongid` – Explicitly set the GID of the anonymous account. Use `anongid-unset` to unset this option.

For example:

```
[admin@snx11000n000]$ cscli cng share nfs clientgroup set -s testshare -N
testlab1 --writeable yes --async yes --subtree-check no
```

4. Apply your changes:

```
[admin@n000]$ cscli cng apply -y
```

CAUTION: When the `-y` or `--yes` parameters are used, all configuration changes that you have made using `cscli cng` commands are committed.

5. If desired, you may check the CNG configuration settings and status:

```
[admin@n000]$ cscli cng show all
[admin@n000]$ cscli cng status
```

Command Reference: *cng show all*, *cng status*

Configure CNG Using the GUI

About this task

By default, all CNG nodes are enabled after the cluster has been installed during manufacture. The following procedure is provided in the event that the CNG nodes have been disabled. These steps explain how to check the node status and enable the CNG nodes if necessary. CNG nodes must be enabled before they can be configured. To verify whether the CNG nodes are enabled or disabled:

Procedure

1. Check the CNG node status:

```
[admin@n000]$ cscli cng show all
```

2. Check the command output for the node status. Following is command output showing the nodes disabled:

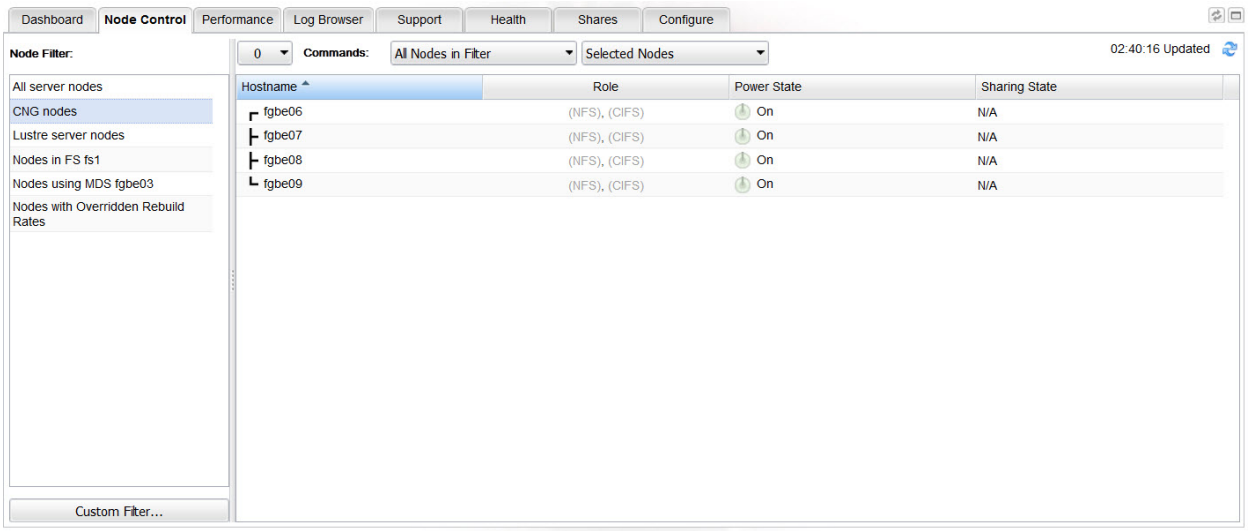
```
CNG nodes:
  disabled: snx11000n[006-009]
```

3. If the nodes are disabled, use the following procedure to enable them.

   a. Launch CSSM (GUI) and log in. The **Dashboard** tab will appear. Select the **Node Control** tab.

   b. In the **Node Filter** panel on the left side of the window, click on CNG nodes to view all available CNG nodes. See the following figure.

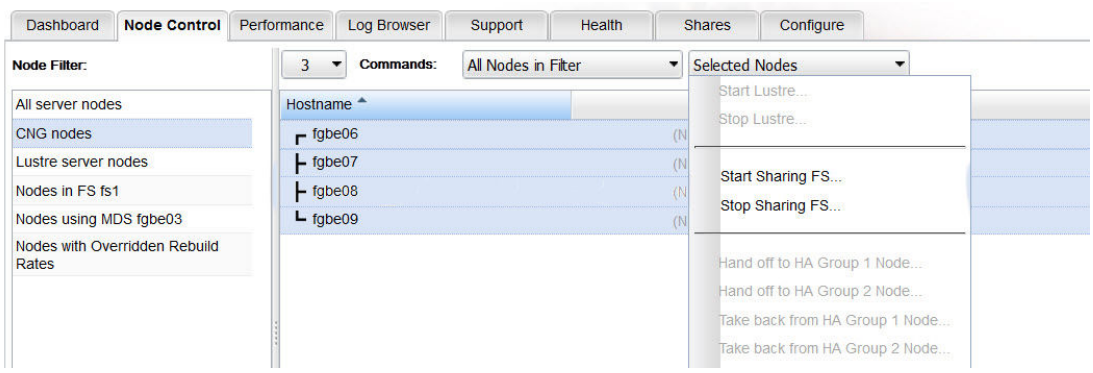
   Figure 5. Node Control Tab Showing CNG Nodes



   c. Select all of the nodes by clicking and highlighting all of the node names.

d. From the **Selected Nodes** menu, select **Start Sharing FS....** See the following figure.

Figure 6. Select CNG Nodes for Sharing



The following window displays, asking you to confirm that you want to start sharing the Lustre filesystem.

Figure 7. Please Confirm Start Sharing FS Dialog

e. Click **Start Sharing FS** to enable the nodes.

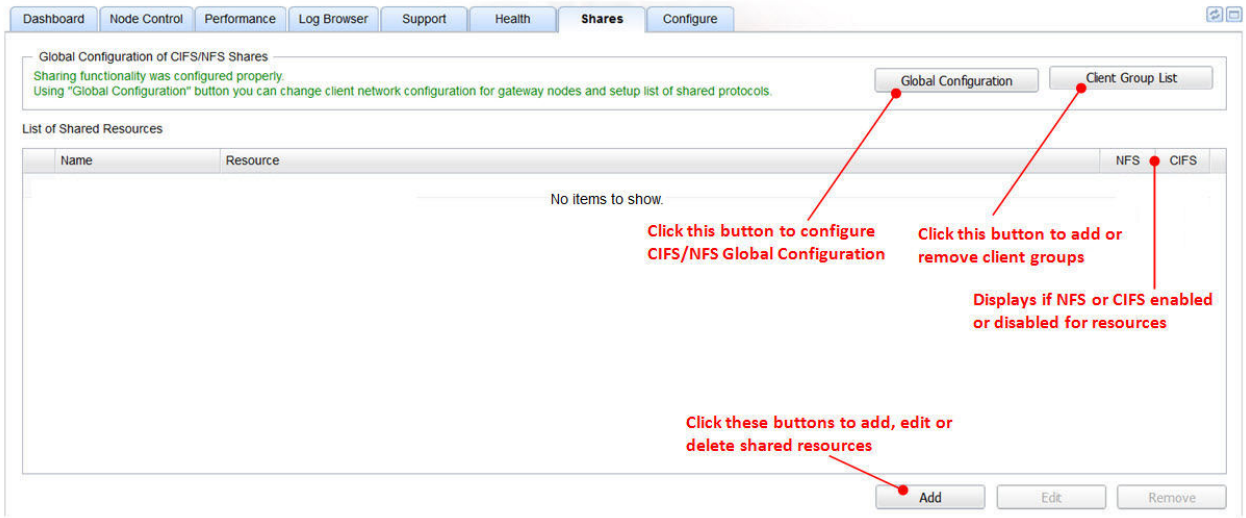
f. If desired, run the CSCLI command `cscli cng show all` again to verify that the nodes are enabled.

Following is command output showing the nodes enabled:

```
CNG nodes:
    enabled: fgbe[306-309]
```

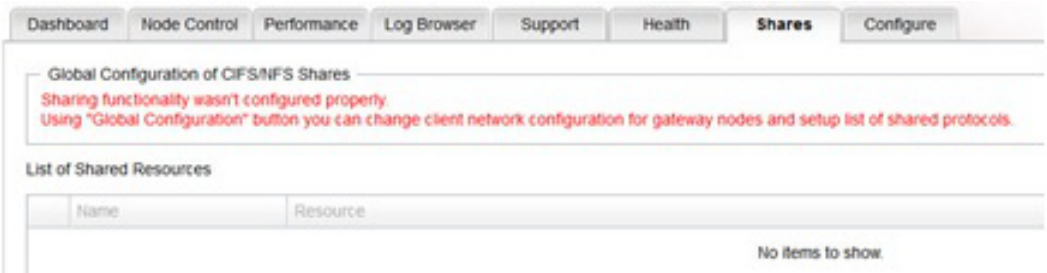
4. Global configuration of CIFS/NFS sharing functionality is done through the CSSM Shares tab. The following diagram shows the Shares tab components, which will list the CNG resources that have been shared. These resources are typically folders in the Lustre filesystem.

Figure 8. Shares Tab



All CNG nodes must be configured before creating shared resources. As illustrated by the message shown below in red, the global configuration for CIFS/NFS sharing is not yet set up.

Figure 9. Message that Global Configuration is Not Set Up



Specify Global Settings for Shares - GUI

About this task

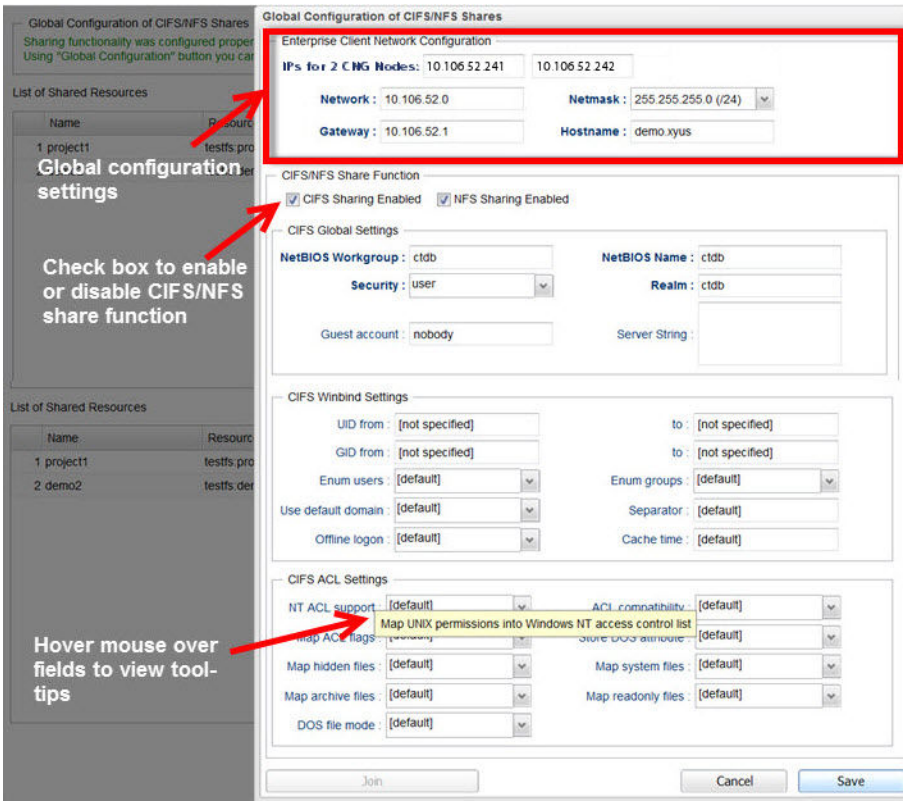
Follow this procedure to specify the global configuration settings for shares.

Procedure

1. On the **Shares** tab, click the **Global Configuration** button.

   The **Global Configuration of CIFS/NFS Shares** window displays. All fields are disabled by default in this window. Click the **Edit** button to enable changing the settings.

Figure 10. CIFS-NFS Shares Window Configuration



2. Specify the following global configuration settings:

a. In the fields next to the label **IPs for** x **CNG Nodes**, enter the IP addresses for the CNG nodes (either two or four) that are to be used for the share nodes on the Enterprise Client Network (ECN).

b. Enter the ECN **Network** IP address, **Netmask**, **Gateway**, and **Hostname**.

c. Click the **CIFS Sharing Enabled** and **NFS Sharing Enabled** checkboxes to enable or disable sharing. You can enable ordisable CIFS or NFS, or both.

3. If you enabled CIFS sharing, set the **CIFS Global Settings** as follows:

- **NetBIOS Workgroup**: the NetBIOS group to which the Samba server belongs (if AD not configured). Examples: `MYDOMAIN`, `XYUS`, etc.
- **NetBIOS Name**: the NetBIOS name of the Samba server. Examples might be CSCNG, CNG200.
- **Realm**: DNS-style AD realm. For example, mydomain.example.com.
- **Security**: depending on the service to be used to authenticate users (such as LDAP), the available options are ads, user, or share. For more information, see *Samba Security Modes* in the *Official Samba HowTo and Reference Guide*.

4. If desired, you may change the default values for **CIFS Winbind Settings** and **CIFS ACL Settings**. For more information about these settings, see the *Official Samba HowTo and Reference Guide*. A summary of these options is shown below:

CIFS Winbind Settings

- **Enum users**: On large installations using `winbindd(8)`, it may be necessary to set this option to **yes** to suppress the enumeration of users through the `setpwent()`, `getpwent()` and `endpwent()` group of system calls.
- **Enum groups**: On large installations using `winbindd`(8) it may be necessary to set this option to yes suppress the enumeration of groups through the `setpwent`(), `getpwent`() and `endpwent`() group of system calls.
- **Use default domain**: Set this parameter to **yes** if the `winbindd(8)` daemon should operate on users without domain component in their username.
- **Separator**: This parameter allows an administrator to define the character used when listing a username of the form of DOMAIN user.
- **Offline logon**: Set to **true** if Winbind should allow login with the `pam_winbind` module using Cached Credentials.
- **Cache time**: Specify the number of seconds the `winbindd(8)` daemon will cache user and group information before querying a Windows NT server again.

The preceding figure shows the fields **UID from:** and **to:** and **GID from:** and **to:**. These fields are deprecated on this screen. They can now only be used with the `cscli cng cifs idmap` commands. See the `-r` range option for *cng cifs idmap add*and *cng cifs idmap set*.

CIFS ACL Settings

- **NT ACL support**: Set to yes if `smbd`(8) should attempt to map UNIX permissions into Windows NT access control lists.
- **ACL compatibility**: This parameter specifies what OS ACL semantics should be compatible with. Normally enter `auto`. (Other values are `win2k` for Windows 2000 and above, and `winnt` for Windows NT 4.)
- **Map ACL flags**: Select **yes** if `smbd(8)` should attempt to map the **inherit** and **protected** access control entry flags stored in Windows ACLs into an extended attribute called `user.SAMBA_PAI`.
- **Store DOS attributes**: If this parameter is set to yes Samba attempts to first read DOS attributes (`SYSTEM`, `HIDDEN`, `ARCHIVE`, or `READ-ONLY`) from a filesystem extended attribute before mapping DOS attributes to UNIX permission bits (such as occurs with `map hidden` and `map readonly`).
- **Map hidden files**: Select **yes** if DOS-style hidden files should be mapped to the UNIX world execute bit.
- **Map system files**: Select **yes** if DOS-style system files should be mapped to the UNIX group execute bit.
- **Map archive files**: Select **yes** if the DOS archive attribute should be mapped to the UNIX owner execute bit.
- **Map readonly files**: Select **yes** if the DOS read only attribute should be mapped from a UNIX filesystem.
- **DOS file mode**: Select **yes** to allow a user who has write access to the file (by whatever means, including an ACL permission) to modify the permissions (including ACL) on the file.

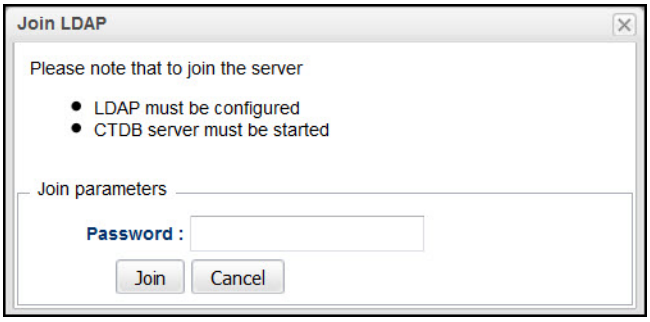
5. Click **Save**.

Once you save the settings; the **Join** button changes to display the relevant server name, based on the specified configuration settings. For example, based on the settings made above in Steps 2 through 4 and using default Winbind and ACL settings, the **Join LDAP** button is activated. If you are configuring CNG for AD you will see a Join AD option.

Figure 11. Join LDAP Button

6. Click **Join LDAP**. The **Join LDAP** window is displayed.

Figure 12. Join LDAP Window

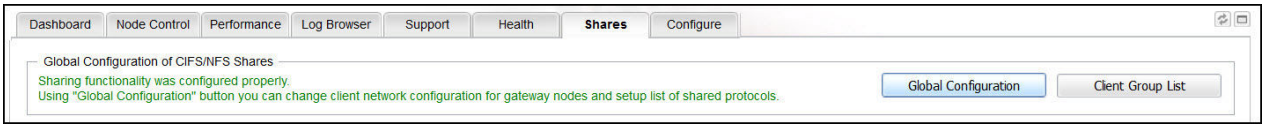


7. Enter the password and click **Join** to join the server.

   The LDAP server must already be configured and the CTDB (Clustered Trivial Database) server must be started in order to join the server.

8. Once the **Join parameters** settings are complete, click **Close**. A message displays confirming that sharing functionality was configured properly.

Figure 13. Sharing Confirmation



Define Client Groups - GUI

About this task

You can define groups of specific enterprise clients by IP address range or hostname. Then, when specifying access permissions and restrictions for shares, you can use a defined client group to refine the sharing level.

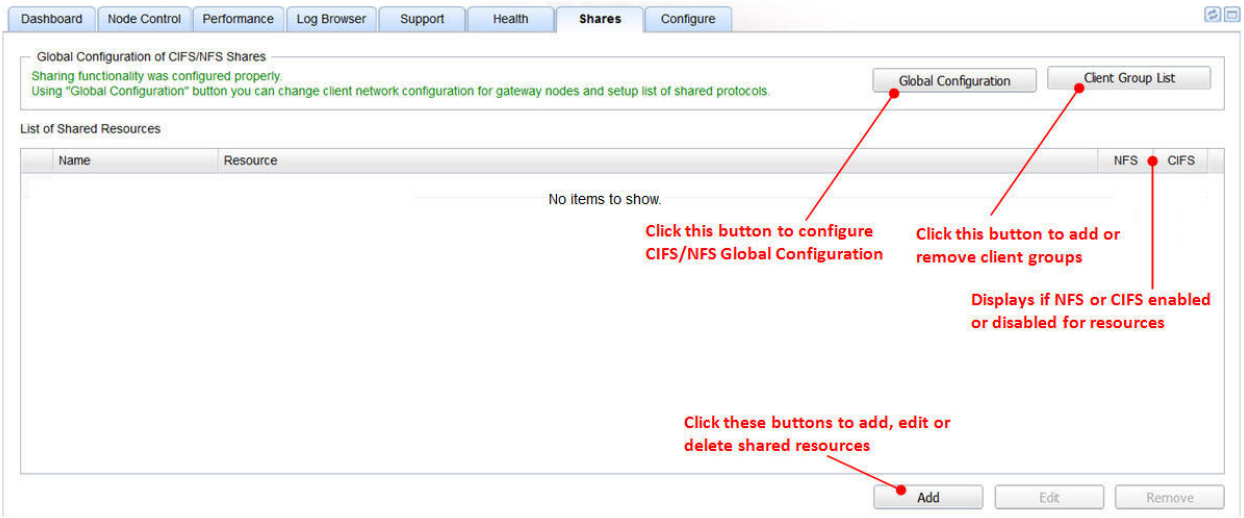
To define a client group:

Procedure

1. Navigate to the CSSM **Shares** tab.

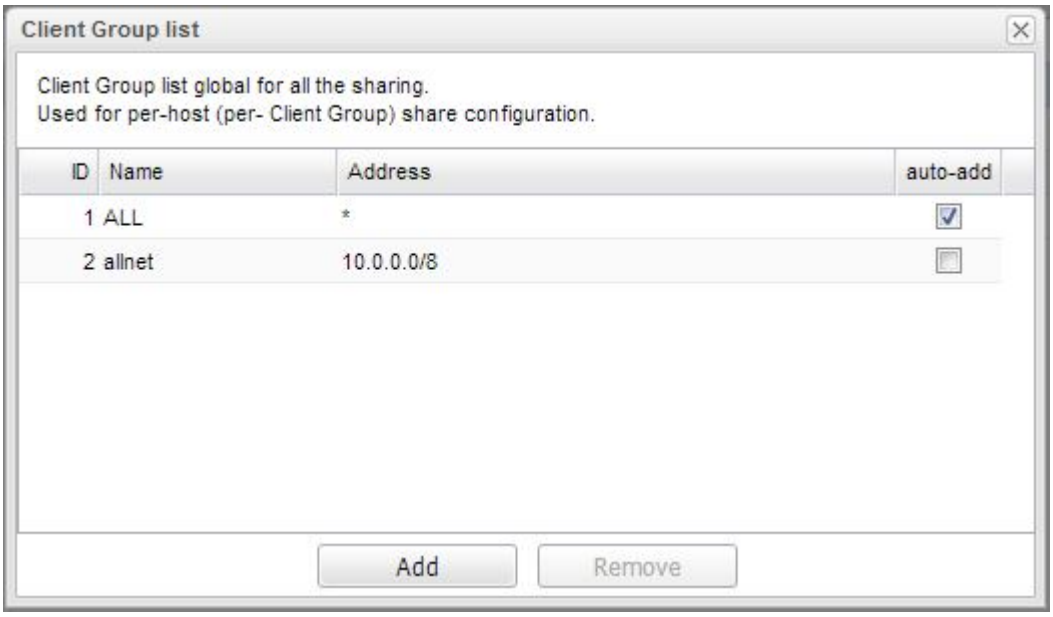
2. Click the **Client Group List** button.

Figure 14. Client Group List Button on Shares Tab



The **Client Group List** window displays.

Figure 15. Add Group on Client Group List Window



3. Click **Add** to define a new client group.

4. Enter a name for the client group.

5. Enter the address of the client group, in the form of an IP address range or hostname.

   An IP address range can be specified in CIDR notation such as xxx.xxx.xxx.xxx/xxx. For example, the CIDR notation 192.168.100.0/22 represents the range of addresses 193.168.100.0 through 192.168.103.255.

6. (Optional) Check **auto-add** to have the client group automatically added to NFS shares when they are defined.

The client group named ALL is predefined on new Sonexion systems and configured by default with the **auto-add** option selected. If you do not want all NFS shares to be configured automatically with this client group, you must uncheck the **auto-add** check box.

7. Click **Add** once more to save the client group.

8. Click **Close** once the client group is saved.

9. To add additional client groups, repeat Steps 3 through 8.

Add Shared Resources - GUI

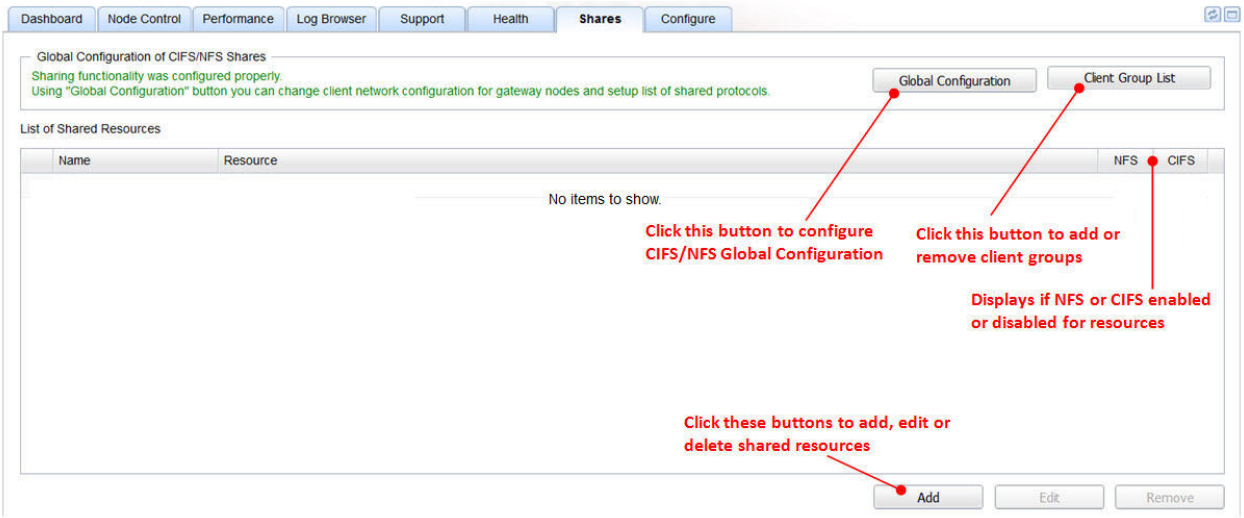
About this task

Follow this procedure to add shared resources that will be available for access by enterprise clients.

Procedure

1. Navigate to the CSSM **Shares** tab.

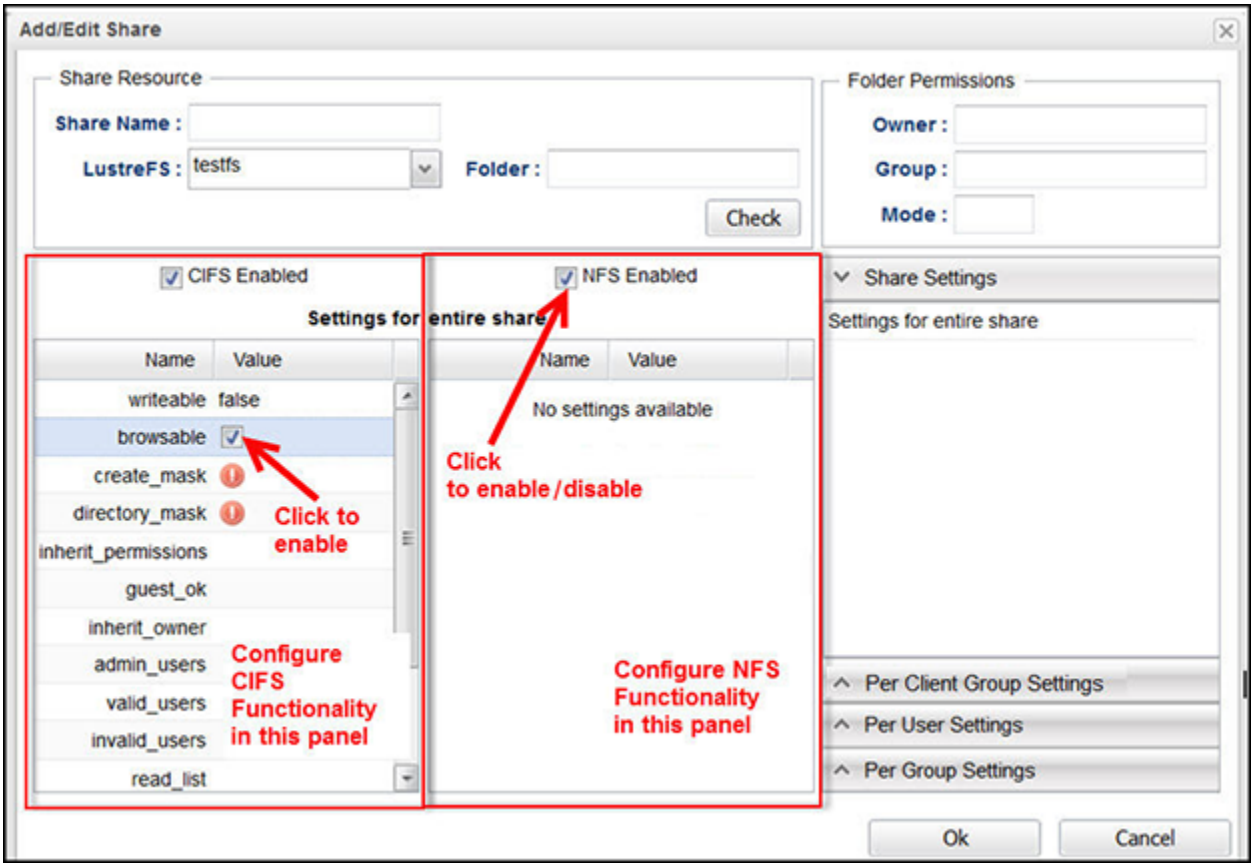
   Figure 16. Shares Tab



2. Click the Add button.

   The **Add/Edit Share** window displays.

Figure 17. Add a Shared Resource



There are four areas in the **Add/Edit Share** window:

- Share Resource
- Folder Permissions
- CIFS Enabled panel and NFS Enabled panel
- Buttons to configure CIFS and NFS Settings for the entire share, and to refine access permissions and restrictions per client group, per user, and per group

3. Set the **Share Resource** information:

a. Enter a name for this shared resource in the **Share Name** field.

b. Select the Lustre filesystem name from the **LustreFS** drop-down arrow.

c. Enter the name of the Lustre file system **Folder** that is to be shared. This can be an existing folder or you can create new folder. When creating a new folder you can skip the following step (3.d).

d. If you specify a folder that already exists, click **Check** to make sure that the folder name you entered is recognized. If the folder name is unknown, the system will display a message that the folder does not exist and will be created.

4. Set the global folder permissions that will be granted to enterprise clients for accessing the shared folder:

a. Enter the **Owner**, **Group**, and **Mode** permissions.

Mode permissions can be entered in standard Linux format, such as 777 for full read/write/execute access for owner/group/user.

5. Enable CIFS, NFS, or both on the shared folder by checking the appropriate checkbox; **CIFS Enabled** or **NFS Enabled**.

   When you check a checkbox, the related configuration settings will become available in the panel below the checkbox. When you uncheck a checkbox, the related configuration options will be grayed out.

6. Specify the global CIFS and NFS Share Settings.

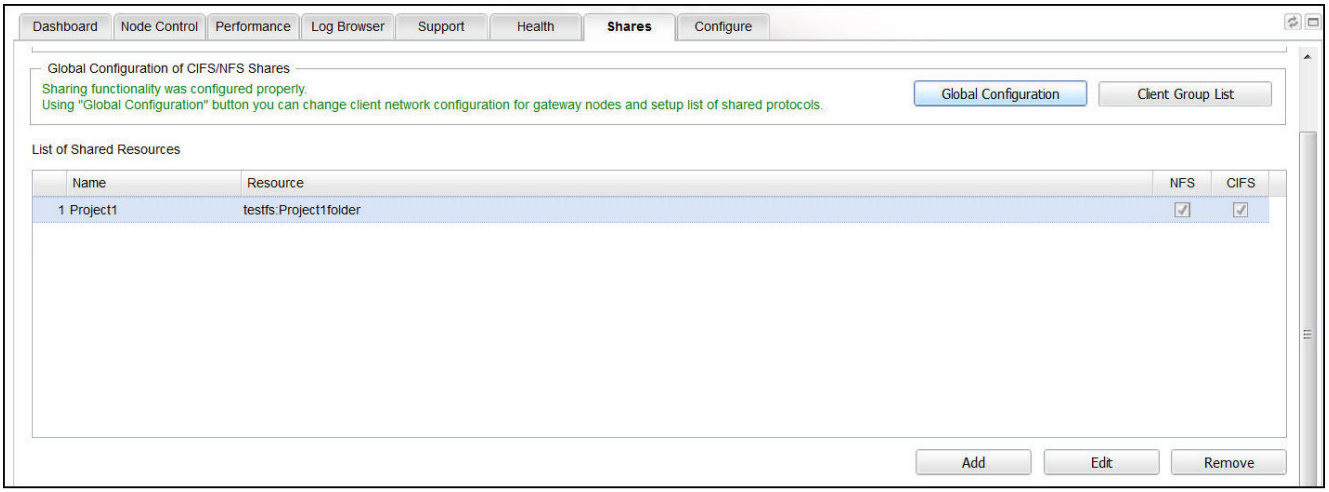
   Enter any values for those settings that you want to configure. Enter the values in the panel that appears below the related checkbox; **CIFS Enabled** or **NFS Enabled**. The setting **Name** is on the left and the **Value** you specify is on the right. For more information about these settings, see the *Official Samba HowTo and Reference Guide*. These settings can be changed any time.

7. If desired, you may refine the access permissions and restrictions **Per Client Group**, **Per User**, and **Per Group**. For more information about per-client-group sharing, see *Specify Permissions and Restrictions by Group - GUI* on page 32.

8. Click OK to create the new shared resource and apply the global share settings.

   The example shown in the following, shows a resource named Project1 that is configured for CIFS and NFS.

   Figure 18. Shared Resource for CIFS and NFS



Specify Permissions and Restrictions by Group - GUI

About this task

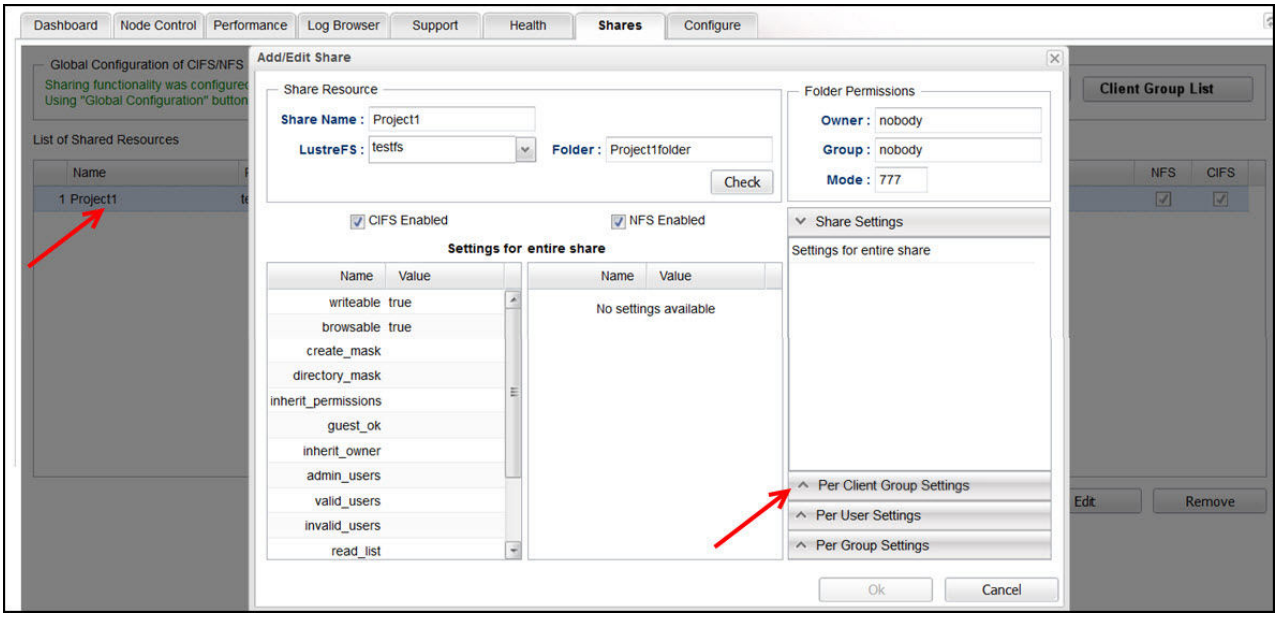
You can refine access permissions and restrictions for client groups that you have already defined. See *Define Client Groups - GUI* on page 28. Follow the procedures in this section to refine sharing access by client group.

Procedure

1. On the CSSM **Shares** tab, double-click the shared resource to which you want to add per client group settings.

2. Click **Per Client Group Settings**. See the following figure.

Figure 19. Set Options for Defined Client Groups

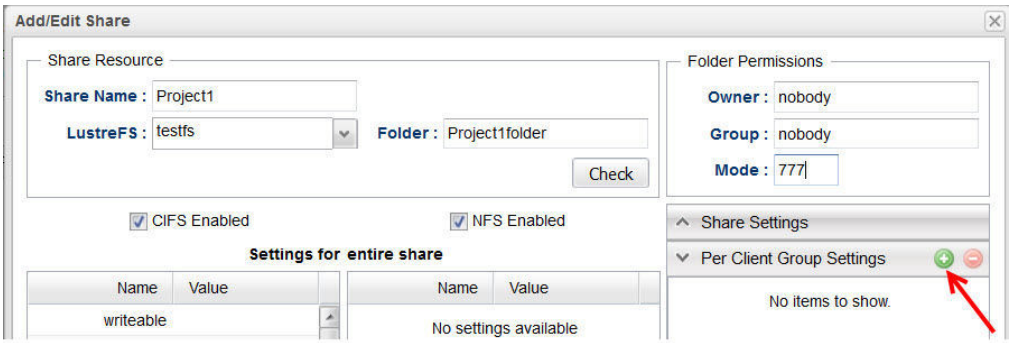


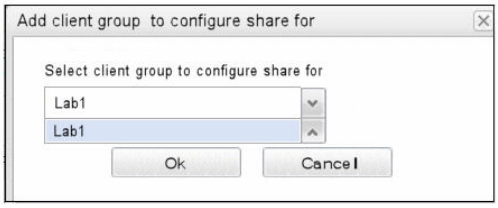
If no per-client-group settings are defined, the panel displays "No host access configured".

3. On the **Add/Edit Share** window, click the button (indicated by a red arrow in the following figure) to add a defined client group.

The window **Add client group to configure share for** is displayed, shown second in the following figure.

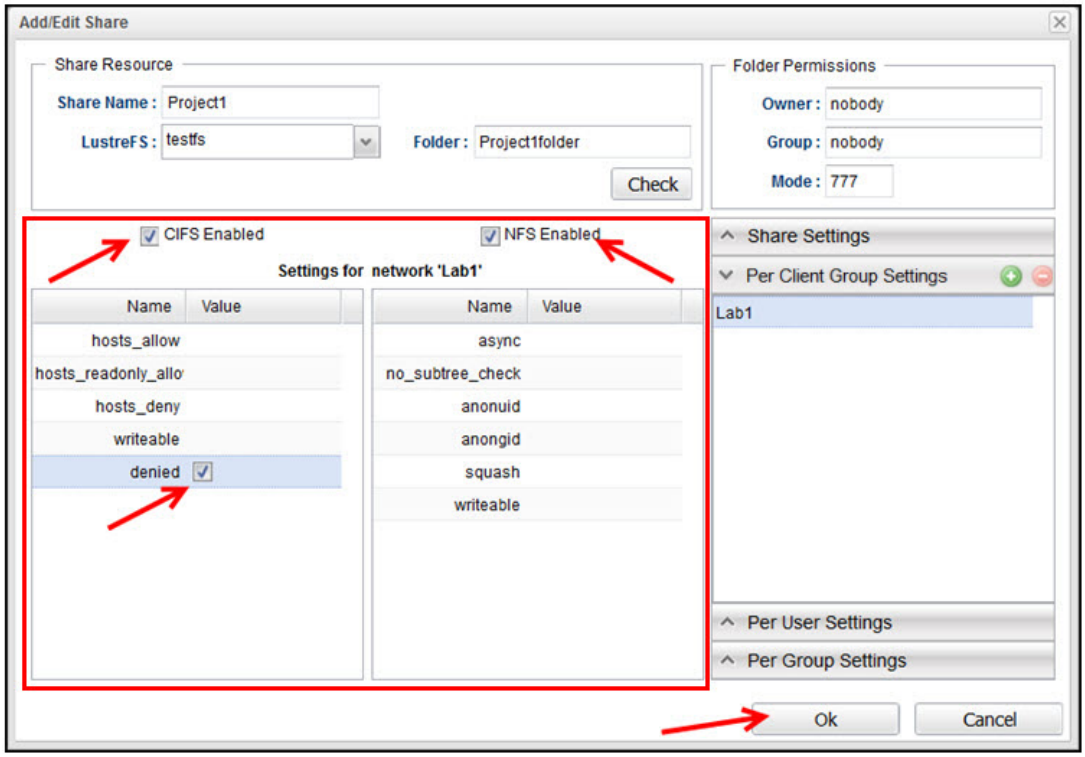
Figure 20. Create Per-client-group Settings





4. Select the client group name from the list of defined client groups, and then click OK. The name of the client group appears under the **Per Client Group Settings** button, as shown in the following figure (using the group name "Lab1").

Figure 21. Client Group Selected for CIFS and NFS Settings



5. As shown in the above figure, check the appropriate checkbox (**CIFS Enabled** or **NFS Enabled**) to enable CIFS, NFS, or both on the shared folder for the enterprise clients that belong to the selected client group.

   When you check a checkbox, the related configuration settings become available in the panel below the checkbox. When you uncheck a checkbox, the related configuration options are grayed out.

6. Specify the CIFS and NFS **Per Client Group Settings** for the enterprise clients that belong to the selected client group.

   Enter any values for settings that you want to configure. Enter values in the panel that appears below the related checkbox, **CIFS Enabled** or **NFS Enabled**. The **Name** setting is on the left, and the Value you specify is on the right. For more information about these settings, see the *Official Samba HowTo and Reference Guide* . In the example settings shown in the preceding figure, selecting the **Denied** checkbox denies access to the shared resource "Project1" for any enterprise clients that are members of the "Lab1" client group. These settings can be changed any time.

7. Repeat Steps 3 through 6 for any additional client groups for which you want to configure access settings to the shared resource.

8. Click **OK** to save the settings.

Test Access to Shared Resource

About this task

An enterprise client such as Windows, Macintosh, or Linux can connect to the server by using its native network connection service and entering a CNG node IP address that was set in step *2.a* on page 26. Once connected to the server, the client user will be prompted to enter a user name and password. The user credentials are validated against the security model that was set up when configuring CNG (that is, LDAP, AD, NIS).

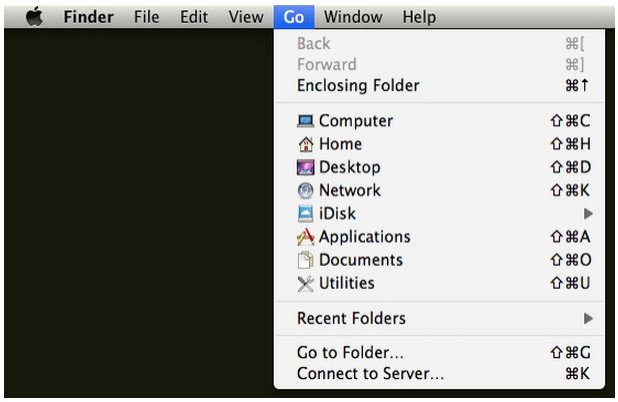
Once the user has connected to the server and been authenticated, they will be able to access files on the shared resource, create folders, and create new files.

For example, the following steps illustrate how a Macintosh client on the ECN can connect to shared resources in the Lustre filesystem.

Procedure

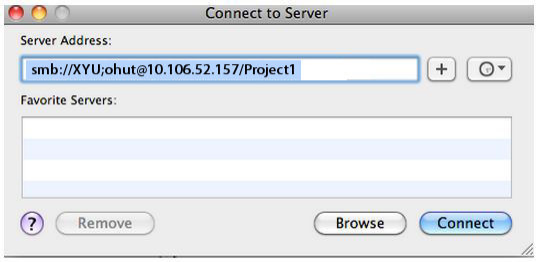
1. Open the **Finder** application. From the menu, choose **Go** > **Network**.

   Figure 22. Macintosh Network Menu



2. In the Connect to Server window, enter the server address for the CNG node(s). Use the smb format for the server address, as shown in this figure:

   Figure 23. Macintosh Connect to Server Menu

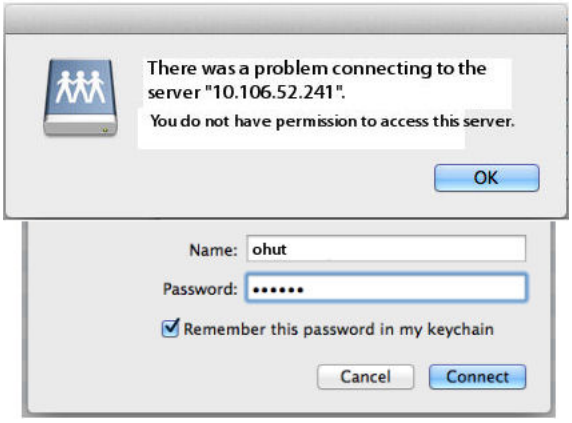


3. Click **Connect**. The server will ask for a username and password..

4. Enter the login credentials and click **Connect**.

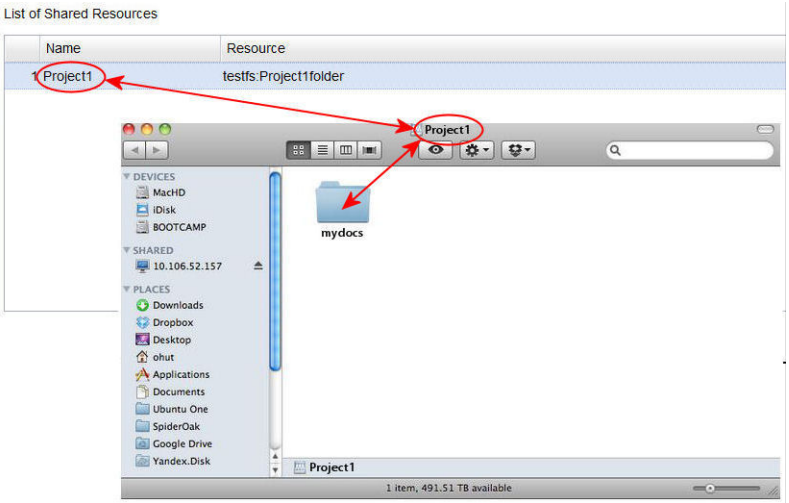
- If the credential are not entered correctly, an error will be displayed:

Figure 24. Macintosh Login Error



- If the credentials are valid, the Finder application will open and display the contents of the shared resource. The following figure shows a MAC connected to the shared resource named Project1, which is a Lustre folder. The MAC user has created a new subfolder named mydocs, which can be seen in the Finder application.

Figure 25. Successful Connection from Macintosh to Shared Resource



Manage CNG After Initial Configuration

IMPORTANT: Most CNG options that are set by CSCLI commands must be saved by running the `cng apply` command. If changes are not saved they may not take effect as expected. In some instances, you must run the `cng apply` command before running another command. It is recommended to save your configuration and share settings frequently to ensure that your changes take effect. Following is the general form of the command:

```
[admin@n000]$ cscli cng apply --restart-cifs -y
```

Command Reference: *cng apply*

Notes:

- `--restart-cifs` – Initiates a service restart after applying the changes to share options. Since some share option changes do not affect connected clients after they are applied, it may be desirable to use this option to initiate a service restart.
- A service restart interrupts any connected enterprise clients, which could cause errors if those clients are actively writing to files when interrupted.
- CSCLI usually informs the user with the following message (shaded text) if the changes being applied might not affect connected clients:

```
[admin@snx11000n000 ~]$ cscli cng share set -f testfs -s mydocs -S documents
cng: share 'documents' is updated.
```

- Changes may not affect currently connected CIFS clients.
- Exercise caution whenever you use the `-y` or `--yes` parameter.

The following procedure topics describe how to use the CSSM and the CLI to perform tasks related to listing and managing shared resources, including:

- List all of the shared resources that have been created
- Show the configuration settings for a specific share
- Change the configuration settings for a specific shared resource
- Enable or disable CIFS and/or NFS sharing
- Change the CIFS settings for a specific shared resource
- List the client groups used for a specific CIFS shared resource
- Add a client group for a specific CIFS shared resource
- List the client groups used for a specific NFS shared resource
- Add a client group for a specific NFS shared resource
- List client groups that have been defined for use with CIFS and NFS shared resources
- Change the options for a specific client group

Manage CIFS/NFS Shares with CLI

Sections in this topic describe the use of CSCLI commands to manage CIFS and NFS shares that have previously been defined:

- *Use of cscli cng share list* on page 38
- *Use of cscli cng share show* on page 38
- *Use of cscli cng share set* on page 39
- *Use of cscli cng share (enable, disable)* on page 40
- *Use of cscli cng share cifs set* on page 41
- *Use of cscli cng share cifs clientgroup (list, set)* on page 42
- *Use of cscli cng share nfs clientgroup (list, set)* on page 43
- *Use of cscli cng clientgroup (list, set)* on page 44

Use of `cscli cng share list`

This command is used to list all of the shared resources that have been created.

The general form of the command is:

```
[admin@n000]$ cscli cng share list
```

Command Reference: *cng share list*

Example

Following is an example of the command and its output, showing that there are 5 shares defined for the testfs filesystem:

```
[admin@snx11000n000 ~]$ cscli cng share list
filesystem share name           path
-------------------------------------------------
testfs     testshare            testshare
testfs     mydocs               documents
testfs     lab2_results         lab2_results
testfs     svn4                 svn4
testfs     brezerk              brezerk
```

Use of `cscli cng share show`

This command is used to show the configuration settings for a specific share.

The general form of the command is:

```
[admin@n000]$ cscli cng share show -f filesystem_name -s share_name -p path
```

Command Reference: *cng share show*

Notes:

- *filesystem_name* – Name of the Sonexion filesystem
- *share_name* – Name given to the shared resource when created
- *path* – Path and folder name to the shared resource on the filesystem

Example

The following example shows the general settings for the share named mydocs on the testfs filesystem. In the command output you can see that the shared Lustre directory is named documents, and the CIFS protocol is enabled (NFS is disabled). The CIFS options show that the shared resource is read only, and guest access is permitted.

```
[admin@snx11000n000 ~]$ cscli cng share show -f testfs -s mydocs
    mydocs:
        uid        : nobody
        gid        : nobody
        mode       : 511
        path       : documents
        filesystem : testfs
        enabled protocols: cifs
        description: documents
        cifs options:
            writeable no
            inherit_permissions yes
            guest_ok yes
            browsable yes
        networks:
            nfs * writeable: true, squash: root
```

Use of `cscli cng share set`

Use this command to modify the configuration settings for a specific shared resource.

The general form of the command is:

```
[admin@n000]$ cscli cng share set -s share_name -f filesystem_name -p path -S
share_name_new -d description -U UID -g GID -M mode
```

Command Reference: *cng share set*

Notes:

- *filesystem_name* – Name of the Sonexion filesystem
- *share_name* – Name given to the shared resource when created
- *path* – Path and folder name for the shared resource on the filesystem
- *share_name_new* – New name for the shared resource

Example

The following example changes the name of the shared resource mydocs on the testfs filesystem to documents. The command output shows that the new share name is `documents`, the shared Lustre directory is named `documents`, and the CIFS protocol is enabled (NFS is disabled). The CIFS options show that the shared resource is read only, and guest access is permitted.

```
[admin@snx11000n000 ~]$ cscli cng share set -f testfs -s mydocs -S documents
cng: share 'documents' is updated.
cng: Note: Changes may not affect currently connected CIFS clients.
    documents:
        uid        : nobody
        gid        : nobody
        mode       : 511
```

```
        path       : documents
        filesystem : testfs
        enabled protocols: cifs
        description: documents
        cifs options:
            writeable no
            inherit_permissions yes
            guest_ok yes
            browsable yes
        networks:
            nfs * writeable: true, squash: root
```

Use of `cscli cng share (enable, disable)`

Use these commands to enable or disable CIFS and/or NFS sharing for a specific shared resource.

- The general form of the enable command is:

  ```
  [admin@n000]$ cscli cng share enable -f filesystem_name -s share_name -p path
  -P protocol
  ```

  Command Reference: *cng share enable*

- The general form of the `disable` command is:

  ```
  [admin@n000]$ cscli cng share disable -f filesystem_name -s share_name -p path
  -P protocol
  ```

  Command Reference: *cng share disable*

Notes:

- *filesystem_name* – Name of the Sonexion filesystem
- *share_name* – Name the shared resource was given when created
- *path* – Path and folder name for the shared resource on the filesystem
- *protocol* – Protocol to enable; options are `cifs` or `nfs`. Omit this option to enable both CIFS and NFS.

Example 1

The following example shows how to enable CIFS exporting for the shared resource named documents on the testfs filesystem:

```
[admin@snx11000n000 ~]$ cscli cng share enable -f testfs -s documents -P cifs
cng: exporting of share 'documents' is enabled by 'cifs'.
```

Example 2

The following example shows how to enable both CIFS and NFS exporting for the shared resource named documents on the testfs filesystem. Note that the `-P` option has been left off of the command:

```
[admin@snx11000n000 ~]$ cscli cng share enable -f testfs -s documents
cng: exporting of share 'documents' is enabled by 'cifs'.
cng: exporting of share 'documents' is enabled by 'nfs'.
```

Example 3

The following example shows how to disable NFS exporting for the shared resource named documents on the `testfs` filesystem:

```
[admin@snx11000n000 ~]$ cscli cng share disable -f testfs -s documents -P nfs
cng: exporting of share 'documents' is disabled by 'nfs'.
```

Example 4

The following example shows how to disable both CIFS and NFS exporting for the shared resource named documents on the testfs filesystem. Note that the `-P` option has been left off of the command:

```
[admin@snx11000n000 ~]$ cscli cng share disable -f testfs -s documents
cng: exporting of share 'documents' is disabled by 'cifs'.
cng: exporting of share 'documents' is disabled by 'nfs'.
```

Use of `cscli cng share cifs set`

Use this command to change the CIFS settings for a specific shared resource.

The general form of the command is:

```
[admin@n000]$ cscli cng share cifs set -s share_name -f filesystem_name -p path
--writeable yes,no --browsable yes,no --guest-ok yes,no --valid-users
'valid_users'
```

Command Reference: *cng share cifs set*

Notes:

- *filesystem_name* – name of the Sonexion filesystem
- *share_name* – the name the shared resource was given when created
- *path* – the path and folder name to the shared resource on the filesystem
- `--writeable` – specify yes to allow the user to write to the share
- `--browsable` – specify yes if the share should be viewable in the list of available shares and in the browse list
- `--guest-ok` – specify yes if no password is required to access the shared resource
- *valid_users* – enter a specification that defines the users who can connect to the shared resource

This command has numerous other options. For more information see `cscli cng share cifs set`.

Example

The following example changes the CIFS share settings for the shared resource named documents on the testfs filesystem, so that 1) the shared resource is writeable and 2) new files and directories created on the shared resource inherit the ownership properties of the shared resource. The command output shows all of the CIFS options that are set after running this command:

```
[admin@snx11000n000 ~]$ cscli cng share cifs set -f testfs -s documents --writeable yes --inherit-owner yes
cng: CIFS configuration of share 'documents' is updated.
cng: Note: Changes may not affect currently connected CIFS clients.
        cifs options:
            writeable yes
            inherit_permissions yes
            inherit_owner yes
            guest_ok yes
            browsable yes
```

Use of `cscli cng share cifs clientgroup (list, set)`

Use these commands to list the client groups used for a specific CIFS shared resource and to add a client group for a specific CIFS shared resource.

- The general form of the `list` command is:

  ```
  [admin@n000]$ cscli cng share cifs clientgroup list -s share_name -f
  filesystem_name -p path
  ```

  Command Reference: *cng share cifs clientgroup list*

- The general form of the `set` command is:

  ```
  [admin@n000]$ cscli cng share cifs clientgroup list -s share_name -f
  filesystem_name -p path -N clientgroup_name --writeable yes,no --denied yes,no
  ```

  Command Reference: *cng share cifs clientgroup set*

Notes:

- *filesystem_name* – Name of the Sonexion filesystem
- *share_name* – Name that the shared resource was given when created
- *path* – Path and folder name for the shared resource on the filesystem
- *clientgroup_name* – Name of the client group to add to a specific CIFS shared resource
- `--writeable` – Specify **yes** to make the share writeable to members of the client group. Specify **no** to make the share read only.
- `--denied` – Specify **yes** to prevent members of the client group from accessing the shared resource. Specify **no** to allow access.

Example 1

The following example indicates that only one client group is set for the CIFS shared resource named documents on the testfs filesystem. The command output displays the client group network specification (`network/cidr`) rather than the client group name.

```
[admin@snx11000n000 ~]$ cscli cng share cifs clientgroup list -f testfs -s documents
            cifs 10.0.0.0/8
```

If there were no client groups set for this shared resource the command output would state, "`cng: There is no any CIFS client groups for the share.`"

Example 2

The following example sets the client group named `lab08` for the shared resource named documents on the `testfs` filesystem. Note that the command output displays the client group network specification (network/cidr) rather than the client group name.

```
[admin@snx11000n000 ~]$ cscli cng share cifs clientgroup set -f testfs -s documents -N lab08
            cifs 10.0.0.0/8
```

Example 3

The following example performs the same task as the previous example, but sets it so that members of the client group are denied access to the shared resource.

```
[admin@snx11000n000 ~]$ cscli cng share cifs clientgroup set -f testfs -s documents -N lab08 --denied yes
            cifs 10.0.0.0/8 denied: true
```

Use of cscli cng share nfs clientgroup (list, set)

Use these commands to list the client groups used for a specific NFS shared resource and to add a client group for a specific NFS shared resource.

The general form of the `list` command is:

```
[admin@n000]$ cscli cng share nfs clientgroup list -s share_name -f
filesystem_name -p path
```

Command Reference: *cng share nfs clientgroup list*

The general form of the `set` command is:

```
[admin@n000]$ cscli cng share nfs clientgroup set -s share_name
     -f filesystem_name -p path -N clientgroup_name --writeable yes,no
     --async yes,no --squash root,no-root,all --subtree-check yes,no
     --anonuid anonuid --anongid anongid
```

Command Reference: *cng share nfs clientgroup set*

Notes:

- *filesystem_name* – Name of the Sonexion filesystem
- *share_name* – Name the shared resource was given when created
- *path* – Path and folder name to the shared resource on the filesystem
- *clientgroup_name* – Name of the client group to add to a specific NFS shared resource
- `--writeable` – Specify **yes** to make the share writeable to members of the client group. Specify **no** to make the share read only.
- `--async` – Specify **yes** to allow the NFS server to reply to requests before any changes made by that request have been committed to stable storage.
- `--squash` – Specify the method of UID/GID squashing. Options are `root`, `no_root`, and `all`. Use `--squash-unset` to unset the squash setting.
- `--subtree` – Specify yes to enable subtree checking; no to disable.
- *anonuid* – UID of the anonymous account
- *anongid* – GID of the anonymous account

Example 1

The following example sets the client group named `lab08` for the shared resource named documents on the testfs filesystem. It also allows the members of the client group to write to the shared resource. Finally, it sets the method of UID/GID squashing to map requests from UID/GID 0 to the anonymous UID/GID. Note that the command output displays the client group network specification (network/cidr) rather than the client group name.

```
[admin@snx11000n000 ~]$ cscli cng share nfs clientgroup set -f testfs -s documents -N lab08 --writeable yes --squash root
           nfs 10.0.0.0/8 writeable: true, squash: root
```

Example 2

This example indicates that one client group is set for the NFS shared resource named documents on the `testfs` filesystem. Note that the command output displays the client group network specification (network/cidr) rather than the client group name.

```
[admin@snx11000n000 ~]$ cscli cng share nfs clientgroup list -f testfs -s documents
            nfs * writeable: true, squash: root
            nfs 10.0.0.0/8 writeable: true, squash: root
```

Use of `cscli cng clientgroup (list, set)`

Use these commands to list client groups that have been defined for use with CIFS and NFS shared resources and to change the options for a specific client group.

- The general form of the `list` command is:

  ```
  [admin@n000]$ cscli cng clientgroup list
  ```

  Command Reference: *cng clientgroup list*

- The general form of the `set` command is:

  ```
  [admin@n000]$ cscli cng clientgroup set -N clientgroup_name --auto-add yes,no
  --network network_spec | --domain domain_glob
  ```

  Command Reference: *cng clientgroup set*

Notes:

- *clientgroup_name* – Name of the client group to be changed
- `--auto-add` – Specify **yes** to have this client group added to NFS shares automatically when they are created. Specify **no** if they are not to be added automatically.
- *network_spec* – Network specification (in `network/cidr form`) for the clients that are members of the client group
- *domain_glob* – Domain glob specification

Example 1

This example of the cng clientgroup list command shows that there are three client groups defined for the CNG installation. Note that the client group named `ALL` is pre-defined.

```
[admin@snx11000n000 ~]$ cscli cng clientgroup list
name            spec            auto-add
ALL             *               yes
testlab4        10.0.0.0/24     yes
lab08           10.0.0.0/8      yes
```

Example 2

The following example shows how to change the settings of the client group named `lab08` so that it is not automatically added to new shares. In the list command that follows, the entry for `lab08` indicates that auto-add is off.

```
[admin@snx11000n000 ~]$ cscli cng clientgroup set -N lab08 --auto-add no
cng: Client group 'lab08' is updated.
[admin@snx11000n000 ~]$ cscli cng apply -y
[admin@snx11000n000 ~]$ cscli cng clientgroup list
name            spec            auto-add
ALL             *               yes
testlab4        10.0.0.0/24     yes
lab08           10.0.0.0/8      no
```

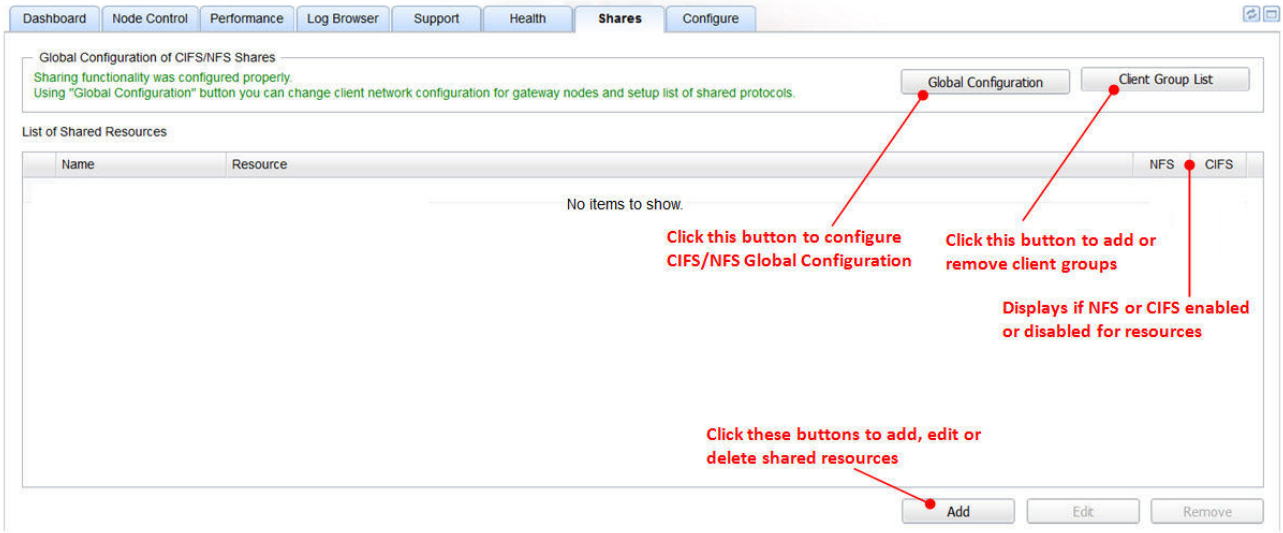
Manage CIFS/NFS Shares with GUI

About this task

This topic introduces the of the CSSM GUI to list and manage CIFS and NFS shares.

To list all shares that have been defined, navigate to the **Shares** tab. The list of shares is shown, displaying the name of each share, the filesystem resource the share is associated with, and whether the share is enabled for CIFS and NFS exporting.

Figure 26. Shares Tab



Other highlights of the **Shares** tab include:

- **Global Configuration** button – Set global configuration options for sharing.
- **Client Group List** button – Open the list of client groups.
- **Add** button – Define a new share.
- **Edit** button – Display and change the settings for A shared resource.
- **Remove** button – Delete a shared resource.

Set Shared Resource, Enable Sharing - GUI

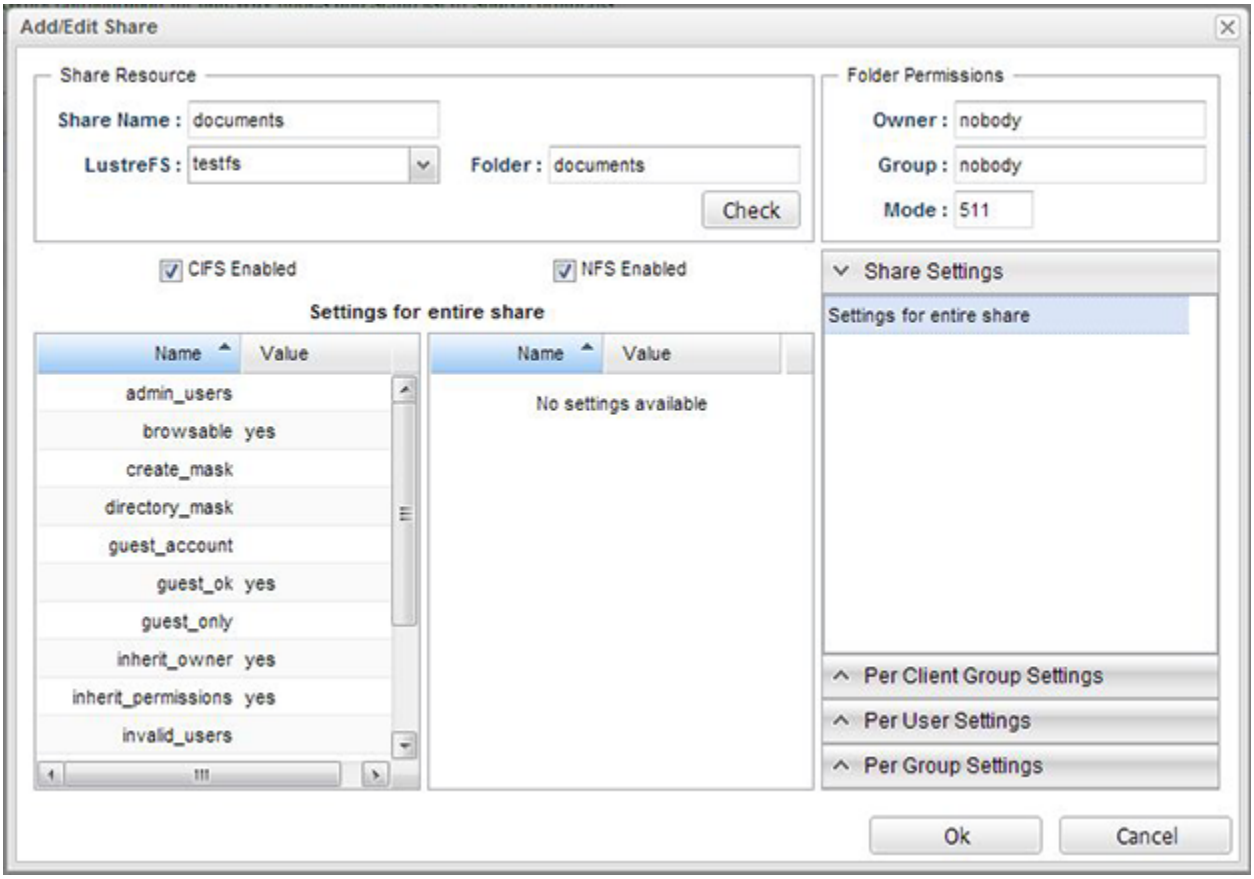
About this task

To display and change the settings for a shared resource, enable or disable CIFS/NFS sharing, and change the CIFS settings for a shared resource:

Procedure

1. Select the name of the share on the **Shares** tab.
2. Click **Edit**. The **Add/Edit Share** window opens.

Figure 27. Add/Edit Share Window



The example window above displays details of the current settings for the shared resource named documents on the testfs filesystem. This share is enabled for CIFS and NFS exporting. Also, for CIFS the shared resource is writable and browsable, inherits owner and permissions, and guest access is allowed.

3. To change the CIFS and/or NFS share settings, make any desired changes on this window. Example:

a. You can change the name for this shared resource in the **Share Name** field.

b. Change **Folder Permissions** by entering new values for the **Owner**, **Group**, and **Mode**.

c. Enable or disable CIFS and/or NFS exporting for the specific shared resource by checking or unchecking the appropriate checkbox; CIFS Enabled or NFS Enabled.

d. Specify the **CIFS Share Settings**.

Enter any values for those settings that you want to configure. For example, you can change the options writable, browsable, inherit_permissions, guest_ok, inherit_owner, etc. For more information about these settings, see the *Official Samba HowTo and Reference Guide* or the command reference: *cng share cifs set*.

These settings can be changed any time.

4. Click **OK** to change the share settings.

Client Groups for a Shared Resource - GUI

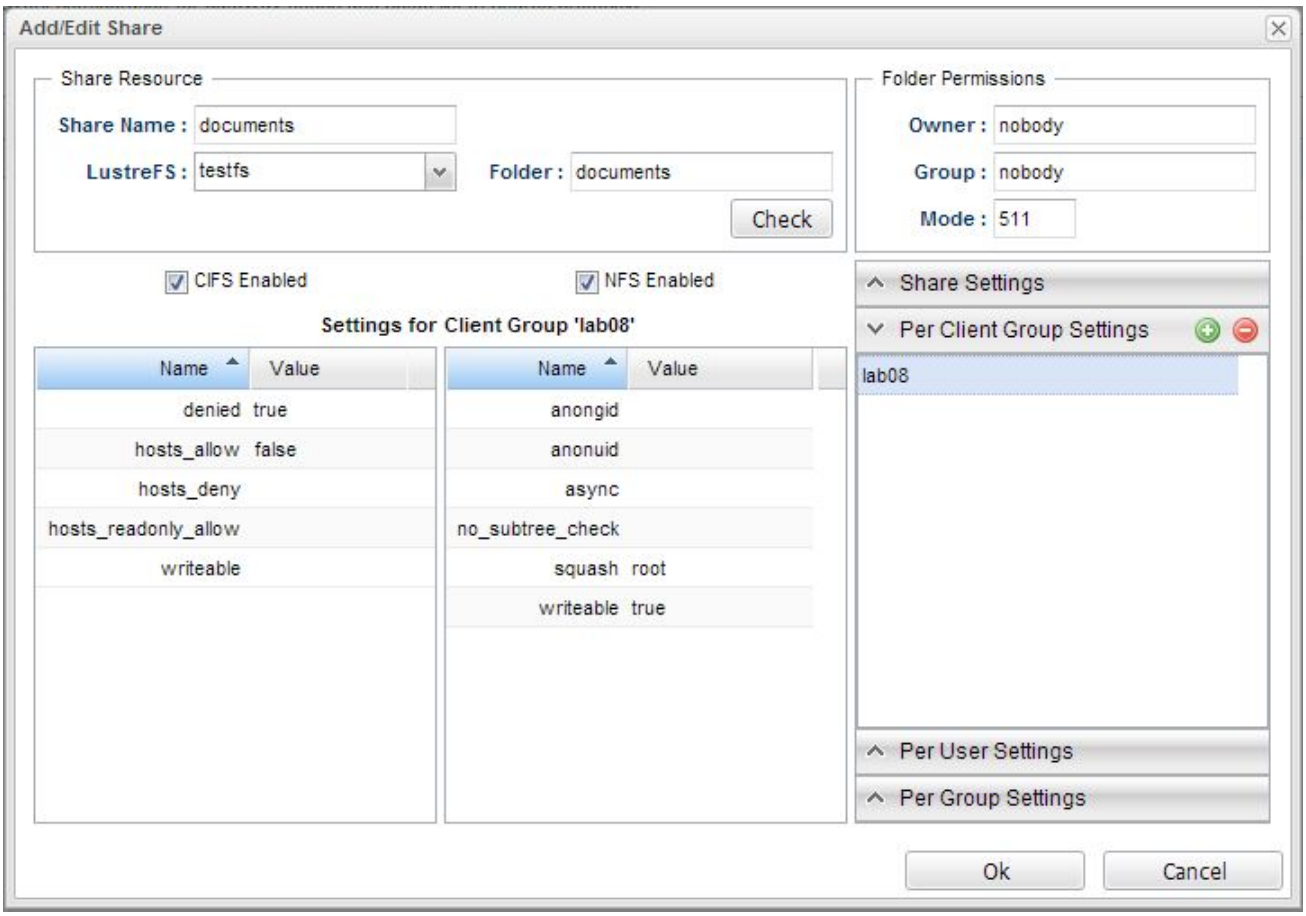
About this task

To display, add, or change the client groups used for a specific shared resource:

Procedure

1. Select the name of the share on the **Share** tab.
2. Click **Edit**. The **Add/Edit Share** window opens.
3. Click **Per Client Group Settings**. The **Add/Edit Share** window changes to display the client groups that are assigned to the share.

   Figure 28. Client Group Settings for Shared Resource



   Client groups used for the share are listed below the **Per Client Group Settings**, and the buttons are displayed. In the example window above, one client group is used, named `lab08`. The CIFS and NFS settings are displayed to the left, beneath the **CIFS Enabled** and **NFS Enabled** check boxes. In this example, for CIFS, access is denied to the members of the client group. For NFS, the shared resource is writeable and the root squash method is set to root.

4. To change the per group share settings, make any desired changes for CIFS and/or NFS in the designated areas

For more information about the available settings, see the *Official Samba HowTo and Reference Guide* or the command reference entries: `cng share cifs clientgroup set` or `cng share nfs clientgroup set`.

5. If desired, you may further refine the access permissions and restrictions **Per User** and **Per Group**.

6. To add a new client group for a shared resource:

   a. While the **Per Client Group Settings** are displayed, click the button.

   b. From the **Add** client group to configure share for window, select the client group you want to add.

   c. Click **OK**.

   d. Set any CIFS and/or NFS share options for the new client group.

7. When you have finished making changes to the per client group share settings, click **OK** to save the changes.

Display Undefined Groups, Change Options

About this task

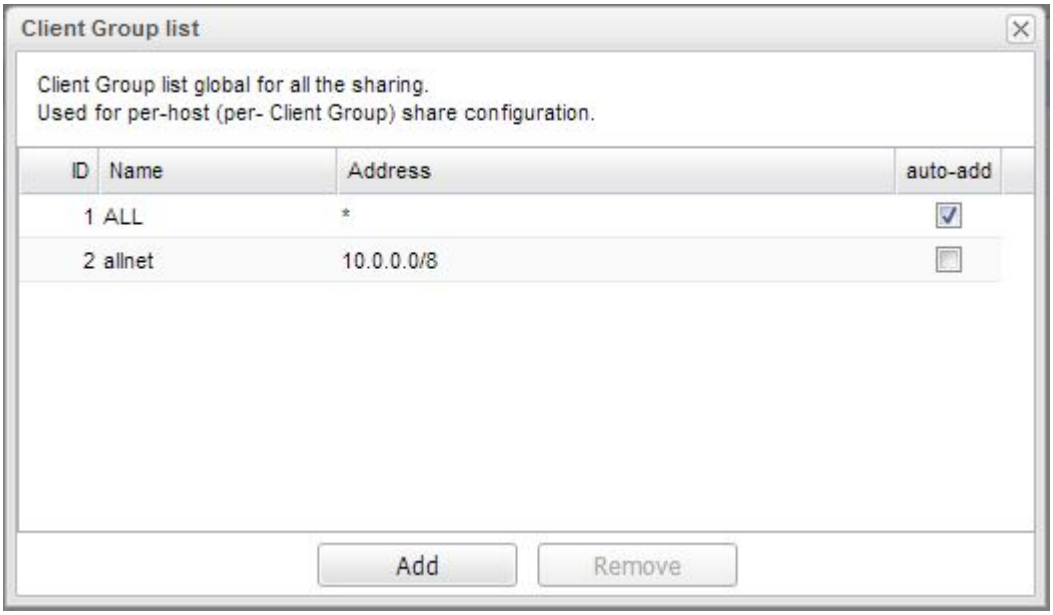
To display a list of client groups that have been defined, and to change the options for a client group:

Procedure

1. On the **Shares** tab click the **Client Group List** button.

   The **Client Group List** window appears.

   Figure 29. Client Group List Window



2. To change the settings for a defined client group:

   a. Select the **Name** field and enter a new name for the client group.

b. Select the **Address** field and enter a new network specification (in CIDR form) that includes the desired members.

c. Check the auto-add check box if you want the client group automatically added to new NFS shares when they are created. Uncheck this option if you do not want the client group added automatically to new NFS shares.

d. Click **Add** to save your changes.

3. To create a new client group:

a. Click **Add**.

b. Specify a name and address for the client group.

c. If desired, check the auto-add check box.

d. Click **Add** once more to save the new client group.

4. Click the **X** in the upper right corner to close the **Client Group List** window.

Delete Shares

About this task

This topic describes how to use the CSSM and the CLI to perform tasks related to listing and deleting shared resources, including:

- List all shared resources that have been created.
- Show configuration settings for a specific share.

Delete Shares Using the CLI

The following CSCLI commands are useful for determining what share you might want to delete and then performing the deletion:

- `cng share list`

  This command provides a list of all shares that have been defined. It is discussed in *cng share list*.

- `cscli cng share delete`

  Use this command to completely remove a shared resource, so that it is no longer available to enterprise clients. This does not delete the filesystem resource itself; it just removes the definition of the share.

  The general form of the command is:

  ```
  [admin@n000]$ cscli cng share delete -f filesystem_name -s share_name -p path
  ```

  Command Reference: *cng share delete*

Notes:

- *filesystem_name* – Name of the Sonexion filesystem
- *share_name* – Name the shared resource was given when created
- *path* – Path and folder name for the shared resource on the filesystem

Example

The following example command deletes the shared resource named testshare from the testfs filesystem.

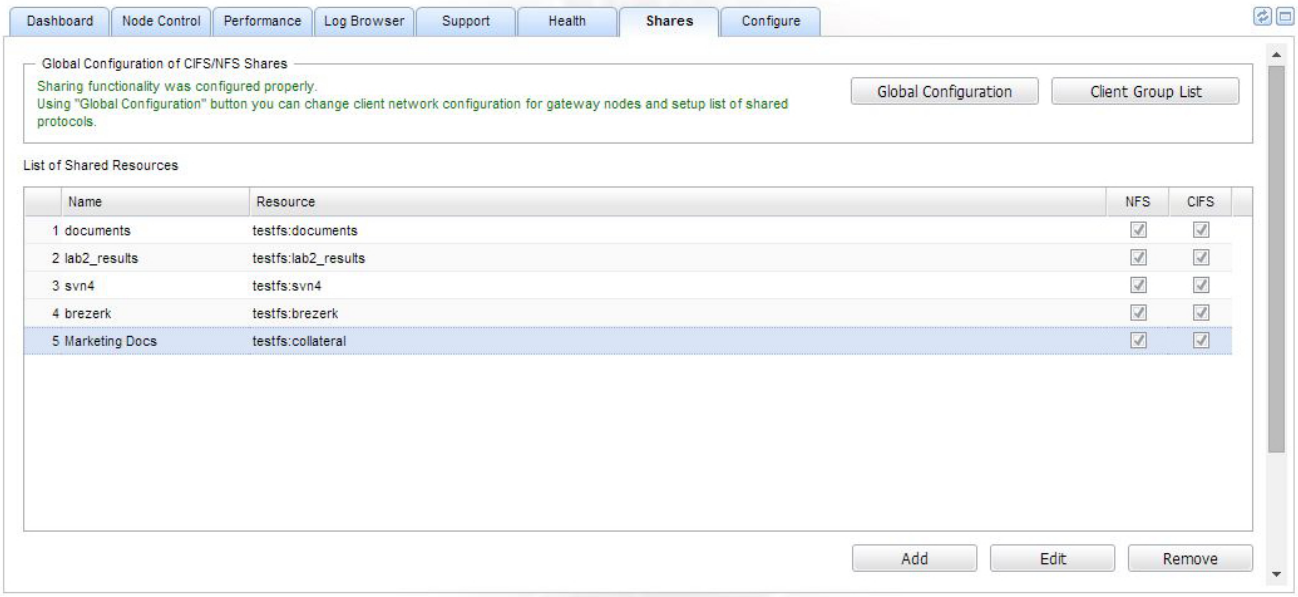
```
[admin@snx11000n000 ~]$ cscli cng share delete -s testshare -f testfs
cng: share 'testshare' is deleted.
```

Delete Shares Using the CSSM GUI

The following steps show how to use the CSSM GUI to list and delete CIFS and NFS shares.

To list all shares that have been defined, navigate to the Shares tab. The list of shares is shown, displaying the name of each share, the filesystem resource the share is associated with, and whether the share is enabled for CIFS and NFS exporting.

Figure 30. Shares Listed on Shares Tab



To delete a shared resource:

Procedure

1. Select the name of the share on the **Shares** tab.
2. Click **Remove**.
3. Click **OK** to confirm that you want to delete the shared resource.

Edit CIFS/NFS Configuration with CLI

The following CSCLI commands are used to display the current CNG configuration settings, change the global configuration settings, change the Winbind settings, and change the **idmap** options:

- *Use of cscli cng show all* on page 51
- *Use of cscli cng cifs globals* on page 53

- *Use of cscli cng cifs winbind* on page 54
- *Use of cscli cng cifs idmap (show, set)* on page 55

Use of `cscli cng show all`

Use this command to display all of the CNG configuration settings.

The general form of the command is:

```
[admin@n000]$ cscli cng show [all,node,winbind,acl,clientgroup,interface]
```

Command Reference: *cng show*

Notes:

- *all* – displays all CIFS/NFS settings
- *node* – displays node settings
- *winbind* – displays the Winbind settings
- *acl* – displays the ACL settings
- *clientgroup* – displays configuration settings for all defined client groups
- *interface* – displays the configuration of the current ECN Export Interface

Example 1

This example of the `cscli cng show all` command illustrates what information is available from the variations of the `cscli cng show` command. For example, in the command output below, the subsection named winbind: that appears beneath the section titled CIFS Global Configuration: is what is returned when you run the `cscli cng show winbind` command.

```
[admin@snx11000n000 ~]$ cscli cng show all
Export protocols:
    CIFS      : enabled
    NFS       : enabled
ECN Configuration:
    DNS name  : ctdb.xyua
    Addresses : ['10.106.52.241', '10.106.52.242']
    Network   : 10.106.52.0
    Netmask   : 24
    Gateway   : 10.106.52.1
CNG nodes:
    enabled: snx11000n[006-007]
ECN Interface Configuration:
    Active Interfaces: eth40
CIFS Global Configuration:
    workgroup     : ctdb
    netbios name  : ctdb
    server string : test server
    security      : user
    realm         : not specified
    guest account : not specified
    log level     : 1
    Winbind:
        winbind separator          : default
        winbind enum users         : default
        winbind enum groups        : default
```

```
        winbind use default domain : default
        winbind offline logon      : default
        winbind cache time         : default
    Windows acl:
        nt acl support       : default
        acl compatibility    : default
        map acl inherit      : default
        map hidden           : default
        map system           : default
        map archive          : default
        map readonly         : default
        store dos attributes : default
        dos filemode         : default
Shares:
    testshare:
        uid        : nobody
        gid        : nobody
        mode       : 777
        path       : testshare
        filesystem : testfs
        enabled protocols: cifs, nfs
        description: Test share with pub access
        cifs options:
            writeable yes
            guest_ok yes
            browsable yes
        networks:
            nfs * writeable: true, squash: all
    mydocs:
        uid        : nobody
        gid        : nobody
        mode       : 511
        path       : documents
        filesystem : testfs
        enabled protocols: cifs
        description: documents
        cifs options:
            writeable no
            inherit_permissions yes
            guest_ok yes
            browsable yes
        networks:
            nfs * writeable: true, squash: root
    lab2_results:
        uid        : nobody
        gid        : nobody
        mode       : 777
        path       : lab2_results
        filesystem : testfs
        enabled protocols: cifs, nfs
        description: Test share with pub access
        cifs options:
            writeable yes
            browsable yes
        networks:
            nfs * writeable: true, squash: root
    svn4:
        uid        : nobody
        gid        : nobody
        mode       : 777
        path       : svn4
```

```
        filesystem : testfs
        enabled protocols: cifs, nfs
        description: Test share with pub access
        cifs options:
            writeable yes
            browsable yes
        networks:
            nfs * writeable: true, squash: root
    brezerk:
        uid        : nobody
        gid        : nobody
        mode       : 777
        path       : brezerk
        filesystem : testfs
        enabled protocols: cifs, nfs
        description: Test share with pub access
        cifs options:
            writeable yes
            browsable yes
        networks:
            nfs * writeable: true, squash: root
```

Example 2

The following example displays only the Winbind settings.

```
[admin@snx11000n000 ~]$ cscli cng show winbind
Winbind:
    winbind separator          : default
    winbind enum users         : yes
    winbind enum groups        : yes
    winbind use default domain : default
    winbind offline logon      : default
    winbind cache time         : default
```

Use of `cscli cng cifs globals`

Use this command to change the CIFS global configuration settings.

The general form of the command is:

```
[admin@n000]$ cscli cng cifs globals -S "description" -N server_name -W
work_group --security security_type --guest-account account_name --realm
realm_name/--realm-unset
```

Command Reference: *cng cifs globals*

Notes:

- *description* – Short description of the Samba server
- *server_name* – NetBIOS name of the Samba server
- *security_type* – Values are: `ads` (Active Directory), `user`, or `share`

  See the *Official Samba HowTo and Reference Guide*for a detailed explanation of the security modes.
- *work_group* – NetBIOS group to which the server belongs. If AD is configured, this is the AD domain.
- *realm_name* – AD realm name and should be set only if the ads security type is used. If security is other than `ads`, use the `--realm-unset` option instead.

NOTE: You cannot define different parameters for different CNG nodes.

Example

This command specifies the NetBIOS group name (to which the Samba server belongs) to `ctdb`, the NetBIOS name of the Samba server to `ctdb`, the Samba server description to `test server1`, and the security type to `user`.

```
[admin@snx11000n000 ~]$ cscli cng cifs globals -W ctdb -N ctdb --security user -S 'test server1'
cng: global cifs configuration is updated.
CIFS Global Configuration:
    workgroup     : ctdb
    netbios name  : ctdb
    server string : test server1
    security      : user
    realm         : not specified
    guest account : not specified
    log level     : 1
    Winbind:
        winbind separator          : default
        winbind enum users         : default
        winbind enum groups        : default
        winbind use default domain : default
        winbind offline logon      : default
        winbind cache time         : default
    Windows acl:
        nt acl support       : default
        acl compatibility    : default
        map acl inherit      : default
        map hidden           : default
        map system           : default
        map archive          : default
        map readonly         : default
        store dos attributes : default
        dos filemode         : default
```

IMPORTANT: For more detailed information about setting options for Guest, LDAP, and AD authentication, see *Security Type Examples* on page 16.

Use of `cscli cng cifs winbind`

Use this command to change the configuration settings for Winbind.

The general form of the command is:

```
[admin@n000]$ cscli cng cifs winbind --enum-users yes,no --enum-groups yes,no--
use-default-domain yes,no --offline-logon true,false --cache-timeenum wb_ctime
```

Command Reference: *cng cifs winbind*

Notes:

- `--enum_users` – On large installations using `winbindd`(8), it may be necessary to specify yes to suppress the enumeration of users through the `setpwent()`, `getpwent()` and `endpwent()` group of system calls.
- `--enum_groups` – On large installations using `winbindd`(8) it may be necessary to specify **yes** to suppress the enumeration of groups through the `setpwent()`, `getpwent()` and `endpwent()` group of system calls.
- `--use-default-domain` – Specify **yes** if the `winbindd`(8) daemon should operate on users without domain component in their username.

Example

This example changes the Winbind `enum-users` and `enum-groups` options to **yes**.

```
[admin@snx11000n000 ~]$ cscli cng cifs winbind --enum-users yes --enum-groups yes
Winbind:
    winbind separator          : default
    winbind enum users         : yes
    winbind enum groups        : yes
    winbind use default domain : default
    winbind offline logon      : default
    winbind cache time         : default
```

Use of `cscli cng cifs idmap (show, set)`

Use these commands to display the current **idmap** setting and to change or set the idmap setting.

- The general form of the show command is:

  ```
  [admin@n000]$ cscli cng cifs idmap show
  ```

  Command Reference:*cng cifs idmap show*

- The general form of the set command is:

  ```
  [admin@n000]$ cscli cng cifs idmap set -d domain -b backend -r range
  ```

  Command Reference:*cng cifs idmap set*

Notes:

- *domain* – Name of the domain. * is the default domain.
- *backend* – Back end that handles domain authentication. Options are `ad`, `adex`, `autorid`, `hash`, `ldap`, `nss`, `rid`, `tdb`, `tdb2`.
- *range* – UID range

Example 1

The output from this example command indicates that the **idmap** backend is set to **tdb2** for the UID range 1000000 to 1999999 on the default domain.

```
[admin@snx11000n000 ~]$ cscli cng cifs idmap show
    idmap config * : backend = tdb2
    idmap config * : range = 1000000-1999999
```

Example 2

This example command changes the **idmap** backend from **tdb2** to rid for the same user range on the default domain, as shown in the previous example.

```
[admin@snx11000n000 ~]$ cscli cng cifs idmap set -d * -b rid -r 1000000-1999999
       idmap config * : backend = rid
       idmap config * : range = 1000000-1999999
```

Edit CIFS/NFS Configuration with GUI

About this task

This procedure describes how to use the CSSM GUI to list and change the CIFS/NFS global configuration settings, including:

- Display the current CNG configuration settings
- Change the CIFS global configuration settings
- Change the Winbind settings

> **IMPORTANT:** Configuring and changing idmap options. idmap options cannot currently be configured or changed using CSSM. Instead, you must use the related CSCLI commands.

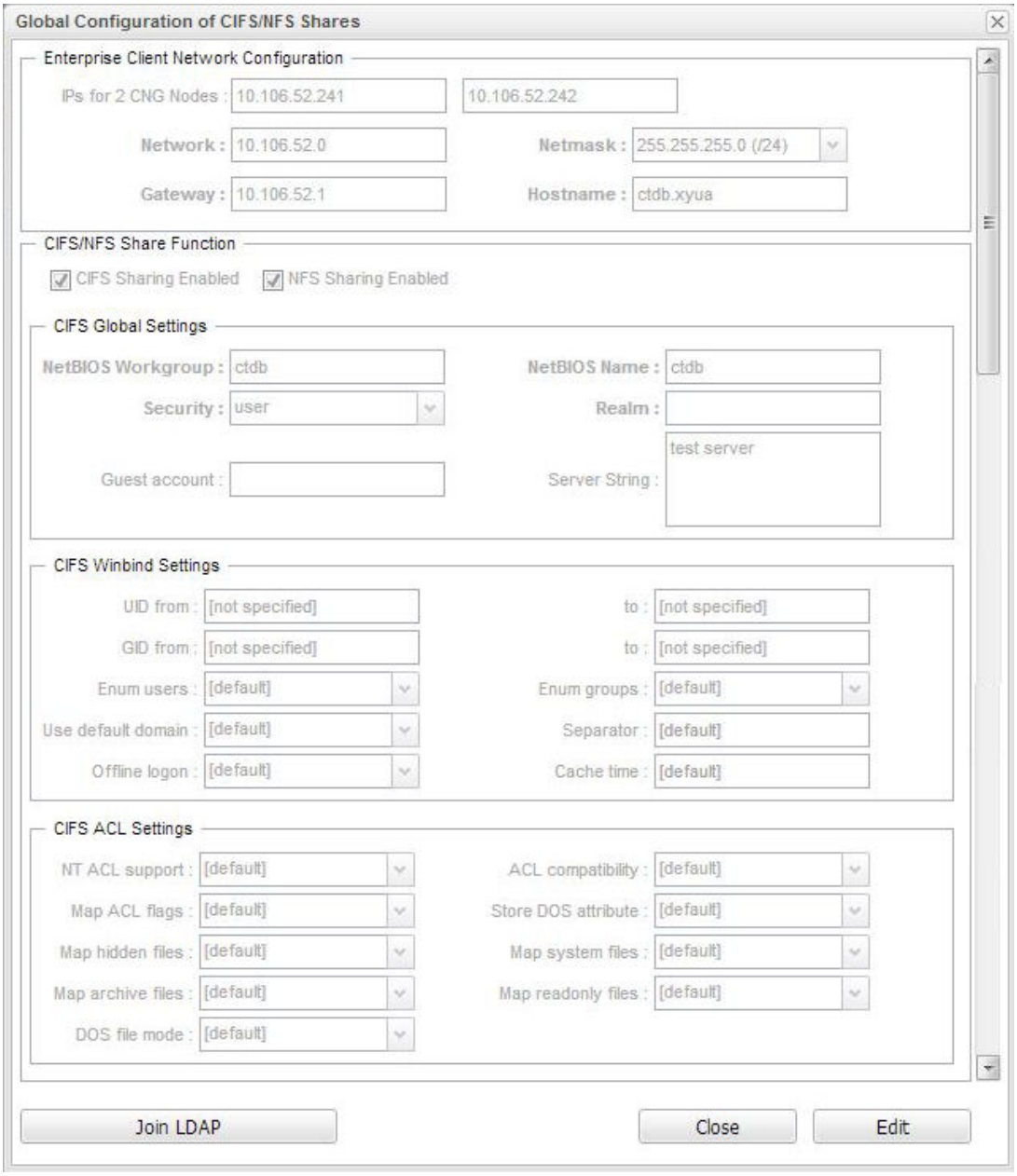
To display the current configuration settings:

Procedure

1. Navigate to the **Shares** tab, and then click **Global Configuration**.

   The Global Configuration of **CIFS/NFS Shares** window is displayed.

Figure 31. Global Configuration of CIFS/NFS Shares



This window displays the global configuration information in several groups of information:

- Enterprise Client Network Configuration
- CIFS/NFS sharing options
- CIFS Global Settings
- CIFS Winbind Settings
- CIFS ACL Settings

2. To change any of the global configuration settings, click Edit to enter edit mode.

Most global configuration options will be set during the initial CNG configuration. For more information about configuring these options, see *Specify Global Settings for Shares - GUI* on page 25. There may be times when you need to make changes to your configuration, such as:

- Adding a new user security model or changing the security model used. For example, switching from LDAP to Active Directory.
- Changing the site's network equipment or infrastructure, such that it requires changing the CNG node IP addresses on the ECN
- Modifying settings when troubleshooting sharing functionality and performance One of the most common reasons to change the global configuration is to enable or disable CIFS and/or NFS exporting. See: *Turn On or Off CIFS-NFS Shares Using the GUI*.

Turn On or Off CIFS-NFS Shares

About this task

This topic covers use of both the CSCLI and GUI to turn shares on and off. The GUI is covered below in *Turn On or Off CIFS-NFS Shares Using the GUI*.

Use of `cscli cng cifs (enable, disable)`

Use these commands to enable or disable CIFS exporting.

- The general form of the `enable` command is:

  ```
  [admin@n000]$ cscli cng cifs enable -y
  ```

  Example. This command enables CIFS exporting:

  ```
  [admin@snx11000n000 ~]$ cscli cng cifs enable -y
  cng: CIFS exporting is enabled.
  ```

  Command Reference: cng cifs enable

- The general form of the `disable` command is:

  ```
  [admin@n000]$ cscli cng cifs disable -y
  ```

  Example. This command disables CIFS exporting

  ```
  [admin@snx11000n000 ~]$ cscli cng cifs disable -y
  cng: CIFS exporting is disabled.
  ```

  Command Reference: *cng cifs disable*

Use of cscli cng nfs (enable, disable)

Use these commands to enable or disable NFS exporting.

- The general form of the enable command is:

  ```
  [admin@n000]$ cscli cng nfs enable
  ```

Example. This command enables NFS exporting.

```
[admin@snx11000n000 ~]$ cscli cng nfs enable -y
cng: NFS exporting is enabled.
```

Command Reference: *cng nfs enable*

- The general form of the disable command is:

```
[admin@n000]$ cscli cng nfs disable
```

Example. This command disables NFS exporting.

```
[admin@snx11000n000 ~]$ cscli cng nfs disable -y
cng: NFS exporting is disabled.
```

Command Reference: *cng nfs disable*

Use of cscli cng (enable, disable)

Use these commands to enable or disable all CIFS and NFS exporting.

- The general form of the `enable` command is:

```
[admin@n000]$ cscli cng enable -y
```

The following example shows how to enable all protocol exporting.

```
[admin@snx11000n000 ~]$ cscli cng enable -y
cng: cifs is enabled.
cng: nfs is enabled.
cng: exporting is enabled.
```

Command Reference: *cng enable*

- The general form of the `disable` command is:

```
[admin@n000]$ cscli cng disable -y
```

This example shows how to disable all protocol exporting.

```
[admin@snx11000n000 ~]$ cscli cng disable -y
cng: cifs is disabled.
cng: nfs is disabled.
cng: exporting is disabled.
```

Command Reference: *cng disable*

Turn On or Off CIFS-NFS Shares Using the GUI

This procedure shows the steps to turn all CIFS and/or NFS sharing on or off. To turn CIFS and/or NFS exporting on or off for a specific shared resource, see *Use of `cscli cng share (enable, disable)`*.

Procedure

1. Navigate to the **Shares** tab, and then click **Global Configuration**.

The **Global Configuration** of the **CIFS/NFS Shares** window is displayed.

2. To enable or disable CIFS exporting, check or uncheck the **CIFS Sharing Enabled** check box.
3. To enable or disable NFS exporting, check or uncheck the **NFS Sharing Enabled** check box.
4. Click **Save**.

Monitor CNG Nodes with CLI

Use the following commands to monitor CNG nodes and health:

- `cscli cng status`
- `cscli monitor (nodes, health, elements)`

Use of `cscli cng status`

Use this command to display general status information about the CNG nodes.

The general form of the command is:

```
[admin@n000]$ cscli cng status
```

Command Reference: *cng status*

Example

The following example command output shows that CTDB, Lustre, and SMB are all started on the two CNG nodes, but that NFS is stopped.

```
[admin@snx11000n000 ~]$ cscli cng status
Current CNG status:
ctdb:
    Started: snx11000n006,snx11000n007
lustre:
    Started: snx11000n006,snx11000n007
nfs:
    Stopped: snx11000n006,snx11000n007
smb:
    Started: snx11000n006,snx11000n007
```

Use of `cscli monitor nodes`

Use this command to get general information about the CNG nodes.

The general form of the command is:

```
[admin@n000]$ cscli monitor nodes -n node_spec -g genders_query -N node_status -
y -v
```

Command Reference: "`cscli monitor nodes`", part of *Sonexion Administrator's Guide*.

Notes:

- *node_spec* – **pdsh**-style node host names specification for the nodes whose status should be displayed (for example, `snx11000n[100-110,120]`)
- *genders_query* – Node genders attributes query (for example, `mds=primary`)
- *node_status* – Node status to display; options are `down`, `unreachable`, `up`, `pending`

- `-y` – Display output data in YAML format
- `-v` – Verbose mode

Examples

The following command displays information about the two CNG nodes (snx11000n006 and snx11000n007), which are both up.

```
[admin@snx11000n000 ~]$ cscli monitor nodes -n snx11000n[006-007]
snx11000n006:  UP  for 1d  6h 27m 12s  checked 2014-07-22 20:13:33  "PING OK -
Packet loss = 0%, RTA = 0.10 ms"
snx11000n007:  UP  for 1d  6h 27m  7s  checked 2014-07-22 20:10:23  "PING OK -
Packet loss = 0%, RTA = 0.15 ms"
```

The following command displays the node information in YAML format.

```
[admin@snx11000n000 ~]$ cscli monitor nodes -n snx11000n[006-007] -y
nodes:
- in_scheduled_downtime: false
  is_flapping: false
  last_check_time: '2014-07-28 14:46:38'
  node_name: snx11000n006
  notify_off: false
  status: UP
  status_duration: 7d  1h  3m  0s
  status_information: PING OK - Packet loss = 0%, RTA = 0.11 ms
- in_scheduled_downtime: false
  is_flapping: false
  last_check_time: '2014-07-28 14:48:38'
  node_name: snx11000n007
  notify_off: false
  status: UP
  status_duration: 7d  1h  2m 55s
  status_information: PING OK - Packet loss = 0%, RTA = 0.14 ms
```

The following information displays the node information in YAML format using verbose mode.

```
[admin@snx11000n000 ~]$ cscli monitor nodes -n snx11000n[006-007] -y -v
nodes:
- active_checks_enabled: true
  check_duration: 4.0039999999999996
  check_latency: 0.22
  check_type: active
  current_attempt: 1
  current_notification_number: 0
  event_handler_enabled: true
  flap_detection_enabled: true
  flapping_percent_state_change: 0.0
  has_been_checked: true
  in_scheduled_downtime: false
  is_flapping: false
  last_check_time: '2014-07-28 14:56:58'
  last_notification: null
  last_state_change: '2014-07-21 13:48:10'
  last_update: '2014-07-28 14:57:08'
  max_attempts: 3
  modified_attributes: None
  next_scheduled_active_check: '2014-07-28 15:02:08'
  node_comments: []
```

```
  node_display_name: snx11000n006
  node_downtimes: []
  node_has_been_acknowledged: false
  node_name: snx11000n006
  notifications_enabled: true
  notify_off: false
  obsess_over_node: true
  passive_checks_enabled: true
  performance_data: rta=0.100000ms;1300.000000;5000.000000;0.000000 pl=0%;
80;100;0
  state_type: HARD
  status: UP
  status_duration: ' 7d  1h  9m  0s'
  status_duration_in_seconds: 608940
  status_information: PING OK - Packet loss = 0%, RTA = 0.10 ms
- active_checks_enabled: true
  check_duration: 4.0049999999999999
  check_latency: 0.114
  check_type: active
  current_attempt: 1
  current_notification_number: 0
  event_handler_enabled: true
  flap_detection_enabled: true
  flapping_percent_state_change: 0.0
  has_been_checked: true
  in_scheduled_downtime: false
  is_flapping: false
  last_check_time: '2014-07-28 14:53:48'
  last_notification: null
  last_state_change: '2014-07-21 13:48:15'
  last_update: '2014-07-28 14:57:08'
  max_attempts: 3
  modified_attributes: None
  next_scheduled_active_check: '2014-07-28 14:58:58'
  node_comments: []
  node_display_name: snx11000n007
  node_downtimes: []
  node_has_been_acknowledged: false
  node_name: snx11000n007
  notifications_enabled: true
  notify_off: false
  obsess_over_node: true
  passive_checks_enabled: true
  performance_data: rta=0.121300ms;1300.000000;5000.000000;0.000000 pl=0%;
80;100;0
  state_type: HARD
  status: UP
  status_duration: ' 7d  1h  8m 55s'
  status_duration_in_seconds: 608935
  status_information: PING OK - Packet loss = 0%, RTA = 0.12 ms
```

Use of cscli monitor elements

Use this command to get general information about the CNG node elements.

The general form of the command is:

```
[admin@n000]$ cscli monitor elements -n node_spec -g genders_query -N node_status
-U element_status -S element_filter -y -v
```

Command Reference: "`cscli monitor elements`", part of *Sonexion Administrator's Guide*.

Notes:

- *node_spec* – **pdsh**-style node host names specification for the nodes whose status should be displayed (for example, `snx110000n[100-110,120]`)
- *genders_query* – Node genders attributes query (e.g., `mds=primary`)
- *node_status* – Node status to display; options are down, unreachable, up, pending
- *element_status* – Element status to display; options are `unknown`, `warning`, `ok`, `critical`, `pending`
- *element_filter* – Specify a search value to find information for element names. The entry is case sensitive and regular expressions are allowed.
- `-y` – Display output data in YAML format
- `-v` – Verbose mode

If you call this command without any options, you may get thousands of elements on a large system.

Example

The following command displays detailed information about the elements for the two CNG nodes (snx11000n006 and snx11000n007). Note how much information is provided.

```
[admin@snx11000n000 ~]$ cscli monitor elements -n snx11000n[006-007]
snx11000n006  "BMC Status":  OK  for 7d  0h 54m 34s  checked 2014-07-21 17:43:16  "All FRU's are operating normally"
snx11000n006  "CIFS Statistics":  OK  for 7d  0h 54m 34s  checked 2014-07-28 14:37:20  "Summary: Total Throughput 0.7 B/s"
snx11000n006  "CIFS/NFS Gateway Health":  OK  for 2d 23h 12m  2s  checked 2014-07-28 14:37:20  "OK 4 services are runnning. 0 services are stopped"
snx11000n006  "Current Load":  OK  for 7d  0h 54m 35s  checked 2014-07-28 14:37:20  "OK - load average: 0.01, 0.01, 0.01"
snx11000n006  "Current Users":  OK  for 7d  0h 54m 34s  checked 2014-07-28 14:37:20  "USERS OK - 0 users currently logged in"
snx11000n006  "Fan Statistics":  OK  for 7d  0h 54m 34s  checked 2014-07-21 17:43:16  "Summary: 7 Fan Sensors available. All Sensors readings are within normal
operating levels"
snx11000n006  "NFS Statistics":  OK  for 7d  0h 54m 34s  checked 2014-07-28 14:37:20  "Summary: Total Throughput 0.0 B/s"
snx11000n006  "Network statistics":  OK  for 7d  0h 54m 34s  checked 2014-07-28 14:37:20  "NET OK - (Rx/Tx) eth0=(10.0GB/10.1GB), eth40=(272.0MB/9.2kB), ib0=(2172.0B/
584.0B), lo=(44.2MB/44.2MB), meth0=(10.0GB/10.1GB), meth1=(192.1kB/468.0B)"
snx11000n006  "Node Status":  PENDING  for 1d 11h 21m 37s+  checked N/A  "Service is not scheduled to be checked..."
snx11000n006  "RAM usage":  OK  for 7d  0h 54m 34s  checked 2014-07-28 14:37:20  "OK - 4.4% (2876172 kB) used."
snx11000n006  "Thermal Statistics":  OK  for 7d  0h 54m 34s  checked 2014-07-21 17:43:16  "Summary: 3 Temperature Sensors available. All Sensors readings are within
normal operating levels"
snx11000n006  "Total Processes":  OK  for 7d  0h 54m 34s  checked 2014-07-28 14:37:20  "PROCS OK: 2585 processes with STATE = RSZDT"
snx11000n007  "BMC Status":  OK  for 7d  0h 54m 30s  checked 2014-07-28 14:41:20  "All FRU's are operating normally"
snx11000n007  "CIFS Statistics":  OK  for 7d  0h 54m 30s  checked 2014-07-28 14:41:20  "Summary: Total Throughput 0.8 B/s"
snx11000n007  "CIFS/NFS Gateway Health":  OK  for 2d 23h 27m 50s  checked 2014-07-28 14:41:20  "OK 4 services are runnning. 0 services are stopped"
snx11000n007  "Current Load":  OK  for 7d  0h 54m 31s  checked 2014-07-28 14:41:20  "OK - load average: 0.00, 0.00, 0.01"
snx11000n007  "Current Users":  OK  for 7d  0h 54m 31s  checked 2014-07-28 14:41:20  "USERS OK - 0 users currently logged in"
snx11000n007  "Fan Statistics":  OK  for 7d  0h 54m 30s  checked 2014-07-28 14:41:20  "Summary: 7 Fan Sensors available. All Sensors readings are within normal
operating levels"
snx11000n007  "NFS Statistics":  OK  for 7d  0h 54m 30s  checked 2014-07-28 14:41:20  "Summary: Total Throughput 0.0 B/s"
snx11000n007  "Network statistics":  OK  for 7d  0h 54m 31s  checked 2014-07-28 14:41:20  "NET OK - (Rx/Tx) eth0=(10.4GB/10.6GB), eth40=(272.7MB/38.7kB), ib0=(2228.0B/
584.0B), lo=(44.5MB/44.5MB), meth0=(10.4GB/10.6GB), meth1=(192.8kB/468.0B)"
snx11000n007  "Node Status":  PENDING  for 1d 11h 21m 38s+  checked N/A  "Service is not scheduled to be checked..."
snx11000n007  "RAM usage":  OK  for 7d  0h 54m 31s  checked 2014-07-28 14:41:20  "OK - 3.2% (2122612 kB) used."
snx11000n007  "Thermal Statistics":  OK  for 7d  0h 54m 30s  checked 2014-07-28 14:41:20  "Summary: 3 Temperature Sensors available. All Sensors readings are within
normal operating levels"
snx11000n007  "Total Processes":  OK  for 7d  0h 54m 31s  checked 2014-07-28 14:41:20  "PROCS OK: 932 processes with STATE = RSZDT"
```

Use of `cscli monitor health`

Use this command to get overall health information and a status summary.

The general form of the command is:

```
[admin@n000]$ cscli monitor health -y
```

Command Reference: "`monitor` Command for System Health", part of *Sonexion Administrator's Guide*.

Notes:

- `-y` – Display output data in YAML format
- `-v` – Verbose mode

Examples

- The following command displays the general health information about the Sonexion system.

```
[admin@snx11000n000 ~]$ cscli monitor health
Nodes:
up: 14  down: 0    unreachable: 0  pending: 0  total: 14
Elements:
ok: 130  warning: 0  critical: 3  unknown: 1  pending: 7  total: 141
```

- The following command displays the general health information in YAML form.

```
[admin@snx11000n000 ~]$ cscli monitor health -y
health:
  elements:
    critical: 3
    ok: 130
    pending: 7
    total: 141
    unknown: 1
    warning: 0
  nodes:
    down: 0
    pending: 0
    total: 14
    unreachable: 0
    up: 14
```

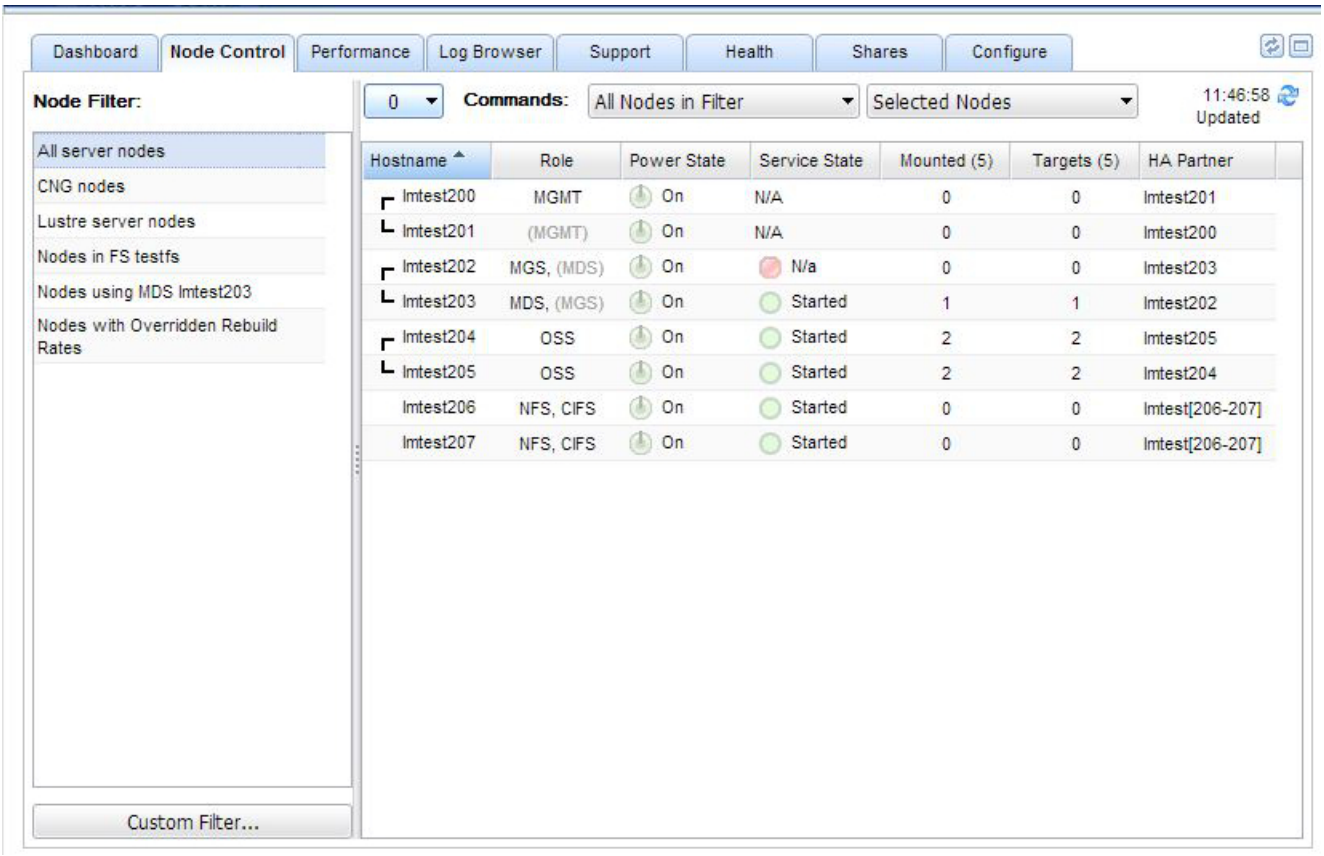
Monitor CNG Nodes Using Node Control Tab

About this task

Use the CSSM **Node Control** tab to check general CNG node status, such as power state and sharing state. You can also perform node operations such as shutting down, rebooting, power cycling, forcing power off, forcing reset, and powering on.

The main **Node Control** view shows all nodes in the system. You can filter the node view by selecting one of the predefined node filters in the left pain, or you can set a custom filter.

Figure 32. Node Control Tab Showing All Nodes



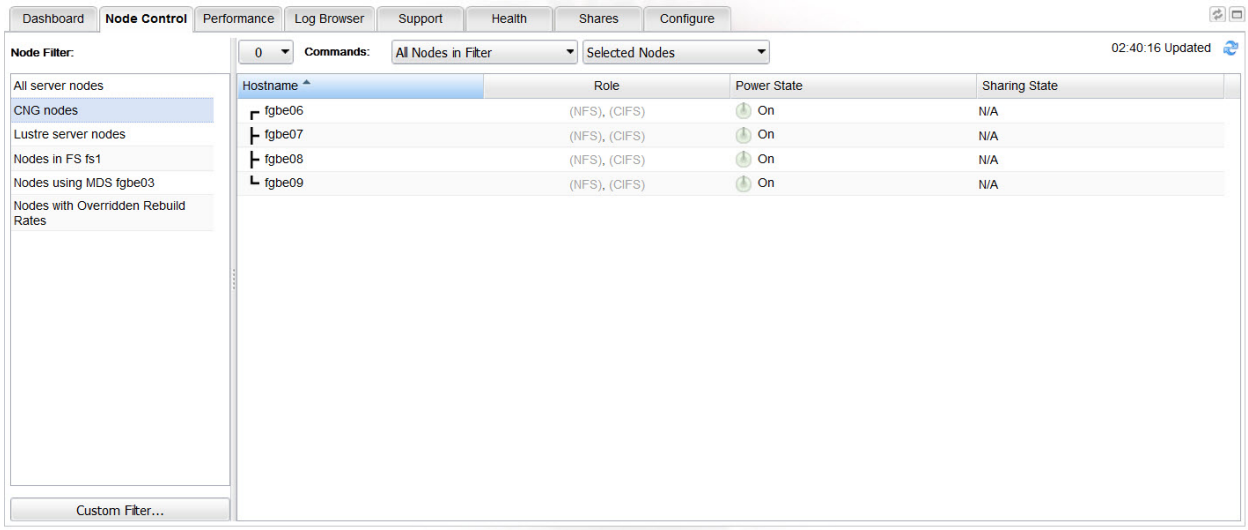
To work with only the CNG nodes:

Procedure

1. Click the **CNG Nodes** option under **Node Filter**.

   The list of nodes is narrowed to show only CNG nodes, as shown in the diagram below.

Figure 33. Node Control Tab Showing CNG Nodes Only



2. To perform actions on all of the CNG nodes, select a command from the **All Nodes** in Filter list. The options are **Shutdown All**, **Reboot All**, **Cycle All**, **Force Power Off All**, **Force Reset All**, and **Power On All**.

3. To perform actions on selected CNG nodes:

   a. Click on each node so that it is highlighted.

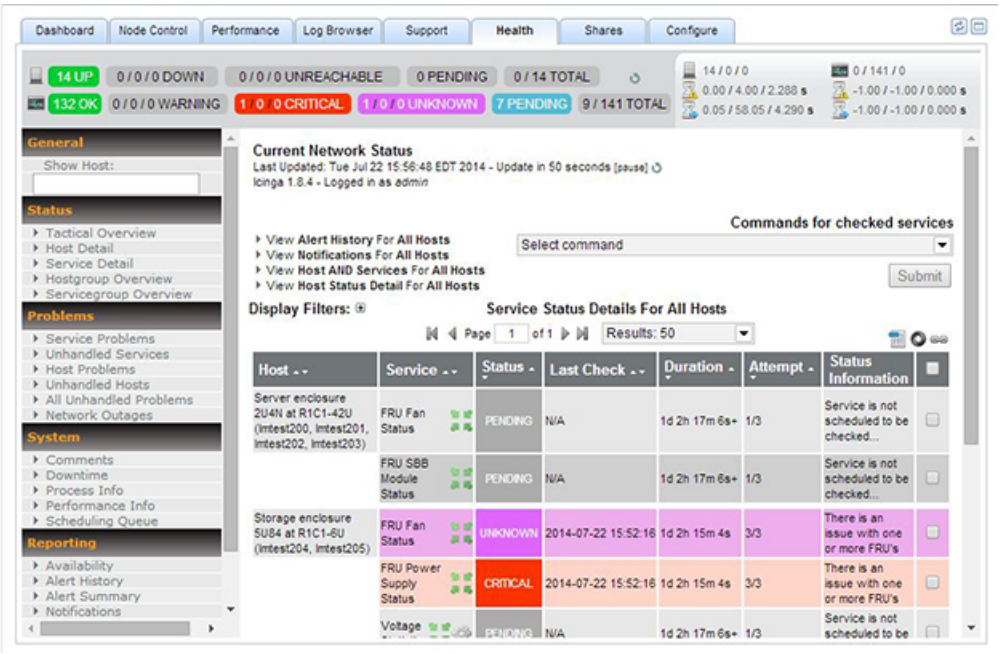
   b. Select a command from the **Selected Nodes** list. The options are **Shutdown**, **Reboot**, **Cycle**, **Power Off**, **Reset**, **Power On**, and **Minimum Rebuild Rates**.

CSSM Health Tab

Use the CSSM **Health** tab to monitor network services (SMTP, POP3, HTTP, NNTP, PING, etc.), host resources (CPU load, disk usage, etc.), and components (ArrayDevices, BMC, Batteries, PSUs, Cooling Fans, I/O modules, disk drives, enclosure electronics, etc.).

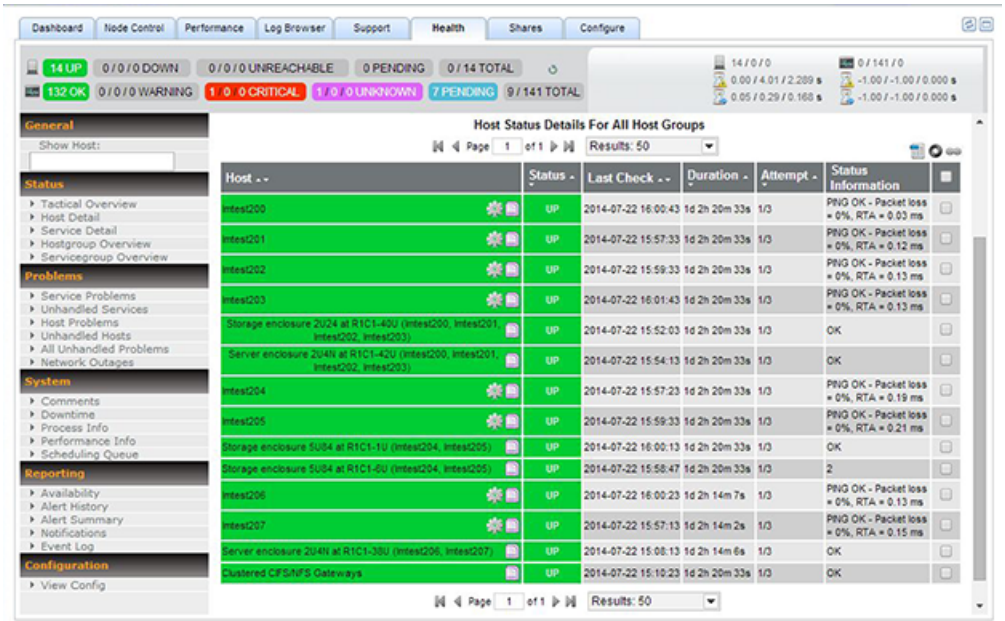
As shown in the following figure, the initial view of the **Health** tab shows the current network status and the service status details for all hosts.

Figure 34. CSSM Health Tab



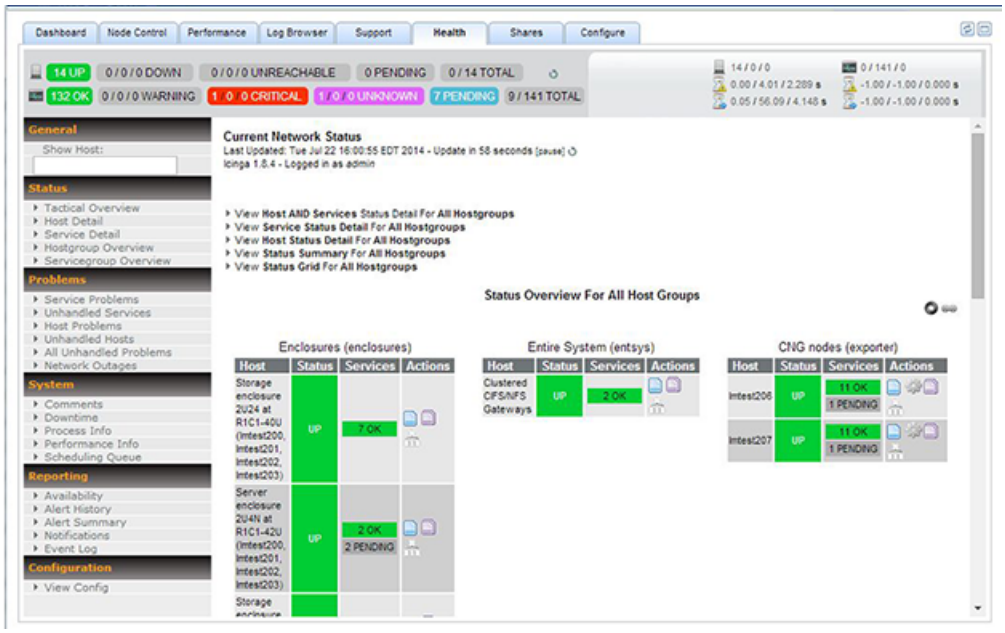
Under **Status** in the left pane, click **Host Detail** to display the general status of the system nodes, shown in the following figure. It shows the CNG nodes, with status **Up**, near the bottom of the list.

Figure 35. Host Status Detail for All Host Groups



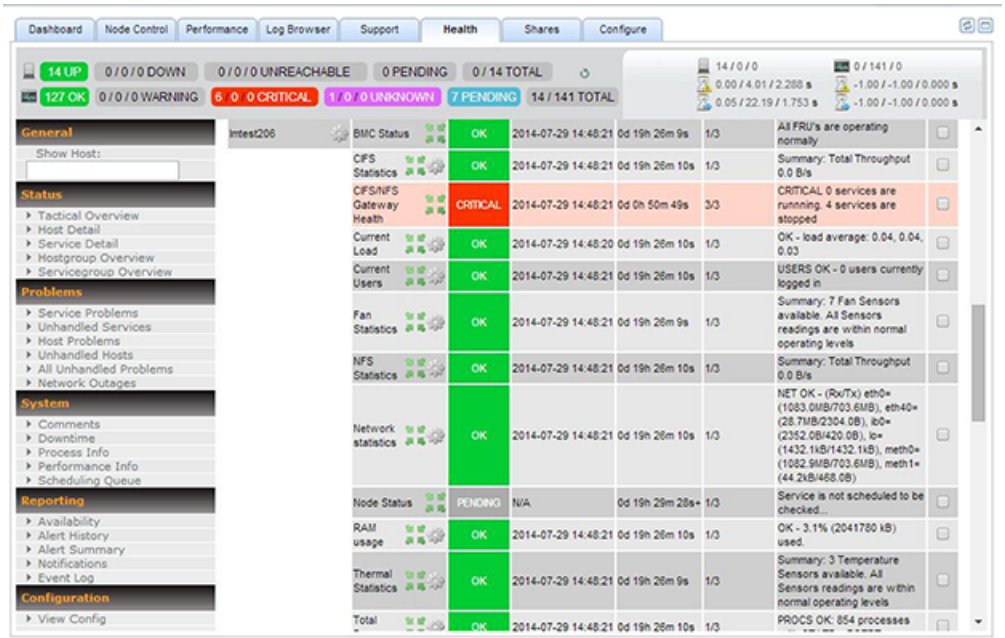
Under **Status** in the left pane, click **Hostgroup Overview** for a different view of the status. The CNG nodes are grouped on the right side of the window, as shown in the following figure.

Figure 36. Host Group Overview Showing CNG Nodes



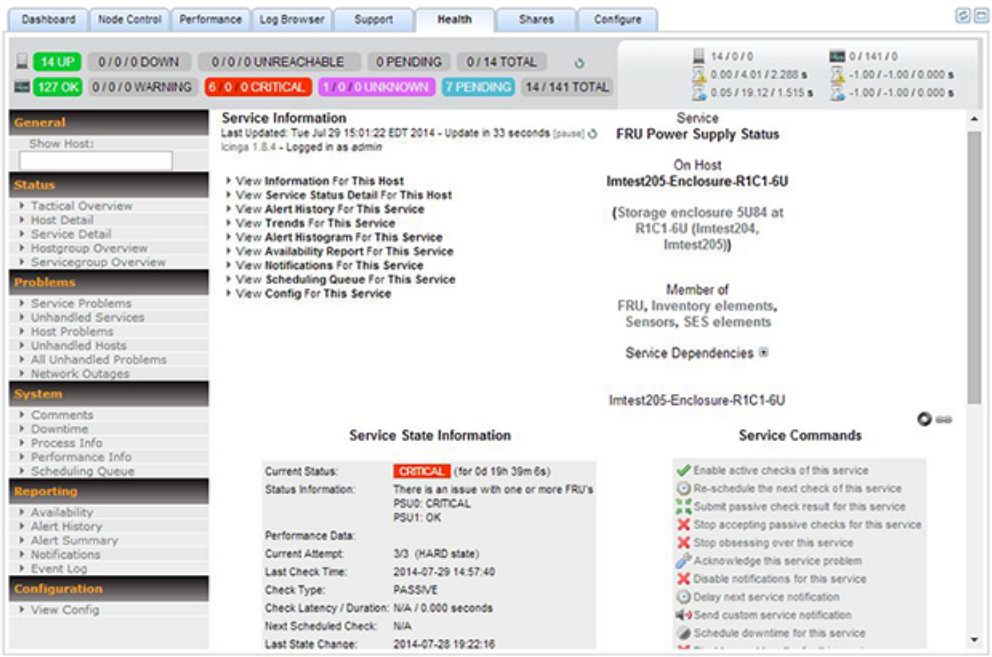
Under **Status** in the left pane, click Service Detail to display detailed service information for all the hosts. In the diagram below, the display has been scrolled down to show service detail status for the CNG node snx11000n006. Note that all services for this CNG node, except **CIFS/NFS Gateway Health**, show status as **OK**. However, **CIFS/NFS Gateway Health** shows status as **Critical**, with 0 services running and 4 services stopped.

Figure 37. Health Tab Service Details



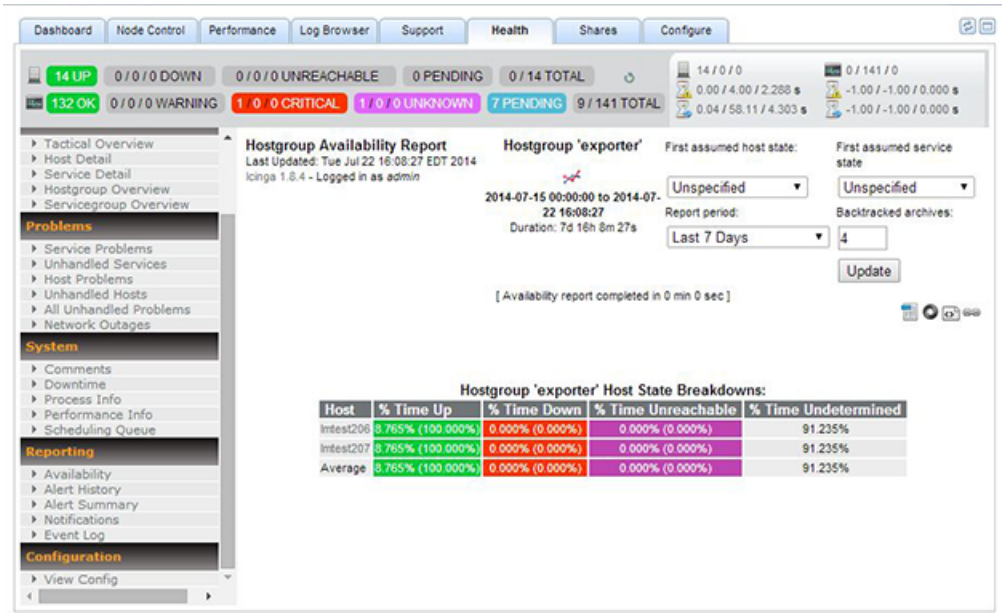
Click on the **CIFS/NFS Gateway Health** link to access additional details about the critical problem. As shown in this next figure, the problem appears to be a faulty PSU (PSU0) in the CNG chassis.

Figure 38. Service Data Showing Faulty PSU in CNG



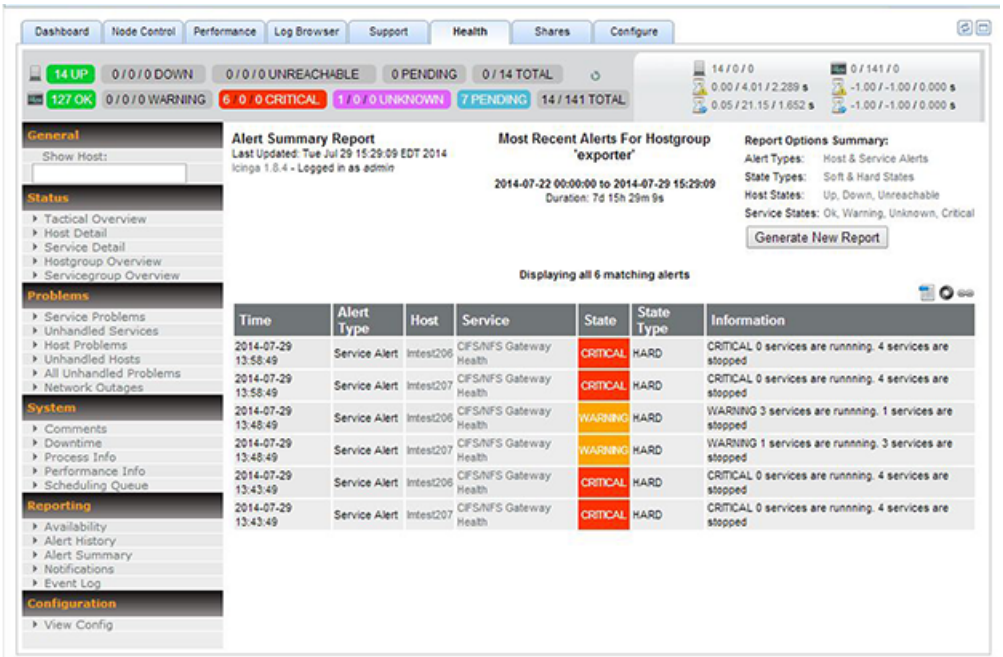
Under **Reports** in the left pane, click **Availability**, and then create an availability report for the CNG nodes (Histgroup=exporter) over a one week period. It will look like the following following.

Figure 39. Availability Report for CNG Nodes



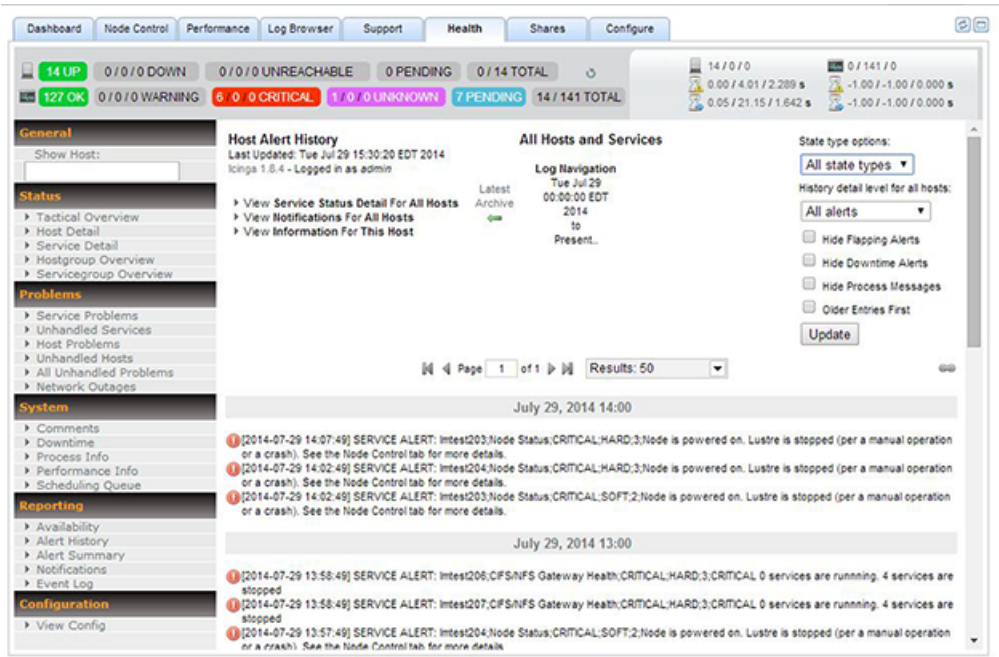
Under **Reporting** in the left pane, click **Alert Summary**, and then create an alert summary report for the CNG nodes (Hostgroup=exporter) over a one week period. It will look like the following figure.

Figure 40. Alert Summary Report for CNG Nodes



Under **Reporting** in the left pane, click **Alert History** for a historical report of alerts for the system. In the diagram below, note the two entries in the report for the CNG nodes, snx11000n006 and snx11000n007.

Figure 41. Alert History Report



Once you have determined a likely cause of any health issues with the CNG nodes, you can initiate corrective action, such as replacing the faulty power supply in the CNG node in the examples above.

CLI cscli cng Reference

This topic describes the `cscli cng` command and its primary subcommands (those with no secondary subcommands). The `cng` command is used to configure, manage, and monitor CNG nodes and export settings.

> **NOTE:** Earlier releases of Sonexion used the command `cscli export`, which is now deprecated and replaced by `cscli cng`. Existing scripts using the older command still work, but it is recommended to update scripts to use `cscli cng`.
>
> **IMPORTANT:** The following names are not allowed in `cscli cng` commands: `nobody`, `root`, and `admin`. For example, using `--guest-account nobody` with the `cscli cng cifs globals` command fails with AD.

Synopsis

```
$ cscli cng [-h] {status, setfacl, enable, join, clientgroup, show, ecn, share,
node, getfacl, disable, nfs, internaldb, interface, apply, cifs, network, ctdb}
```

Positional Arguments	Description
cng apply	Apply configuration changes.
cng cifs	Manages global CIFS configuration.
cng clientgroup	Manage CIFS/NFS client groups.
cng disable	Disable exporting of all protocols.
cng ecn	Configure CIFS/NFS Gateway nodes on Enterprise Client Network.
cng enable	Enables exporting of all protocols.
cng getfacl	Get file ACL.
cng interface	Configure ECN Export Interface.
cng internaldb	Manually manage backup and restore of the internal database used for CIFS/NFS configuration.
cng join	Joins CNG nodes to the Active Directory or LDAP.
cng nfs	Manage global NFS gateway configuration.
cng node	Operations with CNG nodes.
cng setfacl	Set/modify file ACL.
cng share	Configure CIFS/NFS shares.
cng show	Shows CIFS/NFS gateway configuration.
cng status	Show status of CIFS/NFS gateway services.
`ctdb`	This command is deprecated. See `ecn` option above.

Positional Arguments	Description
`network`	This command is deprecated. See `clientgroup` option above.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

Following are the primary subcommands which have no secondary subcommands.

cng apply

The `cng apply` subcommand applies the configuration changes.

IMPORTANT: Exercise caution before using the `-y` or `--yes` parameter.

Synopsis

```
$ cscli cng apply [-h] --restart-cifs -y
```

Optional Arguments	Description
`--restart-cifs`	Initiates a service restart after applying the changes to share options
`-h | --help`	Displays the help message and exits.
`-y | --yes`	Confirms applying.

cng ctdb

The `cng ctdb` subcommand is deprecated. Use `cng ecn` instead.

Synopsis

```
$ cscli cng ctdb [-h] [-D dnsname] [-A ip_address] [-N network] [-M netmask] [-G
gateway_host]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-D dnsname | --dns dnsname`	The CTDB DNS name.
`-A ip_address | --addr ip_address | --address ip_address`	Displays a list of valid ECN IP addresses that CTDB will manage.
`-N network | --network network`	The enterprise client network.
`-M netmask | --netmask netmask`	The enterprise client network netmask.
`-G gateway_host | --gateway gateway_host`	The enterprise client network default gateway.

cng disable

The `cng disable` subcommand disables exporting of all protocols.

Synopsis

```
$ cscli cng disable [-h] -y
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-y | --yes`	Confirms disabling.

cng ecn

The `cng ecn` subcommand configures CIFS/NFS Gateway nodes on the Enterprise Client Network.

Synopsis

```
$ cscli cng ecn [-h] [-D dnsname] [-A ip_address] [-N network] [-M netmask] [-G
gateway_host]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-D dnsname | --dns dnsname`	The fully-qualified domain name to be shared by all CIFS/NFS Gateway nodes.
`-A ip_address | --addr ip_address | --address ip_address`	An IP address to be used for a CIFS/NFS Gateway node on the Enterprise Client Network. Repeat this parameter for each node, providing a unique IP for each.
`-N network | --network network`	The Enterprise Client Network; IP network containing enterprise clients.
`-M netmask | --netmask netmask`	The Netmask for Enterprise Client Network.
`-G gateway_host | --gateway gateway_host`	Default router for IP traffic on the enterprise client network.

cng enable

The `cng enable` subcommand enables exporting of all protocols.

Synopsis

```
$ cscli cng enable [-h] -y
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-y | --yes`	Confirms enabling.

cng getfacl

The `cng getfacl` subcommand gets the ACL file.

Synopsis

```
$ cscli cng getfacl [-h] -p path
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-p` *`path`*	Displays the file path.

cng setfacl

The `cng setfacl` subcommand sets or modifies file ACL.

Synopsis

```
$ cscli cng setfacl [-h] (-m | -x) -s acl_spec -p path
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-m`	Modifies the ACL entries of a file or directory.
`-x`	Removes ACL entries.
`-s` *`acl_spec`*	The ACL entries.
`-p` *`path`*	The file path.

cng status

The `cng status` subcommand displays the status of CIFS/NFS gateway services.

Synopsis

```
$ cscli cng status [-h]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng cifs

The `cng cifs` subcommand manages global CIFS configuration.

Synopsis

```
$ cscli cng cifs [-h] {enable,idmap,winbind,disable,acl,globals} ...
```

Positional Arguments	Description
cng cifs acl	Configure windows ACL settings.
cng cifs disable	Disable CIFS exporting.
cng cifs enable	Enable CIFS exporting.
cng cifs globals	Configure global CIFS settings.
cng cifs idmap	Configure idmap settings.
cng cifs winbind	Configure winbind CIFS settings.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng cifs acl

The command `cng cifs acl` is a second-level subcommand of the `cng cifs` subcommand.

Synopsis

```
$ cscli cng cifs acl [-h] [--nt-acl-support {yes,no}]
[--acl-compatibility {auto,winnt,win2k}]
[--map-acl-inherit {yes,no}] [--map-hidden {yes,no}]
[--map-system {yes,no}] [--map-archive {yes,no}]
[--map-readonly {yes,no}] [--store-dos-attributes {yes,no}]
[--dos-filemode {yes,no}]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`--nt-acl-support {yes,no}`	This boolean parameter controls whether `smbd`(8) will attempt to map UNIX permissions into Windows NT access control lists.
`--acl-compatibility {auto,winnt,win2k}`	This parameter specifies what OS ACL semantics should be compatible with. Possible values are `winnt` for Windows NT 4, `win2k` for Windows 2000 and above and auto.
`--map-acl-inherit {yes,no}`	This boolean parameter controls whether `smbd`(8) will attempt to map the **inherit** and **protected** access control entry flags stored in Windows ACLs into an extended attribute called `user.SAMBA_PAI`.
`--map-hidden {yes,no}`	This controls whether DOS-style hidden files should be mapped to the UNIX world execute bit.
`--map-system {yes,no}`	This command controls whether DOS-style system files should be mapped to the UNIX group execute bit.
`--map-archive {yes,no}`	This controls whether the DOS archive attribute should be mapped to the UNIX owner execute bit.
`--map-readonly {yes,no}`	This controls how the DOS read-only attribute should be mapped from a UNIX filesystem.

Optional Arguments	Description
`--store-dos-attributes {yes,no}`	If this parameter is set, Samba attempts to first read DOS attributes (`SYSTEM`, `HIDDEN`, `ARCHIVE` or `READ-ONLY`) from a filesystem extended attribute, before mapping DOS attributes to UNIX permission bits (such as occurs with map hidden and map read only).
`--dos-filemode {yes,no}`	Enabling this parameter allows a user who has write access to the file (by whatever means, including an ACL permission) to modify the permissions (including ACL) on it.

cng cifs enable

The command `cng cifs enable` is a second-level subcommand of the `cng cifs` subcommand.

Synopsis

```
$ cscli cng cifs enable [-h] -y
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-y | --yes`	Confirms enabling.

cng cifs disable

The command `cng cifs disable` is a second-level subcommand of the `cng cifs` subcommand.

IMPORTANT: Exercise caution before using the `-y` or `--yes` parameter.

Synopsis

```
$ cscli cng cifs enable [-h] -y
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-y | --yes`	Confirms disabling.

cng cifs globals

The command `cng cifs globals` is a second-level subcommand of the `cng cifs` subcommand. Also see *Use of cscli cng cifs globals* on page 53.

Synopsis

```
$ cscli cng cifs globals [-h] [-W workgroup] [-S description]
[-N server_name] [--security {ads,user,share}] [--realm realm_name]
[--realm-unset] [--log-level log_level] [--log-level-unset]
[--guest-account guest_account | --guest-account-unset]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-W workgroup | --workgroup workgroup`	NetBIOS group to which the server belongs (if no AD was configured).
`-S description | --server-string description`	Descriptive string for the Samba server.
`-N server_name | --netbios-name server_name`	The NetBIOS name of the Samba server.
`--security {ads,user,share}`	This option affects how clients respond to Samba.
`--realm realm_name`	Kerberos realm option to be used.
`--realm-unset`	Unsets kerberos realm option.
`--log-level log_level`	Set CIFS log level (following debug classes are currently implemented: `all`, `tdb`, `printdrivers`, `lanman`, `smb`, `rpc_parse`, `rpc_srv`, `rpc_cli`, `passdb`, `sam`, `auth`, `winbind`, `vfs`, `idmap`, `quota`, `acls`, `locking`, `msdfs`, `dmapi`, `registry`).
`--log-level-unset`	Unset log level.
`--guest-account guest_account`	Username used to access services that are specified as guest-OK.
`--guest-account-unset`	Unsets guest account parameter.

cng cifs winbind

The command `cng cifs` windbind is a second-level subcommand of the `cng cifs` subcommand. Also see *Use of cscli cng cifs winbind* on page 54.

Synopsis

```
$ cscli cng cifs winbind [-h] [--separator separator]
[--separator-unset] [--enum-users {yes,no}] [--enum-users-unset]
[--enum-groups {yes,no}] [--enum-groups-unset]
[--use-default-domain {yes,no}] [--use-default-domain-unset]
[--offline-logon {true,false}] [--offline-logon-unset]
[--cache-time wb_ctime] [--cache-time-unset]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`--separator separator`	This parameter allows an administrator to define the character used when listing a username of the form of a DOMAIN user.
`--separator-unset`	Unsets separator parameter.

Optional Arguments	Description
`--enum-users {yes,no}`	On large installations using `winbindd`(8) it may be necessary to suppress the enumeration of users through the `setpwent`(), `getpwent`() and `endpwent`() group of system calls.
`--enum-users-unset`	Unsets enumeration users parameter.
`--enum-groups {yes,no}`	On large installations using `winbindd`(8) it may be necessary to suppress the enumeration of groups through the `setgrent`(), `getgrent`() and `endgrent`() group of system calls.
`--enum-groups-unset`	Unsets enumeration groups parameter.
`--use-default-domain {yes,no}`	This parameter specifies whether the `winbindd`(8) daemon should operate for users without domain component in their username.
`--use-default-domain-unset`	Unsets use default domain parameter.
`--offline-logon {true,false}`	This parameter is designed to control whether Winbind should allow to login with the pam_winbind module using Cached Credentials.
`--offline-logon-unset`	Unsets offline logon parameter.
`--cache-time` *`wb_ctime`*	This parameter specifies the number of seconds the `winbindd`(8) daemon will cache user and group information before querying a Windows NT server again.
`--cache-time-unset`	Unsets cache time parameter.

cng cifs idmap

The command `cng cifs idmap` is a second-level subcommand of the `cng cifs` subcommand. Also see *Use of cscli cng cifs idmap (show, set)* on page 55.

Synopsis

```
$ cscli cng cifs idmap [-h] {show,add,set,delete}
```

Positional Arguments	Description
cng cifs idmap add	Add a new idmap mapping.
cng cifs idmap delete	Delete domain mapping.
cng cifs idmap set	Add a new idmap mapping
cng cifs idmap show	Show a list of idmap mappings.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng cifs idmap add

The command `cng cifs idmap add` is a second-level subcommand of the `cng cifs idmap` subcommand.

Synopsis

```
$ cscli cng cifs idmap add [-h] -d domain
-b {ad,adex,autorid,hash,ldap,nss,rid,tdb,tdb2} -r range
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-d domain | --domain domain`	Domain name (* is default domain).
`-b {ad, adex, autorid, hash, ldap, nss, rid, tdb, tdb2} | --backend {ad, adex, autorid, hash, ldap, nss, rid, tdb, tdb2}`	Backend that handles domain authentication.
`-r range | --range range`	User id range (for instance 1-1000).

cng cifs idmap delete

The command `cng cifs idmap delete` is a second-level subcommand of the `cng cifs idmap` subcommand.

Synopsis

```
$ cscli cng cifs idmap delete [-h] -d domain
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-d domain, --domain domain`	Domain name (* is default domain)

cng cifs idmap set

The command `cng cifs idmap set` is a second-level subcommand of the `cng cifs idmap` subcommand.

Synopsis

```
$ cscli cng cifs idmap set [-h] -d domain
-b {ad,adex,autorid,hash,ldap,nss,rid,tdb,tdb2} -r range
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-d domain, --domain domain`	Domain name (* is default domain)
`-b {ad,adex, autorid, hash, ldap, nss, rid, tdb, tdb2} | --backend {ad, adex,`	Backend that handles domain auth.

Optional Arguments	Description
`autorid, hash, ldap, nss, rid, tdb, tdb2}`	
`-r` *`range`* `| --range` *`range`*	User id range (for instance 1-1000)

cng cifs idmap show

The command `cng cifs idmap show` is a second-level subcommand of the `cng cifs idmap` subcommand.

Synopsis

```
$ cscli cng cifs idmap show [-h]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng clientgroup

The `cng clientgroup` subcommand manages CIFS and NFS client groups.

Synopsis

```
$ cscli cng clientgroup [-h] {add,set,list,delete}
```

Positional Arguments	Description
cng clientgroup add	Adds a new client group for share.
cng clientgroup delete	Deletes share client group.
cng clientgroup list	Displays a list of share client groups.
cng clientgroup set	Modifies client group for share.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng clientgroup add

The command `cng clientgroup add` is a second-level subcommand of the `cng clientgroup` subcommand.

Synopsis

```
$ cscli cng clientgroup add [-h] -N clientgroup_name [--auto-add {yes,no}] (--
network network_spec | --domain domain_glob)
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-N clientgroup_name | --name` *`clientgroup_name`*	The unique client group name.
`--auto-add {yes,no}`	The auto-add flag automatically configures this client group for NFS on new shares when they are added.
`--network` *`network_spec`*	The network specification (network/cidr).
`--domain` *`domain_glob`*	The domain glob specification.

cng clientgroup delete

The command `cng clientgroup delete` is a second-level subcommand of the `cng clientgroup` subcommand.

Synopsis

```
$ cscli cng clientgroup delete [-h] -N clientgroup_name
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-N clientgroup_name | --name` *`clientgroup_name`*	The unique client group name.

cng clientgroup list

The command `cng clientgroup list` is a second-level subcommand of the `cng clientgroup` subcommand. Also see *Use of cscli cng clientgroup (list, set)* on page 44.

Synopsis

```
$ cscli cng clientgroup list [-h]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng clientgroup set

The command `cng clientgroup set` is a second-level subcommand of the `cng clientgroup` subcommand. Also see *Use of cscli cng clientgroup (list, set)* on page 44.

Synopsis

```
$ cscli cng clientgroup set [-h] -N clientgroup_name [--auto-add {yes,no}] (--
network network_spec | --domain domain_glob)
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-N clientgroup_name | --name` *`clientgroup_name`*	The unique client group name.
`--auto-add {yes,no}`	The auto-add flag will automatically configure this client group for NFS on new shares when they are added.
`--network` *`network_spec`*	The network specification (network/cidr).
`--domain` *`domain_glob`*	The domain glob specification.

cng interface

The command `cng interface` configures the ECN Export Interface.

Synopsis

```
$ cscli cng interface [-h] {set,show}
```

Positional Arguments	Description
`set`	Sets ECN Export Interface configuration.
`show`	Shows current ECN Export Interface configuration.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng interface set

The command `cng interface set` is the subcommand of `cng interface` subcommand.

Synopsis

```
$ cscli cng interface set [-h] -p {0,1}
```

Optional Arguments	Description
`-h | --help`	Shows the help message and exits.
`-p {0,1} | --port {0,1}`	Single port to use.

cng interface show

The command `cng interface show` is the subcommand of `cng interface` subcommand.

Synopsis

```
$ cscli cng interface show [-h]
```

Optional Arguments	Description
`-h | --help`	Shows the help message and exits.

cng internaldb

The command `cng internaldb` manually manages backup and restore of the internal database used for CIFS/NFS configuration. In most situations, these commands should not be needed.

Synopsis

```
$ cscli cng internaldb [-h] {restore,backup,list,show}...
```

Positional Arguments	Description
cng internaldb backup	Creates a new backup.
`list`	Displays a list of available backup files.
cng internaldb restore	Restores files from the backup.
cng internaldb show	Displays a list of files in the backup file.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng internaldb backup

The command `cng internaldb backup` is a second-level subcommand of the `cng internaldb` subcommand.

Synopsis

```
$ cscli cng internaldb backup [-h] [-y]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-y | --yes`	Confirms the action.

cng internaldb restore

The command `cng internaldb restore` is a second-level subcommand of the `cng internaldb` subcommand.

Synopsis

```
$ cscli cng internaldb restore [-h] -f filename [-p files] [-y]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-f filename | --backup-file filename`	Name of backup file to be restored.
`-p files | --partial-file files`	Displays list of files for partial restore.
`-y | --yes`	Confirms the action.

cng internaldb show

The command `cng internaldb show` is a second-level subcommand of the `cng internaldb` subcommand.

Synopsis

```
$ cscli cng internaldb show [-h] -f filename
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-f filename | --backup-file filename`	Displays the path to the backup file.

cng join

The `cng join` subcommand joins CNG nodes to the Active Directory or LDAP.

Synopsis

```
$ cscli cng join [-h] {ad,ldap}
```

Positional Arguments	Description
`ad`	Joins AD.
`ldap`	Joins LDAP.

Optional Arguments	Description
`-h | --help`	Shows the help message and exits.

cng join ad

The command `cng join ad` is a second-level subcommand of the `cng join` subcommand.

Synopsis

```
$ cscli cng join ad [-h] -U user [-P passwd]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

Optional Arguments	Description
`-U user | --user user`	Active Directory account login name.
`-P passwd | --password` *`passwd`*	Active Directory account password.

cng join ldap

The command `cng join ldap` is a second-level subcommand of the `cng join` subcommand.

Synopsis

```
$ cscli cng join ldap [-h] -P passwd
```

Optional Arguments	Description
`-h| --help`	Displays the help message and exits.
`-P passwd | --password` *`passwd`*	LDAP account password.

cng nfs

The command `cng nfs` manages global NFS gateway configuration.

Synopsis

```
$ cscli cng nfs [-h] {enable,disable}
```

Positional Arguments	Description
`enable`	Enables NFS exporting.
`disable`	Disables NFS exporting.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng nfs enable

The command `cng nfs enable` is a second-level subcommand of the `cng nfs` subcommand.

Synopsis

```
$ cscli cng nfs enable [-h] -y
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-y | --yes`	Confirms enabling.

cng nfs disable

The command `cng nfs disable` is a second-level subcommand of the `cng nfs` subcommand.

IMPORTANT: Exercise caution before using the `-y` or `--yes` parameter.

Synopsis

```
$ cscli cng nfs enable [-h] -y
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-y | --yes`	Confirms disabling.

cng node

The command `cng node` subcommand performs operations with CNG nodes.

Synopsis

```
$ cscli cng node [-h] {enable,disable}
```

Positional Arguments	Description
`enable`	Enables CNG nodes.
`disable`	Disables CNG nodes.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng node enable

The command `cng node enable` is a second-level subcommand of the `cng node` subcommand.

Synopsis

```
$ cscli cng node enable [-h] (-n node_spec | -a)
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-n node_spec | --node node_spec`	PDSH node specification.
`-a | --all`	Enable all cng nodes.

cng node disable

The command `cng node disable` is a second-level subcommand of the `cng node` subcommand.

Synopsis

```
$ cscli cng node disable [-h] (-n node_spec | -a)
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-n node_spec | --node node_spec`	PDSH node specification.
`-a | --all`	Disable all cng nodes.

cng share

The `cng share` subcommand configures CIFS/NFS shares.

Synopsis

```
$ cscli cng share [-h] {set,cifs,show,list,disable,add,enable,nfs,delete}
```

Positional Arguments	Description
cng share add	Adds a new share.
cng share cifs	Modifies share CIFS configuration.
cng share delete	Deletes share.
cng share disable	Disables share.
cng share enable	Enables share.
cng share list	Displays list of shares.
cng share nfs	Modifies share NFS configuration.
cng share set	Modifies share options.
cng share show	Displays share configuration.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng share add

The command `cng share add` is a second-level subcommand of the `cng share` subcommand.

Synopsis

```
$ cscli cng share add [-h] -s share_name -p path -f filesystem_name [-d
description] -U uid -G gid -M mode [--disable]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-s share_name | --share share_name`	Name of the share.
`-p path | --path path`	Relative path to the shared folder.
`-f filesystem_name | --fs filesystem_name`	Underlying cluster filesystem name.
`-d description | --description description`	The description of the share.
`-U uid | --uid uid`	User ID of the share.
`-G gid | --gid gid`	Group ID of the share.
`-M mode | --mode mode`	Mode of the share.
`--disable`	Disables share (default it is enabled).

cng share delete

The command `cng share delete` is a second-level subcommand of the `cng share` subcommand.

Synopsis

```
$ cscli cng share delete [-h] [-f filesystem_name] (-s share_name | -p path)
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-f filesystem_name | --fs filesystem_name`	Displays the underlying cluster filesystem name.
`-s share_name | --share share_name`	Displays the name of the share.
`-p path | --path path`	Displays the relative (within underlying cluster filesystem) path to the shared folder.

cng share disable

The command `cng share disable` is a second-level subcommand of the `cng share` subcommand. Also see *Use of cscli cng share (enable, disable)* on page 40.

Synopsis

```
$ cscli cng share disable [-h] [-P protocol] [-f filesystem_name] (-s share_name
| -p path)
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-P protocol | --protocol protocol`	Displays the protocol.

Optional Arguments	Description
`-f filesystem_name | --fs filesystem_name`	Displays the underlying cluster filesystem name.
`-s share_name | --share share_name`	Displays the name of the share.
`-p path | --path path`	Displays the relative (within underlying cluster filesystem) path to the shared folder.

cng share enable

The command `cng share enable` is a second-level subcommand of the `cng share` subcommand. Also see *Use of `cscli cng share (enable, disable)`*.

Synopsis

```
$ cscli cng share enable [-h] [-P protocol] [-f filesystem_name] (-s share_name
| -p path)
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-P protocol | --protocol protocol`	Displays the protocol.
`-f filesystem_name | --fs filesystem_name`	Displays the underlying cluster filesystem name.
`-s share_name | --share share_name`	Displays the name of the share.
`-p path | --path path`	Displays the relative (within underlying cluster filesystem) path to the shared folder.

cng share list

The command `cng share list` is a second-level subcommand of the `cng share` subcommand. Also see *Use of cscli cng share list* on page 38.

Synopsis

```
$ cscli cng share list [-h]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng share set

The command `cng share set` is a second=level subcommand of the `cng share` subcommand. Also see *Use of `cscli cng share set`*.

Synopsis

```
$ cscli cng share set [-h] -s share_name [-p path] [-f filesystem_name] [-S
share_name] [-d description] [-U uid] [-G gid] [-M mode]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-s share_name | --share share_name`	Displays the name of the share.
`-p path | --path path`	Displays the relative path to the shared folder.
`-f filesystem_name | --fs filesystem_name`	Displays the underlying cluster filesystem name.
`-S share_name | --new-share share_name`	Displays the new share name.
`-d description, | --description description`	Displays the share description.
`-U uid | --uid uid`	Displays the user ID.
`-G gid | --gid gid`	Displays the group ID.
`-M mode | --mode mode`	Displays the mode.

cng share show

The command `cng share show` is a second-level subcommand of the `cng share` subcommand. Also see *Use of cscli cng share show* on page 38.

Synopsis

```
$ cscli cng share show [-h] [-f filesystem_name] (-s share_name | -p path)
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-f filesystem_name | --fs filesystem_name`	The underlying cluster filesystem name.
`-s share_name | --share share_name`	Name of the share.
`-p path | --path path`	Relative (within underlying cluster filesystem) path to the shared folder.

cng share cifs

The command `cng share cifs` is a second-level subcommand of the `cng share` subcommand.

Synopsis

```
$ cscli cng share cifs [-h] {set,clientgroup}
```

Positional Arguments	Description
`set`	Modifies the share CIFS configuration.
cng share cifs clientgroup	Manages the share CIFS per client group configuration.
`network`	The command is deprecated. See `clientgroup` above.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng share cifs set

The command `cng share cifs set` is a second-level subcommand of the `cng share cifs` subcommand. Also see *Use of cscli cng share cifs set* on page 41.

Synopsis

```
$ cscli cng share cifs set [-h] [-f filesystem_name] (-s share_name | -p path)
[--writeable {yes,no} | --writeable-unset]
[--browsable {yes,no} | --browsable-unset]
[--create-mask create_mask | --create-mask-unset]
[--directory-mask directory_mask | --directory-mask-unset]
[--inherit-permissions {yes,no} | --inherit-permissions-unset]
[--inherit-owner {yes,no} | --inherit-owner-unset]
[--guest-ok {yes,no} | --guest-ok-unset]
[--guest-only {yes,no} | --guest-only-unset]
[--guest-account guest_account | --guest-account-unset]
[--admin-users admin_users | --admin-users-unset]
[--valid-users valid_users | --valid-users-unset]
[--invalid-users invalid_users | --invalid-users-unset]
[--read-list read_list | --read-list-unset]
[--write-list write_list | --write-list-unset]
[--max-connections max_connections | --max-connections-unset]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-f filesystem_name | --fs` *filesystem_name*	The underlying cluster filesystem name.
`-s` *share_name* `| --share` *share_name*	Name of the share.
`-p` *path* `| --path` *path*	Relative (within the underlying cluster filesystem) path to the shared folder.
`--writeable {yes,no}`	Inverted synonym for read only.
`--writeable-unset`	Unset writeable parameter.
`--browsable {yes,no}`	This controls whether this share is seen in the list of available shares in a net view and in the browse list.
`--browsable-unset`	Unset browsable parameter.

Optional Arguments	Description
`--create-mask` *`create_mask`*	This parameter may be thought of as a bit-wise MASK for the UNIX modes of a file. Any bit not set here will be removed from the modes set on a file when it is created.
`--create-mask-unset`	Unset create mask parameter.
`--directory-mask` *`directory_mask`*	This parameter is the octal modes which are used when converting DOS modes to UNIX modes when creating UNIX directories.
`--directory-mask-unset`	Unset directory mask parameter.
`--inherit-permissions {yes,no}`	New directories inherit the mode of the parent directory, including bits such as setgid. New files inherit their read/write bits from the parent directory.
`--inherit-permissions-unset`	Unset inherit permissions parameter.
`--inherit-owner {yes,no}`	This option allows the Samba administrator to specify that the ownership for new files and directories should be controlled by the ownership of the parent directory.
`--inherit-owner-unset`	Unset inherit owner parameter.
`--guest-ok {yes,no}`	If this parameter is yes for a share, then no password is required to connect to the share.
`--guest-ok-unset`	Unset guest ok parameter.
`--guest-only {yes,no}`	If this parameter is yes for a share, then the guest can connect to the share.
`--guest-only-unset`	Unset guest only parameter.
`--guest-account` *`guest_account`*	This is a username which will be used for access to services which are specified as guest ok.
`--guest-account-unset`	Unset guest account parameter.
`--admin-users` *`admin_users`*	Users who can perform operations as root.
`--admin-users-unset`	Unset admin users parameter.
`--valid-users` *`valid_users`*	Users who can connect to a share.
`--valid-users-unset`	Unset valid users parameter.
`--invalid-users` *`invalid_users`*	Users who will be denied access to a share.
`--invalid-users-unset`	Unset invalid users parameter.
`--read-list` *`read_list`*	Users who have read-only access to a writable share.
`--read-list-unset`	Unset read list parameter.
`--write-list` *`write_list`*	Users who have read/write access to a read-only share.
`--write-list-unset`	Unset write list parameter.

Optional Arguments	Description
`--max-connections` *`max_connections`*	Maximum number of connections for a share at a given time.
`--max-connections-unset`	Unset max connections parameter.

cng share cifs clientgroup

The command `cng share cifs clientgroup` is a second-level subcommand of the *cng share cifs* on page 92 subcommand.

Synopsis

```
$ cscli cng share cifs clientgroup [-h] {set,list,delete}
```

Positional Arguments	Description
cng share cifs clientgroup delete	Deletes a CIFS client group.
cng share cifs clientgroup list	Displays a list of available CIFS client groups.
cng share cifs clientgroup set	Creates/modifies CIFS client group options.

Optional Arguments	Description
`-h` | `--help`	Displays the help message and exits.

cng share cifs clientgroup delete

The command `cng share cifs clientgroup delete` is a second-level subcommand of the `cng share cifs clientgroup` subcommand.

Synopsis

```
$ cscli cng share cifs clientgroup delete [-h] [-f filesystem_name] (-s
share_name | -p path) | -N clientgroup_name
```

Optional Arguments	Description
`-h` | `--help`	Displays the help message and exits.
`-f filesystem_name` | `--fs` *`filesystem_name`*	The underlying cluster filesystem name.
`-s` *`share_name`* | `--share` *`share_name`*	Name of the share.
`-p` *`path`* | `--path` *`path`*	Relative (within the underlying cluster filesystem) path to the shared folder.
`-N` *`clientgroup_name`* | `--clientgroup` *`clientgroup_name`*	The client group name.

cng share cifs clientgroup list

The command `cng share cifs clientgroup list` is the second level subcommand of the `cng share cifs clientgroup` subcommand. Also see *Use of cscli cng share cifs clientgroup (list, set)* on page 42.

Synopsis

```
$ cscli cng share cifs clientgroup list [-h] [-f filesystem_name] (-s share_name
| -p path)
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-f filesystem_name | --fs` *`filesystem_name`*	The underlying cluster filesystem name.
`-s` *`share_name`* `| --share` *`share_name`*	Name of the share.
`-p` *`path`* `| --path` *`path`*	Relative (within the underlying cluster filesystem) path to the shared folder.

cng share cifs clientgroup set

The command `cng share cifs clientgroup set` is the second level subcommand of the `cng share cifs clientgroup` subcommand. Also see *Use of cscli cng share cifs clientgroup (list, set)* on page 42.

Synopsis

```
$ cscli cng share cifs clientgroup set [-h] [-f filesystem_name] (-s share_name
| -p path) | -N clientgroup_name [--writeable {yes,no}] [--denied {yes,no}]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-f filesystem_name | --fs` *`filesystem_name`*	The underlying cluster filesystem name.
`-s` *`share_name`* `| --share` *`share_name`*	Name of the share.
`-p` *`path`* `| --path` *`path`*	Relative (within the underlying cluster filesystem) path to the shared folder.
`-N` *`clientgroup_name`* `| --clientgroup` *`clientgroup_name`*	The client group name.
`--writeable {yes,no}`	If the share is writeable for the client group, yes or no.
`--denied {yes,no}`	If the access from the client group is denied, yes or no.

cng share nfs

The command `cng share nfs` is a second-level subcommand of the `cng share` subcommand.

Synopsis

```
$ cscli cng share nfs [-h] {clientgroup}
```

Positional Arguments	Description
`clientgroup`	Manages the share NFS per network configuration.
`network`	This command is deprecated. See clientgroup above.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng share nfs clientgroup

The command `cng share nfs clientgroup` is a second-level subcommand of the `cng share nfs` subcommand. Also see *Use of cscli cng share nfs clientgroup (list, set)* on page 43.

Synopsis

```
$ cscli cng share nfs clientgroup [-h] {set,list,delete}
```

Positional Arguments	Description
`delete`	Deletes an NFS client group.
`list`	Displays a list of available NFS client groups.
`set`	Creates/modifies NFS client group options.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng share nfs clientgroup delete

The command `cng share nfs clientgroup delete` is a second-level subcommand of the `cng share nfs clientgroup` subcommand.

Synopsis

```
$ cscli cng share nfs clientgroup delete [-h] [-f filesystem_name] (-s share_name
| -p path) -N clientgroup_name
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-f filesystem_name | --fs` *`filesystem_name`*	The underlying cluster filesystem name.
`-s` *`share_name`* `| --share` *`share_name`*	Name of the share.

Optional Arguments	Description
`-p` *`path`* `|` `--path` *`path`*	Relative (within the underlying cluster filesystem) path to the shared folder.
`-N` *`clientgroup_name`* `|` `--network` *`clientgroup_name`*	The client group name.

cng share nfs clientgroup list

The command `cng share nfs clientgroup list` is a second-level subcommand of the `cng share nfs clientgroup` subcommand.

Synopsis

```
$ cscli cng share nfs clientgroup list [-h] [-f filesystem_name] (-s share_name
| -p path)
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-f filesystem_name | --fs` *`filesystem_name`*	The underlying cluster filesystem name.
`-s` *`share_name`* `|` `--share` *`share_name`*	Name of the share.
`-p` *`path`* `|` `--path` *`path`*	Relative (within the underlying cluster filesystem) path to the shared folder.

cng share nfs clientgroup set

The command `cng share nfs clientgroup set` is a second-level subcommand of the `cng share nfs clientgroup` subcommand.

Synopsis

```
$ cscli cng share nfs clientgroup set [-h] [-f filesystem_name] (-s share_name |
-p path) -N clientgroup_name [--writeable {yes,no}] [--async {yes,no}] [--squash
{root,no_root,all} | --squash-unset] [--subtree-check {yes,no}] [--anonuid
anonuid | --anonuid-unset] [--anongid anongid | --anongid-unset]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-f filesystem_name | --fs` *`filesystem_name`*	The underlying cluster filesystem name.
`-s` *`share_name`* `|` `--share` *`share_name`*	Name of the share.
`-p` *`path`* `|` `--path` *`path`*	Relative (within the underlying cluster filesystem) path to the shared folder.
`-N` *`clientgroup_name`* `|` `--network` *`clientgroup_name`*	Client group name.

Optional Arguments	Description
`--writeable {yes,no}`	Whether or not the share is writeable from network
`--async {yes,no}`	This option allows the NFS server to violate the NFS protocol and reply to requests before any changes made by that request have been committed to stable storage.
`--squash {root,no_root,all}`	Sets method of uid/gid squashing:root - map requests from uid/gid 0 to the anonymous uid/gidno_root - turn off root squashingall - map all uids and gids to the anonymous user.
`--squash-unset`	Unsets squash option and use defaults.
`--subtree-check {yes,no}`	This option disables/enables subtree checking, which has mild security implications, but can improve reliability in some circumstances.
`--anonuid` *`anonuid`*	This option explicitly sets the uid of the anonymous account.
`--anonuid-unset`	Unsets the anonuid option.
`--anongid` *`anongid`*	This option explicitly sets the gid of the anonymous account.
`--anongid-unset`	Unsets the anongid option.

cng show

The `cng show` subcommand displays the CIFS/NFS gateway configuration.

Synopsis

```
$ cscli cng show [-h] {node,all,winbind,acl,interface,clientgroup}
```

Positional Arguments	Description
cng show acl	Displays ACL settings.
cng show all	Displays all CIFS/NFS settings.
cng show clientgroup	Displays client groups.
cng show interface	Displays current ECN Export Interface configuration.
cng show node	Displays node settings.
cng show windbind	Displays winbind settings.
`network`	This subcommand is deprecated. See `clientgroup` above.

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng show acl

The command `cng show acl` is a second-level subcommand of the cng show subcommand.

Synopsis

```
$ cscli cng show acl -h
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng show all

The command `cng show all` is a second-level subcommand of the `cng show` subcommand.

Synopsis

```
$ cscli cng show all [-h]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

Also see *Use of cscli cng show all* on page 51.

cng show clientgroup

The command `cng show clientgroup` is a second-level subcommand of the `cng show` subcommand.

Synopsis

```
$ cscli cng show clientgroup [-h]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng show interface

The command `cng show interface` is a second-level subcommand of the `cng show` subcommand.

Synopsis

```
$ cscli cng show interface [-h]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng show node

The command `cng show node` is a second-level subcommand of the `cng show` subcommand.

Synopsis

```
$ cscli cng show node -h
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.

cng show windbind

The command `cng show windbind` is a second-level subcommand of the `cng show` subcommand.

Synopsis

```
$ cscli cng show winbind [-h]
```

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.