Sonexion™ 2000 Administration Guide

(2.0.0)

S-2537 Rev C

Contents

1 About the Sonexion™ 2000 Administration Guide 5
2 Hardware Architecture 7
3 Software Architecture 10
4 What Is and Is Not Supported in Sonexion 2.0.0 12
5 Log In to CSM 14
6 SSA 16
6.1 SSA Introduction 16
6.2 Configure SSA for Sonexion 16
6.2.1 SSA Prerequisites 17
6.2.2 Install the Sonexion SSA RPM Packages 17
6.3 Sonexion InfiniBand Fabric Plugins 23
7 Configure LNet Fine-Grained Routing for XC Systems 25
7.1 Routing and Bandwidth Matching for Sonexion Systems 28
7.2 External Server Node: Sonexion 3000 or ClusterStor L300 System Recommended Parameters 30
7.3 Use CLCVT to Configure Fine-Grained Routing Files 32
7.3.1 CLCVT Prerequisite Files 32
7.3.2 Create the CLCVT info.file-system-identifier File 33
7.3.3 Create the CLCVT client-system.hosts File 36
7.3.4 Create the CLCVT client-system.ib File 36
7.3.5 Create the CLCVT cluster-name.ib File 37
7.3.6 Create the CLCVT client-system.rtrIm File 38
7.3.7 Generate ip2nets and Routes Information 39
7.4 Place the LNet Configuration on a CLE6 System 42
7.4.1 Copy Files for External Lustre Fine-grained Routing 42
7.4.2 Configure LNet Routers for Fine-Grained Routing 43
7.5 Create Custom LNet Configuration for Sonexion 45
8 Change Network Settings 48
8.1 Create Custom LNet Configuration for Sonexion 48
8.2 Change DNS Resolver Configuration 50
8.3 Change Externally Facing IP Addresses 50
8.4 Configure LDAP over TLS in Daily Mode 51
8.5 Change LDAP Settings in Daily Mode 53
8.6 Configure NIS Support in Daily Mode 55
8.7 Change NIS Settings in Daily Mode 56
9 Change the NTP Configuration 58

10 Check File Systems Using `e2fsck` Command 59
11 Disk Watching Daemon 61
12 Distributed Namespace (DNE) Feature 64
13 Support Bundles 67
13.1 Collect Support Files Manually 68
13.2 Create Support Bundles Using CLI 69
13.3 Import a Support File 70
13.4 Download a Support File 71
13.5 Delete a Support File 71
13.6 View Support Files 72
13.7 Interpret Support Bundles 73
14 Troubleshooting Overview 75
14.1 Management Node Shutdown Corrupts Database 75
14.2 Invalid Puppet Certificate on Diskless Node Boot-up 77
14.3 Upgrade Error "Metadata File does not Match Checksum" 79
14.4 Capture Dumps for Mellanox Host Adaptors and Switches 79
14.5 Reseat a Problematic High-Speed Network Cable 80
14.6 RAIDs Assembled Incorrectly on Nodes 82
14.6.1 Prepare to Recover a Failed RAID Array 83
14.6.2 Recover a Failed RAID Array 85
14.7 Non-Responsive Server 89
14.8 `conman` to GEM Causes "Port In Use" Message 89
14.9 HA Failover on xyvinc Malfunction 90
14.10 Performance Tab Appears Blank 92
14.11 Isolate a Slow Disk 93
14.12 Warning After cscli unmount: "Database assertion: created a new connection but..." 94
15 CSCLI Introduction 96
15.1 CSCLI Command Summary 98
15.2 Administrative Commands 101
15.3 Administrative Settings Commands 104
15.4 `alerts` Display Command 107
15.5 `alerts_config` Command 110
15.6 `alerts_notify` Command 116
15.7 `async_journal` Command 117
15.8 Configuration Commands 118
15.9 `dm` and `dwd` Unresponsive Drives (UD) Commands 120
15.10 Filter Commands 123
15.11 `ibstat_check` Command 124

15.12 lustre_users Commands ... 124
15.12.1 lustre_users show Command ... 125
15.12.2 lustre_users apply Command ... 126
15.12.3 lustre_users local Commands ... 126
15.12.4 lustre_users upcall Authentication Commands ... 128
15.12.5 lustre_users order Commands ... 129
15.12.6 lustre_users nis Service Commands ... 130
15.12.7 lustre_users ad Service Commands ... 131
15.12.8 lustre_users ldap Service Commands ... 133
15.12.9 lustre_users Commands, New vs. Old ... 134
15.13 lustre_network Commands ... 135
15.14 lustre_perf Command ... 137
15.15 monitor Command for System Health ... 138
15.16 monitor Command Examples ... 139
15.17 netfilter_level Command ... 143
15.18 Network Setup Commands ... 144
15.19 Node Control Commands ... 145
15.20 Node Position and Management Commands ... 148
15.21 RAID Management Commands ... 150
15.22 Rack Management Command ... 152
15.23 service_console Commands ... 153
15.24 service_console configure Command ... 160
15.25 System Software Update Commands ... 163
15.26 support_bundle Command ... 163
16 GEM CLI Commands ... 166
16.1 ddump ... 167
16.2 getboardid ... 168
16.3 getmetisstatus ... 168
16.4 getvpd ... 168
16.5 help ... 169
16.6 ipmi_power ... 170
16.7 ipmi_setosboot ... 170
16.8 logdump ... 171
16.9 report_faults ... 171
16.10 settime ... 172
16.11 ver ... 172

1 About the Sonexion™ 2000 Administration Guide

The *Sonexion™ 2000 Administration Guide, (2.0.0 Rev C) S-2537* procedures help administrate and troubleshoot Cray Sonexion 900, 1600, and 2000 storage systems (software releases 2.0 and earlier).

Record of Revision

Publication	Date	Description
Sonexion 2000 Administration Guide (2.0.0 Rev C) S-2537	May 2018	Updated database schema example for LDAP settings.
Sonexion™ Administrator's Guide (1.5.0) S-2537	October 2014	Supports software release 1.5.0 for Sonexion models 900, 1600, and 2000.
Sonexion™ Administrator's Guide (1.4.0 Rev A) S-2537	July 2014	Supports software release 1.4.0 for Sonexion models 900, 1600, and 2000.
Sonexion™ Administrator's Guide (1.3.1 Rev B) S-2537	March 2014	Supports software release 1.3.1 for Sonexion models 900, 1600, and 2000.

Scope and Audience

The procedures presented in this manual are to be carried about by site administrators (employed by Cray Inc., or the customer organization) where Sonexion systems are installed.

Typographic Conventions

`Monospace`	Indicates program code, reserved words, library functions, command-line prompts, screen output, file/path names, and other software constructs.
`Monospaced Bold`	Indicates commands that must be entered on a command line or in response to an interactive prompt.
`Oblique` or *`Italics`*	Indicates user-supplied values in commands or syntax definitions.
Proportional Bold	Indicates a **GUI Window**, **GUI element**, cascading menu (**Ctrl**→**Alt**→**Delete**), or key strokes (press **Enter**).
\ (backslash)	At the end of a command line, indicates the Linux® shell line continuation character (lines joined by a backslash are parsed as a single line).

Trademarks

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, URIKA, and YARCDATA. The following are trademarks of Cray Inc.: APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYDOC, CRAYPAT, CRAYPORT, DATAWARP, ECOPHLEX, LIBSCI, NODEKARE. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.

2 Hardware Architecture

The Sonexion storage system provides an integrated hardware and software implementation of the Lustre® file system using a rack-scale appliance model. All hardware and software is pre-installed, pre-configured, and pre-tested before delivery to the end user. For small- and medium-scale installations Sonexion ships to customers as a pre-configured, rack-level, integrated cluster that can be incrementally expanded by adding additional components and racks as needed. In this form Cray maintains close control of both the hardware and software configuration. Large-scale installations consisting of six or more racks typically incorporate some custom configuration, especially related to InfiniBand (IB) network cabling, to integrate with the end-user infrastructure.

The Sonexion 2000 offers these features:

- SSU delivering 7.5 GB/s
- Linear speed and capacity scaling
- Integrated metadata controller
- Integrated controller and storage
- Six terabyte drives
- Built-in FDR InfiniBand (IB) and gigabit Ethernet (GigE) switches
- Six storage modules per rack
- Redundant power supplies

The Sonexion 2000 hardware architecture consists of a pre-configured, rack-level storage cluster that can be easily expanded using modular storage node building blocks. The principal hardware components include:

- Metadata Management Unit (MMU)
- Scalable Storage Unit (SSU)
- Network Fabric Switches
- Management Switch

The Sonexion solution is differentiated from other file system solutions by its innovative MMU and SSU architectures.

Metadata Management Unit

The Metadata Management Unit (MMU) is a quad-node server which contains the two management (MGMT) nodes, the MGS node, the MDS node, and one shelf of high-availability shared storage. The central point of management for the entire storage cluster, the MMU runs CSSM software, manages network request handling, and monitors every storage element within the cluster. Sonexion interface ports support InfiniBand fabric network interface technology connections and 1GbE management network connections.

The MMU is fully redundant and fault-tolerant. Each node is configured for active-passive failover, with an active instance of the server running on one system and a passive instance of the node running on the peer system. If an active node fails, for example, the MDS goes down, then the passive MDS node takes over the MDT operations of the failed MDS. The shared storage of the MMU supports a combination of Small Form Factor (SFF) SAS HDD, protected using RAID 1, for management data, file system data, and journal acceleration.

Sonexion supports InfiniBand connections to the MGMT, MDS, and MGS nodes. Additionally, each server connects, via Ethernet, to dedicated private management networks supporting Intelligent Platform Management Interface (IPMI).

Scalable Storage Unit

The core building block is the Scalable Storage Unit (SSU). Each SSU is configured with identical hardware and software components, and hosts two OSS nodes. The SSU contains two OSSs, with RAID-protected, high availability shared storage, and interface ports to support InfiniBand data networks and 1GbE management network connections.

The OSSs are Storage Bridge Bay (SBB) compliant with SAS expanders that enable both modules to directly access all drives in the enclosure (a differentiator among fully fault-tolerant systems). The OSSs connect through a common midplane, eliminating the need for extra external cables, and share a redundant, high-speed interconnect across the midplane for failover services.

The SSU is fully redundant and fault-tolerant, thus ensuring maximum data availability. Each OSS serves as a Lustre node, accessing the disk as shared OST storage and providing active-active failover. If one OSS fails, the active module manages the OSTs and the disk operations of the failed node. In non-failure mode, the I/O load is balanced between modules. The SSU's shared storage consists of high-capacity SAS disk drives, configured in a RAID 6 array to protect against double disk failures and drive failure during rebuilds.

Network Fabric Switches

The Network Fabric switches (InfiniBand) manage I/O traffic and provide network redundancy throughout the Sonexion solution. To maximize network reliability, the OSSs in the SSU are connected to redundant network switches. If one switch fails, the second module in the SSU (connected to the active switch) manages the OSTs of the module connected to the failed switch.

To maintain continuous management connectivity within the solution, the network switches are fully redundant at every point and interconnected to provide local access from the MGMT, MDS, and MGS nodes to all storage nodes.

Management Switches

The Management switches are a dedicated local network on dual 1GbE switches, used for configuration management and health monitoring of all components in the Sonexion system. The management network is private and not used for data I/O in the cluster. This network is also used for IPMI traffic to the SSU's OSSes, enabling them to be power-cycled by CSSM.

3 Software Architecture

Sonexion 2000 software architecture consists of an integrated, multi-layer software stack:

- Cray Sonexion System Manager (CSSM)
- Management Nodes (MGMT)
- Lustre file system
- Management Server (MGS)
- Metadata Server (MDS)
- Object Storage Server (OSS)
- Lustre Clients
- Data protection layer (RAID)
- Unified System Management (USM) software, including Generic Enclosure Management (GEM) software
- Linux OS Additional DNE (Distributed Namespace) Units (ADU)

Sonexion 2000 systems run Lustre software in a Scientific Linux 6.2 environment. The file system is fully integrated with the Sonexion System Manager, GEM, and RAID layers in the stack.

System Manager (CSM)

The Sonexion system management software (CSM) provides a user-interface UI to the Sonexion infrastructure. The UI simplifies cluster installation and configuration and provides consolidated management and control of the entire storage cluster. CSM also provides distributed component services to manage and monitor system hardware and software.

CSM includes wizards to guide the processes of configuration and node provisioning. Once the cluster is running, use the GUI to manage the storage environment with these functions:

- Start and stop file systems

- Manage node failover
- Monitor node status
- Collect and browse performance data

The dashboard reports errors and warnings for the storage cluster and provides tools to aid in troubleshooting, including cluster-wide statistics, system snapshots, and Lustre syslog data.

To maximize availability, CSM works with USM, the platform's integrated management software, to provide comprehensive system health monitoring, error logging, and fault diagnosis. Users are alerted to changing system conditions and degraded or failed components.

Data Protection Layer (RAID)

Sonexion uses Redundant Array of Independent Disks (RAID) to provide different data protection layers throughout the system. For new builds, the RAID subsystem has been changed to GridRAID technology.

Unified System Management (USM) Firmware

Extensive Sonexion system diagnostics are managed by USM management firmware, which runs on each OSS in the scaleable storage unit (SSU). USM monitors and controls the SSU's hardware infrastructure and overall system environmental conditions, providing a range of services including SES and high-availability (HA) capabilities for system hardware and software. USM offers these key features:

- Manages system health, providing random-access services (RAS) that cover all major components such as disks, fans, power-supply units (PSUs), Serial Attached SCSI (SAS) fabrics, PCI (Peripheral Component Interconnect) buses, memories, and CPUs, and provides alerts, logging, diagnostics, and recovery mechanisms
- Power control of hardware subsystems that can be used to individually power-cycle major subsystems and provide additional RAS capabilities – this includes drives, servers, and enclosures
- Fault-tolerant firmware upgrade management
- Monitoring of fans, thermals, power consumption, voltage levels, AC inputs, field-replaceable unit (FRU) presence, and health
- Efficient adaptive cooling to keep the SSU in optimal thermal condition, using as little energy as possible
- Extensive event capture and logging mechanisms to support file system failover capabilities and to allow for post-failure analysis of all major hardware components

4 What Is and Is Not Supported in Sonexion 2.0.0

What is Supported

Qualified Functionality

- Installation and deployment of Sonexion 2000 systems with the following SSU configurations:
  - Single or Multi-SSU
  - SSU + Single ESU
- High Availability
  - Dual Management Network Switch Redundancy (DMN)
  - PDU Redundancy
  - SSU Node Failover/Failback
- Chrome, FireFox and Safari browsers for Windows, Linux and MacOS
- All custom GridRAID Optimal Performance and SCSI code updates and patches
- Lustre Performance Monitoring of LMT with Lustre 2.5.1
- Support File Bundle Collection
- ADU/DNE Addition Procedure
- CSCLI
- GridRAID
- CNG
- SSU Addition (Single SSU and SSU + Single ESU)
- FRU Replacements
- Internet Explorer 10/11

New Functionality

- Qualification of the following new firmware:
  - Mellanox FW 9.2.8000 (SX6025)
  - Mellanox ConnectX3 VPI firmware for Sonexion 2000: 2.32.5100 (IB only)
  - Mellanox ConnectX3 VPI firmware for Intel MMU: 2.30.8000 (IB only)
- Qualification of the Intel 2U Firmware/BIOS version: SE5C600.86B.02.03.0003.041920141333
- New Drive Support (see Supported Hardware Platform Components and References)
- RAS DDIC (Disk Drive In Carrier) Serviceability Feature
- ADU/DNE Addition Procedure

- ADU/DNE Management and Monitoring (GUI and CSCLI)
- Automated InfiniBand fabric Switch Configuration (Brocade only)
- Dashboard Support for CNG nodes
- Dashboard Support for ADU nodes

Hardware

- 60A (48A Derated) US PDU within the Sonexion rack

What is NOT Supported

- Ethernet fabric for data
- Software upgrades from any Sonexion 1.x releases to Sonexion 2.0.0
- Software upgrades to future releases from Sonexion 2.0.0

5 Log In to CSM

Use either of the following procedures to begin using Sonexion System Manager (CSM), the primary administrative interface for the Sonexion system.

Log In to CSM via the CSCLI

To use the CSCLI interface log in to the active MGMT node via SSH:

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

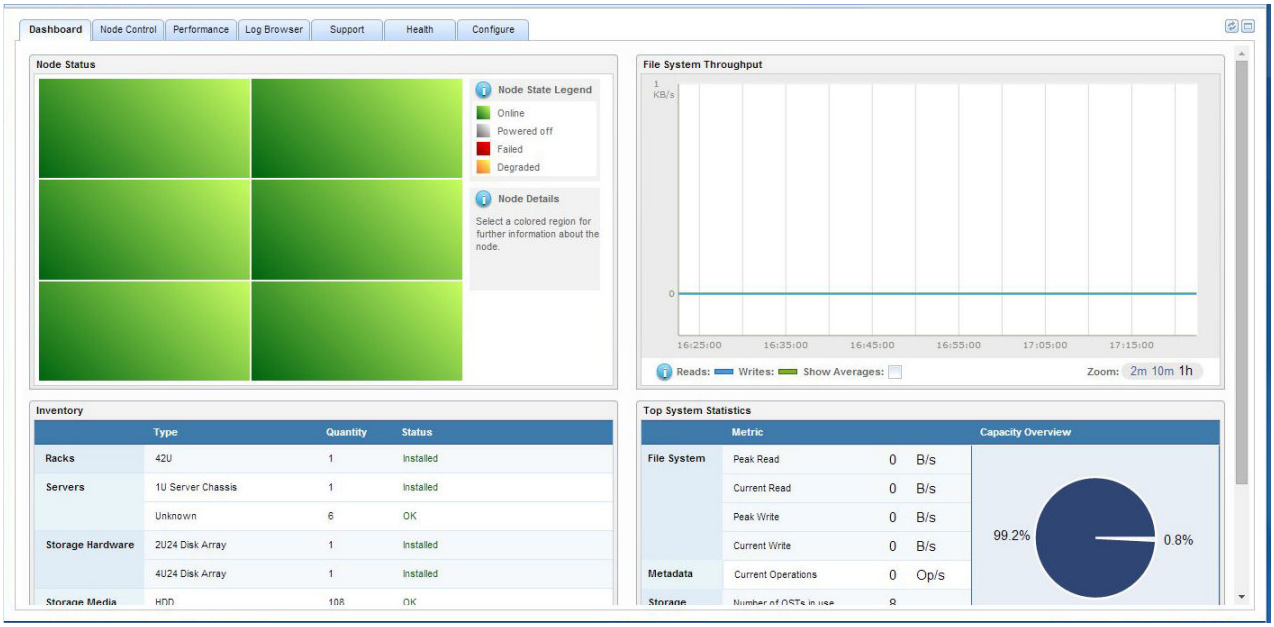
Log In to CSM via the GUI

Access the CSM GUI using a direct connection between the site's IP network and the MGMT nodes. Log in to the GUI using a browser on a PC or on a terminal that is connected to the site's IP network. CSM supports Chrome, Firefox, or Internet Explorer browsers.

1. Open a browser window.
2. Enter the CSM IP address in the browser's URL field, and press **Enter**.
3. Enter the administrator password used to access CSSM at the site, and press **Enter**.

   A **What's New** window will appear. Close this window to see the **CSSM Dashboard** tab open in the browser window.

   Figure 1. Dashboard Tab After Login



The **Dashboard** tab displays individual widgets to view status and performance data. From the Dashboard, users can survey high-level system health and resolve issues with individual nodes and take further action by quickly linking to the **Node Control** tab.

6 SSA

6.1 SSA Introduction

The System Snapshot Analyzer (SSA) software supports analytics technology for Sonexion® systems.

SSA collects, analyzes, and uploads (if uploading is enabled) product health, configuration, or triage information to Cray. Uploaded information captured by SSA is analyzed using a sophisticated analytics platform that detects and enumerates changes over time, changes in the health state of the system, and processes triage information to assist with case resolution. Through automation, SSA is designed to improve the overall customer experience by reducing the manual effort and time required to report and resolve support issues.

The SSA Sonexion shepherd client software manages the collection, first-level analysis, and secure transport of support information back to Cray.

For additional information and references on SSA, please visit *http://www.cray.com/support*. For details on how to activate an SSA account and download SSA software, refer to Cray SFDC/Case Management Knowledge *Article 4546*, *Getting Started with the Cray System Snapshot Analyzer (SSA)*.

Request Support for SSA

To request support, contact a Cray support representative or file a service case against the Sonexion SSA component. Please submit request for enhancements (RFEs) or defect reports (bugs) against SSA. Feedback and suggestions are valued and welcomed.

6.2 Configure SSA for Sonexion

Download the Sonexion SSA 1.6.5 shepherd software by following guidance in *Cray Knowledge Article 4546*, *Getting Started with the Cray System Snapshot Analyzer (SSA)*.

A CrayPort account is required to download software and activate an SSA account. SSA account activation is required to obtain authentication credentials for use during configuration and to enable snapshot uploads to Cray.

Visit *https://crayport.cray.com* if you have active support entitlements and would like to register for CrayPort accounts.

Shepherd Overview

Sonexion SSA shepherd software is delivered in two packages. One contains the shepherd command line interface (`ssacli`) and libraries. The other contains plugins for a specific Cray platform (e.g., Cray XE, Cray XC,

or Sonexion). These packages are revision matched, with the plugin package dependent on the base shepherd software.

The shepherd operates in three, progressive stages—collect, snapshot, and upload.

Collect Stage	The collect stage is responsible for utilizing plugins to perform collection and analysis of system information, and to prepare for subsequent stages.
Snapshot Stage	The snapshot stage searches for collections which have not been previously processed. It encodes them into a network friendly format in preparation for upload to Cray.
Upload Stage	The upload stage then takes any snapshots which have not been uploaded and attempts to upload them to Cray for further analysis and processing.

The upload takes place over a secure network connection, using transport layer security (TLS), and is further authenticated using credentials from customer SSA accounts.

The shepherd uses a configurable purge policy to remove older collections or snapshots.

The shepherd is either invoked on-demand or automatically on a schedule via `cron`. Information on supported run configurations of the shepherd and their purpose are covered in *Collect and Upload a Snapshot* on page 21.

6.2.1 SSA Prerequisites

SSA Upload Account Activation

The activation of an SSA account in CrayPort is required in order to obtain the organization name and passphrase to use in the configuration of SSA for the upload of snapshot information to Cray.

Network Connectivity

The shepherd application only initiates outbound network connection—over TCP/IP (version 4) TCP port 443—to the network host `ssa.cray.com`. Cray does not initiate an inbound connection to the customer network/system. The outbound network session is established only long enough to submit a snapshot of information to Cray. Then it terminates.

To communicate with the Cray upload system, outbound connectivity must be provided from the user site as described. Optionally, a local network proxy (HTTP, SOCKS) can be utilized through features in the shepherd application. For additional details about how to configure a proxy, refer to the comments in the `shepherd.conf` file distributed with the shepherd.

Configuration of the `sudo` Utility

The shepherd leverages the functionality provided by the `sudo` utility to remove privileges and provide an audit trail of application activity. The `root` user must be able to execute any command, as any user, on installation nodes via `sudo`. Sudo must be configured so that the `root` user is not prompted for a password to execute `sudo`.

Disable `requiretty`

In addition, `requiretty` must be disabled to enable `sudo` execute scheduled jobs via SSA (e.g., those not run directly on a real TTY, by an interactive user).

6.2.2 Install the Sonexion SSA RPM Packages

Prerequisites

This procedure must be performed as the `root` user on the Sonexion management node.

About this task

This procedure installs the base SSA shepherd application and system-specific plugins for systems.

Time Required: Approximately 30 minutes. SSA RPMs can be installed during customer operations and while the Lustre file systems are running.

Packages for Sonexion are used in the examples for illustration only. Always download the latest shepherd software specific to the system.

Perform the RPM installation steps on the primary management node, then on the secondary management node. The SSA software is not synchronized between the two management nodes. The configuration and SSA client (shepherd) versions must be the same on each management node.

The SSA installation process creates cron entries in `/etc/cron.d/cray-ssa/` for scheduled operation.

Procedure

1. Log in to the management node as `root`.
2. If upgrading SSA, disable SSA using *Enable or Disable SSA*.
3. Install the RPM packages:

```
# rpm -ivh \
cray-ssa-shepherd-1.6.5-0.x86_64.rpm cray-ssa-shepherd-sonexion-plugins-1.6.5-0.x86_64.rpm
Preparing...                          ################################# [100%]
   1:cray-ssa-shepherd        ################################# [ 50%]
Active mgmt node, attempting to create /mnt/mgmt/var/opt/cray/ssa directory structure ...
Setting Alternatives (update-alternatives) ...
   2:cray-ssa-shepherd-sonex################################# [100%]
```

   Installation of the RPMs yields messages associated with the `alternatives` software. For information about how to manage the version of shepherd used, refer to the `man` page for `update-alternatives` and *Specify a Different Version of SSA* (if the RPM for the platform uses `alternatives`).

4. Repeat the steps above on the secondary management node.
5. Exit and `sudo` to `root` on each management node to invoke the new `root` user environment.

6.2.2.1 Edit the Sonexion Shepherd Configuration File

Prerequisites

This procedure must be performed as the `root` user.

About this task

The Sonexion shepherd uses a single configuration file, `/opt/cray/ssa/default/etc/shepherd.conf`. The configuration file is structured into sections and contained within square brackets, for example `[control]`. Each section and each of its related parameters contain a header of descriptive configuration information. Sections may contain individual or groups of related parameters. Prior to running the shepherd, edit this file on the primary management node.

The steps below describe commonly configured parameters for desired shepherd features. It is imperative that both the primary and secondary management nodes have the same shepherd configuration entries. After the completion of the procedure below on the primary management node, copy the configuration file over to the secondary management node.

Always make a backup copy of the shepherd configuration file (`shepherd.conf`) for reference.

> **IMPORTANT:** Review the SSA release notes for versions that have compatible configuration files. If the version being installed is compatible with the previous version, simply make a backup copy of the new version's configuration file and overwrite it with the existing shepherd configuration file.

Procedure

1. Login to the management node as `root`.
   - If using an existing compatible configuration file, copy the file to the appropriate location as shown and skip to step 8.

     ```
     MGMT0#  cd /opt/cray/ssa/default/etc
     MGMT0#  cp -a shepherd.conf shepherd.conf-dist
     MGMT0#  cp -a /opt/cray/ssa/ssa_version/etc/shepherd.conf .
     ```
   - If this is an initial installation, proceed to step 2.
2. From the primary management node, edit `/opt/cray/ssa/default/etc/shepherd.conf` file.
3. Enable the shepherd master operation mode. This configuration setting allows all stages (collection, snapshot, upload) to be active.

   Set `master_enabled: true` in the `[control]` section.

   ```
   [control]
   master_enabled: true
   ```
4. Set system identification information in the `[sysinfo]` section.
   a. Set the serial number of the system.

      ```
      [sysinfo]
      serial_num: 99999
      ```
   b. Set the system type.

      ```
      [sysinfo]
      system_type: SNX3000
      ```
   c. Set the system name.

```
[sysinfo]
system_name: prod
```

d. Set a short system description.

```
[sysinfo]
system_description: SNX3000 PROD
```

5. Set upload information in the `[upload]` section.

> **IMPORTANT:** If SSA must be run in local mode, (no information is uploaded to Cray), see *Configure SSA for Local Only Mode* and skip steps 5 through 7.

The `upload_server`, `upload_org`, and `upload_pw` parameters in the `[upload]` section must be set before using the snapshot or upload stages.

a. Set the organization received when the SSA account was activated that snapshots will be uploaded for.

```
[upload]
upload_org: cto01
```

b. Set the password received when the SSA account was activated.

```
[upload]
upload_pw: SuperSecretPassword
```

These parameters must be set for a successful authentication and upload of SSA information to Cray.

6. Optional: Set, in the `[upload]` section, the upload server (`upload_server`) address to an IP address. Or, if using IP name resolution, set an additional X.509 (SSL/TLS) certificate validation option.

a. Optional: If using DNS or a local resolution method (e.g., `/etc/hosts`), set `verify_x509_host` to `true`.

This enables additional protection within the shepherd to validate that the subject name in the X.509 SSL/TLS certificate matches that of the server (ssa.cray.com), allows resolution of the DNS name ssa.cray.com, adds a small amount of security to the upload process, and should be enabled if using a suitable resolution method. The certificate chain for the SSA upload system is maintained locally within the shepherd application (isolated from other certificate stores on the system). The CA bundle file is located in `/opt/cray/ssa/default/etc/ssl/ssa.pem`.

```
[upload]
verify_x509_host: true
```

b. Optional: If not using DNS and if a manually configured local resolution method is not desired, set the `upload_server` to an IP address.

The upload system uses, at the time of this writing, a single IP address, `136.162.62.191`. This IP address should resolve via a DNS PTR reference to an A record ending in `.cray.com`. Any changes in SSA upload addressing will be communicated directly to customers by Cray.

1. Set `upload_server` to IP address (using the documented IP above).

```
[upload]
upload_server: 136.162.62.191
```

2. Ensure that `verify_x509_host` is set to false.

```
[upload]
verify_x509_host: false
```

7. Save the file and exit.

After the configuration file is saved (with `master_enabled: true`) the `cron` schedule for the shepherd will be activated. The `cron` schedule can be located in `/opt/cray/ssa/default/etc/cray-ssa`. It is symbolically linked from `/etc/cron.d/cray-ssa`.

8. Validate the configuration.

```
MGMT0# ssacli --check_conf
[stdout] Configuration File and CLI Options Valid.
```

The `ssacli` command should only be executed on the primary management node. If output differs from the above, review the messages presented and make corrections to the configuration file.

9. Use `scp` to copy the configuration from the primary management node to the secondary management node.

```
MGMT0# scp -p /opt/cray/ssa/default/etc/shepherd.conf \
snx99999n001:/opt/cray/ssa/default/etc/shepherd.conf
```

6.2.2.2 Collect and Upload a Snapshot

Prerequisites

IMPORTANT: This procedure is not necessary if upgrading to a newer version of SSA software.

This procedure must be performed on the Sonexion primary management node as `root`. The output examples in this section have been reduced for the sake of brevity. They will vary depending on the task being performed.

About this task

SSA terminology:

Run Set — Each (shepherd) plugin is associated with zero or more plugin run sets. A run set is an alpha-numeric, textual label for a configuration that the shepherd uses to select plugins for execution. If `ssacli` is not invoked with `--runset` options, the default run set is `default`. Plugins can, and often are, associated with multiple plugin run sets.

Output Channel — Every shepherd collection and related snapshot is associated with exactly one output channel (channel). A channel is an alpha-numeric label. If `ssacli` is not invoked with the `--channel` option, the default channel is `default`. Channels allow information collected and reported by SSA to be categorized by use.

Release 1.6.5 contains four run sets, each targeting a different SSA use case. These run sets are provided in the table.

Table 1. Supported Shepherd Run Configurations on Sonexion Systems

Purpose	Run Set	Output Channel	Scheduled/On-Demand	Frequency
Baselining and detecting change in product configurations	`default`	`default`	Either	Once daily

Purpose	Run Set	Output Channel	Scheduled/On-Demand	Frequency
Product health monitoring	`health`	`health`	Scheduled	Every 15 minutes
Seagate telemetry information gathering	`seagate_telemetry`	`seagate_telemetry`	Scheduled	Every 15 minutes (offset by 7 minutes from `health`)
Product support bundle capture	`triage`	`triage`	On-demand	As requested

To observe behavior for the `default` run set and channel, perform each stage of the shepherd process separately and review the shepherd output on the system console.

IMPORTANT: This procedure is not necessary if upgrading to a newer version of SSA software.

Procedure

1. Execute a collection for the `default` run set and channel.

   The execution time can vary between several minutes to up to 20 minutes on the largest Sonexion systems (more than 100 SSUs).

```
[root@snx99999n000]# ssacli --collect
[stdout] UI master_control status is (enabled)
[stdout] MASTER CONTROLS -> (M:True, C:True, S:True, U:True)
[stdout] UI CONTROLS -> (C:True, S:False, U:False)
[stdout] Shepherd Session: 1515075962
[stdout] Exclusive run: Lock file created @ /mnt/mgmt/var/opt/cray/ssa/lock/ssa.lock_channel-
default_device-snx99999n000_SNX2000_99999
[stdout] COLLECT stage start
[stdout] PLOAD: 28 plugin source modules loaded
[stdout] Collection Session:   '1515075962'
[stdout] Collection Directory: '/mnt/mgmt/var/opt/cray/ssa/collection/snx99999n000_SNX2000_99999/
default/1515075962'
[stdout] Collection Channel:   'default'
[stdout] Run Sets:             '['default']'
[stdout] Explicit Plugins:     'None'
...
[stdout] ** Entering run-level 20 **
[stdout] Plugin 'shepherd.encode.diagnostic' started
[stdout] Plugin 'shepherd.encode.diagnostic' stopped, return 0, time 9.67
[stdout] 25333557.0 raw bytes collected via directives.
[stdout] no collection directories meet purge requirements
[stdout] COLLECT stage stop (normally)
[stdout] Collection output directory @ /mnt/mgmt/var/opt/cray/ssa/collection/
snx99999n000_SNX2000_99999/default/1515075962
[stdout] shepherd session stop successfully
[stdout] run took 100.70 seconds
```

   If the `COLLECT` stage stops either normally or as stated with survivable exception, the collection process was successful. The shepherd creates a text report named `collection_report.txt` in the collection output directory located
   in: `/mnt/mgmt/var/opt/cray/ssa/collection/<collection_device>/default/<timestamp>`

   This report provides a high-level summary of the collection, including:

   - Information on the shepherd
   - Amount of storage consumed by the collection

- The status of health checks the shepherd performed during the collection
- Platform summary information
- A plugin execution summary trace

This report can be useful to local operators in reviewing system status and high-level configuration. Survivable exceptions are part of the shepherd design. Individual plugins can fail in controlled ways, report these failures, and continue operation.

2. Execute a snapshot for the `default` run set and channel.

```
[root@snx99999n000]# ssacli --snapshot
[stdout] UI master_control status is (enabled)
[stdout] MASTER CONTROLS -> (M:True, C:True, S:True, U:True)
[stdout] UI CONTROLS -> (C:False, S:True, U:False)
[stdout] Shepherd Session: 1515076208
[stdout] Exclusive run: Lock file created @ /mnt/mgmt/var/opt/cray/ssa/lock/ssa.lock_channel-
default_device-snx99999n000_SNX2000_99999
[stdout] Starting SNAPSHOT stage
[stdout] Added '/mnt/mgmt/var/opt/cray/ssa/collection/snx99999n000_SNX2000_99999/default/1515075962'
to snapshot source list
[stdout] Est 18130548 bytes needed to snapshot, based on raw storage of 1 collection(s)
[stdout] Snapshot encoding dir created at '/mnt/mgmt/var/opt/cray/ssa/snapshot/default/isodx/staq01/
Linux/out/snx99999n000_SNX2000_99999/1515075962'
[stdout] no snapshot directories meet purge requirements
[stdout] Stopping SNAPSHOT stage normally
[stdout] shepherd session stop successfully
[stdout] run took 6.75 seconds
```

The `SNAPSHOT` stage should complete normally. Report other status messages to Cray support.

3. Invoke an upload of the `default` run set and channel.

```
[root@snx99999n000]# ssacli --upload
[stdout] UI master_control status is (enabled)
[stdout] MASTER CONTROLS -> (M:True, C:True, S:True, U:True)
[stdout] UI CONTROLS -> (C:False, S:False, U:True)
[stdout] Shepherd Session: 1515076262
[stdout] Exclusive run: Lock file created @ /mnt/mgmt/var/opt/cray/ssa/lock/ssa.lock_channel-
default_device-snx99999n000_SNX2000_99999
[stdout] Starting UPLOAD stage
[stdout] Upload Organization: staq01
[stdout] Upload Server: 136.162.62.191
[stdout] Upload Device: snx99999n000_SNX2000_99999
[stdout] Stopping UPLOAD stage normally
[stdout] shepherd session stop successfully
[stdout] run took 14.16 seconds
```

The `UPLOAD` stage should complete successfully. If the stage does not complete successfully, ensure the connectivity requirement has been met. Then report the issue to Cray support.

If the process above completes successfully, the first snapshot of the system support information is uploaded to Cray.

> **NOTE:** Cray recommends that the `triage` step be completed on an initial install of SSA to make sure there are no issues when collecting a triage snapshot.

4. Optional: If this is an initial installation, repeat steps 1 through 3 for the `triage` run set and channel. Add command line options `--runset=triage` and `--channel=triage` to each of the `ssacli` command lines for `--collect`, `--snapshot`, and `--upload`.

6.3 Sonexion InfiniBand Fabric Plugins

The 1.6.5 Sonexion SSA release includes two plugins to collect and reset InfiniBand error counters. Error counters that are collected and reset are from all nodes that are connected to the Sonexion InfiniBand fabric and logged into the subnet manager (SM). The plugin to collect the error counters is executed as part of the SSA triage,default, and health run sets. The health run set will be executed every 15 minutes. As of the 1.6.0 release, the plugin to reset the InfiniBand error counters (`cluster.network.infiniband.faberrcnt.reset plugin`) is disabled by default.

Sites that have no other mechanism of clearing the Infiniband error counters should enable the `cluster.network.infiniband.faberrcnt.reset` plugin. Not clearing the Infiniband error counters on a regular basis will result in erroneous data being collected.

For the procedure to disable the `cluster.network.infiniband.faberrcnt.reset` plugin, please refer to Cray *SFDC Article 6454*, *SSA Sonexion: SSA Sonexion 1.6.0 InfiniBand fabric enable reset plugin*.

7 Configure LNet Fine-Grained Routing for XC Systems

Tasks for Configuring Fine-grained Routing

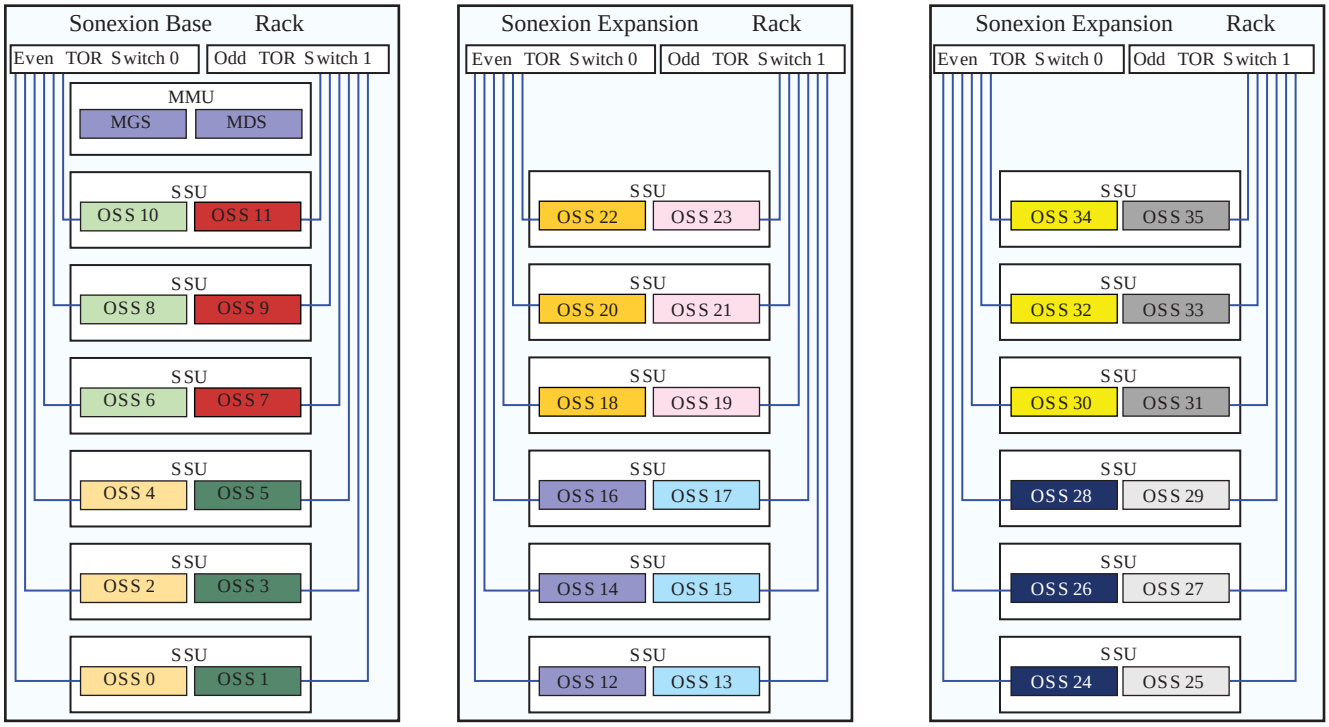
The following tasks must be completed to configure fine-grained routing (FGR) on Cray and Sonexion systems:

- Use bandwidth matching to get the router-to-server ratio
- Determine the IP addressing scheme to support the system configuration on the Cray and Sonexion systems
- Use CLCVT to generate the LNet configuration (`lnet.conf`, `routes.conf`, and `ip2nets.conf`)
- Configure IP addressing on IB interfaces (IPoIB)
- Place the LNet configuration the CLE system
- Place LNet configuration on the Sonexion
- Verify and test the configuration

Sonexion File Systems

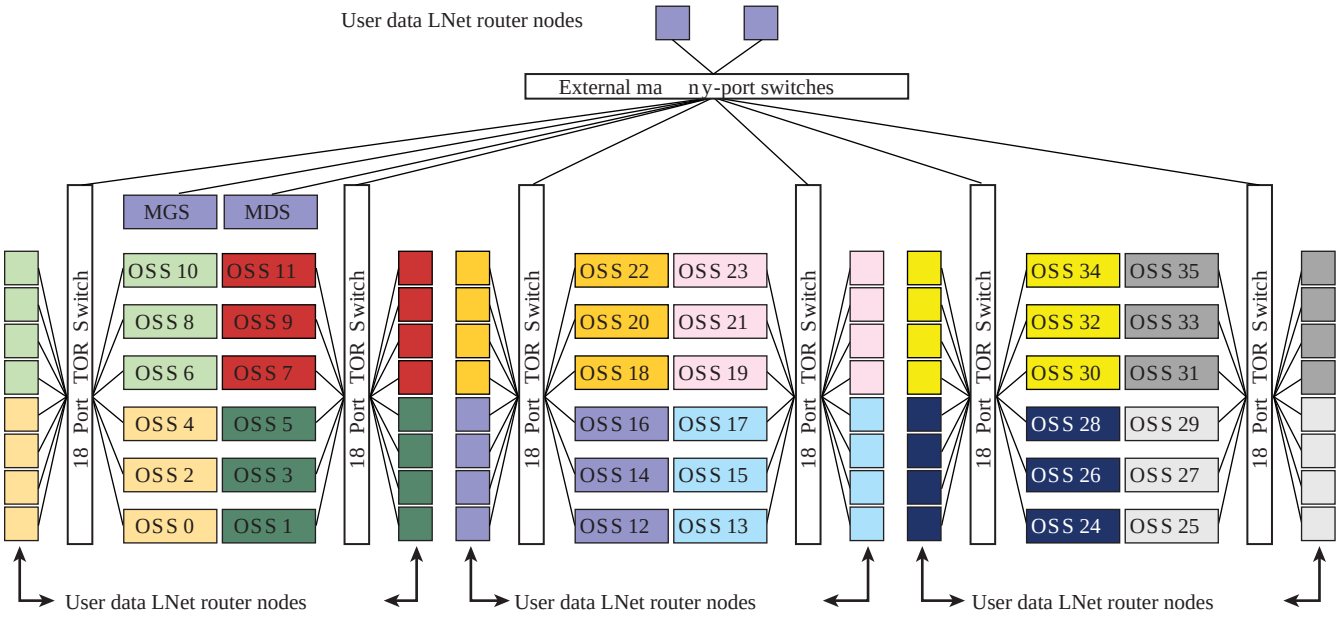
Sonexion file systems are made of two basic building blocks: scalable storage units (SSUs) and a metadata management unit (MMU). Each SSU houses disk drives and two OSSs that are connected to an even or odd top-of-rack IB switch and then to the system storage network. The MMU houses the MDS/MGS server pair and each file system includes an MDS/MGS server pair.

Figure 2. Sonexion File System Components



LNet FGR 3-Rack Example on page 26 shows how FGR is configured for three Sonexion racks. Each color represents an LNet group that is optimized to match the maximum bandwidth capacity for a Sonexion OSS. Four Cray LNet router nodes support three Sonexion OSSs in this example.

Figure 3. LNet FGR 3-Rack Example



InfiniBand Infrastructure

Cray uses industry standard fourteen data rate (FDR) InfiniBand infrastructure (although LNet supports other HSNs). The FDR InfiniBand network infrastructure (IBNI) is typically composed of dual InfiniBand core switches, and redundant pathways. There are two independent paths for all functionality accessed through the IBNI. This is accomplished with the use of active-active pairing for every function. In the event of an IBNI (or node) failure removing access to a node, the active partner(s) will automatically take over operation. Once the failure is resolved, the failed node is placed back into service. All components can be serviced, to include removal/replacement, in the event of a single failure without loss of access to the file system.

InfiniBand subnet routing must be implemented that does not result in credit loops, (deadlock avoidance) and that high-bandwidth connections take paths through the subnet to avoid bandwidth conflicts. Connections from the HPC systems are typically split, to allow maximum bandwidth in the event of a total switch failure.

Lustre LNet routers can connect two distinct InfiniBand fabrics. Any number of LNet routers may be used in parallel to achieve the desired performance level. The hardware to create such an LNet router can be as simple as a 1U rackmount server with two InfiniBand HCA interfaces and standard Lustre software.

XC Series Client Interface

The connection from the XC Series system to the Sonexion is accomplished with a set of LNet router nodes attached to the internal Aries® HSN. LNet routers typically have FDR InfiniBand connections to director class switches (evenly split between the two). For any given file system, the Lustre clients use all appropriate and available LNet routers in a round-robin fashion. The effect of this is to add together the performance of each useable LNet router for a total available bandwidth number. The LNet routers demonstrate nearly linear performance scaling as they act in parallel.

A single XC Series I/O blade supports two independent I/O nodes that each run the Cray Linux Environment (CLE). Each I/O node supports two PCIe Gen3 x8 slots, each can support single or dual-port IB HCAs, to provide two compute node LNet routers.

A single port reaches the maximum available PCIe bandwidth. Each HCA is capable of approximately 6GB/s of raw IB data transfer. In the LNet router configuration, this translates into approximately 5.5GB/s of Lustre data traffic. However—due to interference between the two HCAs and the limits of the HSN interface—when both HCAs are active with Lustre traffic, the total is 8.3GB/s (or 4.15GB/s for each of the two IB links).

> **IMPORTANT:** Due to routing restrictions imposed by both Lustre LNet and the Internet Protocol (IP), which provides the underlying node addressing, each of the two HCA ports must be assigned to a different IP subnet and LNet o2ib instance number.

In addition, the IP subnet must be different for the HSN interface. This allows the LNet router to distinguish the appropriate exit path for any packet coming into the router. The IBNI typically includes dual-core switches and is designed to maintain the highest possible performance during any failure. Therefore, each of the two HCAs on a LNet router node are connected to different core switches. When a switch fails, the normal 4.15GB/s available to each port can become 5.5GB/s on the port surviving the failure.

The address assignments for each file system must use unique subnets. Therefore, any given LNet router IB link services only the file system that matches the router-assigned subnet. Because the two links on a single LNet router node must have different subnets, it follows that each LNet router node services two different file systems.

Metadata Server Restrictions

Lustre metadata servers (MDS/MGS) require special consideration. The traffic for metadata is very latency sensitive, but does not use much bandwidth. If this traffic were to be interleaved with normal read/write data

traffic, it would be subject to the transmission delays imposed by large data transfers. These are redundant connections, and either of the two links for a single file system is sufficient to support operations. Each of the two links for a single file system is connected to a different core switch, allowing operations to proceed in the event of either a LNet router or a switch failure.

Distributing links across I/O blades accommodates system service procedures. If an LNet router node fails, the entire I/O blade may be serviced to resolve the problem. Placement of I/O blades purposed as LNet routers is distributed across different cabinets so that any failure affecting an entire cabinet will only take a single pair of LNet router nodes out of service.

7.1 Routing and Bandwidth Matching for Sonexion Systems

Cray recommends two IB links per router node FDR InfiniBand links on XC Series systems.

Sonexion bandwidth is evenly spread across all OSS nodes. Peak usable bandwidth is 2.4GB/s per OSS. External connections to the TOR switches must allow for this peak plus additional overhead, for a total of 3.3GB/s per OSS. A single InfiniBand FDR link can carry two OSSs worth of traffic (nominally 6.8GB/s). It is desirable to have an even number of links per TOR switch. The bandwidth of the HSN LNet nodes are matched to the bandwidth of the of the OSS nodes on the IB network for optimal throughput.

> **RESTRICTION:** When two IB interfaces are used for each LNet router node, each IB interface must be on separate IPv4 subnets.

Sonexion storage systems are composed of two basic building blocks—a single Metadata Management Unit (MMU) and one or more Scalable Storage Units (SSUs). An optional building block, the additional distributed namespace environment (DNE) unit (ADU), is also available. The MMU consists of two management servers (MGS) and two metadata servers (MDSs) and either a 2U24 or 5U84 drive enclosure. An SSU consists of two OSSs with a 5U84 drive enclosure.

An ADU consists of two MDSs with a 2U24 drive enclosure. The Sonexion 1600 MDRAID and GridRAID systems provide 5GB/s per SSU sustained, and 6GB/s per SSU peak. The Sonexion 2000 system provides 7.5GB/s per SSU sustained, and 9 GB/s per SSU peak. Ensuring sufficient network bandwidth to each OSS is a key requirement in the design of Lustre networks.

An XC Series I/O blade has one Aries Network Interface Controller (NIC) that provides an I/O module (IBB) with 17GB/s of I/O bandwidth. That IBB includes two nodes (LNet routers), so each node can achieve 8.5GB/s. Each LNet router node can support one or two, single- or dual-port IB host channel adapters (HCAs). Each active HCA port (`ib0` and `ib2`, for instance) must be assigned to a different LNet (cannot bond them to a single LNet). Therefore, a single LNet router node services two different LNets. A single FDR (Fourteen Data Rate) IB HCA is capable of 5.5GB/s of LNet traffic. Therefore, if busy, the two IB HCAs on a single LNet router split 8.5GB/s and achieve 4.25GB/s per IB port.

Because a single FDR IB link provides sufficient bandwidth for a single Sonexion 2000 OSS, the ratio of *n* IB links to *n* servers would work. However, in the case of the Sonexion 1600, this results in wasted bandwidth. *Bandwidth Capabilities of 6 Sonexion 2000 OSSs (3 SSUs) (22.5GB/s) 5 IB Links from Single HCA Routers (27.50GB/s)* on page 29 indicates that six Sonexion 1600 OSSs can deliver 18GB/s peak I/O bandwidth. If six single HCA LNet router links are assigned to service the six servers, then the fabric is capable of 33GB/s. This is nearly double the bandwidth that six OSSs can deliver. If IB links from LNet routers with dual HCAs are assigned, then the configuration would provide 23% more bandwidth than what is needed.

Assigning two single HCA LNet router links (or three IB links from dual HCA routers) to every four Sonexion-1600 servers is ideal from the perspective of minimizing wasted bandwidth. This would either result, however, in FGR groups that span more than one TOR switch or would require additional sub-optimal FGR groups that contain just two servers. If only FGR groups are considered (where the number of servers in each FGR group is evenly

divisible by the number of servers in each TOR switch) then this restricts the configurations to ratios with one, two, three or six servers.

With the above constraints in mind, suitable ratios are:

- Sonexion 1600
  - XC 40 Single HCA: 2:3
  - XC 40 Dual HCA: 5:6, or n:n+1
- Sonexion 2000
  - XC 40 Single HCA: n:n
  - XC 40 Dual HCA: n:n

Sonexion 2000 Bandwidth Matching

The bolded table cells in each table show the best grouping of Cray XC Series LNet nodes and Sonexion OSS nodes over single or dual InfiniBand HCAs. Always assign more bandwidth in LNet router nodes than is provided by the disk I/O to/from the OSSs.

- Based on the IOR benchmark with Cray chosen parameters and read/write averaging per SSU, the bandwidth is 6.0 GB/sec per SSU (3.0GB/s per OSS) for 2TB drives and requires an LNet ratio of 1:1
- Based on the IOR benchmark with Cray chosen parameters and read/write averaging per SSU, without an ESU, the bandwidth is 7.5GB/s per SSU (3.75GB/s per OSS) for 4TB and 6TB drives and requires an LNet ratio of 1:1
- Based on the IOR benchmark with Cray chosen parameters and read/write averaging per SSU, with an ESU, the bandwidth is 9.0 GB/sec per SSU (4.5GB/s per OSS) for 4TB and 6TB drives and requires an LNet ratio of 7:6

Table 2. Bandwidth Capabilities of 6 Sonexion 2000 OSSs (3 SSUs) (22.5GB/s) 5 IB Links from Single HCA Routers (27.50GB/s)

	1	2	3	4	5	6
Sonexion 1600 OSS	3.00	6.00	9.00	12.00	15.00	18.00
Sonexion 2000 OSS	3.75	7.50	11.25	15.00	18.75	***22.50***
Single HCA	5.50	11.00	16.50	22.00	***27.50***	33.00
Dual HCA	4.20	8.40	12.60	16.80	21.00	25.20

Table 3. 6 Sonexion 2000 OSSs (3 SSUs) (22.5GB/s) 6 IB Links from Dual-HCA Routers (25.20GB/s)

	1	2	3	4	5	6
Sonexion 1600 OSS	3.00	6.00	9.00	12.00	15.00	18.00
Sonexion 2000 OSS	3.75	7.50	11.25	15.00	18.75	***22.50***
Single HCA	5.50	11.00	16.50	22.00	27.50	33.00

	1	2	3	4	5	6
Dual HCA	4.20	8.40	12.60	16.80	21.00	***25.20***

Sonexion 3000 Bandwidth Matching

SSU Performance for IB with EDR LDN - Based on the IOR benchmark with Cray parameters and read/write averaging:

- 9GB/s per SSU for all 7200 RPM drive sizes (4.5GB/s per OSS)
- 10GB/s per SSU with an ESU, for all drive types (5GB/s per OSS)
- 12GB/s per SSU with 10K RPM HPC drives, (6GB/s per OSS)

The following table shows the number of physical connections (i.e. cables) required for connecting the TOR switches in each rack to a XC system.

Table 4. Number of Physical Connections Required for Sonexion 3000 LNet Routing

Number of SSUs	Single FDR HCA LNet Connections	Dual FDR HCA LNet Connections	Single EDR HCA Connections
7.2 RPM Drives			
1	1	2	1
2	2	3	2
3	3	4	2
4	4	5	3
5	5	6	3
6	6	7	4
7	7	8	4
10K HPC Drives			
1	2	2	1
2	3	3	2
3	4	4	2
4	5	6	3
5	6	7	4
6	7	8	4
7	8	10	5

7.2 External Server Node: Sonexion 3000 or ClusterStor L300 System Recommended Parameters

For Sonexion 3000 or ClusterStor L300 systems, the `peer_credits` setting must be consistent across all InfiniBand (IB) peers on the Lustre network. When routers and/or external Lustre clients have mlx5-based Host Channel Adapters (HCAs), `map_on_demand` must be set to 0. In addition, Cray recommends `peer_credits` and `concurrent_sends` be set to 16. Thus, when connecting to a Sonexion 3000/ClusterStor L300, the recommended parameters are generally the same as those for the Sonexion 900, 1600, and 2000—except that `map_on_demand` must be set to 0, and `peer_credits` and `concurrent_sends` should be set to 16 for all IB peers on the Lustre network.

If an IB peer must have access to an mlx4-based file system (i.e. Sonexion 900, Sonexion 1600, and Sonexion 2000) and an mlx5-based file system (i.e. Sonexion 3000), the `ko2iblnd` parameters of all mlx4 peers must match the `ko2iblnd` mlx5-peer parameters to ensure shared mlx4- and mlx5-peer function. For example, in a system where an external Login node needs access to a Sonexion 2000 and Sonexion 3000, all mlx4- and mlx5-peer `ko2iblnd` parameters should match the LNet parameters recommended for a Sonexion 3000.

For systems that have a Sonexion 3000/ClusterStor L300 running at running software version 2.1-SU003 or greater, `peer_credits` can be increased to 84 and `concurrent_sends` can be increased to 42. All IB peers within the Lustre network must be able to support these same values if they are to be used.

Cray recommends an object-based disk (OBD) timeout of 100 seconds, which is the default value. Set this parameter using the `lctl conf_param` command on the management server (MGS). For example:

```
$ lctl conf_param fs_name.sys.timeout=100
$ cat /proc/sys/lustre/timeout
100
```

`ko2iblnd:timeout`

Default **10**. The `o2iblnd` timeout in seconds. Cray recommends setting this to 10 seconds.

`ko2iblnd:peer_timeout`

Default **0**. Number of seconds without aliveness news it takes to declare a peer dead. Cray recommends setting this to 0.

`ko2iblnd:keepalive`

Default **30**. Idle time in seconds before sending a keepalive. Cray recommends setting this to 30.

`ko2iblnd:credits`

Default **2048**. Number of concurrent sends allowed by `o2iblnd`. Shared by all CPU partitions (CPT). Cray recommends setting this to 2048.

`ko2iblnd:ntx`

Default **2048**. Number of message descriptors allocated for each pool. Cray recommends setting this to 2048.

`ko2iblnd:peer_credits` 16

Enter the value for the `ko2iblnd` parameter `peer_credits`. This is the number of concurrent sends to a single peer. This value must be the same on all external login clients and the Lustre file system servers.

`ko2iblnd:concurrent_sends` 16

Determines send work queue sizing. If this option is omitted, the default is calculated based on the values of `peer_credits` and `map_on_demand`. This value must be the same on the external login clients and the Lustre file system servers.

`ko2iblnd:map_on_demand`

Default **0**. Controls the use of fast memory registration (FMR). Cray recommends setting this value to 0 for InfiniBand HCAs or a value of 32 for Intel® Omni-Path (OPA) host fabric interfaces (HFI).

lnet:router_ping_timeout

Default **50**. Number of seconds to wait for the reply to a router health query. Cray recommends using the default value of 50 seconds.

lnet:live_router_check_interval

Default **60**. Number of seconds between live router health checks. Cray recommends leaving this at the default value of 60 seconds. A value less than or equal to 0 disables pinging of live routes.

lnet:dead_router_check_interval

Default **60**. Number of seconds between dead router health checks. Cray recommends using the default value of 60 seconds. A value less than or equal to 0 disables pinging of dead routes.

lnet:avoid_asym_router_failure 1

Avoid asymmetrical router failures (0 to disable; 1 to enable).

7.3 Use CLCVT to Configure Fine-Grained Routing Files

The `clcvt` command, available on the boot node and the system management workstation (SMW), aids in the configuration of Lustre networking (LNet) fine-grained routing (FGR).

The `clcvt` command requires several file-system-specific input files and generates LNet kernel module configuration information that can be used to configure the servers, routers, and clients for that file system. The utility can also create cable maps (in HTML, CSV, and human-readable formats) and validate cable connection on Sonexion systems.

See the `clcvt(8)` `man` page for detailed information.

7.3.1 CLCVT Prerequisite Files

The `clcvt` command requires several prerequisite files in order to compute the `ip2nets` and `routes` information for the specific configuration. The prerequisite files must be placed in an empty directory on the boot node or SMW, depending on where `clcvt` will be run.

Deciding how to assign which routers to which object storage servers (OSSs), what fine grained routing (FGR) ratios to use, which interface on which router to use for a Lustre networking (LNet) group, and router placement are all things that can vary greatly from site to site. LNet configuration is determined as the system is ordered and configured. See a Cray representative for the site-specific values. Use *Routing and Bandwidth Matching for Sonexion Systems* on page 28 as a guide.

info.*file-system-identifier*	A file with global file system information for the *`cluster-name`* server machine and each client system that will access it.
***client-system*.hosts**	A file that maps the client system (such as the Cray mainframe) IP addresses to unique host names, such as the boot node `/etc/hosts` file. The *`client-system`* name must match one of the clients in the `info.`*`file-system-identifier`* file.

***client-system*.ib**	A file that maps the client system LNet router InfiniBand IP addresses to system hardware `cnames`. The *`client-system`* name must match one of the `clients` in the `info.`*`file-system-identifier`* file. This file must be created by an administrator.
***clustername*.ib**	A file that maps the Lustre server InfiniBand IP addresses to cluster (for example, Sonexion) host names. The *`clustername`* name must match the `clustername` in the `info.`*`file-system-identifier`* file. This file must be created by an administrator.
***client-system*.rtrIm**	A file that contains `rtr -Im` command output (executed on the SMW) for the *`client-system`*.

7.3.2 Create the CLCVT info.file-system-identifier File

Create the `info.`*`file-system-identifier`* file manually. This file contains global file system information for the Lustre server machine and each client system that must have access. Based on the ratio of OSS to LNet routers in the configuration, the [*`clustername`*] section (`snx11029n` in this example) and each [*`client-system`*] section (`hera` in this example) defines which servers and routers belong to each InfiniBand (IB) subnet.

This file is in the form of a `ini` style file, and the possible keywords in the `[info]` section include `clustername`, `ssu_count`, and `clients`. Refer to the `clcvt man` page for detailed information and usage.

clustername	Defines the base name used for all file system servers. The example show a Sonexion file system `snx11029n`. Thus, all server hostnames will be `snx11029n`*`NNN`*. *`NNN`* is a three-digit number starting at `000` and `001` for the primary and secondary Cray Sonexion management servers (MGMT), `002` for the MGS, `003` for the MDS, `004` for the first OSS, and counting up from there for all remaining OSSs.
ssu_count	Defines how many SSUs make up a Sonexion file system. If this is missing, then this is not a Sonexion file system but a CLFS installation.
clients	Defines a comma-separated list of mainframe names that front-end this file system (`hera` in this example).

fgr_ratio	Determines the Fine-Grained Routing ratio (see *Routing and Bandwidth Matching for Sonexion Systems* on page 28. Set this to *`M`*`:`*`N`*, where *`M`* is the number of routers per group and *`N`* is the number of servers per group. If set, `oss_per_switch` and `ssu_per_rack` must be equal to or greater than *`N`*, and they must be set to a whole-number multiple of *`N`*. Changing the `ssu_per_rack` and `oss_per_switch` default settings can accommodate seven SSUs per rack.
	
	`ssu_per_rack = 7`
	`oss_per_switch = 7`

The `info.`*`file-system-identifier`* file requires a [*`client-system`*] section for each client system listed in the `clients` line of the `[info]` section to describe the client systems and a [*`clustername`*] section to describe the Lustre server system.

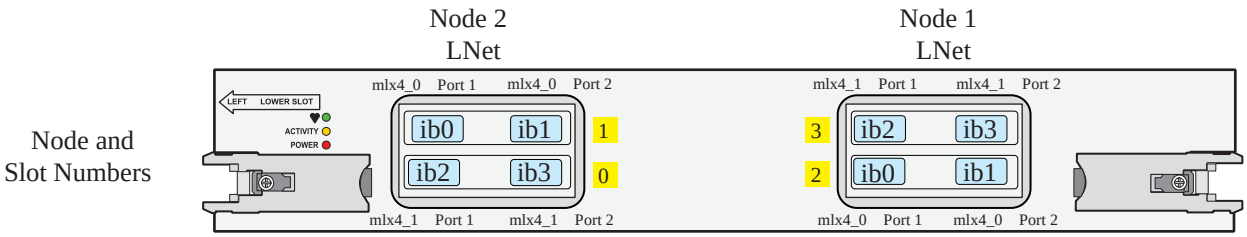
Each of these sections contain a literal LNet network wildcard in the format of *`LNET-name`*:*`IP-wildcard`* which instructs the LNet module to match a host IP address to *`IP-wildcard`* and, if it matches, instantiate LNet *`LNET-name`* on them.

The hostname fields in the [*`client-system`*] section of this file are fully qualified interface specifications of the form *`hostname`*`(ib`*`n`*`)`, where `(ib0)` is the assumed default if not specified.

XC systems support multiple IB interfaces per LNet router. Configure the second IB interface and append the interface names (`ibn`) to the `cname` for the routers (as shown in the example). These interface names must also be appended to the *`client-system`*`.ib` file.

XC40 IB port assignments are shown in the illustration.

Figure 4. XC40 IB Port Assignments



Example: `info.`*`file-system-identifier`* File

This example configures an `info.snx11029n` file that supports the example file system in *XC40 with Two FDR IB Interfaces Per LNet Router* on page 35. It is configured with four FDR IB interfaces (two per router node) from each XC40 LNet router blade and includes a total of 12 SSUs for the file system.

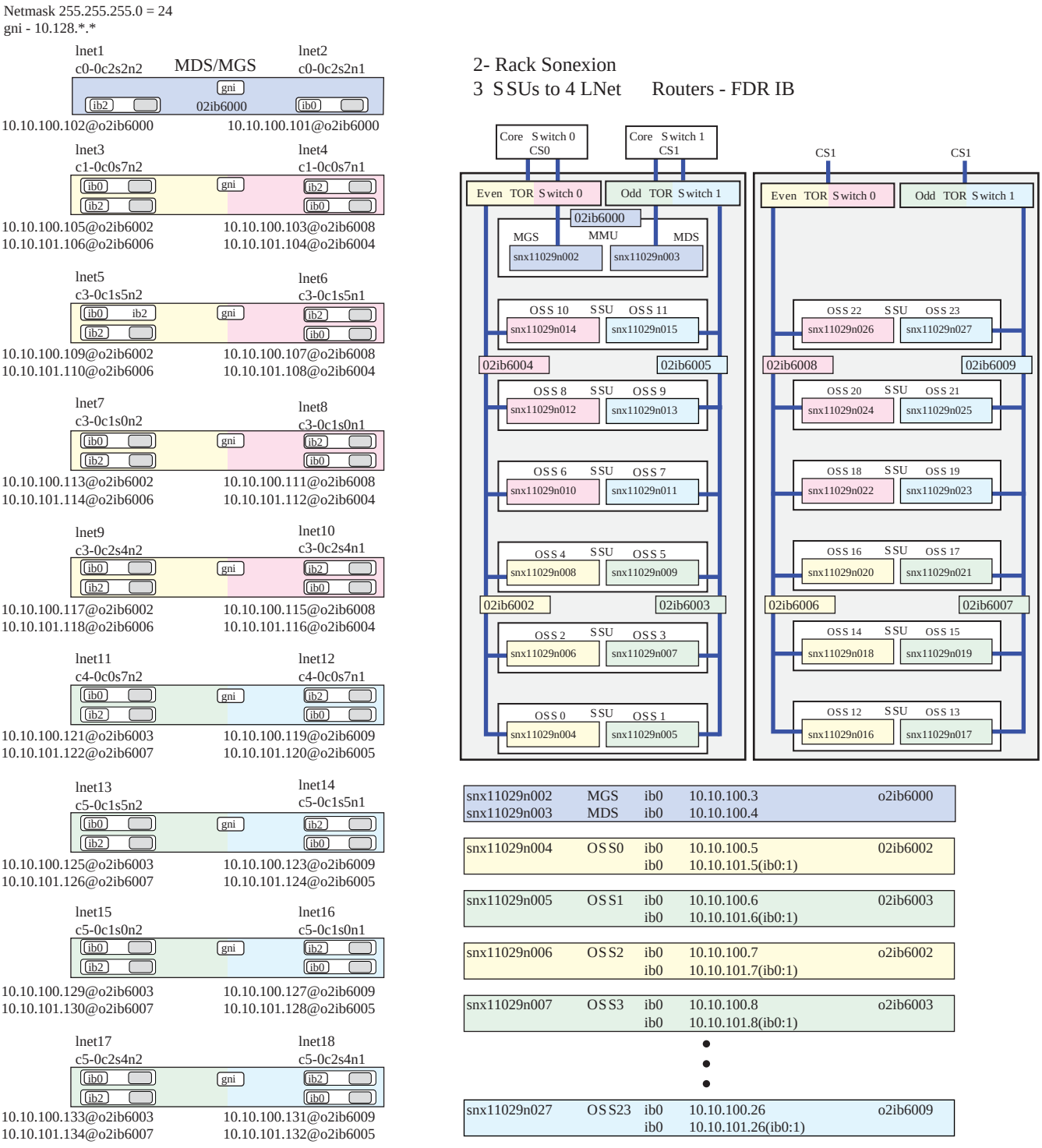
The Sonexion OSSs that connect to the ib2 interface of the LNet routers in this example must have an IP address on the ib2 subnet. The ib0:1 alias is added as label for the ib2 interfaces because there is only single IB interface on each Sonexion OSS.

```
# This section describes the size of this filesystem.
[info]
clustername = snx11029n
SSU_count = 12
clients = hera
fgr_ratio = 4:3
# other optional keywords not used in this example
# oss_count
# oss_per_switch
# sonexion
# ssu_per_rack
# fgr

[hera]
lnet_network_wildcard = gni1:10.128.*.*

# Because of our cabling assumptions and naming conventions, we only
# need to know which XIO nodes are assigned to which LNETs.  From that
# our tool can actually generate a "cable map" for the installation folks.
o2ib6000: c0-0c2s2n2(ib2), c0-0c2s2n1(ib0) ; MGS and MDS
o2ib6002: c1-0c0s7n2(ib0), c3-0c1s5n2(ib0), c3-0c1s0n2(ib0), c3-0c2s4n2(ib0) ; OSSs 0/2/4
o2ib6003: c4-0c0s7n2(ib0), c5-0c1s5n2(ib0), c5-0c1s0n2(ib0), c5-0c2s4n2(ib0) ; OSSs 1/3/5
o2ib6004: c1-0c0s7n1(ib0), c3-0c1s5n1(ib0), c3-0c1s0n1(ib0), c3-0c2s4n1(ib0) ; OSSs 6/8/10
o2ib6005: c4-0c0s7n1(ib0), c5-0c1s5n1(ib0), c5-0c1s0n1(ib0), c5-0c2s4n1(ib0) ; OSSs 7/9/11
o2ib6006: c1-0c0s7n2(ib2), c3-0c1s5n2(ib2), c3-0c1s0n2(ib2), c3-0c2s4n2(ib2) ; OSSs 12/14/16
o2ib6007: c4-0c0s7n2(ib2), c5-0c1s5n2(ib2), c5-0c1s0n2(ib2), c5-0c2s4n2(ib2) ; OSSs 13/15/17
o2ib6008: c1-0c0s7n1(ib2), c3-0c1s5n1(ib2), c3-0c1s0n1(ib2), c3-0c2s4n1(ib2) ; OSSs 18/20/22
o2ib6009: c4-0c0s7n1(ib2), c5-0c1s5n1(ib2), c5-0c1s0n1(ib2), c5-0c2s4n1(ib2) ; OSSs 19/21/23

[snx11029n]
lnet_network_wildcard = o2ib0:10.10.100.*

o2ib6000: snx11029n002(ib0), snx11029n003(ib0) ; MGS and MDS
o2ib6002: snx11029n004(ib0), snx11029n006(ib0), snx11029n008(ib0) ; OSSs 0/2/4
o2ib6003: snx11029n005(ib0), snx11029n007(ib0), snx11029n009(ib0) ; OSSs 1/3/5
o2ib6004: snx11029n010(ib0), snx11029n012(ib0), snx11029n014(ib0) ; OSSs 6/8/10
o2ib6005: snx11029n011(ib0), snx11029n013(ib0), snx11029n015(ib0) ; OSSs 7/9/11
o2ib6006: snx11029n016(ib0:1), snx11029n018(ib0:1), snx11029n020(ib0:1) ; OSS 12/14/16
o2ib6007: snx11029n017(ib0:1), snx11029n019(ib0:1), snx11029n021(ib0:1) ; OSS 13/15/17
o2ib6008: snx11029n022(ib0:1), snx11029n024(ib0:1), snx11029n026(ib0:1) ; OSS 18/20/22
o2ib6009: snx11029n023(ib0:1), snx11029n025(ib0:1), snx11029n027(ib0:1) ; OSS 19/21/23
```

The example shows two interfaces for the MDS/MGS nodes on a single blade. If the metadata server must be on a 3rd IPv4 subnet, then the additional interfaces would be placed on other LNet nodes.

Move the `info.snx11029n` file to the working directory where the `clcvt` command will be run.

Figure 5. XC40 with Two FDR IB Interfaces Per LNet Router



7.3.3 Create the CLCVT client-system.hosts File

Use the `/etc/hosts` file from the XC boot node to create the *client-system*`.hosts` file (`hera.hosts`). Copy the `/etc/hosts` file from the boot node to a working directory and include the LNet router nodes. A typical `/etc/hosts` file is shown below.

Example *client-system*`.hosts` File

```
#
# hosts         This file describes a number of hostname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.
# Syntax:
#
# IP-Address  Full-Qualified-Hostname  Short-Hostname
#

127.0.0.1       localhost

# special IPv6 addresses
::1     ipv6-localhost  localhost       ipv6-loopback

fe00::0 ipv6-localnet

ff00::0 ipv6-mcastprefix
ff02::1 ipv6-allnodes
ff02::2 ipv6-allrouters
ff02::3 ipv6-allhosts

10.128.0.1      nid00000        c0-0c0s0n0      dvs-0
10.128.0.2      nid00001        c0-0c0s0n1      boot001 boot002
10.128.0.31     nid00030        c0-0c0s0n2      #ddn_mds
10.128.0.32     nid00031        c0-0c0s0n3      hera-rsip2
10.128.0.3      nid00002        c0-0c2s2n1      lnet2
10.128.0.4      nid00003        c0-0c2s2n2      lnet1
10.128.0.29     nid00018        c1-0c0s7n1      lnet4
10.128.0.30     nid00019        c1-0c0s7n2      lnet3
10.128.0.45     nid00024        c3-0c1s0n1      lnet8
10.128.0.46     nid00025        c3-0c1s0n2      lnet7
10.128.0.57     nid00036        c3-0c1s5n1      lnet6
10.128.0.58     nid00037        c3-0c1s5n2      lnet5
10.128.0.67     nid00046        c3-0c2s4n1      lnet10
10.128.0.68     nid00047        c3-0c2s4n2      lnet9
10.128.0.75     nid00054        c4-0c0s7n1      lnet12
10.128.0.76     nid00055        c4-0c0s7n2      lnet11
10.128.0.85     nid00067        c5-0c1s5n1      lnet14
10.128.0.86     nid00068        c5-0c1s5n2      lnet13
10.128.0.91     nid00074        c5-0c1s0n1      lnet16
10.128.0.92     nid00075        c5-0c1s0n2      lnet15
10.128.0.101    nid00093        c5-0c2s4n1      lnet18
10.128.0.102    nid00094        c5-0c2s4n2      lnet17
. . .
```

Move the *client-system*`.hosts` file to the working directory on the SMW or boot node where the `clcvt` command will be run.

7.3.4 Create the CLCVT client-system.ib File

The *client-system*`.ib` file contains a client-system LNet router InfiniBand (IB) IP address to `cname` mapping information in an `/etc/hosts` style format. The hostname field in this file is a fully-qualified interface specification of the form *hostname*`(ib`*n*`)`, where `(ib0)` is the assumed default if not specified. This file must be created by an administrator.

XC systems can support multiple IB interfaces per router—configure the second IB interface and append the interface names (`ib`*n*) to the `cname` for the routers. The LNet router IB IP addresses should be within the same subnet as the Lustre servers (MGS/MDS/OSS). One possible address assignment scheme would be to use a contiguous set of IP addresses, with `ib0` and `ib2` on each node having adjacent addresses. These interface names must be appended to the `info.`*`file-system-identifier`* file.

Example: *`client-system`*`.ib` File

This example (`hera.ib`) configures two FDR IB cards per XC system LNet router.

```
#
# This is the /etc/hosts-like file for Infiniband IP addresses
# on "hera".
#
10.10.100.101    c0-0c2s2n1(ib0)
10.10.100.102    c0-0c2s2n2(ib2)
10.10.100.103    c1-0c0s7n1(ib2)
10.10.101.104    c1-0c0s7n1(ib0)
10.10.100.105    c1-0c0s7n2(ib0)
10.10.101.106    c1-0c0s7n2(ib2)
10.10.100.107    c3-0c1s5n1(ib2)
10.10.101.108    c3-0c1s5n1(ib0)
10.10.100.109    c3-0c1s5n2(ib0)
10.10.101.110    c3-0c1s5n2(ib2)
10.10.100.111    c3-0c1s0n1(ib2)
10.10.101.112    c3-0c1s0n1(ib0)
10.10.100.113    c3-0c1s0n2(ib0)
10.10.101.114    c3-0c1s0n2(ib2)
10.10.100.115    c3-0c2s4n1(ib2)
10.10.101.116    c3-0c2s4n1(ib0)
10.10.100.117    c3-0c2s4n2(ib0)
10.10.101.118    c3-0c2s4n2(ib2)
10.10.100.119    c4-0c0s7n1(ib2)
10.10.101.120    c4-0c0s7n1(ib0)
10.10.100.121    c4-0c0s7n2(ib0)
10.10.101.122    c4-0c0s7n2(ib2)
10.10.100.123    c5-0c1s5n1(ib2)
10.10.101.124    c5-0c1s5n1(ib0)
10.10.100.125    c5-0c1s5n2(ib0)
10.10.101.126    c5-0c1s5n2(ib2)
10.10.100.127    c5-0c1s0n1(ib2)
10.10.101.128    c5-0c1s0n1(ib0)
10.10.100.129    c5-0c1s0n2(ib0)
10.10.101.130    c5-0c1s0n2(ib2)
10.10.100.131    c5-0c2s4n1(ib2)
10.10.101.132    c5-0c2s4n1(ib0)
10.10.100.133    c5-0c2s4n2(ib0)
10.10.101.134    c5-0c2s4n2(ib2)
```

Move the *`client-system`*`.ib` file to the working directory on the SMW or boot node where the `clcvt` command will be run.

7.3.5 Create the CLCVT cluster-name.ib File

The CLCVT *`cluster-name`*`.ib` file contains Lustre server InfiniBand (IB) IP addresses to cluster (Sonexion) host name mapping information in a `/etc/hosts` style format. This file must be created by an administrator.

The Sonexion servers that connect to the ib2 interface of the routers in this example must have an IP address on the ib2 subnet. The ib0:1 alias is added as a label for the ib2 interfaces to support the single IB interface on each Sonexion OSS.

Example *`cluster-name`*`.ib` File*`snx11029n`*`.ib`

```
#
# This is the /etc/hosts-like file for InfiniBand IP addresses
```

```
# on the Sonexion known as "snx11029n".
#
10.10.100.1    snx11029n000    #mgmnt
10.10.100.2    snx11029n001    #mgmnt
10.10.100.3    snx11029n002    #mgs
10.10.100.4    snx11029n003    #mds
10.10.100.5    snx11029n004    #first oss, oss0
10.10.100.6    snx11029n005    #oss1
10.10.100.7    snx11029n006    #oss2
10.10.100.8    snx11029n007    #oss3
10.10.100.9    snx11029n008    #oss4
10.10.100.10   snx11029n009    #oss5
10.10.100.11   snx11029n010    #oss6
10.10.100.12   snx11029n011    #oss7
10.10.100.13   snx11029n012    #oss8
10.10.100.14   snx11029n013    #oss9
10.10.100.15   snx11029n014    #oss10
10.10.100.16   snx11029n015    #oss11
10.10.101.17   snx11029n016(ib0:1)    #oss12
10.10.101.18   snx11029n017(ib0:1)    #oss13
10.10.101.19   snx11029n018(ib0:1)    #oss14
10.10.101.20   snx11029n019(ib0:1)    #oss15
10.10.101.21   snx11029n020(ib0:1)    #oss16
10.10.101.22   snx11029n021(ib0:1)    #oss17
10.10.101.23   snx11029n022(ib0:1)    #oss18
10.10.101.24   snx11029n023(ib0:1)    #oss19
10.10.101.25   snx11029n024(ib0:1)    #oss20
10.10.101.26   snx11029n025(ib0:1)    #oss21
10.10.101.27   snx11029n026(ib0:1)    #oss22
10.10.101.28   snx11029n027(ib0:1)    #oss23
```

Move the *cluster-name*.ib file to the working directory on the SMW or boot node where the `clcvt` command will be run.

7.3.6 Create the CLCVT client-system.rtrIm File

About this task

The *client-system*.rtrIm file contains output from the `rtr -Im` command as executed from the SMW. When capturing the command output to a file, use the `-H` option to remove the header information from `rtr -Im` or open the file after capturing and delete the first two lines.

Procedure

1. Log on to the SMW as `crayadm`.
2. Run the `rtr -Im` command and capture the output (without header information) to a file.

```
crayadm@smw> rtr -Im -H > client-system.rtrIm
```

Example *client-system*.rtrIm File

```
  4        4       c0-0c2s2n1    c0-0c2s2g0     0  0  0
  5        5       c0-0c2s2n2    c0-0c2s2g0     0  0  0
221      221       c1-0c0s7n1    c1-0c0s7g0     0  3  7
222      222       c1-0c0s7n2    c1-0c0s7g0     0  3  7
641      769       c3-0c1s0n1    c3-0c1s0g0     1  4  0
642      770       c3-0c1s0n2    c3-0c1s0g0     1  4  0
661      789       c3-0c1s5n1    c3-0c1s5g0     1  4  5
662      790       c3-0c1s5n2    c3-0c1s5g0     1  4  5
721      849       c3-0c2s4n1    c3-0c2s4g0     1  5  4
722      850       c3-0c2s4n2    c3-0c2s4g0     1  5  4
797      1053      c4-0c0s7n1    c4-0c0s7go     2  0  7
798      1054      c4-0c0s7n2    c4-0c0s7go     2  0  7
1025     1281      c5-0c1s0n1    c5-0c1s0g0     2  4  0
```

```
1046     1302      c5-0c1s0n2    c5-0c1s0g0     2  4  0
1045     1301      c5-0c1s5n1    c5-0c1s5g0     2  4  5
1046     1302      c5-0c1s5n2    c5-0c1s5g0     2  4  5
1077     1365      c5-0c2s4n1    c5-0c2s4g0     2  4  13
1078     1366      c5-0c2s4n2    c5-0c2s4g0     2  4  13
```

3. Move the *client-system*.rtrIm (hera.rtrIm) file to the working directory where the clcvt command will be run.

7.3.7 Generate ip2nets and routes Information

After the prerequisite files have been created, generate the persistent-storage file with the clcvt generate action. This portable file will then be used to create ip2nets and routes directives for the servers, routers, and clients.

The following procedures frequently use the --split-routes=4 flag, which prints information that can be loaded into ip2nets and routes files. This method of adding modprobe.conf directives is particularly valuable for large systems where the directives might otherwise exceed the modprobe buffer limit.

- *Create the CLCVT persistent-storage File* on page 39
- *Create ip2nets and Routes Information for Compute Nodes* on page 39
- *Create ip2nets and routes Information for Service Node Lustre Clients (MOM and Internal Login Nodes)* on page 40
- *Create ip2nets and routes Information for LNet Router Nodes* on page 41
- *Create ip2nets and routes Information for Lustre Server Nodes* on page 42

7.3.7.1 Create the CLCVT persistent-storage File

Procedure

1. Move all prerequisite files to an empty directory on the boot node or SMW (the clcvt command is only available on the boot node or the SMW).

   The working directory should include the following files.

```
crayadm@smw$ ll
total 240
-rw-rw-r-- 1 crayadm crayadm  23707 Oct  8 14:27 hera.hosts
-rw-rw-r-- 1 crayadm crayadm    548 Oct  8 14:27 hera.ib
-rw-rw-r-- 1 crayadm crayadm  36960 Oct  8 14:27 hera.rtrIm
-rw-rw-r-- 1 crayadm crayadm   1077 Feb  8 14:27 info.snx11029n
-rw-rw-r-- 1 crayadm crayadm    662 Feb  8 14:27 snx11029n.ib
```

2. Create the persistent-storage file. (Use the --debug flag to display debugging information.)

```
crayadm@smw$ clcvt generate
INFO:LNET_filesystem.load: clustername = snx11029n
INFO:LNET_filesystem.load: This is a SONEXION
INFO:LNET_filesystem.load: 12 SSU, 6 SSU per rack
INFO:LNET_filesystem.load: You have chosen to use Fine Grained Routing.
INFO:LNET_filesystem.load: The ratio of LNET router connections to OSSs is '4:3'.
```

7.3.7.2 Create ip2nets and Routes Information for Compute Nodes

Procedure

1. Execute the `clcvt` command with the `compute` flag to generate directives for the compute nodes.

```
crayadm@smw$ clcvt compute --split-routes=4
# Place the following line(s) in the appropriate 'modprobe' file.
#vvvv
```

```
o2ib6008 1 [221,641,661,721]@gni1
o2ib6009 1 [797,1025,1045,1077]@gni1
o2ib6000 2 [221,222,641,642,661,662,721,722,797,798,1025,1026,1045,1046,1077,1078]@gni1
o2ib6002 2 [221,641,661,721]@gni1
o2ib6003 2 [797,1025,1045,1077]@gni1
o2ib6004 2 [222,642,662,722]@gni1
o2ib6005 2 [798,1026,1046,1078]@gni1
o2ib6006 2 [221,641,661,721]@gni1
o2ib6007 2 [797,1025,1045,1077]@gni1
o2ib6008 2 [222,642,662,722]@gni1
o2ib6009 2 [798,1026,1046,1078]@gni1
#^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

2. Place the command output in the appropriate `modprobe` configuration file.

7.3.7.4 Create ip2nets and routes Information for LNet Router Nodes

Procedure

1. Execute the `clcvt` command with the `router` flag to generate directives for the LNet router nodes.

```
crayadm@smw$  clcvt router --split-routes=4
# Place the following line(s) in the appropriate 'modprobe' file.
#vvvv
```

```
o2ib6009 2 [798,1026,1046,1078]@gni1
#^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

2. Place the command output in the appropriate `modprobe` configuration file.

7.3.7.5 Create ip2nets and routes Information for Lustre Server Nodes

Procedure

1. Execute the `clcvt` command with the `server` flag to generate directives for the Lustre server nodes.

```
crayadm@smw$ clcvt server --split-routes=4
# Place the following line(s) in the appropriate 'modprobe' file.
#vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
options lnet ip2nets=/path/to/ip2nets-loading/filename
options lnet routes=/path/to/route-loading/filename
#^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
# Place the following line(s) in the appropriate ip2nets-loading file.
#vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
o2ib6 10.10.*.*
o2ib6000 10.10.100.[3,4]
o2ib6002 10.10.100.[5,7,9]
o2ib6003 10.10.100.[6,8,10]
o2ib6004 10.10.100.[11,13,15]
o2ib6005 10.10.100.[12,14,16]
o2ib6006(ib0:1) 10.10.101.[17,19,21]
o2ib6007(ib0:1) 10.10.101.[18,20,22]
o2ib6008(ib0:1) 10.10.101.[23,25,27]
o2ib6009(ib0:1) 10.10.101.[24,26,28]
#^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
# Place the following line(s) in the appropriate route-loading file.
#vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
gni1 1 10.10.100.[101,102]@o2ib6000
gni1 1 10.10.100.[105,109,113,117]@o2ib6002
gni1 1 10.10.100.[121,125,129,133]@o2ib6003
gni1 1 10.10.101.[104,108,112,116]@o2ib6004
gni1 1 10.10.101.[120,124,128,132]@o2ib6005
gni1 1 10.10.101.[106,110,114,118]@o2ib6006
gni1 1 10.10.101.[122,126,130,134]@o2ib6007
gni1 1 10.10.100.[103,107,111,115]@o2ib6008
gni1 1 10.10.100.[119,123,127,131]@o2ib6009
gni1 2 10.10.100.[105,109,113,117,121,125,129,133]@o2ib6000
gni1 2 10.10.101.[104,108,112,116,120,124,128,132]@o2ib6000
gni1 2 10.10.101.[104,108,112,116]@o2ib6002
gni1 2 10.10.101.[120,124,128,132]@o2ib6003
gni1 2 10.10.100.[105,109,113,117]@o2ib6004
gni1 2 10.10.100.[121,125,129,133]@o2ib6005
gni1 2 10.10.100.[103,107,111,115]@o2ib6006
gni1 2 10.10.100.[119,123,127,131]@o2ib6007
gni1 2 10.10.101.[106,110,114,118]@o2ib6008
gni1 2 10.10.101.[122,126,130,134]@o2ib6009
#^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

2. Place the command output in the appropriate `modprobe` configuration file for the Lustre servers. (For more information, refer to the site-specific Lustre server documentation.)

7.4 Place the LNet Configuration on a CLE6 System

7.4.1 Copy Files for External Lustre Fine-grained Routing

Prerequisites

This procedure is only for systems that use an external Lustre file system. It assumes the following:

- Fine-grained routing (FGR) files have been generated by `clcvt`
- Cray LNet configuration service (`cray_lnet`) has been configured with FGR

About this task

This procedure places the `ip2nets.conf` and `routes.conf` files in the CLE config set for the LNet routers.

Procedure

1. Create an `lnet` directory under `roles` in the CLE config set directory structure.

   This example uses a config set named *p0*. Substitute the correct config set name for this site.

   ```
   smw# mkdir -p /var/opt/cray/imps/config/sets/p0/files/roles/lnet
   ```

2. Confirm the file names of the fine-grained routing files.

   It is possible that these two files were created with names other than `ip2nets.conf` and `routes.conf`. Check these two settings in the `cray_lnet` configuration service to see what file names are used (example settings are for config set *p0* and a file system with key *sonexion*).

   ```
   smw# cfgset search -l advanced -s cray_lnet -t fgr_routes p0
   # 2 matches for 'fgr_routes' from cray_lnet_config.yaml
   #-------------------------------------------------------
   cray_lnet.settings.fgr_routes.data.sonexion.ip2nets_file: ip2nets.conf
   cray_lnet.settings.fgr_routes.data.sonexion.routes_file: routes.conf
   ```

3. Copy the `ip2nets.conf` and `routes.conf` files to the `lnet` directory.

   ```
   smw# cd directory_containing_ip2nets.conf_and_routes.conf

   smw# cp -p ip2nets.conf routes.conf /var/opt/cray/imps/config/sets/p0/files/roles/lnet
   ```

7.4.2 Configure LNet Routers for Fine-Grained Routing

Prerequisites

Fine-grained routing files have been configured using `clcvt`.

About this task

This procedure modifies the *Cray System Management Configuration Worksheet* for the `cray_lnet` service in the config set to configure fine-grained routing (FGR).

Procedure

1. As `root`, use `cfgset` to modify the `cray_lnet` service in the configuration set.

   ```
   smw# cfgset update --service cray_lnet --mode interactive partition
     cray_lnet        [ status: enabled ]  [ validation: valid ]
   ```

```
--------------------------------------------------------------------------
  Selected     #      Settings               Value/Status (level=basic)
--------------------------------------------------------------------------
                      ko2iblnd
                1)      peer_credits         63
                2)      concurrent_sends     63
                      local_lnet
                3)      lnet_name            gni1
                4)      ip_wildcard          10.128.*.*
                5)    flat_routes            [ 6 sub-settings unconfigured, select
                                             and enter C to add entries ]
                6)    fgr_routes             [ 5 sub-settings unconfigured, select
                                             and enter C to add entries ]
--------------------------------------------------------------------------
```

2. Enter 6, then C to configure fine-grained routing (`fgr_routes`).

3. Enter + and type the Sonexion server designation (`snx11029n`) to configure `cray_lnet.settings.fgr_routes`.

```
cray_lnet.settings.fgr_routes.data.dest_name
[<cr>=set '', <new value>, ?=help, @=less] $ snx11029n
CUG
```

4. Enter + to configure `cray_lnet.settings.fgr_routes.data.snx11029n.routers`.

```
cray_lnet.settings.fgr_routes.data.snx11029n.routers
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add routers (Ctrl-d to exit) $ c0-0c2s2n1
Add routers (Ctrl-d to exit) $ c0-0c2s2n2
Add routers (Ctrl-d to exit) $ c1-0c0s7n1
Add routers (Ctrl-d to exit) $ c1-0c0s7n2
Add routers (Ctrl-d to exit) $ c3-0c1s0n1
Add routers (Ctrl-d to exit) $ c3-0c1s0n2
...
Add routers (Ctrl-d to exit) $
```

5. Specify the name of the `ip2nets.conf` file for this FGR configuration. (The file must be placed in the config set on the SMW in `/var/opt/cray/imps/config/sets/`*`config_set`*`/files/roles/lnet/`.)

```
cray_lnet.settings.fgr_routes.data.snx11029n.ip2nets_file
[<cr>=set '', <new value>, ?=help, @=less] $ ip2nets.conf
```

6. Specify the name of the `routes.conf` file for this FGR configuration. Place `routes.conf` in the `config_set` on the SMW `/var/opt/cray/imps/config/sets/`*`config_set`*`/files/roles/lnet/`.

```
cray_lnet.settings.fgr_routes.data.snx11029n.routes_file
[<cr>=set '', <new value>, ?=help, @=less] $ routes.conf
```

7. Follow the guidance in the `cray_lnet` worksheet and set each configuration value for the system as needed.

8. Review the configuration settings:

```
Configured Values:
1) 'snx11029n'
    a) routers:
        c0-0c2s2n1
        c0-0c2s2n2
        c1-0c0s7n1
        c1-0c0s7n2
        ...
    b) ip2nets_file: ip2nets.conf
    c) routes_file: routes.conf
    d) ko2iblnd_peer_credits: 63
    e) ko2iblnd_concurrent_sends: 63
```

9. Reboot the nodes associated with these configuration changes and integrate the new configuration into the default config set (`p0`).

Configure Sonexion system with this FGR scheme.

7.5 Create Custom LNet Configuration for Sonexion

Prerequisites

For a new system, first complete the setup procedures described in the *Sonexion Field Installation Guide*.

About this task

Follow this procedure to create a custom LNet configuration on the Sonexion system while in the "daily mode".

Procedure

1. Log in to the primary management mode.
2. Change to `root`.

```
$ sudo su -
```

3. Stop the Lustre file system.

```
# cscli unmount -f file_system_name
```

4. Use the following steps to change the `o2ib` index. First, start the MySQL client and connect to the `t0db` database.

```
# mysql t0db
```

5. Display the `mgsNID`, `nidFormat`, and `nidIndex` entries.

```
mysql> select * from property where name in ('nidFormat', 'mgsNID', 'nidIndex');
+-----+--------------------+-----------+--------------------+-----------+
| id  | context            | name      | value              | attr_type |
+-----+--------------------+-----------+--------------------+-----------+
|  22 | snx11000n:beConfig | nidFormat | l%s@o2ib%d         | str       |
| 106 | snx11000n:beConfig | nidIndex  | 3                  | int       |
| 109 | snx11000n:beConfig | mgsNID    | lsnx11000n002@o2ib0 | str       |
+-----+--------------------+-----------+--------------------+-----------+
       3 rows in set (0.00 sec)
```

6. Set the `o2ib` index by modifying the `nidIndex` entry.

```
mysql> update property set value=desired_odib_index where name='nidIndex';
```

For example:

```
mysql> update property set value=2 where name='nidIndex';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

7. Set the `mgsNID` entry to match the `o2ib` index.

```
mysql> update property set \
value='original_value@o2ibdesired_o2ib_index' \
where name='mgsNID';
```

For example:

```
mysql> update property set value='lsnx11029n002@o2ib0' where name='mgsNID';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

8. Verify the changes.

```
mysql> select * from property where name in ('nidFormat', 'mgsNID', 'nidIndex');
+-----+--------------------+-----------+--------------------+-----------+
| id  | context            | name      | value              | attr_type |
+-----+--------------------+-----------+--------------------+-----------+
|  22 | snx11000n:beConfig | nidFormat | l%s@o2ib%d         | str       |
| 106 | snx11000n:beConfig | nidIndex  | 2                  | int       |
| 109 | snx11000n:beConfig | mgsNID    | lsnx11000n002@o2ib0 | str       |
+-----+--------------------+-----------+--------------------+-----------+
       3 rows in set (0.00 sec)
```

9. Close the MySQL session.

```
mysql> quit
```

10. Run `puppet`.

```
# /opt/xyratex/bin/beUpdatePuppet -sa
```

11. Run the `beSystemNetConfig.sh` script on the primary management node and wait for it to finish.

```
# /opt/xyratex/bin/beSystemNetConfig.sh \
-c file_location/lnet.conf -r file_location/routes.conf \
-i file_location/ip2nets.conf clustername
```

12. Verify that the customized LNet configuration has been applied.

    a. List the `nids`.

    ```
    # pdsh -g lustre lctl list_nids | sort
    ```

    b. List the nodes and targets.

    ```
    # cscli fs_info
    ```

13. Start the Lustre file system and wait for the targets to mount on all system nodes.

```
# cscli mount -f file_system_name
```

14. Modify `modprobe.conf` to support IB Aliases.

    The following examples intercept the LNet module and setup IB aliases before the module is loaded. **Each example below is a single line** in `lnet.conf`.

    a. Add `ib0:1` to `10.10.101` subnet.

    ```
    ## Create aliases.
    ## Intercepts the Lnet module and sets up the aliases before the module is loaded install lnet

    /sbin/ip -o -4 a show ib0 | /usr/bin/awk '/inet/{s=$4;sub("10\\.10\\.100\
    \.","10.10.101.",s);print "/sbin/ip address add dev ib0 label ib0:1",s}' | /bin/sh ;/sbin/
    modprobe --ignore-install lnet $CMDLINE_OPTS
    ```

    b. Remove all `ib0:1` aliases when LNet module is unloaded.

```
## Remove all ib0:1 aliases when lnet module is unloaded

remove lnet /sbin/modprobe -r --ignore-remove lnet && /sbin/ip -o -4 a show label ib0:1 | awk
'{print "/sbin/ip address del dev ib0 label ib0:1",$4}' | /bin/sh
```

c. Configure a metadata server on a 3rd IPv4 subnet.

```
install lnet if nodeattr mds; then /sbin/ip -o -4 a show ib0 | /usr/bin/awk '/inet/{s=$4;sub("10\
\.10\\.100\\.","10.10.102.",s);print "/sbin/ip address add dev ib0 label ib0:1",s}' | /bin/sh ;
else /sbin/ip -o -4 a show ib0 | /usr/bin/awk '/inet/{s=$4;sub("10\\.10\\.100\
\.","10.10.101.",s);print "/sbin/ip address add dev ib0 label ib0:1",s}' | /bin/sh ; fi ; /sbin/
modprobe --ignore-install lnet $CMDLINE_OPTS
```

8 Change Network Settings

8.1 Create Custom LNet Configuration for Sonexion

Prerequisites

For a new system, first complete the setup procedures described in the Sonexion Field Installation Guide.

About this task

Follow this procedure to create a custom LNet configuration on the Sonexion system while in the "daily mode".

Procedure

1. Log in to the primary management mode.
2. Change to `root`.

   ```
   $ sudo su -
   ```

3. Stop the Lustre file system.

   ```
   # cscli unmount -f file_system_name
   ```

4. Use the following steps to change the `o2ib` index. First, start the MySQL client and connect to the `t0db` database.

   ```
   # mysql t0db
   ```

5. Display the `mgsNID`, `nidFormat`, and `nidIndex` entries.

   ```
   mysql> select * from property where name in ('nidFormat', 'mgsNID', 'nidIndex');
   +-----+--------------------+-----------+--------------------+-----------+
   | id  | context            | name      | value              | attr_type |
   +-----+--------------------+-----------+--------------------+-----------+
   |  22 | snx11000n:beConfig | nidFormat | l%s@o2ib%d         | str       |
   | 106 | snx11000n:beConfig | nidIndex  | 3                  | int       |
   | 109 | snx11000n:beConfig | mgsNID    | lsnx11000n002@o2ib0 | str       |
   +-----+--------------------+-----------+--------------------+-----------+
          3 rows in set (0.00 sec)
   ```

6. Set the `o2ib` index by modifying the `nidIndex` entry.

   ```
   mysql> update property set value=desired_odib_index where name='nidIndex';
   ```

   For example:

```
mysql> update property set value=2 where name='nidIndex';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

7. Set the `mgsNID` entry to match the `o2ib` index.

```
mysql> update property set \
value='original_value@o2ibdesired_o2ib_index' \
where name='mgsNID';
```

For example:

```
mysql> update property set value='lsnx11029n002@o2ib0' where name='mgsNID';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

8. Verify the changes.

```
mysql> select * from property where name in ('nidFormat', 'mgsNID', 'nidIndex');
+-----+--------------------+-----------+---------------------+-----------+
| id  | context            | name      | value               | attr_type |
+-----+--------------------+-----------+---------------------+-----------+
|  22 | snx11000n:beConfig | nidFormat | l%s@o2ib%d          | str       |
| 106 | snx11000n:beConfig | nidIndex  | 2                   | int       |
| 109 | snx11000n:beConfig | mgsNID    | lsnx11000n002@o2ib0 | str       |
+-----+--------------------+-----------+---------------------+-----------+
       3 rows in set (0.00 sec)
```

9. Close the MySQL session:

```
mysql> quit
```

10. Run `puppet`:

```
# /opt/xyratex/bin/beUpdatePuppet -sa
```

11. Run the `beSystemNetConfig.sh` script on the primary management node and wait for it to finish:

```
# /opt/xyratex/bin/beSystemNetConfig.sh \
-c file_location/lnet.conf -r file_location/routes.conf \
-i file_location/ip2nets.conf clustername
```

12. Use this step and the next step to verify that the customized LNet configuration has been applied. First, list the node NIDs:

```
# pdsh -g lustre lctl list_nids | sort
```

13. List the nodes and targets:

```
# cscli fs_info
```

14. Start the Lustre file system and wait for the targets to mount on all system nodes:

```
# cscli mount -f file_system_name
```

Modify modprobe.conf to support IB Aliases

15. The following examples intercept the LNet module and setup IB aliases before the module is loaded. **Each example below is a single line** in `lnet.conf`.

Add ib0:1 to 10.10.101 subnet:

```
## Create aliases.
## Intercepts the Lnet module and sets up the aliases before the module is loaded install lnet
```

```
/sbin/ip -o -4 a show ib0 | /usr/bin/awk '/inet/{s=$4;sub("10\\.10\\.100\\.","10.10.101.",s);print "/
sbin/ip address add dev ib0 label ib0:1",s}' | /bin/sh ;/sbin/modprobe --ignore-install lnet
$CMDLINE_OPTS
```

Remove all ib0:1 aliases when LNet module is unloaded:

```
## Remove all ib0:1 aliases when lnet module is unloaded

remove lnet /sbin/modprobe -r --ignore-remove lnet && /sbin/ip -o -4 a show label ib0:1 | awk
'{print "/sbin/ip address del dev ib0 label ib0:1",$4}' | /bin/sh
```

Configure a metadata server on a 3rd IPv4 subnet:

```
install lnet if nodeattr mds; then /sbin/ip -o -4 a show ib0 | /usr/bin/awk '/inet/{s=$4;sub("10\\.
10\\.100\\.","10.10.102.",s);print "/sbin/ip address add dev ib0 label ib0:1",s}' | /bin/sh ; else /
sbin/ip -o -4 a show ib0 | /usr/bin/awk '/inet/{s=$4;sub("10\\.10\\.100\\.","10.10.101.",s);print "/
sbin/ip address add dev ib0 label ib0:1",s}' | /bin/sh ; fi ; /sbin/modprobe --ignore-install lnet
$CMDLINE_OPTS
```

8.2 Change DNS Resolver Configuration

About this task

Follow this procedure to change the DNS resolver, the system service that translates URLs into IP addresses.

Procedure

1. SSH into the primary MGMT node:

   ```
   $ ssh -l admin primary_MGMT_node
   ```

2. Sudo to `root`:

   ```
   [MGMT0]$ sudo su -
   ```

3. Update the DNS settings in the `t0db` database:

   ```
   [MGMT0]$ mysql t0db -e "replace into property(context,name,value,attr_type)
   values (\"$(nodeattr -VU cluster):beSystemNetConfig\",\"nameServers\",\
   "xx.xx.xx.xx yy.yy.yy.yy\",\"str\")"
   ```

   Where *xx.xx.xx.xx* and *yy.yy.yy.yy* are the IP addresses of the primary and secondary DNS servers, respectively.

4. Propagate the settings:

   ```
   [MGMT0]$ /opt/xyratex/bin/beUpdatePuppet -s -g mgmt
   ```

8.3 Change Externally Facing IP Addresses

About this task

Use this procedure to change externally facing IP addresses, for customers to change one or both of the IP addresses of a Sonexion system after it has been installed. Each MGMT node binds an Ethernet interface to one of these externally facing IP addresses. On release 1.5.0 and later, that interface is `eth1`, which is used in the following examples.

Procedure

1. Log in to the secondary MGMT node.
2. SSH to the primary MGMT node:

   ```
   $ ssh -l admin primary_MGMT_node
   ```

3. Sudo to `root`:

   ```
   [admin@n000]$ sudo su -
   ```

4. Edit the Ethernet configuration file and change as desired:

   ```
   [root@n000]# vi /etc/sysconfig/network-scripts/ifcfg-eth1
   ```

   If the system was initially configured to use dynamic (DHCP) IP addresses, the file will look like this:

   ```
   DEVICE=eth1
   BOOTPROTO=dhcp
   ONBOOT=yes
   ```

   If the system was configured to use static IP addresses, it will look like this:

   ```
   DEVICE=eth1
   BOOTPROTO=static
   IPADDR=xx.xx.xx.xx
   NETMASK=255.255.x.x
   GATEWAY=xx.xx.xx.xx
   ONBOOT=yes
   ```

   Where *xx.xx.xx.xx* is the valid IP address.

5. Toggle the Ethernet interface:

   ```
   [root@n000]# ifdown eth1; ifup eth1
   ```

6. Exit out of the SSH session for the primary MGMT node.
7. Log in to the primary MGMT node and SSH into the secondary MGMT node.
8. Repeat steps *3* through *5* to configure the externally facing IP address on the secondary MGMT node.

8.4 Configure LDAP over TLS in Daily Mode

About this task

Set up an LDAP/TLS server.

Procedure

1. In the file `/etc/puppet/modules/ldap/templates/ldap.conf.erb`:
   a. Add the line:

      ```
      tls_cacert /etc/openldap/cacerts/ca.crt
      ```

      or whatever file has the CA certificate.
   b. Change the line:

      ```
      uri ldap://<%= system['serverName'] %>:<%= system['serverPort'] %>/
      ```

      to:

      ```
      uri ldaps://<%= system['serverName'] %>:<%= system['serverPort'] %>/
      ```

2. In the file `/etc/puppet/modules/ldap/templates/nslcd.conf.erb`:
   a. Add the line:

      ```
      tls_cacertfile /etc/openldap/cacerts/ca.crt
      ```

      or whatever file has the CA certificate.
   b. Change the line:

      ```
      uri ldap://<%= system['serverName'] %>:<%= system['serverPort'] %>/
      ```

      to:

      ```
      uri ldaps://<%= system['serverName'] %>:<%= system['serverPort'] %>/
      ```

3. Put the CA certificate file in the appropriate `%=_system['serverName']_%` directory in the image on **n001**:

   ```
   /mnt/nfsdata/images/2.0.0-51/appliance.x86_64/etc/openldap/cacerts/
   ```

   Additional Customer LDAP Parameters

   If your customer's LDAP server requires parameters not included in the procedure above, use the following steps to add these changes to the file's `/etc/puppet/modules/ldap/templates/ldap.conf.erb` and `/etc/puppet/modules/ldap/templates/nslcd.conf.erb`.

   NOTE: The customer parameter can be added anywhere in the two files except between "<%" and "%>".

Example: The customer-secure LDAP server requires the following parameter setting:

```
TLS_REQCERT never
```

4. Log in to MGMT node n000 as root .

5. Add the parameter to the following files, noting that it can be added anywhere in the two files except between "<%" and "%>":
   - `/etc/puppet/modules/ldap/templates/ldap.conf.erb`
   - `/etc/puppet/modules/ldap/templates/nslcd.conf.erb`

6. Once the parameter is added, puppet configuration needs to be updated.

   On MGMT node n000 as root, execute the following commands:

```
pdsh -g mgmt puppetd -tv
pdsh -g mds puppetd -tv
```

7. After puppet configuration has been updated login to the servers n000, n001, n002, n003 and verify that `/etc/ldap.conf and /etc/openldap/ldap.conf` files have been modified as expected.

8. To verify LDAP functionality, execute the following command from each node n000, n001, n002, n003.

```
L_GETIDENTITY_TEST=1 /usr/sbin/l_getidentity_nss snx11168-MDT0000 uid
```

8.5 Change LDAP Settings in Daily Mode

About this task

Use this procedure to change the LDAP settings, which are stored in the `t0db` database table `ldap_setup`. The following columns are present in this table and are used to configure LDAP:

Table 5. Database LDAP Settings

Setting	Use
`server_name`	The LDAP server
`port`	The port that the LDAP server listens on (typically 389)
`base_dn`	The base DN to search
`user_dns`	Search patch for user information
`group_dns`	Search path for group information
`bind_dns`	DN to bind to the LDAP directory
`password`	Password to use with `bind_dn`

To change the LDAP settings of a running cluster, it is necessary to change the corresponding field with the `update MySQL` command and then run `beUpdatePuppet -sa`. The following example shows how to change the base DN to search.

> **IMPORTANT:** As of SU10, the preferred method is to use the `cscli lustre_users ldap` commands, which are described in the CSCLI reference section or use the CSM GUI, **Configure** tab **LDAP** settings window.

Procedure

1. SSH into the primary MGMT node.

```
$ ssh -l admin primary_MGMT_node
```

2. Sudo to root:

```
[MGMT0]$ sudo su -
```

3. Print the existing configuration:

   a. List the contents of `/etc/nslcd.conf`.

```
[MGMT0]# cat /etc/nslcd.conf
#
# LDAP Trinity
#
# Auto generated by puppet
# Do not change it manually
#

timelimit 120
bind_timelimit 120
idle_timelimit 3600

# Workaround for names <3 char length. see TRT-1832
validnames /^[a-z0-9._@$][a-z0-9._@$ \~-]*[a-z0-9._@$~-]$/i

#pam_password md5
#bind_policy soft
#ldap_version 3

uri ldap://172.30.12.19:389/
base dc=datacenter,dc=cray,dc=com
base passwd ou=People,dc=datacenter,dc=cray,dc=com
```

   b. List the LDAP settings in the `t0db` database table `ldap_setup`.

```
[MGMT0]#  mysql t0db -e "select * from ldap_setup;"
```

```
+---------------+-----------------------------+---------+----------+------------+----------+------------+-------------+---------+
| ldap_setup_id | base_dn                     | bind_dn | password | cluster_id | tls_cert | tls_pvtkey | tls_ca_cert | mapping |
+---------------+-----------------------------+---------+----------+------------+----------+------------+-------------+---------+
|             1 | dc=datacenter,dc=cray,dc=com | NULL    | NULL     |          1 | NULL     | NULL       | NULL        | default |
+---------------+-----------------------------+---------+----------+------------+----------+------------+-------------+---------+
```

4. Change the `base_dn` setting:

```
[MGMT0]# mysql t0db -e "update ldap_setup set
base_dn='dc=new_ldap,dc=example,dc=com'"
```

Where *new_ldap* is the new LDAP server.

5. Update `puppet`:

```
[MGMT0]# /opt/xyratex/bin/beUpdatePuppet -sa
```

6. Repeat step 3 and check for the new value(s).

8.6 Configure NIS Support in Daily Mode

Prerequisites

- NIS must not have been previously configured during Site Configuration mode. If it is already configured, see *Change NIS Settings in Daily Mode*.
- Both the NIS Domain name and the IP address of NIS servers reachable (pingable) from the Management Nodes must be known.

About this task

This procedure applies to releases 1.3.1 and later. Sonexion 1.3.1 supports LDAP and NIS but was intended only to support configuration of NIS during the Customer Wizard phase of the installation.

Procedure

1. Enable support for NIS:

```
[root@n000]# mysql t0db -e 'update filesystem set lustre_upcall="nis"'
```

2. Configure the IP address(es) of NIS servers:

```
[root@n000]# mysql t0db -e 'insert into property (context, name, value,
attr_type) values ("lustre:upcall", "nis_server", "xx.xx.xx.xx","str")'
```

To specify more than one IP address, ensure that they are separated by a single space.

3. Configure the name of the NIS domain:

```
[root@n000]# mysql t0db -e 'insert into property (context, name, value,
attr_type) values ("lustre:upcall", "nis_domain", "xxxxxxxxxx", "str")'
```

Where *xxxxxxxxxx* is the value of `nisdomainname` on the relevant server.

4. Update `puppet`:

```
[root@n000]# beUpdatePuppet -sa
```

5. Check that `/etc/puppet/data/CSSM/nis.yaml` has been updated on the primary and secondary management nodes and contains the following lines (example):

```
[root@n000]# pdsh -a cat /etc/puppet/data/CSSM/nis.yaml 2>/dev/null | dshbak -c
----------------
```

```
MGMT[00-01]
----------------
lustre_nis:
nis_domain: xxxxxxxxxx
nis_server: [xx.xx.xx.xx]
```

6. Verify that `/etc/yp.conf` on the MMU nodes contains the same information (example):

```
[root@n000]# pdsh -a cat /etc/yp.conf 2>/dev/null | dshbak -c
----------------
MGMT[00-03]
----------------
#
#
# CSSM Lustre NIS
#
#
# Auto generated by puppet
# Do not change it manually
#
domain xxxxxxxxxx server xx.xx.xx.xx
```

7. Run these final checks on all MMU nodes:

```
[root@n000]# service ypbind status
[root@n000]# ypwhich
[root@n000]# ypwhich -m
```

8.7 Change NIS Settings in Daily Mode

Prerequisites

NIS has previously been configured, either during customer wizard or following the instructions in *Configure NIS Support in Daily Mode*.

About this task

Use this procedure, which applies to releases 1.3.1 and later, to change NIS settings using the CSCLI interface in Daily Mode.

Procedure

1. SSH into the primary MGMT node.

```
$ ssh -l admin primary_MGMT_node
```

2. Sudo to `root`:

```
[admin@n000]$ sudo su -
```

3. Print the existing configuration:

```
[root@n000]# service ypbind status
[root@n000]# ypwhich
[root@n000]# ypwhich -m
[root@n000]# cat /etc/yp.conf
[root@n000]# mysql t0db -e 'select * from property where name = "nis_domain" or
name = "nis_server"'
```

Sample output:

```
[root@snx11000n000 ~]# pdsh -g mgmt,mds ypwhich | dshbak -c
----------------
snx11000n[000-003]
----------------
172.30.74.10

[root@snx11000n000 ~]# pdsh -g mgmt,mds ypwhich -m | dshbak -c
----------------
snx11000n[000-003]
----------------
auto_sw_linux_cf ra.us.cray.com
auto_sw_linux_sea ra.us.cray.com
auto_users ra.us.cray.com
auto_master_linux_mh ra.us.cray.com
auto_master_linux_cf ra.us.cray.com
```

4. Change to IP address of the NIS Server:

```
[root@n000]# mysql t0db -e 'update property set value =
"xx.xx.xx.xx yy.yy.yy.yy" where name = "nis_server"'
```

One or more IP addresses can be specified for NIS master and its reachable NIS slaves.

5. Change the NIS domain name of the NIS server.

```
[root@n000]# mysql t0db -e 'update property set value = "xxxxxxxxxx" where name
= "nis_domain"'
```

6. Update puppet:

```
[root@n000]# beUpdatePuppet -sa
```

This is required after the previous changes.

7. Repeat step *3* to verify the new NIS server settings.

9 Change the NTP Configuration

About this task

Use this procedure to change the Network Time Protocol (NTP) configuration on a Sonexion system without re-running the customer wizard. (Since the wizard makes many changes, it can introduce unnecessary complexity.) The following procedure has these characteristics:

- A change in the **t0db** MySQL database is required.
- As with any direct changes to the database, the database is first backed up.
- Changes are performed in a script session.
- It is best if there are not large differences between the old and new server times.

Procedure

1. Back up the database:

   ```
   # mysqldump t0db > t0db.bak
   ```

2. Determine what is currently in use:

   ```
   # mysql t0db -e "select * from property where name = 'timeServers';"
   ```

3. Update the database:

   ```
   # mysql t0db -e "update property set value = 'list' where name =
   'timeServers';"
   ```

   Where *list* is a space-separated list of timeserver IPs.

4. Propagate changes:

   ```
   # beUpdatePuppet -sa
   ```

10 Check File Systems Using e2fsck Command

About this task

Perform the following steps on the Sonexion filesystem. In order to run e2fsck on a device or set of devices, the device must not be mounted. In a Lustre environment, even a read-only e2fsck runs incorrectly on a mounted and running device.

Procedure

1. Stop Lustre by running the following on the MGMT node:

```
[root@n000]# cscli unmount -f fs_name
```

2. SSH to the OSS node containing the devices where the e2fsck is to be run, **n004** in this example:

```
[root@snx11000n000]# ssh snx11000n004
[root@snx11000n004]#
```

3. Use the following steps to build the disk arrays to be accessible as mountable devices. The simplest method is to build all the arrays on the system. A Sonexion using MDRAID has four devices; a Sonexion using GridRAID has only one device.

```
[root@snx11000n004]# mdraid-activate -md
mdadm: /dev/md/snx11000n004:md132 has been started with 2 drives.
mdadm: /dev/md/snx11000n004:md133 has been started with 2 drives.
mdadm: /dev/md/snx11000n004:md2 has been started with 10 drives.
assembled snx11000n004:/dev/md2 in 1 tries
mdadm: /dev/md/snx11000n004:md140 has been started with 2 drives.
mdadm: /dev/md/snx11000n004:md141 has been started with 2 drives.
mdadm: /dev/md/snx11000n004:md6 has been started with 10 drives.
assembled snx11000n004:/dev/md6 in 1 tries
mdadm: /dev/md/snx11000n004:md128 has been started with 2 drives.
mdadm: /dev/md/snx11000n004:md129 has been started with 2 drives.
mdadm: /dev/md/snx11000n004:md0 has been started with 10 drives.
assembled snx11000n004:/dev/md0 in 1 tries
mdadm: /dev/md/snx11000n004:md136 has been started with 2 drives.
mdadm: /dev/md/snx11000n004:md137 has been started with 2 drives.
mdadm: /dev/md/snx11000n004:md4 has been started with 10 drives.
assembled snx11000n004:/dev/md4 in 1 tries
```

In the above output, the OST devices are named /dev/md# in the lines beginning assembled snx11###n###:md# in 1 tries. From the example above, the device of interest is this:

```
assembled snx11000n004:md2 in 1 tries
```

The other drives are journaling and backup devices.

4. Confirm that OST devices are not mounted, by running `mount` and verifying that they are not present. Only four devices should be mounted for MDRAID systems, or one for GridRAID systems. The following output indicates they are **not** mounted:

```
[root@snx11000n004]# mount
none on /proc type proc (rw)
none on /dev/shm type tmpfs (rw)
none on /tmp type tmpfs (rw)
none on /mnt type tmpfs (rw,size=512k)
none on /WIBS type tmpfs (rw,size=512k)
none on /data type tmpfs (rw,size=512k)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/sda2 on /mnt/persistent_ha_data type ext4 (rw)
/mnt/persistent_ha_data/dirs/etc/ha.d on /etc/ha.d type none (rw,bind)
/mnt/persistent_ha_data/dirs/var/lib/heartbeat on /var/lib/heartbeat type none (rw,bind)
/mnt/persistent_ha_data/dirs/var/lib/mdraidscripts on /var/lib/mdraidscripts type none (rw,bind)
/mnt/persistent_ha_data/dirs/var/lib/pengine on /var/lib/pengine type none (rw,bind)
/dev/md132 on /WIBS/snx11000n004:md2 type ext2 (rw)
/dev/md128 on /WIBS/snx11000n004:md0 type ext2 (rw)
/dev/md140 on /WIBS/snx11000n004:md6 type ext2 (rw)
/dev/md136 on /WIBS/snx11000n004:md4 type ext2 (rw)
```

   For comparison, the following output indicates an **incorrect condition**, in that the devices **md0**, **md2**, **md4**, and **md6** are mounted.

```
[root@snx11000n004]# mount
none on /proc type proc (rw)
none on /dev/shm type tmpfs (rw)
none on /tmp type tmpfs (rw)
none on /mnt type tmpfs (rw,size=512k)
none on /WIBS type tmpfs (rw,size=512k)
none on /data type tmpfs (rw,size=512k)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/sda2 on /mnt/persistent_ha_data type ext4 (rw)
/mnt/persistent_ha_data/dirs/etc/ha.d on /etc/ha.d type none (rw,bind)
/mnt/persistent_ha_data/dirs/var/lib/heartbeat on /var/lib/heartbeat type none (rw,bind)
/mnt/persistent_ha_data/dirs/var/lib/mdraidscripts on /var/lib/mdraidscripts type none (rw,bind)
/mnt/persistent_ha_data/dirs/var/lib/pengine on /var/lib/pengine type none (rw,bind)
/dev/md132 on /WIBS/snx11000n004:md2 type ext2 (rw)
/dev/md128 on /WIBS/snx11000n004:md0 type ext2 (rw)
/dev/md140 on /WIBS/snx11000n004:md6 type ext2 (rw)
/dev/md136 on /WIBS/snx11000n004:md4 type ext2 (rw)
/dev/md2 on /data/snx11000n004:md2 type lustre (rw,errors=panic,journal_checksum)
/dev/md0 on /data/snx11000n004:md0 type lustre (rw,errors=panic,journal_checksum)
/dev/md6 on /data/snx11000n004:md6 type lustre (rw,errors=panic,journal_checksum)
/dev/md4 on /data/snx11000n004:md4 type lustre (rw,errors=panic,journal_checksum)
```

5. Run the `e2fsck` command as follows. A basic `e2fsck` task is to run a read-only check of the device to diagnose any errors, such as a simple non-invasive action on **md2**.

   Run the command in the form:

```
[root@nxxx]$ e2fsck -nvf device_to_be_checked
```

   Example:

```
[root@snx11000n004]# e2fsck -nvf /dev/md2
```

   Analysis of `e2fsck` output is described elsewhere. If the output from the above commands indicates the need for a read/write `e2fsck`, use the same command with the appropriate options.

6. Deactivate the RAID to clean up and return the system to a state ready to mount for normal Lustre operation:

```
[root@snx11000n004]# mdraid-deactivate -md
```

11 Disk Watching Daemon

This topic explains the use of the Disk Watching Daemon (DWD) to monitor hard drives in the system.

- **Prerequisites**

  For Sonexion 1.3.1 systems, System Update 019 (SU-019) must be installed for DWD to be available on the system.
- **Service Interruption Level**: Live (can be applied to a live system with no service interruption)
- **System Access Requirements:**

  This procedure is written for use by an admin user and does not require root (super user) access. It is recommended that the technician log in as an administrative user and perform the procedure using CSCLI commands as written, and not log in as root or perform the procedure as a root user.

DWD is a daemon on Sonexion systems that monitors hard drives for symptoms that suggest impending failure. DWD periodically polls all hard drives that are part of a RAID array, and if it finds a symptomatic drive, DWD takes corrective action.

DWD is available on Sonexion systems beginning with software release 1.3.1-019. Note that 1.3.1 systems must have at least System Update 019 (SU-019) installed.

DWD runs on all nodes of a Sonexion system, and monitors every disk drive for three specific failure conditions:

- An excess of timed-out SCSI commands
- Failure of the SMART status check
- Failure during a drive interrogation

DWD Configuration

DWD can be configured to take one of two actions when a drive fails a DWD health check:

- Log the test failure with a suggestion that the drive be removed.
- Log the test failure and then power down the drive, causing the hard drive to be removed from any RAID arrays of which it is a member. After the drive is removed from the RAID array, a replacement hot-spare drive is automatically recruited into the RAID device (for MDRAID configurations only) and a RAID rebuild is triggered. For GridRAID configurations, the drive is removed and a repair operation is triggered to use the built-in hot-spare space of the array.

Even if DWD is configured for this option, the drive is not powered off if the array is already degraded. Instead, DWD reports in syslog:

```
"Array array_name is degraded or without redundancy, no automatic purging".
```

To determine what corrective action DWD will take when a drive fails a health check, run:

```
[admin@n000]$ cscli dwd
```

Sample command and output:

```
[admin@snx11000n000 ~]$ cscli dwd
------------------------------------------------------------
Node Drive power-off on failure Task abortion rate
------------------------------------------------------------
global 0 1
```

In the sample output, the column labeled Drive power-off on failure indicates what action DWD takes when that drive fails one of the health checks.

- A value of 0 (the default) indicates that the drive will not be powered off. The only action that occurs is that a message will be sent to syslog stating that, "You should be Failing out problematic drive."

  For example:

  ```
  Nov 26 18:24:02 cstor01n04 dwd: You should be Failing out problematic drive :
  </sbin/mdadm /dev/md4 --fail /dev/sdo>
  ```

- A value of 1 indicates that DWD will log "Failing out problematic drive" to syslog:

  ```
  Nov 26 18:24:02 cstor01n04 dwd: Failing out problematic drive : </sbin/
  mdadm /dev/md4 --fail /dev/sdo>
  ```

DWD then powers off the drive.

DWD's Drive power-on can be set on failure by using the `cscli dwd -l` command, where `-l` is lethal. For example:

```
[admin@snx11000n000 ~]$ cscli dwd -l yes
dwd: done
[admin@snx11000n000 ~]$ cscli dwd
------------------------------------------------------------
Node Drive power-off on failure Task abortion rate
------------------------------------------------------------
global 1 1
```

At the time of publication, once the "Drive power-off on failure" value is 1, it cannot be set back to 0.

DWD Check 1: Excess Timed-out SCSI Commands

When a disk drive does not respond to an I/O command within 100 seconds, that command is considered failed and the SCSI subsystem attempts to abort it. DWD watches the count of timed-out commands and considers a drive failed after it exceeds a certain number of aborted tasks per (rolling) hour.

This "Task abortion rate" threshold is viewable in CSCLI with the command `cscli dwd`:

Sample command and output:

```
[admin@snx11000n000 ~]$ cscli dwd
------------------------------------------------------------
Node Drive power-off on failure Task abortion rate
------------------------------------------------------------
global 0 1
[admin@snx11000n000 ~]$
```

When the value for the task abortion rate is 1, the actual threshold is automatically set to a value of (queue depth) * 2 + 1, which works out to either 5 or 9, depending on the queue depth setting on the hard drive.

To set the task abortion rate manually, run:

```
[admin@n000]$ cscli dwd -a value
```

Sample command and output:

```
[admin@snx11000n000 ~]$ cscli dwd -a 20
dwd: done
[admin@snx11000n000 ~]$ cscli dwd
------------------------------------------------------------
Node Drive power-off on failure Task abortion rate
------------------------------------------------------------
global 0 20
```

DWD Check 2: Failure of the SMART Status Check

Periodically, DWD runs a SMART inquiry, `smartctl -q -H` *device*, of all drives that are members of a RAID array assembled on the local host. If the SMART status check returns DISK FAILING, the test fails.

Currently, there are no adjustable parameters for this check. A future release of DWD may allow the user to change the test rate (which is currently hardcoded to occur once a day) or to disable this test.

DWD Check 3: Failure During a Drive Interrogation

If a disk drive accumulates 12 SCSI errors within 1 hour, it is subjected to an interrogation. DWD issues the following commands against the drive:

```
mdadm --examine drive
smartctl -l error drive
```

The test fails if the above commands indicate problems with the drive, either because the command timed out or because it returned errors.

At the time of publication, no drive has failed interrogation. This test is included in DWD for completeness, but success on this test should not be assumed to indicate good disk drive health.

No parameters for this check can be set through the `cscli dwd` command.

12 Distributed Namespace (DNE) Feature

Manage DNE in Sonexion

The Distributed NamespacE (DNE) feature is managed by a system administrator with root privileges, via `lfs` commands from any Lustre client node.

As the root user on a client node, use `lfs` commands to assign directories to specific MDTs, so all files and subdirectories added under that directory will be placed on that MDT. Any directory hierarchy not explicitly assigned to an MDT is assigned, by default, to MDT0.

To assign a directory tree to a particular MDT, run:

```
[Client]$ sudo lfs mkdir -i mdtindex remotedir
```

where *`mdtindex`* specifies the MDT (using MDTx syntax) and *`remotedir`* specifies the directory name.

For example, if Lustre is mounted on `/mnt/lustre`, running these commands creates directories alpha on MDT0, bravo on MDT1, and charlie on MDT2.

```
[Client]$ sudo lfs mkdir -i 0 /lustre/alpha
[Client]$ sudo lfs mkdir -i 1 /lustre/bravo
[Client]$ sudo lfs mkdir -i 2 /lustre/charlie
```

By default, the system administrator can create remote sub-directories only on the base MDS (MDT0). However, once remote subdirectories exist, the system administrator can allocate them to separate MDTs (for example, MDT1, MDT2, etc.). Cray does not recommend creating remote subdirectories in parent directories that are **not** hosted on MDT0, because failure of the parent MDT leaves the namespace below it inaccessible. To relax this limitation and enable remote subdirectories to be accessed from any MDT, use the following command:

```
[Client]$ sudo lctl set_param mdd.*.enable_remote_dir=1
```

Purpose of DNE

DNE is a Lustre 2.5+ feature (implemented in Sonexion 2.0.0) that enables multiple MetaData Servers (MDSs) to be used within a single Lustre file system. With DNE, the file system namespace is shared among multiple MetaData Targets (MDTs), which are reachable through one or more MDSs. Each MDS may front multiple MDTs. The function of the MDS is to make metadata stored across the MDTs (that is, the “namespace”) available to Lustre clients.

Each MDT handles a subsection of the namespace; these subsections must be explicitly specified by the system administrators. The subsections consist of a set of directory subtrees. All files and directories created in a subtree assigned to an MDT have their metadata and directory contents stored on that MDT.

Uses for DNE

Spreading the Lustre namespace over many MDSes supports the following five distinct use cases:

Improve System Metadata Performance

Distributing metadata load among multiple servers and drive sets improves overall system metadata performance:

- MDT operations can occur in parallel, with no contention among the drives (each MDT has a distinct set of drives) or within the MDT software stack
- Multiple MDSes provide more memory, CPU power, and execution threads
- Overall metadata throughput of the system scales linearly with the number of MDSes
- Even within a single MDS, two MDTs provide better metadata performance than a single one

  If MDT access patterns are not evenly balanced, system performance will be suboptimal.

Prevent Global System Impacts by Isolating Operations to a Specific MDT

In many situations, a Lustre file system sees concurrent use by a number of different jobs, some of which may have significant metadata components and impacts. If a single job is too disruptive to others, an administrator can assign this job to a more isolated MDT (and its specific namespace subtrees), thereby eliminating systemic impact to other users.

Enable Scalable Metadata Access in Highly Contended Directories

If a job utilizes a large number of files from a single directory (more than 50,000), internal locking and iterative functions can significantly impact operations in that directory. In this case, the directory can be split ("striped") across two or more MDTs to improve metadata performance. However, this optimization requires the "DNE striped directories" enhancement, which is only available in Lustre 2.6+.

Increase Namespace Size Beyond a Single ext4 File System

A single `ldiskfs` (ext4) file system is limited to a maximum of 4 billion files. With DNE, each MDT can have 4B files, allowing scalable access to larger file counts.

Decrease Individual MDS Load by Utilizing Controller Pairs

A pair of MDS servers in a suitable enclosure, such as Cray's Additional DNE Unit (ADU), can utilize dual-ported drives that, in the event of a problem, enable one server to "take over" the MDS service for any MDT in that enclosure. In non-DNE implementations, the paired MDSes act in an active-passive configuration, meaning one MDS is idle while the other must serve the entire metadata load. The DNE feature allows MDTs to operate in an active-active failover configuration, similar to Object Storage Target (OST) layouts. During normal operation, each MDS in the pair serves half of the MDTs (with half of the system's metadata), as compared to a non-DNE configuration. Put more simply, given the same metadata load, utilizing DNE enables the system's metadata load to be split among two (or more) servers while maintaining failover protection.

DNE Implemention in Sonexion

In Cray's Lustre 2.5.1 implementation (available in the Sonexion 2.0.0 release), the DNE feature supports additional MDTs in the logical node architecture, enabling enhanced metadata performance and increased inode support, while providing improved metadata operation load balancing for greater system reliability. Using DNE allows multiple MDTs, operating through multiple MDS nodes, to be configured and operate as part of a single file system.

In Sonexion 2.0.0, DNE-based functionality is implemented as the ADU feature, which provides additional MDS/MDT capability to a file system cluster. Cray's ADU architecture is deployed as a single 2U24 enclosure containing:

- Two ADU DNE servers (Object Storage Servers)
- 22 x 10K HDDs
- Two standard 2U24 power supply units / fan modules

Each ADU enclosure hosts two MDS nodes:

- One MDS node runs on each ADU DNE server; together, the MDSes operate as an HA failover pair.
- During normal operations, each MDS owns and operates one of the two MDTs provided by the ADU.
- During failover operations, when one MDS node fails, the surviving MDS owns and operates services for both MDTs in the ADU.

Each MDT is provisioned as follows:

- One RAID10 (5+5) array, with a total of two RAID10 arrays per ADU enclosure
- 22 HDDs plus two hot spare disks

A Sonexion file system using Cray's Lustre 2.5.1 implementation can be configured with 0 to 8 ADUs, providing up to 16 additional MDTs (not including the primary MDT). The following figure shows the Sonexion architectural framework with 4 ADU servers added (base MDS/MDT [MDT0] with serially-numbered additional MDTs [MDT1, MDT2, MDT3 and MDT4].

Figure 6. ADU/DNE Implementation

13 Support Bundles

Support bundles are collections of event logs from field systems that are collected by Cray personnel and used to debug many Sonexion problems. Sonexion provides a mechanism for collecting support bundles that can be initiated manually or triggered automatically by certain events (e.g., Lustre bugs or failover events). These support bundles should be provided to Cray personnel in the course of requesting technical support.

When a problem is encountered, use the following tabs in the CSSM GUI to assist with resolving the issue:

- The **Health** tab displays details of the host or service alerts and notifications to determine the issues.
- The **Log Browser** tab is used to review the log files for help identifying and diagnosing the issues.
- The **Support** tab shows diagnostic information from the storage cluster, including logs and configuration settings.

Support File Overview

The CSSM **Support** tab provides support functionality for collecting diagnostic information, including logs and configuration settings, on an automatic or manual basis. When a Lustre error occurs, the system automatically collects diagnostic information. Alternatively, Sonexion users can manually collect a diagnostic payload and browse the contents.

The three principal resources for debugging an issue are support bundles, system logs, and GEM logs.

Collection of Sonexion Data in Support Files

When a Lustre error or a system event (such as failover) occurs, Sonexion automatically triggers a process to collect system data and diagnostics, and bundle them in support files. The process waits two minutes before collecting the data to ensure that all consequences of the events and errors are logged. Only one collection process is active at a time.

Multiple errors do not trigger the collection of additional data if the current process is still running or within a two-hour window after the current process was triggered. For example, if a Lustre error occurs at 8:00, triggering data to be collected in support file bundle and the same error occurs one hour later at 9:00, Sonexion does not start a second data collection process related to the later error.

Contents of Support Bundles

Data related to system errors is collected in files, which are packaged together into support bundles. A support bundle is a standard UNIX-compressed file (`tar-gzip`), with files that include:

- System logs for all nodes for the 45-minute period before the error occurred
- List of all cluster nodes and information for each node:
  - Software version
  - Linux kernel and patches

  - Sonexion RPMs
  - OSTs mounted on the node
  - Power states
  - Resource states
  - Relevant processes
  - **Sysrq** data
- Current Apache/WSGI logs from the MGS/MDS
- Application state data (MySQL database dump)
- Diagnostic and performance test logs

Automatic vs. Manual Data Collection

When an error occurs, data collection and the bundling of support files is triggered automatically, and Sonexion users cannot terminate or cancel the operation. Alternately, a user can manually start data collection and create a support bundle. Unlike the automatic process, a manual data collection operation can be canceled.

An operator can also manually start a support bundle collection or import a support bundle, during which time the operator is prompted to select the nodes (defaults to “all”) and a window of time (the default is 45 minutes) for logs. After a confirmation dialog appears and is acknowledged, the process begins immediately, there is no 2-minute wait.

13.1 Collect Support Files Manually

About this task

The Sonexion user can manually start system data collection and create support files as described in this procedure. Unlike the automatic process, the manual data collection operation can be canceled.

Procedure

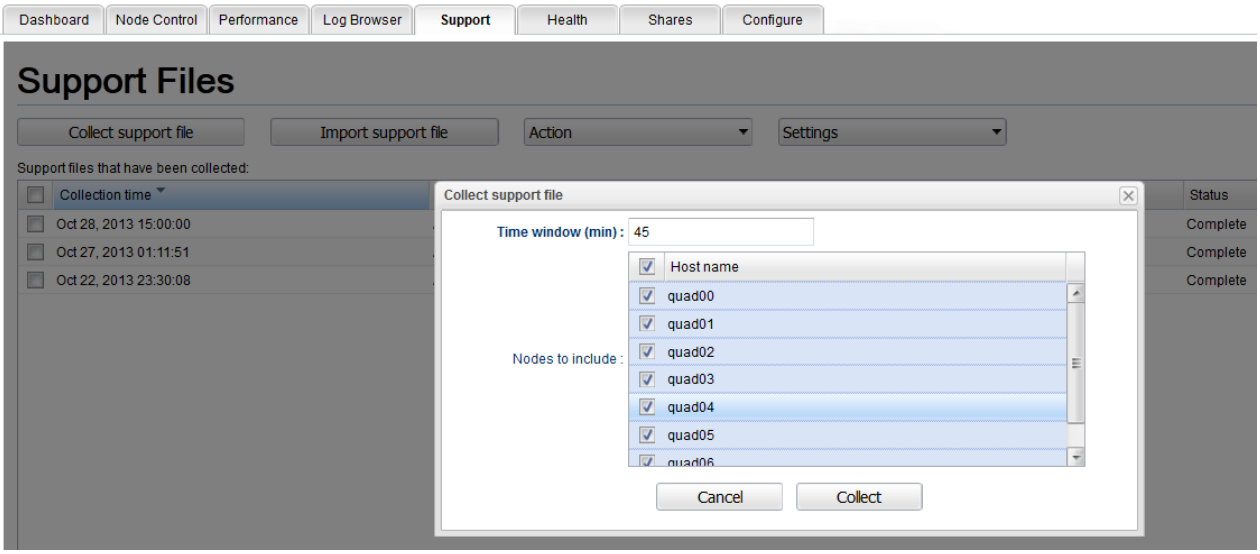
1. Click the CSSM **Support** tab.

   The **Support Files** screen displays.

2. Click the **Collect support file** button.

   The **Collect support file** dialog window opens and lists all nodes in the cluster.

Figure 7. Support Files Screen: Collect File



3. Specify the data collection parameters for the support file:

   a. Select the **time period** to look back for syslog data to be collected. The default is 45 minutes.

   b. Select the **nodes** for which data will be collected (the check box next to **Hostname** selects all nodes).

   c. Click the **Collect** button.

      The data collection process starts using the specified parameters. While collecting data, it displays in the status field “Still collecting, *xx*% complete”. When it is complete, it states “Done.” To terminate the operation at any point, click **Cancel**.

      When the operation is complete, the support file, which is a Zip file containing hundreds of different log files, is created.

13.2 Create Support Bundles Using CLI

About this task

Support bundles can be created manually CLI commands. For further information, see *support_bundle Command*.

Procedure

1. Log into the primary MGMT node via SSH.

```
$ ssh -l admin primary_MGMT_node
```

2. Change to `root` user.

```
$ sudo su -
```

3. Collect the support bundle.
   - To collect the bundle using the default 45 minute time period:

     ```
     [root@n000] cscli support_bundle -c
     ```
   - To collect the bundle using a different time period, *minutes*:

     ```
     [root@n000] cscli support_bundle -c -t minutes
     ```

   For example:

   ```
   [root@n000] cscli support_bundle -c -t 90
   Collecting support bundle: id:4, nodes:all, time-window:90 minutes
   ```

4. Check the status of the data collection. The system response ending "in progress" indicates the export cannot occur yet.

   ```
   [root@n000] cscli support_bundle -e 22
   Support_bundle: Error: Collection of support bundle with id 22 is in progress
   ```

5. Export the support bundle.

   ```
   [root@n000] cscli support_bundle -e 22
   Support_bundle with id 22 saved in file
   support_bundle_2013-08-08_10-54-07_310920.tgz
   ```

13.3 Import a Support File

About this task

Use the Import feature to upload a single support file bundle into CSSM to view its contents.

Procedure

1. Click the CSSM **Support** tab.

   The **Support Files** screen displays.

2. Click the **Import support file** button.

   The **Import support file** dialog window opens.

3. Select a support file to upload.

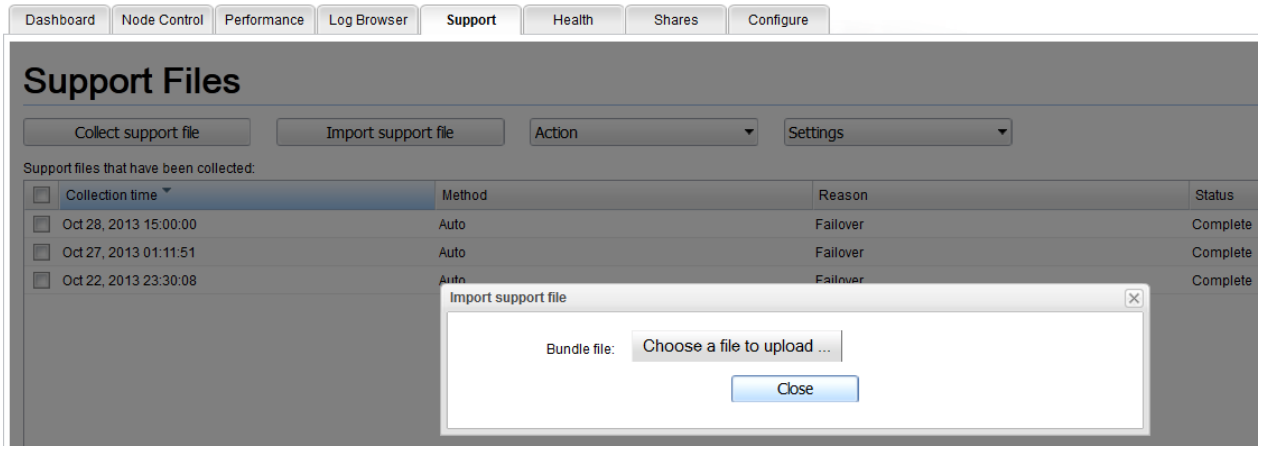
   a. Click the **Choose a file to upload ...** button.

      A list of available support files displays.

   b. Select the support file.

      The selected file opens and can be viewed.

Figure 8. Support Files Screen: Select File



13.4 Download a Support File

About this task

Use the **Download a Copy** feature to save a local copy of the selected support file.

Procedure

1. Click the CSSM **Support** tab.

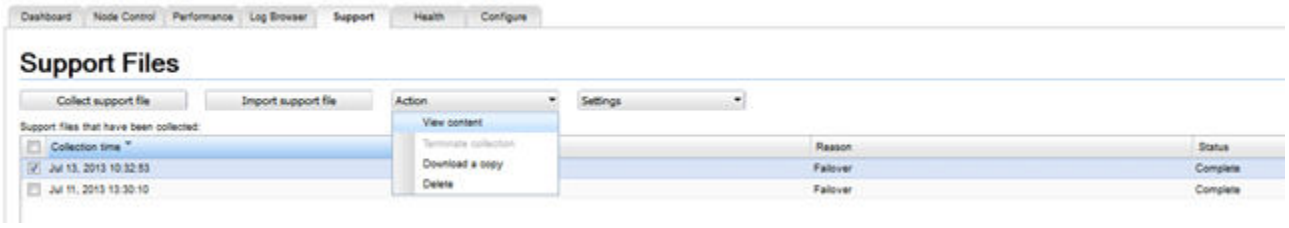
   The **Support Files** screen displays.

2. Select a support file to view.

   a. In the row containing the support file to view (in the **User actions** column) click the **Actions** button.

   b. Select **Download a copy**.

   c. Specify where to save the file on the system, or choose to open and view the file directly.



13.5 Delete a Support File

About this task

Use the **Delete File** feature on the CSSM graphic interface to delete a selected support file:

Procedure

1. Click the CSSM **Support** tab.

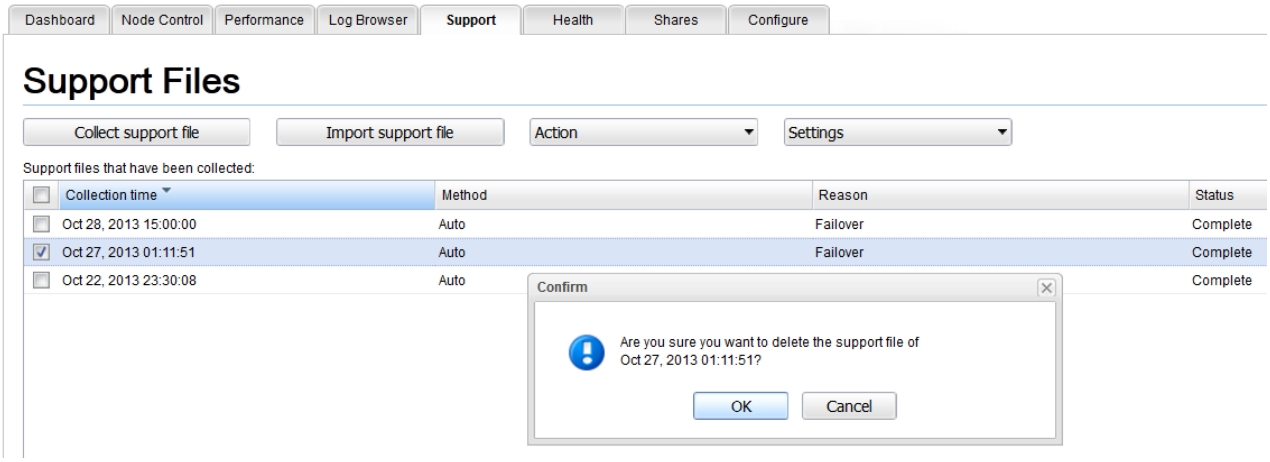
   The **Support Files** screen displays.

2. Click to select a support file.

   a. In the row containing the support file to delete (in the **User actions** column) click the **Actions** button.

   b. Select **Delete**.

   A dialog window appears prompting to confirm the deletion. Click the **Yes** button to delete the file.



13.6 View Support Files

About this task

Use the **Support** tab to view the contents of a support file created either automatically or from manual data collection methods.

To view Sonexion support files:

Procedure

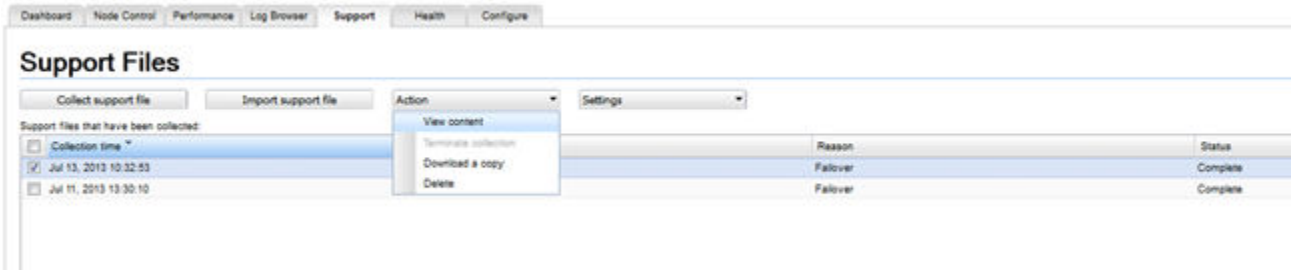
1. Click the CCMS **Support** tab.

   The **Support Files** screen is displayed, listing support files that have been collected by either the automatic or manual data collection process.

2. Select a support file to view.

   a. In the row containing the support file to view (in the **User actions** column), click the **Actions** button.

   b. Select **View Content**.



   The **Support File** content screen opens and displays the following tabs:

   - **System Logs** (default tab) - Lists all the system logs for the cluster when the support file was created.
   - **Node Information** - Lists information for all nodes in the cluster.
   - **Web Logs** - Lists all web logs for the cluster when the support file was created.
   - **Application State** - Shows data tracking the states of the management application, which is being transmitted to the support staff. Do not attempt to use this information, as it may change format from version to version.

13.7 Interpret Support Bundles

Support bundles contain two types of logs: system-wide logs that collect data for the entire system, and node-specific logs that collect data for an individual node. With the exception of lbug_syslog.csv, these log files are not intended for use by Sonexion end users, but may be valuable to Cray personnel and OEMs to better understand system states and behavior.

System-wide Logs

Log File Name	File Overview
`lbug_syslog.csv`	Contains syslog messages, in comma-separated value (CSV) format.
`logs/access.log`	Contains Apache HTTP access data.
`logs/data_tables.sql`	Contains a dump of MySQL database tables. The tables describe internal structures used to manage the cluster, the state of cluster resources, information about hardware, software, firmware, and network configuration, a FRU inventory, etc. The database dump contains all information required to recreate the system state at the time when the support bundle was created.
`logs/error.log`	Contains the Apache error log.
`logs/wsgi_access.log`	Contains records of web service calls made from the CSSM.

Node-specific Logs

Log File Name	File Overview
`nodes/`*`nodename`*`/conman.log`	Contains console data captured by CONsole MANager (`Conman`), a daemon that provides centralized access to node SOL (serial over LAN, IPMI) or real serial consoles. It also provides logging, broadcasting to several consoles or shared console sessions.
`nodes/`*`nodename`*`/crm.log`	Contains state data for the RAID and Lustre resources as seen by `Pacemaker`, an open-source, high-availability resource manager that is suitable for small and large clusters.
`nodes/`*`nodename`*`/dmesg.log`	Contains a dump of kernel messages collected from the node.
`nodes/`*`nodename`*`/fru_dump.yaml`	Contains an inventory of FRUs for the enclosure hosting the node (DDICs, PSU, fans, power supplies, etc). The dump file includes serial numbers for individual FRU equipment, firmware versions, and states such as `OK` or `Failure`).
`nodes/`*`nodename`*`/lspci.log`	Contains a list of PCI devices in a free-form text format generated by the `lspci` tool. `lspci` lists PCI devices and their characteristics. `lspci` can be run in standard or verbose (`-vvv` option) mode.
`nodes/`*`nodename`*`/mdstat.log`	Contains state data of the MDRAID arrays, i.e., content of the `/proc/mdstat` file.
`nodes/`*`nodename`*`/processes.csv`	Contains a list of processes, a snapshot of `top`, which is a standard monitoring program that reports the top consumers of CPU or memory.
`nodes/`*`nodename`*`/sgmap.log`	Contains a list of sg devices and specifies for each device the SCSI address, firmware version, and corresponding block devices.
`nodes/`*`nodename`*`/software_versions.csv`	Contains a list of all installed packages with version information (`rpm -qa` output).
`nodes/`*`nodename`*`/states.csv`	Contains miscellaneous state data, including power, memory, uptime, CPU load, and Lustre targets.

14 Troubleshooting Overview

Troubleshooting issues fall into the following groups, with links to the respective topics.

CSSM issues

- *Management Node Shutdown Corrupts Database* on page 75
- *Invalid Puppet Certificate on Diskless Node Boot-up* on page 77
- *Upgrade Error "Metadata File does not Match Checksum"* on page 79

Networking issues

- *Capture Dumps for Mellanox Host Adaptors and Switches* on page 79
- *Reseat a Problematic High-Speed Network Cable* on page 80

RAID/HA issues

- *RAIDs Assembled Incorrectly on Nodes* on page 82
- *Prepare to Recover a Failed RAID Array* on page 83
- *Recover a Failed RAID Array* on page 85
- *HA Failover on xyvinc Malfunction* on page 90

Other issues

- *Non-Responsive Server* on page 89
- *conman to GEM Causes "Port In Use" Message* on page 89
- *Performance Tab Appears Blank* on page 92
- *Isolate a Slow Disk*

14.1 Management Node Shutdown Corrupts Database

About this task

Affected Releases	1.3.1, 1.4.0, 1.5.0
Affected Products	Sonexion 900, 1300, 1600, 2000
Root Access Required	Yes
Service Interruption Level	Interrupt

If the management node hosting the MySQL server is shut down uncleanly, including by a `stonith` command, the `t0db` database (named `filesystem_`*`filesystem_name`*) on the management nodes may become corrupt.

This can manifest in several ways, including out-of-date information in the performance tab and problems assessing the management database `t0db`. There will usually be errors in the `/var/log/mysqld.log` file indicating the corrupt tables:

```
130220 8:20:28 [ERROR] /usr/libexec/mysqld: Table './filesystem_snx11003/MDS_OPS_DATA' is marked as crashed and should be repaired
130220 8:20:43 [ERROR] /usr/libexec/mysqld: Table './filesystem_snx11003/MDS_OPS_DATA' is marked as crashed and should be repaired
130220 8:20:58 [ERROR] /usr/libexec/mysqld: Table './filesystem_snx11003/MDS_OPS_DATA' is marked as crashed and should berepaired
```

Workaround: To repair the corrupted tables, execute the following procedure. In order to assist in troubleshooting if the procedure fails, please use the `script` command to save the output.

> **IMPORTANT:** For very large tables, repair operations can create temporary files that are larger than the available file system space. It is recommended that the available space be monitored during this procedure using the `df -h` command.

Procedure

1. Log into the primary MGMT node via SSH.

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

2. Sudo to `root`.

   ```
   [admin@n000]$ sudo su -
   ```

3. Collect program output using `script`.

   ```
   [root@n000]# script /tmp/t0db_repair_log
   ```

4. Shut down the `mysql` service.

   ```
   [root@n000]# service mysqld stop
   ```

5. Check all tables in `t0db`.

   ```
   [root@n000]# mysqlcheck filesystem_filesystem_name
   filesystem_fs1.EVENT_DATA OK
   filesystem_fs1.EVENT_INFO OK
   filesystem_fs1.FILESYSTEM_AGGREGATE_DAY
   error : Size of indexfile is: 15360 Should be: 18432
   error : Corrupt
   filesystem_fs1.FILESYSTEM_AGGREGATE_HOUR
   error : Size of indexfile is: 224256 Should be: 251600
   error : Corrupt
   filesystem_fs1.FILESYSTEM_AGGREGATE_MONTH OK
   filesystem_fs1.FILESYSTEM_AGGREGATE_WEEK OK
   filesystem_fs1.FILESYSTEM_AGGREGATE_YEAR OK
   filesystem_fs1.FILESYSTEM_INFO OK
   ```

   Additional output omitted.

6. Repair all tables.

   ```
   [root@n000]# mysqlcheck -s -r filesystem_filesystem_name
   ```

7. Verify that the repair worked and all tables are `OK`.

```
[root@n000]# mysqlcheck filesystem_filesystem_name
filesystem_fs1.EVENT_DATA OK
filesystem_fs1.EVENT_INFO OK
filesystem_fs1.FILESYSTEM_AGGREGATE_DAY OK
filesystem_fs1.FILESYSTEM_AGGREGATE_HOUR OK
filesystem_fs1.FILESYSTEM_AGGREGATE_MONTH OK
filesystem_fs1.FILESYSTEM_AGGREGATE_WEEK OK
filesystem_fs1.FILESYSTEM_AGGREGATE_YEAR OK
filesystem_fs1.FILESYSTEM_INFO OK
```

Additional output omitted.

8. Restart `mysqld`.

```
[root@n000]# service mysqld start
```

9. Exit from the `script` data collection. This will save the output to the specified log file.

```
[root@n000]# exit
```

If problems are encountered during this procedure or persist after this procedure is finished, contact Cray support.

14.2 Invalid Puppet Certificate on Diskless Node Boot-up

About this task

Affected Releases	1.3.1
Affected Products	Sonexion 1300, Sonexion 1600
Root Access Required	Yes
Service Interruption Level	Live

Problem description: If attempts to log into an OSS node using known-good credentials fail, the node is probably experiencing puppet connection problems. Use this procedure to clear up the puppet configuration on the node:

Workaround:

Procedure

1. Log into the primary MGMT node via SSH:

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

2. Change to root user:

```
[admin@n000]$ sudo su -
```

3. Revoke the certificate for the OSS node and remove the certificate files from the management node:

```
[root@n000]# puppetca --clean OSS_nodename
```

4. Log into the OSS node via SSH:

```
[root@n000]# ssh –l admin OSS_nodename
```

   - If the attempt to log in to the node succeeds, go to step 5.
   - If the attempt to log in to the node fails:

     ```
     [root@n000]# find /var/lib/puppet/ssl_persistent -nameoss_nodename.* -delete ii
     ```

     Reboot the OSS node (physically or using `conman`), wait until the node is accessible via SSH, and then go to Step 9.

5. Change to root user:

```
[oss]$ sudo su -
```

6. Remove the SSL certificate and private key from the OSS node:

```
[oss]# rm –rf /var/lib/puppet/ssl/*
```

7. Run the puppet client, to regenerate the private key and request a new signed certificate from the management node:

```
[oss]# puppetd –tv
```

8. Exit back out to the management node:

```
[oss]# exit
```

9. Populate the persistent storage with the node's certificate and private key:

```
[root@n000]# rsync -zaHv --numeric-idshostname:/var/lib/puppet/ssl/ /var/lib/
puppet/ssl_persistent
```

10. The certificate and associated private key files are regular files, like any other. To verify that the persistent directory has the right files, run:

```
for i in $(nodeattr -s diskless); do
 diff -q <(ssh $i cat /var/lib/puppet/ssl/certs/$i.pem)  /var/lib/puppet/
ssl_persistent/certs/$i.pem 2> /dev/null || echo cert for $i is not correct in
persistent storage
done
```

    This checks that the certificate file for each diskless node is the same in that node's `/var/lib/puppet/ssl/certs` directory and the `/var/lib/puppet/ssl_persistent/certs` directory

11. Verify that the current puppet certificate is valid:

```
[root@n000]# puppetd -tv
```

14.3 Upgrade Error "Metadata File does not Match Checksum"

About this task

Affected Releases	1.3.1, 1.4.0, 1.5.0
Affected Products	Sonexion 900, 1300, 1600, 2000
Root Access Required	Yes
Service Interruption Level	Not Applicable

There is a bug in the implementation of the `prepare_updates` routine that makes packages cache unusable.

Workaround: Run the following commands.

Procedure

1. Log into the primary MGMT node via SSH.

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

2. Log in as `root` user.

```
[admin@n000]$ sudo su -
```

3. Execute the following commands.

```
[MGMT]# rm -rf /var/cache/yum/*
[MGMT]# rm -rf /var/cache/yum_image/*
[MGMT]# createrepo /opt/xyratex/release/new_version
```

Where *new_version* is the version to which the software is being upgraded.

14.4 Capture Dumps for Mellanox Host Adaptors and Switches

About this task

Affected Releases	1.3.1, 1.4.0, 1.5.0
Affected Products	Sonexion 900, 1300, 1600, 2000
Root Access	Yes

Service Interruption Level	Live

This procedure is used to capture register dumps from Mellanox Host Channel Adaptors (HCAs) and switches. The information in these dumps is the only effective method of debugging a failed switch or Mellanox HCA problems. Once the dump is captured it can then be attached to the JIRA ticket.

Procedure

1. If debugging an HCA failure, log into the node with the failed HCA; if debugging a switch, log into a node that is still connected to the switch.

   ```
   [Client]$ ssh NODE
   ```

   Where *NODE* is either the node with the failed HCA or the node that is still connected to the switch.

2. Sudo to `root`.

   ```
   [NODE]$ sudo su -
   ```

3. Start MST.

   ```
   [NODE]# mst start
   ```

4. Discover the network.

   ```
   [NODE]# mst ib add
   ```

   This populates `/dev/mst/` with IB devices, including the faulted switch or HCA.

5. Capture the dump.

   ```
   [NODE]# mstdump /dev/mst/device_name > mstdump.out
   ```

   Where *device_name* is the name of the faulted switch or HCA. Switch names start with `SW` and HCAs with `CA`.

6. Attach the `mstdump.out` file to the relevant JIRA ticket.

14.5 Reseat a Problematic High-Speed Network Cable

About this task

Affected Releases	1.3.1, 1.4.0, 1.5.0
Affected Products	Sonexion 1300, Sonexion 1600
Root Access Required	Yes
Service Interruption Level	Interrupt

Problem description: On occasion, a node may lose its connection to the InfiniBand fabric. This problem can occur both on systems running with InfiniBand or with 40-gigabit Ethernet networks.

Loss of connectivity can be caused by an incorrectly seated network cable (leads in the cable/switch not making physical contact), by dust on the leads, or because the cable itself has gone bad. Mellanox cables can only be plugged and unplugged a finite number of times before reaching their lifetime maximum.

On systems using InfiniBand, a bad InfiniBand fabric connection can be diagnosed using the ibcheckerrors command. This command must return cleanly (no new errors reported) for the high speed network to be considered functional.

Workaround: To reseat a cable, complete the following procedure:

Procedure

1. Log into the primary MGMT node via SSH. Run:

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

2. Change to root user:

   ```
   [admin@n000]$ sudo su -
   ```

3. Unmount Lustre:

   ```
   [root@n000]# /opt/xyratex/bin/cscli unmount -c cluster_name -f filesystem_name
   ```

4. Inspect whether the LED switch for the cable is on.

5. Disable HA's InfiniBand fabric querying:

   ```
   [root@n000]# ssh nodename stop_ibstat
   ```

6. Determine the physical location of the cable to be reseated and unplug it.

7. Inspect the cable head for any signs of corrosion or other damage.

8. Blow compressed air over the cable head to remove any dust.

9. Before the cable is re-plugged, verify failover on the node that was unplugged:

   ```
   [root@n000]# crm_mon –1"
   ```

10. Re-plug the cable.

11. If the LED switch for that cable was previously on, verify that it comes back on after the cable has been re-plugged. Depending on how long is required for discovery, this may take up to a minute.

12. Enable HA's InfiniBand fabric querying:

    ```
    [root@n000]# ssh nodename start_ibstat
    ```

13. After reseating the cable, log into the affected node.

14. Replace the cable if it is damaged, or if there are multiple reseats, do not fix the problem.

15. Mount Lustre:

```
[root@n000]# /opt/xyratex/bin/cscli mount -c cluster_name -f filesystem_name
```

14.6 RAIDs Assembled Incorrectly on Nodes

Affected Releases	1.3.1, 1.4.0, 1.5.0
Affected Products	Sonexion 900, 1300, 1600, 2000
Root Access	Yes
Service Interruption Level	Not Applicable

When an MDRAID device fails (for example, as a result of an expander reset temporarily removing several disks) the STONITH resource detects this change within its monitoring interval (10 minutes) and attempts to reassemble the MDRAID device on its OSS node. If the MDRAID device does not rebuild successfully, then the reassembly attempt times out after three minutes and the `STONITH` resource records a `failed actions` message for the OSS node.

The `STONITH` resource then tries to assemble the MDRAID device on the OSS node's HA partner node. If the rebuild is not successful on the HA partner node, then the reassembly attempt times out after three minutes and the STONITH resource records another “failed actions” message for the HA partner node.

After these failed attempts, the `STONITH` resource no longer tries to assemble the MDRAID resource but leaves the first three resources in the group assembled.

> **IMPORTANT:** GridRAID systems running release 1.4 requires SU 11 to be installed before attempting this procedure.

Workaround: Use the following steps to manually recover the RAID. This procedure assumes that onsite personnel have identified the OSS node(s) that control the failed RAID array(s). Please note that:

- Even numbered OSS nodes natively control even numbered MDRAID devices (`md0`, `md2`, `md4`, and md6).
- Odd numbered OSS nodes natively control odd numbered MDRAID devices (`md1`, `md3`, `md5`, and `md7`).
- If a native OSS node is in a failover state, control of the MDRAID devices that it natively controls will migrate to its HA partner node. It is possible to recover the MDRAID device using either of these HA partner OSS nodes.

In the Sonexion solution, an enclosure and two controllers are bundled in the modular SSU. Each controller hosts one OSS node; there are two OSS nodes per SSU. Within an SSU, the OSS nodes are organized in a High Availability (HA) pair with sequential numbers (for example, `node04` / `node05` or `snx11000n004` / `snx11000n005`). If an OSS node goes down because its controller fails, its resources migrate to the HA partner/OSS node in the other controller.

The 84 disk drives in a Sonexion SSU are configured as:

- Eight OSTs, each a RAID6 array consisting of eight data disks and two parity disks

- Two SSDs partitioned to create multiple independent RAID1 slices, used for MDRAID write intent bitmaps and external OST/ldiskfs file system journals
- Two hot spares

The virtual drives defined by the RAID6 arrays are referred to as MDRAID devices, numbered sequentially from 0 through 7, for example, `md0`. Within the STONITH resource, there are resources defined for each MDRAID device, used by the STONITH resource to control the MDRAID device. For example, `snx11000n004_md0-raid` is the resource that controls the MDRAID device `md0`.

14.6.1 Prepare to Recover a Failed RAID Array

About this task

Affected Releases	1.3.1, 1.4.0, 1.5.0
Affected Products	Sonexion 900, 1300, 1600, 2000
Root Access	Yes
Service Interruption Level	Not Applicable

After an MDRAID failure (see *RAIDs Assembled Incorrectly on Nodes* on page 82), recover a failed array using the following steps and continue to *Recover a Failed RAID Array* on page 85.

Procedure

1. Log in to the primary MGMT node via SSH.

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

2. Sudo to `root`.

   ```
   [admin@n000]$ sudo su -
   ```

3. Determine if either of the OSS nodes that control the failed MDRAID device are offline. If so, power on the downed OSS node(s). If both OSS nodes are down, repeat this step on the HA partner node.

   For example:

   ```
   [root@n000]# pm -1 OSS_nodename
   ```

4. Wait several minutes, and then log into the previously downed OSS node via SSH to verify that it is back online. If both OSS nodes were down, repeat this step on the HA partner node.

   ```
   [root@n000]# ssh OSS_nodename
   ```

5. Log into the OSS node that should natively (i.e., even/odd MDs to even/odd OSSs) control the MDRAID with a problem, via SSH.

   ```
   [OSS node]# ssh OSS_nodename
   ```

6. Check `/var/log/messages` to see if there was a report of an invalid `mdraid` superblock. If the following error message is present, it is necessary to zero out the device superblock.

```
md: sdXX does not have a valid v1.2 superblock, not importing!
```

```
[OSS node]# mdadm –zero-superblock /dev/ sdXX
```

7. Use the `crm_mon` utility to verify that a failed actions message was recorded for the failed MDRAID device. Also verify that the first three resources in the failed MDRAID device's resource group have failed over to the HA partner node.

   IMPORTANT: When reviewing the `crm_mon` output, note the failed MDRAID device's resource group name. This information is needed when performing the procedure to recover the failed RAID array.

```
[OSS node]# crm_mon -1r
============
Last updated: Wed Jan 23 17:30:10 2013
Last change: Wed Jan 23 17:16:30 2013 via cibadmin on snx11000n005
Stack: Heartbeat
Current DC: snx11000n004 (8ab209a5-874a-404d-af1c-1afa84cc18a9) - partition with quorum
Version: 1.1.6.1-2.el6-0c7312c689715e096b716419e2ebc12b57962052
2 Nodes configured, unknown expected votes
55 Resources configured.
Online: [ snx11000n004 snx11000n005 ]
snx11000n004-stonith (stonith:external/gem_stonith): Started snx11000n004
snx11000n005-stonith (stonith:external/gem_stonith): Started snx11000n005
snx11000n004_mdadm_conf_regenerate (ocf::heartbeat:mdadm_conf_regenerate):Started snx11000n004
snx11000n005_mdadm_conf_regenerate(ocf::heartbeat:mdadm_conf_regenerate):Started snx11000n005
baton (ocf::heartbeat:baton): Started snx11000n005
snx11000n004_ibstat (ocf::heartbeat:ibstat): Started snx11000n004
snx11000n005_ibstat (ocf::heartbeat:ibstat): Started snx11000n005
Resource Group: snx11000n004_md0-group
snx11000n004_md0-wibr (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md0-jnlr (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md0-wibs (ocf::heartbeat:XYMNTR): Started snx11000n004
snx11000n004_md0-raid (ocf::heartbeat:XYRAID): Stopped
snx11000n004_md0-fsys (ocf::heartbeat:XYMNTR): Stopped
snx11000n004_md0-stop (ocf::heartbeat:XYSTOP): Stopped
Resource Group: snx11000n004_md1-group
snx11000n004_md1-wibr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md1-jnlr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md1-wibs (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md1-raid (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md1-fsys (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md1-stop (ocf::heartbeat:XYSTOP): Started snx11000n005
Resource Group: snx11000n004_md2-group
snx11000n004_md2-wibr (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md2-jnlr (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md2-wibs (ocf::heartbeat:XYMNTR): Started snx11000n004
snx11000n004_md2-raid (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md2-fsys (ocf::heartbeat:XYMNTR): Started snx11000n004
snx11000n004_md2-stop (ocf::heartbeat:XYSTOP): Started snx11000n004
Resource Group: snx11000n004_md3-group
snx11000n004_md3-wibr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md3-jnlr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md3-wibs (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md3-raid (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md3-fsys (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md3-stop (ocf::heartbeat:XYSTOP): Started snx11000n005
Resource Group: snx11000n004_md4-group
snx11000n004_md4-wibr (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md4-jnlr (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md4-wibs (ocf::heartbeat:XYMNTR): Started snx11000n004
snx11000n004_md4-raid (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md4-fsys (ocf::heartbeat:XYMNTR): Started snx11000n004
snx11000n004_md4-stop (ocf::heartbeat:XYSTOP): Started snx11000n004
Resource Group: snx11000n004_md5-group
snx11000n004_md5-wibr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md5-jnlr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md5-wibs (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md5-raid (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md5-fsys (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md5-stop (ocf::heartbeat:XYSTOP): Started snx11000n005
Resource Group: snx11000n004_md6-group
snx11000n004_md6-wibr (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md6-jnlr (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md6-wibs (ocf::heartbeat:XYMNTR): Started snx11000n004
snx11000n004_md6-raid (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md6-fsys (ocf::heartbeat:XYMNTR): Started snx11000n004
```

```
snx11000n004_md6-stop (ocf::heartbeat:XYSTOP): Started snx11000n004
Resource Group: snx11000n004_md7-group
snx11000n004_md7-wibr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md7-jnlr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md7-wibs (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md7-raid (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md7-fsys (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md7-stop (ocf::heartbeat:XYSTOP): Started snx11000n005
Failed actions:
snx11000n004_md0-raid_start_0 (node=snx11000n005, call=134, rc=-2, status=Timed Out): unknown exec error
snx11000n004_md0-raid_start_0 (node=snx11000n004, call=134, rc=-2, status=Timed Out): unknown exec error
```

- If the RAID fails to assemble and generates messages similar to the following, then the RAID recovery procedure has failed, go to step *8* on page 85.

  ```
  mdadm: ignoring /dev/disk/by-id/wwn-0x5000cca01b3d4224 as it reports /dev/disk/by-id/wwn-0x5000cca01b3cf13c as failed
  mdadm: ignoring /dev/disk/by-id/wwn-0x5000cca01b3d7e24 as it reports /dev/disk/by-id/wwn-0x5000cca01b3cf13c as failed
  mdadm: ignoring /dev/disk/by-id/wwn-0x5000cca01b3d6080 as it reports /dev/disk/by-id/wwn-0x5000cca01b3cf13c as failed
  mdadm: ignoring /dev/disk/by-id/wwn-0x5000cca01c375f04 as it reports /dev/disk/by-id/wwn-0x5000cca01b3cf13c as failed
  ```

- If the error messages do not appear, go to step *9* on page 85.

8. Abort this procedure and contact Cray support. Cray support will require the `mdraid` superblock data to be collected in order to debug the problem. In release 1.4.0 and later, this information can be collected using the `mdraid-analyze` script located at `/usr/sbin/`. In release 1.3.1 or earlier, use the `collect_superblock.sh` script to collect this data.

a. Download the `collect_superblock.sh` script from XIC to `/tmp` on the OSS node with the failed raid.

b. Exit the OSS node.

```
[OSS node]# exit
```

c. Run the script.

```
[root@n000]# pdsh -g OSS_node /tmp/collect_superblock.sh /var/lib/
mdraidscripts/mdadm.conf
```

9. If the first three resources in the failed MDRAID device's resource group have failed over to the HA partner node, exit the OSS node and log into the HA partner node via SSH.

```
[OSS node]# exit
[root@n000]# ssh HA_partner_nodename
```

The procedure to prepare for recovering a failed RAID array is now complete. Proceed to *Recover a Failed RAID Array* on page 85.

14.6.2 Recover a Failed RAID Array

Prerequisites

Prepare to Recover a Failed RAID Array on page 83 must be completed prior to beginning this procedure to gather required information.

About this task

Affected Releases	1.3.1, 1.4.0, 1.5.0
Affected Products	Sonexion 900, 1300, 1600, 2000
Root Access	Yes

Service Interruption Level	Not Applicable

This procedure describes how to force assemble an MDRAID device to recover a failed RAID array.

CAUTION: Assembling a RAID array with the `--force` argument can result in data loss or data corruption. This procedure should only be used as a last resort.

Procedure

1. Log into the previously downed OSS node via SSH.

   ```
   [root@n000]# ssh OSS_nodename
   ```

2. Stop the resource group containing the failed MDRAID device.

   ```
   [OSS node]# stop_xyraid resource_group_name
   ```

   Where *resource_group_name* is the resource group name identified in the prerequisite procedure *Prepare to Recover a Failed RAID Array* on page 83.

   For example:

   ```
   [OSS node]# stop_xyraid snx11000n004_md0-group
   ```

3. Allow the resource group resources to be started outside of the STONITH resource.

   ```
   [OSS node]# unmanage_xyraid resource_group_name
   ```

   For example:

   ```
   [OSS node]# unmanage_xyraid snx11000n004_md0-group
   ```

4. Clean the resource group to remove the failed actions.

   ```
   [OSS node]# clean_xyraid resource_group_name
   ```

   For example:

   ```
   [OSS node]# clean_xyraid snx11000n004_md0-group
   Cleaning up snx11000n004_md0-wibr on snx11000n004
   Cleaning up snx11000n004_md0-wibr on snx11000n005
   Cleaning up snx11000n004_md0-jnlr on snx11000n004
   Cleaning up snx11000n004_md0-jnlr on snx11000n005
   Cleaning up snx11000n004_md0-wibs on snx11000n004
   Cleaning up snx11000n004_md0-wibs on snx11000n005
   Cleaning up snx11000n004_md0-raid on snx11000n004
   Cleaning up snx11000n004_md0-raid on snx11000n005
   Cleaning up snx11000n004_md0-fsys on snx11000n004
   Cleaning up snx11000n004_md0-fsys on snx11000n005
   Cleaning up snx11000n004_md0-stop on snx11000n004
   Cleaning up snx11000n004_md0-stop on snx11000n005
   Waiting for 13 replies from the CRMd.............. OK
   ```

5. If during *Prepare to Recover a Failed RAID Array* on page 83 it was determined that the first three resources in the failed MDRAID device's resource group have failed over to the HA partner node, complete this step, otherwise procede to the next step.

a. SSH to the OSS node that natively controls the MDRAID device.

```
[OSS node]# ssh OSS_nodename
```

b. Fail back resources to the OSS node.

```
[OSS node]# failback_xyraid
```

6. Determine if the `--force` argument is necessary to assemble the MDRAID device.

```
[OSS node]# mdraid-activate -d resource_group_name
```

This example shows an unsuccessful attempt to assemble the MDRAID device without the `--force` argument:

```
[root@snx11000n004 ~]# mdraid-activate -d snx11000n004_md0-group
mdadm: /dev/md/snx11000n004:md128 has been started with 2 drives.
mdadm: /dev/md/snx11000n004:md129 has been started with 2 drives.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdraid-activate 358: unable to assemble snx11000n004:md0
```

NOTE: If the assembly attempt was successful, proceed to step *9* on page 88.

7. Assemble the MDRAID device using the `--force` argument.

```
[OSS node]# mdraid-activate -f i_am_sure_i_want_to_do_this,exit -d
resource_group_name
```

This example shows a successful forced assembly of an MDRAID device on 1.3.1:

```
[root@snx11000n004 ~]# mdraid-activate -f i_am_sure_i_want_to_do_this,exit -d snx11000n004_md0-group
mdadm: /dev/md/snx11000n004:md129 has been started with 2 drives.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdadm: failed to RUN_ARRAY /dev/md/snx11000n004:md0: Input/output error
mdadm: Not enough devices to start the array.
/usr/lib/ocf/lib/heartbeat/xrtx-ocf-shellfuncs: line 867: ocf_log: command not found
mdadm: forcing event count in /dev/disk/by-id/wwn-0x5000c500212d3f2b(3) from 19 upto 39
mdadm: clearing FAULTY flag for device 7 in /dev/md/snx11000n004:md0 for /dev/disk/by-id/wwn-0x5000c500212d3f2b
mdadm: Marking array /dev/md/snx11000n004:md0 as 'clean'
mdadm: /dev/md/snx11000n004:md0 has been started with 8 drives (out of 10).
assembled snx11000n004:md0 in 1 tries
```

IMPORTANT: If the MDRAID device failed to assemble, stop and contact Cray Support.

8. Check the MDRAID device.

```
[OSS node]# e2fsck -fp /dev/MDRAID_device
```

9. Execute the `e2fsck` command on the MDRAID device.

   For example:

```
[root@snx11000n004 ~]# e2fsck -fp /dev/md0
testfs-OST0000: recovering journal
testfs-OST0000: 86/7879680 files (2.3% non-contiguous), 509811/31457280 blocks
```

10. Stop the MDRAID device.

```
[OSS node]# mdraid-deactivate resource_group_name
```

11. Manage the MDRAID device's resource group.

```
[OSS node]# manage-xyraid resource_group_name
```

12. Start the MDRAID device's resource group.

```
[OSS node]# start-xyraid -nowait resource_group_name
```

13. Verify that the MDRAID device's resource group started correctly. This can take several minutes.

```
[OSS node]# crm_mon -1
============ 23 18:00:58 2013
Last change: Wed Jan 23 18:00:18 2013 via cibadmin on snx11000n004
Stack: Heartbeat
Current DC: snx11000n005 (8ab209a5-874a-snx11000n004d-af1c-1afa84cc18a9) -
partition with quorum
Version: 1.1.6.1-2.el6-0c7312c689715e096b716419e2ebc12b57962052
2 Nodes configured, unknown expected votes
55 Resources configured.
Online: [ snx11000n004 snx11000n005 ]
snx11000n004-stonith (stonith:external/gem_stonith): Started snx11000n004
snx11000n005-stonith (stonith:external/gem_stonith): Started snx11000n005
snx11000n004_mdadm_conf_regenerate
(ocf::heartbeat:mdadm_conf_regenerate):Started snx11000n004
snx11000n005_mdadm_conf_regenerate
(ocf::heartbeat:mdadm_conf_regenerate):Started snx11000n005
baton (ocf::heartbeat:baton): Started snx11000n005
snx11000n004_ibstat (ocf::heartbeat:ibstat): Started snx11000n004
snx11000n005_ibstat (ocf::heartbeat:ibstat): Started snx11000n005
Resource Group: snx11000n004_md0-group
snx11000n004_md0-wibr (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md0-jnlr (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md0-wibs (ocf::heartbeat:XYMNTR): Started snx11000n004
snx11000n004_md0-raid (ocf::heartbeat:XYRAID): Started snx11000n004
snx11000n004_md0-fsys (ocf::heartbeat:XYMNTR): Started snx11000n004
snx11000n004_md0-stop (ocf::heartbeat:XYSTOP): Started snx11000n004
Resource Group: snx11000n004_md1-group
snx11000n004_md1-wibr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md1-jnlr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md1-wibs (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md1-raid (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md1-fsys (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md1-stop (ocf::heartbeat:XYSTOP): Started snx11000n005
```

```
...
Resource Group: snx11000n004_md7-group
snx11000n004_md7-wibr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md7-jnlr (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md7-wibs (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md7-raid (ocf::heartbeat:XYRAID): Started snx11000n005
snx11000n004_md7-fsys (ocf::heartbeat:XYMNTR): Started snx11000n005
snx11000n004_md7-stop (ocf::heartbeat:XYSTOP): Started snx11000n005
```

14.7 Non-Responsive Server

About this task

Affected Releases	1.3.1, 1.4.0, 1.5.0
Affected Products	Sonexion 1300, 1600, 2000
Root Access	Yes
Service Interruption Level	Failover

A server node failed and is not responding to power manage commands to reboot. Revive the node, using `conman`, and run the `ipmi` command to start the node.

Procedure

1. Log into the primary MGMT node via SSH:

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

2. Log in as `root` user:

   ```
   [admin@n000]$ sudo su -
   ```

3. Log in via console manager:

   ```
   [MGMT]# conman nodename-gem
   ```

4. Start the node:

   ```
   [GEM]# -ipmi_power 4
   ```

14.8 `conman` to GEM Causes "Port In Use" Message

About this task

Affected Releases	1.3.1, 1.4.0, 1.5.0

Affected Products	Sonexion 1300, 1600, 2000
Root Access	Yes
Service Interruption Level	Live/Interrupt

The conman daemon automatically connects to all the x86 CPU targets in its configuration file using BMC (Baseboard Management Controller) SOL (serial on LAN), and then uses these SOL connections to open UART connections to GEM. These SOL streams for x86 and GEM are always running in the background.

When the user opens a terminal to a node using conman, a terminal front end is being attached to the background SOL stream. If the conman daemon is already running on one node and the daemon is started on the other, the open SOL stream is disrupted on the first node.

This disruption actually results in the second node forcing the SOL connection on the first node to close. But this forced closure occurs before the GEM UART resources can be freed, resulting in the "Port Already In Use" message.

SOL does not support multiple connections; therefore, if multiple sessions are started to x86 and GEM on SATI, then the conman service will fail, cause an error, or crash.

Use the following procedure if the "Port already in use" message is encountered.

Procedure

1. Load conman with either the -f or -j options.

   The -f flag specifies that write-access to the console should be forced, thereby stealing the console away from existing clients having write privileges. The original clients are informed by conman of who perpetrated the theft as their connections are terminated.

   The -j flag specifies that write-access to the console should be joined, thereby sharing the console with existing clients having write privileges. The original clients are informed by conman that a new client has been granted write privileges.

2. If either of the above options does not allow conman to attach to GEM, then restart the ser2net service on the node in question.

```
[root@n000]# # ssh node_name
[OSS node]# service ser2net restart
```

14.9 HA Failover on xyvinc Malfunction

About this task

Affected Releases	1.3.1, 1.4.0, 1.5.0
Affected Products	Sonexion 900, 1300, 1600, 2000
Root Access	Yes
Service Interruption Level	Not Applicable

A new OSS controller failed to come online after replacement. The `restore_ha_config.sh` script did not complete because the High-Availability (HA) system has multiple nodes defined with the same hostname.

Follow this procedure to manually remove the extraneous nodes and rerun the `restore_ha_config.sh` script.

Procedure

1. Log in to the new controller.

   ```
   [new_controller]$ ssh –l admin new_controller
   ```

2. Log in as `root` user.

   ```
   [new_controller]$ sudo su -
   ```

3. Check to see if the HA system has multiple nodes defined with the same hostname.

   ```
   [new_controller]# crm_mon -1r
   ```

   If the output has more than one entry with the new hostname and no resources are started on the new node (as shown below), then it is necessary to remove the node that does not correspond to the working node.

   ```
   Online: [ snx11000n005 snx11000n004 ]
   Offline: [ snx11000n004 ]
   ```

4. Retrieve the `uuid` of the new controller.

   ```
   [new_controller]# crm_uuid
   ```

   For example:

   ```
   [root@snx11000n004]# crm_uuid
   e596dc72-84a9-4a61-8873-5cd728bea632
   ```

5. Edit the CIB and remove the entries for this hostname that **do not** match this `uuid`.

   ```
   [new_controller]# crm configure edit
   ```

   This command puts the user in `vi` mode, editing the CIB. At the very top of the CIB are the node definitions, for example:

   ```
   node $id="3334b519-897c-4438-bd65-2750e4094a3a" snx11000n005
   node $id="e596dc72-84a9-4a61-8873-5cd728bea632" snx11000n004
   node $id="e534b532-987e-54a3-8873-4ab8798ab1ba" snx11000n004
   ```

   a. Remove the entire line corresponding to the node that has the same hostname, but **does not** have the uuid (the second field) of the working node. In this example, because the `uuid` of the working `snx11000n004` is `e596dc72-84a9-4a61-8873-5cd728bea632`, the third line must be removed.

   b. Exit the edit session with the sequence **`:wq`**

6. Check to see if the new controller still has multiple nodes defined with the same hostname; it should now only show two online nodes.

   ```
   [new_controller]# crm_mon -1
   ```

7. Stop the heartbeat service on the new controller.

```
[new_controller]# service heartbeat stop
```

8. Rerun `restore_ha_config.sh`, which should now complete successfully.

```
[new_controller]# /opt/xyratex/bin/restore_ha_config.sh
```

14.10 Performance Tab Appears Blank

Affected Releases	1.3.1
Affected Products	Sonexion 1300, Sonexion 1600
Root Access	Yes
Service Interruption Level	Live

Problem: System I/O performance is displayed in the CSSM **Performance** tab. This tab may become blank, even if the system is servicing I/O requests.

Corruption of the **MyISAM** database is not the only problem that can cause a blank **Performance** tab. The illustration below shows the **Performance** tab's location.

Figure 9. Performance Tab

Two known problems can cause this:

- Cerebrod is binding to the wrong Ethernet interface.
- LMT has caused the MyISAM database to become corrupt.

Cause: Cerebrod is binding to the wrong Ethernet interface

This problem is addressed by system upgrade 1.x-TRT-1. (Not to be confused with system upgrade 1.2.0-TRT-1). (Refer to the 1.x-TRT-1 document, Sonexion system upgrade_1.x-TRT-1 (V1.2a).pdf, and the cerebro.conf.patch documentation located on the Cray Information Center.) Use the Legacy system upgrade Checker to verify that system upgrade 1.x-TRT-1 is installed. Go to the Cray Information Center to access the Hotfix Checker.

Workaround: If system upgrade 1.x-TRT-1 is not installed, install it.

Cause: LMT has caused the MyISAM database to become corrupt.

The `myisamchk` command is used to check the integrity of the MyISAM database. The precise command depends on whether or not the manual failover system upgrade has been installed.

If the manual failover script has been installed, use this command to verify that the MyISAM database is not corrupt. Run:

```
[root@n000]# myisamchk --silent /mnt/mgmt/var/lib/mysql/filesystem_*/*.MYI
```

Workaround: If the database is found to be corrupt, the alternatives are attempting either to recover or to recreate the database.

Use these steps to recover an LMT database:

```
[root@n000]# /etc/profile.d/lustre-fs-name.sh 2>/dev/null || FSNAME=( $(mysql t0db -
Nse "SELECT fs_name FROM filesystem") )
[MGMT]# for fs in "${FSNAME[@]}"; do
    mysql -e "repair table filesystem_${fs}.MDS_OPS_DATA"
    mysql -e "repair table filesystem_${fs}.OST_OPS_DATA"
    mysql -e "repair table filesystem_${fs}.OST_DATA"
done
```

If the manual failover script has not been installed, use this command:

```
[root@n000]# myisamchk --silent /var/lib/mysql/filesystem_*/*.MYI
```

Use these steps to recreate an LMT database:

```
[root@n000]# dbs=$(mysql -Nse "show databases  like 'filesystem%'")
[root@n000]# mysql -e "drop database $dbs" /opt/xyratex/bin/create_lmt_databases.sh
[root@n000]# mysql -D mysql -s < /opt/xyratex/package/db/lmt_access.sql
```

Check the system again to verify that the **Performance** tab is available.

14.11 Isolate a Slow Disk

Prerequisites

`iostat` utility. Iostat is part of the `sysstat` rpm. It can be installed on OSS nodes using the following command:

```
$ sudo yum --installroot /mnt/mouse/images/pristine/appliance.x86_64/ install
sysstat
```

About this task

This topic provides step-by-step instructions to isolate slow disk drives on a Sonexion system, and describes how to measure the performance of the drive.

Procedure

1. Log in to an OSS node exhibiting slow performance:
   a. Change to root user:

      ```
      $ sudo su -
      ```

   b. In one shell, run:

      ```
      # iostat -xmctd 5 > iostat_ost00xx.log
      ```

   c. In another shell, run the `obdfilter-survey`.
   d. When the obdfilter-survey is finished, stop iostat and save the file.
2. Collect the SMART data:

   ```
   # for x in /dev/disk/by-id/wwn-*; do echo disk: $x; smartctl –a $x; done >
   smart.log
   ```

3. Collect mdraid status information with the command:

   ```
   # cat /proc/mdstat
   ```

4. Repeat the previous three steps for all slow, faulty, or suspect OSSes on the system.

   Optionally display which drives are assigned to each virtual RAID device, by using the `mdraid-analyze` script located at /usr/sbin/ :

   ```
   sudo /.mdraid-analyze
   ```

14.12 Warning After cscli unmount: "Database assertion: created a new connection but..."

Affected Releases	1.2.0, 1.3.1
Affected Products	Sonexion 1300, Sonexion 1600
Root Access	No
Service Interruption Level	Live

Problem: The following error message appears after a CSCLI unmount:

```
unmount: Database assertion: created a new connection but pool_size is already
reached (4 > 3)!
```

Workaround: This warning indicates the occurrence of a connection leak, meaning that a CSSM instance is using more than three database connections due to a bug in the management software. This warning is benign and can be disregarded.

15 CSCLI Introduction

Sonexion CLI commands are organized by mode; certain commands are available depending on the mode (state) of the Sonexion system. Two modes are relevant to customers: Site Configuration Mode and Daily Mode. A third mode, OEM Mode, is relevant only to Manufacturing and factory personnel. OEM Mode commands are not included in this document.

Site Configuration Mode/Pre-Shipment Mode

Site Configuration (`custWizard`) Mode (also called "Pre-Shipment" mode) commands are available after the Sonexion system has been fully provisioned and before the system runs in Daily Mode. These commands enable users to specify customer configuration settings, apply or reset network cluster settings, obtain FRU information, upgrade Sonexion software on Lustre nodes, and toggle between Site Configuration and Daily Modes.

Daily Mode

Daily Mode commands are available after the Sonexion system has been fully provisioned and configured for customer use. These commands enable users to fully manage the Lustre file system and cluster nodes, including mount/unmount, power-cycle, failover/failback, and control node filters and exports. Daily Mode commands also enable users to obtain FRU information and upgrade software on Lustre nodes.

Guest Mode

A guest account allows non-privileged users to run some commands to obtain information about the system, using read-only access to the system. Depending on the privileges, a subset of CSCLI commands is provided for a guest account.

Set the Mode

To determine which mode is active, run:

```
cscli cluster_mode –s
```

To change modes, run one of the following:

```
cscli cluster_mode --mode daily
cscli cluster_mode --mode custwiz
```

How CSCLI Handles Invalid Parameters

If CSCLI detects multiple invalid parameters, it may report an error for only one of them. After fixing the designated error and re-entering the command, an error is reported for the next invalid parameter, and so on.

For example, if there is a sequence of validation, when the validation of the first argument fails, the validation of upcoming arguments stops and an exception is raised.

IMPORTANT: For all CSCLI commands, exercise caution before using the `-y` or `--force` parameters, even if they are required.

Summary of Changes for Release 2.1.0

No commands were added or changed in this release, when compared to Release 2.0.0.

While no commands were added, changed, or removed in this software release, please note:

- The CNG feature is not supported in Release 2.1.0. To avoid confusion, the section on CNG-specific CSCLI commands has been removed from this document.
- In Release 2.1.0, optional Additional DNE Units (ADUs) are now referred to as optional additional Metadata Management Units (MMUs). CSCLI commands that were previously used for ADUs are applicable to MMUs.

Summary of Changes for Release 2.0

The following table shows commands that were added or changed in the previous release:

Table 6. CLI Changes for Release 2.0

No.	Addition/ Change	CLI Command	Description	Component
1.	Added	`async_journal`	Enables/Queries/Disables asynchronous journal for OST targets	OST Target
2.	Added	`configure_mds`	Configures MDS nodes (as part of ADU addition)	ADU
3.	Updated	`service_console`	Configure service console	Serviceability
4.	Updated	`support_bundle`	Additional log files are available in the support bundle for MDRAID examine output and lustre users/groups or local users/groups for CNG nodes.	Support Bundle Settings
5.	Added	`lustre_network`	Lustre network (LDN) IP addresses management.	Lustre
		`list_ranges`	List ranges of IPs for LDN	
		`find_gaps`	Show unused IPs in ranges of LDN	
		`add_range`	Add new range of IPs into LDN	
		`extend_range`	Extend range of IPs for LDN	
		`remove_range`	Remove range of IPs for LDN	
6.	Added	`lustre_perf`	These commands are used to view the luster performance	Lustre
7.	Added	`show_new_nodes`	Displays the nodes in ADUs when they are added to a system	ADU
8.	Added to existing CSCLI command	`configure_hosts`	Nodes in the ADU (MDS nodes) can be configured via this command	ADU

No.	Addition/ Change	CLI Command	Description	Component
9.	Added to existing CSCLI command	`show_nodes`	Nodes in the ADU (MDS nodes) will also display along with OSS nodes	ADU
10.	Added to existing CSCLI command	`fs_info`	File System information of nodes (MDS) in the ADU will also display	ADU
11.	Added	`rack`	Extends the functionality of `cscli get_rack_position` and `set_rack_position`	Rack
12.	Added	UD Commands	Unresponsive Drives Commands	Disk Drive
		`dm`	DM Service Configuration Management	
		`dwd`	DWD- Daemon Configuration Management	

15.1 CSCLI Command Summary

The following table summarizes Sonexion CLI commands, with columns indicating the mode or modes that include each command.

Table 7. CSCLI Command Summary

Command	Description	Daily Mode	Site Con-fig Mode	Guest Mode
`alerts`	Displays current or historical system health alert notifications	X	X	X
`alerts_config`	Shows and updates the alerts configuration	X	X	
`alerts_notify`	Enables or disables alerts notifications		X	
`apply_network_setup`	Applies a network setup to a Sonexion system		X	
`async_journal`	Enables, queries, and disables asynchronous journal for OST targets	X		
`autodiscovery_mode`	Enables or disables auto-discovery mode on system nodes	X	X	
`batch`	Runs a sequence of CSCLI commands in a batch file	X	X	
`cluster_mode`	Toggles the system between Daily mode and CustWizard (Site Configuration) mode	X	X	

Command	Description	Daily Mode	Site Con-fig Mode	Guest Mode
`configure_hosts`	Configures host names for discovered nodes	X		
`configure_mds`	Configures MDS nodes (as part of ADU addition)	X		
`configure_oss`	Configures a new OSS node	X		
`create_filter`	Creates customer filters for nodes	X		
`delete_filter`	Deletes customer filters for nodes	X		
`dm`	DM service configuration management	X		
`dwd`	DWD daemon configuration management	X		
`failback`	Fails back resources for the specified node	X		
`failover`	Fails over resources to the specified node	X		
`fru`	Retrieves FRU (replacement) information	X	X	X
`fs_info`	Retrieves file system information	X	X	X
`get_rack_position`	Indicates the specified node's position in the Sonexion rack	X		
`ibstat_check`	Disables or enables HA's probing of the high speed network	X		
`ip_routing`	Manages IP routing	X	X	
`list`	Lists all supported commands	X	X	X
`lustre_network`	Lustre network (LDN) IP addresses management commands	X		
`lustre_perf`	Export lustre performance data collected by the Sonexion system	X		
`lustre_users`	Configures Lustre filesystem users	X		
	`lustre_users show`	X		
	`lustre_users local`	X		
	`lustre_users ldap`	X		
	`lustre_users ad`	X		
	`lustre_users nis`	X		
	`lustre_users upcall`	X		
	`lustre_users order`	X		
	`lustre_users apply`	X		

Command	Description	Daily Mode	Site Con-fig Mode	Guest Mode
manage_guest	Manages the guest account	X		
monitor	Monitors the current health of the cluster nodes and elements	X	X	X
mount	Mounts the Lustre file system in the cluster	X	X	
netfilter_level	Manages the netfilter level	X	X	
power_manage	Specifies node power management options	X		
rack	Used to list, show, create, delete, move and rename racks.	X		
raid_check	Enables RAID checks on RAID devices	X		
cmd_rebuid_rate	Manages the RAID rebuild rate	X	X	
reset_network_setup	Resets the network setup of an existing Sonexion system		X	
restore_mgmt	Enables MGMT node recovery. When enabled, MGMT node that boots is restored from the latest good backup.	X		
cmd_service_console	Configure service console	X		
set_admin_passwd	Changes administrator user password on an existing Sonexion system	X	X	
set_date	Manages the date setting on the Sonexion system	X		
set_network	Specifies a Sonexion network setup		X	
set_node_version	Changes the image for a diskless node	X		
set_rack_position	Changes a given node position in the Sonexion rack	X		
set_timezone	Manages the timezone setting on the Sonexion system	X		
show_filters	Shows customized and predefined node filters	X		X
show_network_setup	Shows a Sonexion network setup		X	X
show_new_nodes	Displays a table with new OSS nodes and their resources	X		
show_node_versions	Shows the current software version on the specified nodes	X		
show_nodes	Displays node information	X		X
show_update_versions	Shows available software versions in the Sonexion Management Server repository	X		
show_version_nodes	Shows all nodes at the specified software version	X		
sm	Manages the InfiniBand Subnet Manager	X	X	
support_bundle	Manages support bundles and support bundle settings	X	X	X

Command	Description	Daily Mode	Site Con-fig Mode	Guest Mode
`syslog`	Retrieves `syslog` entries	X	X	X
`unmount`	Unmounts Lustre clients or targets on the file system	X	X	
`update_node`	Updates the software version on the specified node	X		

15.2 Administrative Commands

Sonexion administrative commands include functions to get file system and cluster node information, retrieve syslog entries, show FRU information and list available commands.

File System Information Command

Modes: Site configuration, Guest, Daily

The `fs_info` command shows all file system information. File System information of nodes (MDS) in the ADU will also display if they are configured.

Synopsis

```
$ cscli fs_info [-h] [-f fs_name] |-c cluster_name |--cluster cluster_name
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-f fs_name |--fs fs_name`	Displays the file system name.
`-c cluster_name|--cluster cluster_name`	This parameter is deprecated. It is supported only for backward compatibility.

FRU Information Command

Modes: Site configuration, Guest, Daily

The `fru` command retrieves information about Field Replaceable Units (FRUs) in the Sonexion system. FRUs are grouped into the following element 'types': ArrayDevice, BMC, Cooling, Enclosure, Enclosure_Electronics, PSU and Battery. FRU information can be retrieved per element type, on a per node basis, or for all nodes in the system.

Synopsis

```
$ cscli fru [-h] (-a | -n node_spec) [-t
{ArrayDevice,BMC,Cooling,Enclosure,Enclosure_Electronics,PSU,Battery}] [-i index] [-
l [history]]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-a|--all`	Displays FRUs (including status) grouped by type, for all nodes in the system.
`-n` *`node_spec`*`|--nodes` *`node_spec`*	Displays FRUs (including status) grouped by element type, for a specified node(s) in the system.
`-t {ArrayDevice,` *`bmc`*`, Cooling, Enclosure, Enclosure_Electronics, PSU, Battery}`	Displays FRUs (including status) for the specified element type. Examples of element types: array device, BMC, PSU, battery.
`-i` *`index`* `|--index` *`index`*	Displays FRUs (including status) for specified elements within a list of elements of the same type.
`-l [`*`history`*`] | --history [`*`history`*`]`	Displays FRU history (default is 10 lines of history).

Cluster Mode Command

Modes: Site configuration, Daily

The `cluster_mode` command toggles the Sonexion system among multiple system modes: daily, custWizard or pre-shipment.

Synopsis

```
$ cscli cluster_mode [-h] [-s] [--mode {daily,custwiz,pre-shipment}] [--db-only]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-s |--status`	Displays the status of the cluster.
`--mode {daily,custwiz,pre-shipment}`	Switches to the specified mode. Switches to either daily mode, site configuration mode or pre-shipment mode.
`--db-only`	Update only the database. Does not sync nodes via puppet. Valid only with '--mode' argument.

List Commands in Mode

Modes: Site configuration, Guest, Daily

The `list` command shows a list of available commands in the current Sonexion mode.

Synopsis

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

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

Display Log Information

Modes: Site configuration, Guest, Daily

The `syslog` command displays Lustre log entries.

Synopsis

```
$ cscli syslog [-h] [-m max] [-F] [-d duration] [-s start_time] [-e end_time] [-r]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-m max |--max=max`	Specifies the maximum number of entries to return.
`-F |--follow`	Polls for future messages. Only valid without `-e`, `-r` arguments.
`-d duration |--duration=duration`	Specifies duration (in seconds) for which to follow output. Only valid with `–F` argument.
`-s start_time |--start_time=start_time`	Specifies the earliest time for which messages should be received.
`-e end_time |--end_time end_time`	Specifies the latest time for which messages should be received.
`-r |--reverse`	Sorts entries in descending order (by time).

Set Administrator Password Command

Modes: Site configuration, Daily

The `set_admin_passwd` command changes and sets an administrator password.

Synopsis

```
$ cscli set_admin_passwd [-h] [-p password]
```

Optional Arguments	Description
`-h|--help`	Displays the help message and exits.
`-p|--password`	Specify the new administrator password string.

Batch Execution Command

Modes: Site configuration, Daily

The `batch` command runs a sequence of CSCLI commands in a batch file.

Synopsis

```
$ cscli batch [-h] –b batch_file
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-b batch_file|--batch-file batch_file`	Specifies the command batch file.

IP Routing Command

Modes: Site configuration, Daily

The `ip_routing` command manages IP routing to and from the system database.

Synopsis

```
$ cscli ip_routing [arguments]
```

where arguments can be **one** of the following lines:

```
--show|-s [--loadable]

 --load path_to_file

 --add | -a  --dest destination_ip --prefix prefix_len --router router_ip

 --update | -u --route-id route_id [--dest destination_ip] [--prefix prefix_len] [--
router router_ip]

 --delete | -d --route-id route_id

 --clear | -c

 --apply | -a
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-s |--show`	Shows the current IP routing table in the database.
`--loadable`	Prints the routing table in loadable format (use with the `–show` argument).
`-c |--clear`	Clears the routing table in the database.
`--apply`	Applies IP routing.
`--load` *load*	Loads the IP routing table from a file to the database.
`-a |--add`	Inserts IP routing in the database.
`-u |--update`	Updates IP routing in the database.
`-d |--delete`	Deletes IP routing from the database.
`--dest` *dest*	Specifies the destination IP address.
`--prefix` *prefix*	Specifies the prefix length (0-32).
`--router` *router*	Specifies the router IP address.
`--route-id` *route_id*	Specifies the route identifier (see `ip_routing –show`).

15.3 Administrative Settings Commands

Manage the Administrative Password

Modes: Site configuration, Guest, Daily

The `set_admin_passwd` command sets the Sonexion system administrator's user password.

Synopsis

```
$ cscli set_admin_passwd [-h] -p password
```

Option	Description
`-h |--help`	Prints the help message and exits.
`-p |--password`	Sets the system administrator's password.

Manage the System Date

The `set_date` command manages the date on the Sonexion system.

IMPORTANT: Exercise caution before using the `--force` parameter.

Synopsis

```
$ cscli set_date [-h] [-s new_date] [--force-ntp]
```

Optional Arguments

Option	Description
`-h |--help`	Displays the help message and exits.
`-s new_date |--set new_date`	Specifies the new date in the format: *MMDDhhmmCCYY.ss*
`--force-ntp`	Forces NTP configuration.

Manage the System Timezone

Modes: Daily

The `set_timezone` command manages the timezone on the Sonexion system.

Synopsis

```
$ cscli set_timezone [-h] [-s new_timezone] [-l]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-s new_timezone |--set new_timezone`	Specifies the new time zone location name. For example, "America/Los_Angeles".
`-l |--list`	Lists the available timezones.

Changing the time zone is optional. By default, the time zone is set to PDT (UTC/GMT -7). If necessary, the time zone setting can be changed on the Sonexion systems. However after using the `cscli set_timezone` command on the system, it is not propagated completely to all components. Rebooting is the safest way to ensure that all processes are using the same time zone. Thus it is recommended to do a full system restart of the system.

Manage the InfiniBand Subnet Manager

Modes: Site configuration, Guest, Daily

The `sm` command manages (enables, disables or prioritizes) the InfiniBand Subnet Manager (SM) integrated with the Sonexion system. The local SM ensures that InfiniBand is properly configured and enabled for use. In situations in which Sonexion is connected to a larger InfiniBand network that already uses a subnet manager, the local SM should be disabled. The `sm` command can also be used to modify subnet manager priorities.

Synopsis

```
$ cscli sm [-h] (-e | -d) [-P {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}]|-c cluster_name |--cluster cluster_name
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-e |--enable`	Enables the IB storage manager.
`-d |--disable`	Disables the IB storage manager used with the system.
`-P {0,1,2,3,4,5,6,7,8,9,10,11,12,13, 14,15}|--priority {0,1,2,3,4, 5,6,7,8,9,10, 11,12,13,14,15}`	Sets the priority [0..15] of the IB storage manager used with the system.
`-c cluster_name|--cluster cluster_name`	This parameter is deprecated. It is supported only for backward compatibility.

Guest Account Management

Modes: Daily

A guest account allows a non-privileged user to run some commands to obtain information about the system (read-only access to the appliance). Depending on the privileges CSCLI provides a limited subset of commands to the “guest” account.

The “Guest” account has access to export but that displays only the subcommands.

The `cscli manage_guest` command is part of the "user management" component and is available for an "admin" account only. This command lets you enable or disable a "guest" account access using UI or/and CSCLI and change "guests" password.

Synopsis

```
$ cscli manage_guest [-h] [-s] [--enable-shell | --disable-shell] [--enable-web | --disable-web] [--password password | --password-disable]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-s |--show`	Displays guest account information.

Optional Arguments	Description
`--enable-shell`	Enables shell for guest account.
`--disable-shell`	Disables shell for guest account.
`--enable-web`	Enables web for guest account.
`--disable-web`	Disables web for guest account.
`--password` *`password`*	Guest account password.
`--password-disable`	Clears password and blocks account until a new password is set. This option also sets “disabled” flags in cscli and GUI for guest account.

15.4 `alerts` Display Command

Modes: Site configuration, Guest, Daily

The Sonexion `alerts` subcommand displays current and historic health alerts for system nodes and elements, and thresholds for system alerts.

Synopsis

```
cscli alerts [-h] {elements_active,nodes,elements,nodes_active,thresholds}
```

Positional Arguments	Description
`nodes`	Displays alert history for nodes.
`elements`	Displays alert history for elements.
`nodes_active`	Displays current alerts for nodes.
`elements_active`	Displays current alerts for elements.
`thresholds`	Displays editable alert thresholds and their current settings.

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

`alerts elements_active` Subcommand

Synopsis

```
cscli alerts elements_active [-h] [-y] [-v] [-x] [-n node_spec | -g genders_query]
[-S element_filter]
```

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

Optional Arguments	Description
`-y |--yaml`	Outputs data in YAML format.
`-v |--verbose`	Outputs extra data.
`-x |--unhandled`	Displays alerts for notifications that have not been turned off. (Default value is all alerts are shown.)
`-n` *`node_spec`* `|--nodes` *`node_spec`*	Specifies pdsh-style node hostnames (e.g. `node[100-110,120]`).
`-g` *`genders_query`* `|--genders` *`genders_query`*	Specifies node genders attributes query (e.g. `mds=primary`).
`-S` *`element_filter`* `|--search` *`element_filter`*	Specifies the element filter so a search can be done by element name. The pattern is case-sensitive. Regular expressions allowed. For example, Fan Statistics, Power Statistics, Thermal Statistics, Voltage Statistics, etc.

`alerts nodes` Subcommand

Synopsis

```
cscli alerts nodes [-h] [-y] [-s start_time] [-e end_time] [-m limit] [-n
node_name] [-N {down,unreachable,up}]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-y |--yaml`	Outputs data in YAML format.
`-s` *`start_time`* `|--start-time` *`start_time`*	Specifies the alert start time in ISO-8601 format. If `--start-time` is not specified, then `--end-time` is ignored and the "last 7 days" period is used.
`-e` *`end_time`* `|--end-time` *`end_time`*	Specifies the alert end time in ISO-8601 format. (Default value is “now”.)
`-m` *`limit`* `|--max` *`limit`*	Specifies the maximum number (limit) of alerts to display.
`-n` *`node_name`* `|--node` *`node_name`*	Specifies the node for which to display alerts. **Pdsh**-style node masks are not allowed here.
`-N {down,unreachable,up}|--node` *`status`*	Specifies node status.

`alerts elements` Subcommand

Synopsis

```
cscli alerts elements [-h] [-y] [-s start_time] [-e end_time] [-m limit] [-n
node_name] [-U {unknown,warning,ok,critical}]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-y |--yaml`	Outputs data in YAML format.
`-s start_time |--start-time start_time`	Specifies the start time filter in ISO-8601 format. If `--start-time` is not specified, `--end-time` is ignored and the "last 7 days" period is used.
`-e end_time |--end-time end_time`	Specifies the end time filter in ISO-8601 format (default value is "now").
`-m limit |--max limit`	Specifies the maximum number (limit) of items to display.
`-n node_name |--node node_name`	Specifies the node for which to display items. Pdsh-style node masks are not allowed here.
`-U {unknown,warning,ok,critical}| --element status`	Specifies the element's status.

`alerts nodes_active` Subcommand

Synopsis

```
cscli alerts nodes_active [-h] [-y] [-v] [-x] [-n node_spec | -g genders_query]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-y |--yaml`	Outputs data in YAML format.
`-v |--verbose`	Outputs extra data.
`-x |--unhandled`	Displays alerts for notifications that have not been turned off (default is all alerts are shown).
`-n node_spec |--nodes node_spec`	Specifies pdsh-style node hostnames (e.g. node[100-110,120]).
`-g genders_query |--genders genders_query`	Specifies node genders attributes query (e.g. `mds=primary`).

`alerts threshold` Subcommand

Synopsis

```
$ cscli alerts thresholds [-h] [-y]
```

Threshold fields:

Field	Description
name	Short identifier of the threshold

Field	Description
`description`	Describes the threshold and gives tips on how to modify it
`gender`	Type of nodes to which the threshold is applied
`warning`	Value of the warning threshold
`critical`	Value of the critical threshold

Possible gender values:

Value	Specifies
all	All nodes; general node type that can be overwritten by more specific node types
`mgmt`	Management nodes (primary and secondary)
`mds`	Metadata Servers
`oss`	Object Storage Servers

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-y |--yaml`	Outputs data in YAML format.

15.5 `alerts_config` Command

Modes: Site configuration, Daily

The `alerts_config` command allow viewing and updating of the alerts configuration.

Synopsis

```
cscli alerts_config [-h] {email_off,thresholds,email_update, email_server_update,
email_delete, email_add,email_on,email_server,emails}
```

Positional Arguments	Description
`email_off`	Turns off notifications for notification subscribers.
`thresholds`	Sets the current value of an threshold. This value can be edited.
`email_update`	Sends an email alert with an update.
`email_server_update`	Sends an email alert with a server update.
`email_delete`	Deletes the email.
`email_add`	Adds a new notification subscriber.
`email_on`	Turns on notifications for subscribers.

Positional Arguments	Description
`email_server`	Displays the relay SMTP server configuration.
`emails`	Lists the alert notification subscribers.

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

`email_off` Subcommand

The `email_off` command turns off notifications for subscribers.

Synopsis

```
cscli alerts_config email_off [-h] -u email
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-u email |--user email`	Displays subscriber email. Notifies subscribers have new mail in `/var/spool/mail` or admin.

`thresholds` Subcommand

Current thresholds are applied to the monitoring configuration only if the `--apply-config` option is used. It may take about 15 seconds to apply the configuration threshold changes.

If a group of changes needs to be made to the thresholds, edit a few threshold values and then add the `--apply-config` option to the last edit to set all the changes at once.

The new thresholds applied to monitoring configuration take effect a few minutes after they are applied when the next scheduled node check is performed.

The only editable thresholds are those listed in the output of the `cscli alerts thresholds` command.

Synopsis

```
cscli alerts_config thresholds [-h] -t threshold_name -g gender_name [-W
warning_threshold_value] [-C critical_threshold_value] [-A]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-t` *`threshold_name`* `|--threshold` *`threshold_name`*	Displays the name of the threshold.
`-g` *`gender_name`* `|--gender` *`gender_name`*	Displays the gender name of the threshold.
`-W` *`warning_threshold_value`* `|--warning` *`warning_threshold_value`*	Displays the warning threshold value.
`-C` *`critical_threshold_value`* `|--critical` *`critical_threshold_value`*	Displays the critical threshold value.

Optional Arguments	Description
`-A |--apply-config`	Applies the threshold configuration.

If the `--apply-config` command is used the current thresholds are applied only to the monitoring configuration. It may take about 15 seconds to apply the configuration threshold changes.

To make a group of changes to the thresholds, edit a few threshold values and then add the `--apply-config` option with the last edit to set all the changes at once.

The new thresholds applied to monitoring configuration take effect a few minutes after they are applied when the next scheduled node check is performed.

The only editable thresholds are those listed in the output of the `cscli alerts thresholds` command.

`email_update` Subcommand

The `email_update` command updates the existing subscriber's notification.

Notification Levels

The `level` option sets the alerts trigger for an email to be sent to a subscriber. The possible level option values are:

- **Critical** - Notify elements critical or node down statuses
- **Warning** - Notify elements warning statuses
- **Unknown** - Notify elements unknown statuses
- **Ok** - Notify when elements and nodes recover from problems
- Any combination of the above (comma-separated)
- **None** - No notifications (similar to `cscli alerts_config email_off`)
- **All** - Send all notifications, including notifications
- When a node/element is flapping between statuses
- When a node/element is in scheduled downtime

When the email notification level for a subscriber is set to "OK", an email for an "OK" alert is not sent to the subscriber. For the system to send email for "OK" alerts, make sure that at least one additional severity of alerts is given.

Notification Periods

The Notification periods are as follows:

- 24x7 - Notify always
- Workhours - Notify only during working days and hours (in the timezone of the server).

Synopsis

```
cscli alerts_config email_update [-h] -u email [-M email] [-N user_full_name]  [-P
{24x7,workhours}] [-L level]
```

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

Optional Arguments	Description
`-u email |--user email`	Displays subscriber email. Notifies you have new mail in `/var/spool/mail` or `admin`.
`-M email |--email email`	Displays the email address.
`-N user_full_name |--name user_full_name`	Displays a longer name or description for the subscriber.
`-P {24x7,workhours} |--period {24x7,workhours}`	Displays the time periods at which the subscriber is notified. possible values: {24x7,workhours}
`-L level, |--level level`	Displays notification level; possible values: any comma-separated combination of `{critical,ok,unknown,warning}`, or `all`, or `none`.

`email_server_update` Subcommand

The `email_server_update` command configures the SMTP server to send alerts to external email addresses.

Synopsis

```
cscli alerts_config email_server_update [-h] -s smtp_server_address [--port port] [-S email_from] [-d domain] [-u smtp_user] [-p smtp_password]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-s smtp_server_address |--server smtp_server_address`	Displays an IP address or hostname of the (relay) SMTP server.
`--port port`	SMTP server port (default: 25)
`-S email_from|--sender email_from`	Displays the senders email address. If `--domain` is set, the default value for the sender is *`cluster_name@domain`*. If `--domain` is not set, the sender's email address is required.
`-d domain |--domain domain`	Displays the internet hostname of the mail system to be used with email addresses that have no "@".
`-u smtp_user, --user smtp_user`	Specifies the username if the SMTP server requires authentication.
`-p smtp_password |--password smtp_password`	The password if the SMTP server requires authentication.

`email_delete` Subcommand

The `email_delete` command deletes notifications to subscribers.

Synopsis

```
cscli alerts_config email_delete [-h] -u email
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-u email |--user email`	Displays subscriber email. Notifies you have new mail in `/var/spool/mail` or admin.

`email_add` Subcommand

The `email_add` command adds a new notification subscriber.

Synopsis

```
cscli alerts_config email_add [-h] -M email [-N user_full_name] [-P
{24x7,workhours}] [-L level]
```

Notification Levels

The level option sets the alerts trigger for email to be sent to a subscriber. Possible level option values are:

- **Critical** - Notify elements critical or node down statuses
- **Warning** - Notify elements warning statuses
- **Unknown** - Notify elements unknown statuses
- **Ok** - Notify when elements and nodes recover from problems
- Any combination of the above (comma-separated)
- **None** - No notifications (similar to "cscli alerts_config email_off")
- **All** - Send all notifications, including notifications when a node/element is flapping between statuses, or when a node/element is in scheduled downtime

Notification Periods

Possible Notification Periods:

- 24x7 - Notify always
- Workhours - Notify only during working days and hours (in the timezone of the server) .

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-M email|--email email`	Displays subscriber email. Notifies you have new mail in `/var/spool/mail` or admin. email address.
`-N user_full_name |--name user_full_name`	Displays a longer name or description for the subscriber.

Optional Arguments	Description
`-P {24x7,workhours} |--period {24x7,workhours}`	The time periods at which the subscriber is notified. Possible values: `{24x7,workhours}` (default: `24x7`).
`-L level |--level level`	The notification level. Possible values: any comma-separated combination of: `{critical,ok,unknown,warning}`, or `all`, or `none` (default: `all`).

email_on Subcommand

The email_on command turns on notifications for subscribers.

Synopsis

```
cscli alerts_config email_on [-h] -u email
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-u email |--user email`	Displays subscriber email. Notifies you have new mail in `/var/spool/mail` or admin.

email_server Subcommand

The email_server command displays the relay smtp server configuration.

Synopsis

```
$ cscli alerts_config email_server [-h]
```

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

emails Subcommand

The emails command displays a list of alert notifications to the subscribers.

Notification Levels

The level option sets the alerts trigger for email to be sent to a subscriber. Possible level option values are:

- **Critical** - Notify elements critical or node down statuses
- **Warning** - Notify elements warning statuses
- **Unknown** - Notify elements unknown statuses
- **Ok** - Notify when elements and nodes recover from problems
- Any combination of the above (comma-separated)
- **None** - No notifications (similar to "cscli alerts_config email_off")
- **All** - Send all notifications, including notifications

- When a node/element is flapping between statuses
- When a node/element is in scheduled downtime

Synopsis:

```
cscli alerts_config emails [-h] [-y] [-v] [-u email]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-y |--yaml`	Outputs data in YAML format.
`-v |--verbose`	Outputs extra data in verbose mode.
`-u email |--user email`	Displays subscriber email.Notifies you have new mail in `/var/spool/mail` or `admin`.

15.6 `alerts_notify` Command

Modes: Site configuration, Daily

The Sonexion `alerts_notify` command turns alert notifications on or off.

Synopsis

```
cscli alerts_notify [-h] {on,off} ...
```

Positional Arguments	Description
`on`	Sets the alert notification on.
`off`	Sets the alert notification off.

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

alerts_notify on

The `alerts_notify on` command turns alert notifications on.

Synopsis

```
cscli alerts_notify on [-h] (-n node_spec | -g genders_query) [-S element_filter | -
E element_name]
```

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

Positional Arguments	Description
`-n node_spec |--node |--node_spec |--nodes node_spec`	Looks through passed hostname elements. Looks for pdsh style nodes host names (e.g. node[100-110,120]).
`-g genders_query|--genders genders_query`	Displays the node genders attributes query (e.g. mds=primary).
`-S element_filter |--search element_filter`	This command searches by element name. The pattern is case sensitive. Regular expressions allowed.
`-E element_name|--element element_name`	Displays the element name.

alerts_notify off

The `alerts_notify off` command turns alert notifications off.

Synopsis

```
cscli alerts_notify off [-h] (-n node_spec | -g genders_query) [-S element_filter | -E element_name] [-C comment]
```

Positional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-n node_spec |--node |--node_spec |--nodes node_spec`	Looks through passed hostname elements. Looks for pdsh style nodes host names (e.g. `node[100-110,120]`).
`-g genders_query|--genders genders_query`	Displays the node genders attributes query (e.g. mds=primary).
`-S element_filter |--search element_filter`	This command searches by element name. The pattern is case sensitive. Regular expressions allowed.
`-E element_name|--element element_name`	Displays element name.
`-C comment|--comment comment`	Displays a brief description of the action being taken.

15.7 `async_journal` Command

Modes: Daily

The `async_journal` command can be used to enable, query or disable OST targets.

Synopsis

```
cscli async_journal [-h] [-s] [--enable] [--disable]
```

where:

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-s |--status`	Displays the status of asynchronous journal.
`--enable`	Enables asynchronous journal for OSTs.
`--disable`	Disables asynchronous journal for OSTs.

15.8 Configuration Commands

The configuration commands specify the MAC address and hostname for a given node and configure OSS nodes.

Configure Hosts Command

Modes: Daily

The Sonexion `configure_hosts` command configures the MAC address and host names for the discovered node. Nodes in the ADU (MDS nodes) can be configured via this command

Synopsis

```
cscli configure_hosts [-h] -m mac_address --hostname hostname [-f]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-m` *`mac_address`* `|--mac` *`mac_address`*	The node *`mac_address`*
`--hostname` *`hostname`*	The new node hostname.
`-f |--force`	Forces the mode (to skip hostname validation).

Configure MDS Nodes Command

Modes: Daily

The `configure_mds` command is used to add and configure new MDS nodes (as part of the ADU addition) in the Sonexion system. The command is used in two modes:

- Bind MD device with Lustre FS (by name)
- Apply all bindings (configuration), i.e., perform formatting of MD devices as Lustre targets, configuration of HA, and so forth.

To bind MD device with Lustre FS (by name) the following options can be used, for example:

```
# cscli configure_mds -c snx11000n -n snx11000n06 --bind-arrays md0:snx11000n
# cscli configure_mds -c snx11000n -n snx11000n07 --bind-arrays md1:snx11000n
```

For applying all bindings (configuration) the following options can be used, for example:

```
# cscli configure_mds -c snx11000n -n snx11000n06 --apply-config
```

Synopsis

```
cscli configure_mds [-h] -n node_spec (-A | -b bind_arrays) |-c cluster_name |--
cluster cluster_name
```

where:

Optional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-n node_spec | --nodes node_spec`	Specifies the hostname of the new MDS node (in genders style).
`-a | --apply-config`	Applies the configuration to the new MDS node.
`-b bind_arrays | --bind-arrays bind_arrays`	Specifies comma-separated pairs of array-file system bindings. Each binding should be in this format: *array*`:`*file_system_name*. The *array* variable can be a genders-style string such as **md[0-3]**.
`-c cluster_name | --cluster cluster_name`	This parameter is deprecated. It is supported only for backward compatibility.

Configure OSS Nodes Command

Modes: Daily

The `configure_oss` command configures new OSS nodes in the system.

Synopsis

```
cscli configure_oss [-h] -n node_spec (-A | -b bind_arrays) |-c cluster_name |--
cluster cluster_name
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-n node_spec |--nodes node_spec`	Specifies the hostname of the new OSS node (in genders style).
`-A |--apply-config`	Applies the configuration to the new OSS node.
`-b bind_arrays |--bind-arrays bind_arrays`	Specifies comma-separated pairs of array-file system bindings. Each binding should be in this format:array:file_system_name. The array variable can be a genders-style string, e.g. `md[0-3]`.
`-c cluster_name| --cluster cluster_name`	This parameter is deprecated. It is supported only for backward compatibility.

Restore Management Nodes Command

Modes: Daily

The `restore_mgmt` command enables MGMT node recovery. When enabled, the MGMT node that boots is restored from the latest good backed-up nodes. This command is used to format and copy data to the internal drives on the MGMT nodes.

When run on MGMT0 node with the `--enable` parameter, it copies data from the MGMT node backup image (which is created using a nightly cron job) to the internal drive on the MGMT1 node. When run with the `--enable` command on the MGMT1 node, it will do likewise on the MGMT0 internal drive.

The `--disable` command switches from recovery mode to normal boot mode. It is automatically invoked once the enable is finished and under most circumstances should not be manually invoked.

If run without either `--enable` or `--disable`, the command will print its help information.

Synopsis

```
cscli restore_mgmt [-h] [-s] [--enable | --disable]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-s |--show`	Displays the MGMT recovery status.
`|--enable`	Enables the MGMT recovery, boot MGMT normally.
`|--disable`	Disables the MGMT recovery, boot MGMT normally.

Show New OSS Nodes Command

Modes: Daily

The `show_new_nodes` command displays a table of new OSS nodes and their resources.

Synopsis

```
cscli show_new_nodes [-h] [-v] |-c cluster_name |--cluster cluster_name
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-v, --verbose`	Specifies the verbose mode.
`-c cluster_name|--cluster cluster_name`	This parameter is deprecated. It is supported only for backward compatibility.

15.9 dm and dwd Unresponsive Drives (UD) Commands

Modes: Daily

To address unresponsive drive issues in Sonexion, systems the diskmonitor tool can modify maximum reading errors threshold for arrays. Additionally, the dwd (disk watcher) daemon was created to allow to power down drives in case if they are failing or modify threshold for SCSI task-abort. The following commands let customers modify those values:

- `cscli dwd`
- `cscli dm`

Daemon Configuration Command

Modes: Daily

The `cscli dwd` command may be used to configure disk watcher daemon (DWD) to allow or disallow it from powering down failed drives or to update SCSI task-abort threshold, where 0 sets to "ignore" those completely, '1' sets it to autocalculation mode and any other value will be treated as an actual numeric treshold.

Synopsis

```
cscli dwd [-h] [-n nodes] [--reset] [-l {yes,no}] [-a dwd_abrt_limit]
```

where:

Option	Description
`-h |--help`	Displays the help message and exits.
`-n nodes| --node nodes`	pdsh style node hostnames. Global configuration will be set without this argument.
`--reset`	Resets configuration to default values.
`-l {yes,no} | --lethal {yes,no}`	Allows (if set to 'yes') or disallows (if set to 'no') DWD to power down ("kill") failing drive. For GridRAID, this is the reconstruction rate.
`-a dwd_abrt_limit | --abort-limit dwd_abrt_limit`	Configures how DWD will treat SCSI task Aborts. This parameter expects integer (number) as an argument. Value of '0' will disable monitoring completely, value of '1' will enable auto-calculation and any other value will be treated as a threshold.

When called without arguments, current settings will be printed.

Default settings for DWD:

```
[root@snx11000n000 ~]# cscli dwd
------------------------------------------------------------
Node    Drive power-on on failure       Task abortion rate
------------------------------------------------------------
global              0                               1
Sample message for setting abort-task limit:
[root@snx11000n000 ~]# cscli dwd -a 15
dwd: done
Sample message for setting drive power down trigger:
[root@snx11000n000 ~]# cscli dwd -l yes
dwd: done
Sample output for DWD after setting new values:
[root@snx11000n000 ~]# cscli dwd
------------------------------------------------------------
Node    Drive power-on on failure       Task abortion rate
------------------------------------------------------------
global              1                               15
Those values can also be reset at any time similar to cscli dm command:
[root@snx11000n000 ~]# cscli dwd --reset
dwd: done
```

DM Service Configuration Management

The `cscli dm` command allows to update maximum read error threshold for all arrays or only specifically for RAID6 arrays.

Synopsis

```
cscli dm [-h] [-n nodes] [--reset] [-g max_read_errs] [-m max_read_errs_r6]
```

where:

- `-g` is for global limits
- `-m` for setting RAID6 limits only

Additionally there is a “-reset” option to unset all thresholds to default. If no options were provided output would be similar to following:

Option	Description
`-h |--help`	Displays the help message and exits.
`-n nodes |--node nodes`	pdsh style nodes hostnames. Global configuration will be set without this argument.
`--reset`	Resets configuration to default values.
`-g max_read_errs | --global max_read_errs`	Sets maximum read errors value threshold before the drive is considered as failed. This is global limit for both GridRaid and MDRaid.
`-m max_read_errs_r6 | --mdraid-limit max_read_errs_r6`	Sets max read errors value threshold before drive would be considered as failed. This is a MDRaid specific parameter to override the global limit if needed.

If no options were provided, output would be similar to:

```
[root@snx11000n00 ~]# cscli dm
------------------------------------------------------------------------
Node    Max read errors(Gridraid or mdraid)     Max read errors(mdraid(raid6)
specific)
------------------------------------------------------------------------
global              3500                        3500
```

Example output for changing values

Setting new limit for RAID6 arrays:

```
[root@snx11000n000 ~]# cscli dm -m 5000
dm: done
Setting new limit for all type of arrays:
[root@snx11000n000 ~]# cscli dm -g 7000
dm: done
Output after setting new values:
[root@snx11000n000 ~]# cscli dm
------------------------------------------------------------------------
Node    Max read errors(GridRAID or mdRAID)     Max read errors(mdraid(raid6)
specific)
```

```
------------------------------------------------------------------------
global              7000                          5000
```

To go back to default settings reset command may be used:

```
[root@snx11000n000 ~]# cscli dm --reset
dm: done
```

15.10 Filter Commands

The Sonexion filter commands create, show, and delete a customer-nodes filter.

Create a Filter

Modes: Daily

The `create_filter` command creates a customer nodes filter.

Synopsis

```
cscli create_filter [-h] -i filter_sid -F filter_name -e filter_expr
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-i filter_sid |--id filter_sid`	Displays the symbol identifier of the filter.
`-F filter_name |--name filter_name`	Displays the filter name.
`-e filter_expr |--expression filter_expr`	Displays the filter expression. Examples: "host1,host2", "host[1-3]", "mds=primary".

Show Filters

Modes: Guest, Daily

The `show_filters` command shows all filters.

Synopsis

```
cscli show_filters [-h] [-P] [-C]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-P |--predefined`	Displays only predefined filters.
`-C |--custom`	Displays only custom filters.

Delete a Filter

Modes: Daily

The `delete_filter` command deletes a customer nodes filter.

Synopsis

```
cscli delete_filter [-h] -i filter_sid
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-i filter_sid |--id filter_sid`	Displays the symbol identifier of the filter.

15.11 `ibstat_check` Command

Modes: Daily

The `ibstat_check` command, part of HA stack settings, disables and enables HA's probing of high speed network(s). This command is available for the administrator account only.

Synopsis

```
cscli ibstat_check [-h] [-n nodes] (--enable | --disable)
```

Option	Description
`-h |--help`	Displays the help message and exits.
`-n nodes |--node nodes`	Displays `pdsh`-style nodes hostnames. **NOTE:** Lustre server nodes only.
`--enable`	Enables HA's probing of the Lustre data network.
`--disable`	Disables HA's probing of the Lustre data network.

15.12 `lustre_users` Commands

Modes: Daily

The advanced users (Lustre Users) commands support local user and group authentication commands. The following are available for Lustre users.

The `lustre_users` commands are used to configure the Lustre filesystem users settings. You can select the upcall method, configure services, and order user lookup across different services.

Synopsis

```
cscli lustre_users [-h] {ad,show,ldap,upcall,nis,apply,local,order}
```

If you are unable to connect, the following error message displays: “Unable to bind (connect) to (for example) an LDAP service using the parameters provided. Please check the URI (including port), Bind DN, and password.”

Positional Arguments	Description
`show`	Displays detailed information about all Lustre Users settings.
`apply`	Apply Lustre users configuration.
`local`	Define and manage local users and groups.
`ldap`	Management of LDAP settings.
`ad`	Management of AD settings.
`nis`	Management of NIS settings.
`order`	Instruct Lustre upcall to process user/group services in the order specified.
`upcall`	Configure the underlying method of Lustre upcall.
`-h |--help`	Displays the help message and exits.

15.12.1 `lustre_users show` Command

The Sonexion `lustre_users show` command is a subcommand of the `lustre_users` command. This command shows detailed information about all Lustre Users settings.

Synopsis

```
cscli lustre_users show [-h]
```

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

Example:

```
# cscli lustre_users show
Lustre Users
   Upcall: generic
   Order: local, directory
   LDAP:
      Servers: ldap://dc.xyua:3268
      Base DN: dc=dc,dc=xyua
      Bind DN: administrator@dc.xyua
      Password: ********
      User DNs: cn=Users,dc=dc,dc=xyua
      Group DNs: not defined
      Hosts DNs: not defined
      TLS Cert: TLS cert is valid.
         Version: 3
         Subject: CN=cfw-dc0.cfw-ad.no-more.kiev.ua
         Issuer: DC=ua, DC=kiev, DC=no-more, DC=cfw-ad, CN=ca
         Serial: 106310341937692867035147
      TLS Private Key: TLS pvtkey is valid, 1024 bits
      TLS CA Cert: TLS CA cert is valid.
         Version: 3
         Subject: C=UA, ST=Ukraine, L=Kiev, O=No More BBS, OU=Software Department,
```

```
CN=NoMore Root CA v3/emailAddress=ca@no-more.kiev.ua
         Issuer: C=UA, ST=Ukraine, L=Kiev, O=No More BBS, OU=Software Department,
CN=NoMore Root CA v3/emailAddress=ca@no-more.kiev.ua
         Serial: 9313703399412187718
   NIS: (not configured)
   Local: (not configured)
```

15.12.2 `lustre_users apply` Command

The Sonexion `lustre_users apply` subcommand applies the Lustre users configuration.

Synopsis

```
cscli lustre_users apply [-h] [-y]
```

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

15.12.3 `lustre_users local` Commands

The Sonexion `lustre_users local` command is a subcommand of the `lustre_users`command. This command defines users and groups locally, using files in the SysV standard formats of `/etc/passwd` and `/etc/group`.

Synopsis

```
cscli lustre_users local [-h] {clear,show,set_all,get_all}
```

Positional Arguments	Description
`get_all`	Downloads Lustre users/groups files.
`show`	Displays Lustre users/groups settings.
`set_all`	Uploads Lustre users/groups files.
`clear`	Resets Lustre users/groups to pristine state.

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

`lustre_users local get_all` Subcommand

The `lustre_users local get_all` command is a second-level subcommand of the `lustre_users local` subcommand.

This command downloads the current locally-defined users and/or groups, as files in the SysVstandard, colon-separated form of `/etc/passwd` and `/etc/group`. If there are no locally-defined users or groups, a template file is downloaded.

Synopsis

```
cscli lustre_users local get_all [-h] [-u [users_file]] [-g [groups_file]] [-y]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-u [users_file]|--users-file [users_file]`	The name that is used to save the users file (default `lustre-users.txt`)
`-g [groups_file] |--groups-file [groups_file]`	The name that is used to save the groups file (default lustre=`groups.txt`)
`-y |--yes`	Overwrites an existing file of the same name without prompt.

`lustre_users local set_all` Subcommand

The Sonexion `lustre_users local set_all` command is a second-level subcommand of the `lustre_users local` subcommand.

This command uploads a new user and/or group file, which will entirely replace the previous set of defined users and/or groups. It is strongly recommended to download the current definitions to a file, then add, delete, or edit entries in that file in order to upload changes.

> **IMPORTANT:** It is strongly recommended to download the current definitions to a file, then add, delete, or edit entries in that file in order to upload changes.

Synopsis

```
cscli lustre_users local set_all [-h] [-u users_file] [-g groups_file][-y]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-u users_file|--users-file users_file`	A path to a file in SysV '/etc/passwd' format.
`-g groups_file|--groups-file groups_file`	A path to a file in SysV '/etc/group' format.
`-y |--yes`	Overwrites an existing file of the same name without prompt.

`lustre_users local clear` Subcommand

The `lustre_users local clear` command is a second-level subcommand of the `lustre_users` local subcommand.

This command resets Lustre users/groups to pristine state.

Synopsis

```
cscli lustre_users local clear [-h] [-y]
```

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

Optional Arguments	Description
-y |--yes	Confirms the reset.

lustre_users local show Subcommand

The lustre_users local show command is a second-level subcommand of the lustre_users local subcommand.

This command resets displays the local users and groups.

Synopsis

```
cscli lustre_users local show [-h]
```

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

Output:

```
# cscli lustre_users local show
Local:
  Number of Groups: 3
  Number of Users: 3
```

15.12.4 lustre_users upcall Authentication Commands

The Sonexion lustre_users upcall subcommand configures the underlying method of Lustre upcall. The lustre_users upcall command must have a unique user ID. The administrator receives a warning if the number of UIDs exceed 5000 users. This operation does not fail if the threshold is exceeded.

Synopsis

```
cscli lustre_users upcall [-h] {set,show}
```

Positional Arguments	Description
-show	Displays the underlying method of Lustre upcall.
-set	Configures the underlying method of Lustre upcall.

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

lustre_users upcall set Subcommand

The lustre_users upcall set command is a subcommand of the lustre_users upcall command. Select the underlying upcall method for Lustre users and groups: ad_only for Active Directory only, or generic for any of LDAP, NIS, or locally defined users and groups.

The Lustre filesystem must be unmounted in order to change upcall type.

Synopsis

```
cscli lustre_users upcall set [-h] -u {ad_only,generic,none}
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-u {ad_only,generic}| --upcall {ad_only,generic,none}`	The `ad_only` command enables only Active Directory users and groups only. The `generic` command allows for any of LDAP, NIS, or locally-defined users and groups. The `none` command disables all Lustre upcall commands, which may result in file permission errors.

`lustre_users upcall show` Subcommand

The `lustre_users upcall show` command is a second-level subcommand of the `lustre_users upcall` subcommand.

This command shows the underlying method of Lustre upcall.

Synopsis

```
cscli lustre_users upcall show [-h]
```

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

15.12.5 `lustre_users order` Commands

The Sonexion `lustre_users order` subcommand configures Lustre filesystem users. The command selects upcall method, configure services, and order for user lookup across different services. This command instructs Lustre upcall to process user or group services in the order specified.

Synopsis

```
cscli lustre_users order [-h] {set,show}
```

Positional Arguments	Description
`show`	Displays the order for looking up services when more than one is defined.
`set`	Instructs Lustre upcall to process user/group services in the order specified.

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

`lustre_users order set` Subcommand

The `lustre_users order set` command is a subcommand of the `lustre_users order` command. This command sets the order for looking up services when more than one is defined.

Synopsis

```
cscli lustre_users order set [-h] {local,nis,directory} [{local,nis,directory} ...]
```

Positional Arguments	Description
`local,nis,directory`	Sets local, NIS and directory services.

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

`lustre_users order show` Subcommand

The `lustre_users order show` command is a second-level subcommand of the `lustre_users order` subcommand.

This command shows the order for looking up services when more than one is defined.

Synopsis

```
cscli lustre_users order show [-h]
```

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

15.12.6 `lustre_users nis` Service Commands

The Sonexion `lustre_users nis` command is a subcommand of the `lustre_users` command. This command configures users and groups via the Network Information Service (NIS).

Synopsis

```
cscli lustre_users nis [-h] {clear,set,show}
```

Positional Arguments	Description
`show`	Displays NIS settings.
`set`	Sets NIS configuration.
`clear`	Removes NIS configuration.

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

`lustre_users nis set` Subcommand

The `lustre_users nis set` command is a subcommand of the `lustre_users nis` command.

Synopsis

```
cscli lustre_users nis set [-h] [-s nis_server] [-d nis_domain]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-s nis_server |--nis_server nis_server`	Specifies this option 1 to 3 times using IP address or fully-qualified domain name.
`-d nis_domain |--nis_domain nis_domain`	NIS domain. Example: nisdomain.

`lustre_users nis clear` Subcommand

The `lustre_users nis clear` command is a second-level subcommand of the `lustre_users nis` subcommand.

This command removes all NIS configuration.

Synopsis

```
cscli lustre_users nis clear [-h] [-y]
```

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

`lustre_users nis show` Subcommand

The `lustre_users nis show` command is a second-level subcommand of the `lustre_users nis` subcommand.

Synopsis

```
cscli lustre_users nis show [-h]
```

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

15.12.7 `lustre_users ad` Service Commands

The Sonexion `lustre_users ad` command is a subcommand configures users and groups via an external AD service.

Synopsis

```
cscli lustre_users ad [-h] {clear,set,show}
```

Positional Arguments	Description
`show`	Displays AD settings.
`set`	Sets AD configuration.
`clear`	Removes all AD configurations.

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

`lustre_users ad show` Subcommand

The `lustre_users ad show` command is a second-level subcommand of the `lustre_users ad` subcommand.

Synopsis

```
cscli lustre_users ad show [-h]
```

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

`lustre_users ad set` Subcommand

The `lustre_users ad set` command is a subcommand of the `lustre_users ad` command.

Synopsis

```
cscli lustre_users ad set [-h] [-l ldap_uri] [-b base_dn] [-i bind_dn] [-p password]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-l` *`ldap_uri`* `|--ldap-uri` *`ldap_uri`*	URI, for example: ldap://127.0.0.1:389.
`-b` *`base_dn`* `|--base-dn` *`base_dn`*	Base Domain Name.
`-i` *`bind_dn`* `|--bind-dn` *`bind_dn`*	Bind Domain Name.
`-p` *`password`*`, |--password` *`password`*	Bind password.

`lustre_users ad clear` Subcommand

The `lustre_users ad clear` command is a second-level subcommand of the `lustre_users ad` subcommand.

This command removes all AD configuration.

Synopsis

```
cscli lustre_users ad clear [-h] [-y]
```

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

15.12.8 `lustre_users ldap` Service Commands

The Sonexion `lustre_users ldap` command is a subcommand of the `lustre_users` command. This command configures users and groups via an external LDAP service.

Synopsis

```
cscli lustre_users ldap [-h] {clear,set,show}
```

If you are unable to connect, the following error message displays: “Unable to bind (connect) to (for example) an LDAP service using the parameters provided. Please check the URI (including port), Bind DN, and password.”

Positional Arguments	Description
`show`	Displays LDAP settings.
`set`	Sets LDAP configuration.
`clear`	Removes all LDAP configurations.

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

`lustre_users ldap set` Subcommand

The `lustre_users ldap set` command is a subcommand of the `lustre_users ldap` command.

Synopsis

```
cscli lustre_users ldap set [-h] [-l ldap_uri] [-b base_dn]  [-u user_dn] [-G
group_dn] [-s hosts_dn] [-i bind_dn] [-p password] [--pvt-key filepath] [--tls-cert
filepath] [--tls-ca filepath]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-l` *`ldap_uri`* `|--ldap-uri` *`ldap_uri`*	URI. For example: `ldap://127.0.0.1:389`
`-b` *`base_dn`* `|--base-dn` *`base_dn`*	Base domain name
`-u` *`user_dn`* `|--user-dn` *`user_dn`*	User domain name
`-G` *`group_dn`* `|--group-dn` *`group_dn`*	Group domain name
`-s` *`hosts_dn`* `|--hosts-dn` *`hosts_dn`*	Hosts domain name
`-i` *`bind_dn`* `|--bind-dn` *`bind_dn`*	Bind domain name

Optional Arguments	Description
`-p password, |--password password`	Bind password.
`--pvt-key filepath`	Private key file (pem).
`--tls-cert filepath`	TLS certificate file (crt).
`--tls-ca filepath`	TLS ca certificate file (crt).

`lustre_users ldap clear` Subcommand

The `lustre_users ldap clear` command is a second-level subcommand of the `lustre_users ldap` subcommand. This command removes all LDAP configuration.

Synopsis

```
cscli lustre_users ldap clear [-h] [-y]
```

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

`lustre_users ldap show` Subcommand

The `lustre_users ldap show` command is a second-level subcommand of the `lustre_users ldap` subcommand.

Synopsis

```
cscli lustre_users ldap show [-h]
```

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

15.12.9 `lustre_users` Commands, New vs. Old

Table 8. New and Old Sonexion `lustre_users` Commands Comparison

New (2.0) Commands	Old (1.3.1) Commands
`lustre_users`	None

`lustre_users show`	`get_lustre_users_ldap`
	`get_lustre_users_nis`
	`get_lustre_users_upcall`

`lustre_users local`	None
`lustre_users ldap set`	`set_lustre_users_ldap`

New (2.0) Commands	Old (1.3.1) Commands
`lustre_users ldap clear`	`set_lustre_users_ldap -C`
`lustre_users nis set`	`set_lustre_users_nis`
`lustre_users nis clear`	`set_lustre_users_nis -c`
`lustre_users upcall`	`set_lustre_users_upcall`
`lustre_users order`	None
`lustre_users apply`	None
`lustre_users ad set`	`set_lustre_users_ad`

15.13 `lustre_network` Commands

Modes: Daily

The Sonexion `lustre_network` commands are used for Hidden Service Protocol (HSP) IP address management.

Synopsis

```
cscli lustre_network [-h]
{add_range,list_ranges,find_gaps,remove_range,extend_range}
```

where:

Positional Arguments	Description
`list_ranges`	List ranges of IPs for InfiniBand fabric
`find_gaps`	Show unused IPs in ranges of InfiniBand fabric
`add_range`	Add new range of IPs into InfiniBand fabric
`extend_range`	Extend range of IPs for InfiniBand fabric
`remove_range`	Remove range of IPs for InfiniBand fabric

`list_ranges`

The `list_ranges` subcommand is part of the `lustre_network` command.

Synopsis

```
cscli lustre_network list_ranges [-h]
```

where:

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

find_gaps

The `find_gaps` command is a subcommand of the `lustre_network` command.

Synopsis

```
cscli lustre_network  find_gaps [-h]
```

where:

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

add_range

The `add_range` command is a subcommand of the `lustre_network` command.

Synopsis

```
cscli lustre_network add_range [-h] -f from_ip -t to_ip [-h]
```

where:

Optional Arguments	Description
`-h |--help`	Displays the help message and exits
`-f from_ip | --from_ip` *`from_ip`*	Displays first IP address in the range
`-t` *`to_ip`* `| --to_ip` *`to_ip`*	Displays last IP address in the range

extend_range

The `lustre_network extend_range` command is a subcommand of the `lustre_network` command.

Synopsis

```
cscli lustre_network extend_range [-h] -i id [-f from_ip] [-t to_ip] [-h]
```

where:

Optional Arguments	Description
`-h |--help`	Displays the help message and exits
`-i id | --id` *`id`*	Displays the range ID
`-f` *`from_ip`* `| --from_ip` *`from_ip`*	Displays first IP address in the range
`-t` *`to_ip`* `| --to_ip` *`to_ip`*	Displays last IP address in the range

remove_range

The `lustre_network remove_range` command is a subcommand of the `lustre_network` command.

Synopsis

```
cscli lustre_network remove_range [-h]
```

where:

Optional Arguments	Description
`-h | --help`	Displays the help message and exits
`-i id | --id` *id*	Displays the range ID

15.14 lustre_perf Command

Modes: Daily

The Sonexion `lustre_perf` command is used to view the Lustre file system performance in the Sonexion system.

Synopsis

```
cscli lustre_perf subcommand [option] [-h]
```

where:

Positional Arguments	Description
`|--version`	Displays twisted version and exits
`-h |--help`	Displays the help message and exits

LustrePerf CLI client

The LustrePerf CLI client is a subcommand of the `lustre_perf` command.

Optional Arguments	Description
`fetch`	This command exports historical lustre data between start time and end time to the local filesystem. Use `cscli lustre_perf list` to find the location of the resulting output.
`ltop`	Displays live information about a Lustre file system . Use the `--help` command for more details.
`list`	Lists the full path of any existing log files. Use the `--help` command for more details.
`status`	Returns the status of the last run command (or the currently running command if it is non blocking and a process is still running). Use the `--help` command for more details.
`abort`	Aborts the currently running export job. Use the `--help` command for more details.
`clean`	Deletes all export files in the export folder. Use the `--help` command for more details.

15.15 `monitor` Command for System Health

Modes: Site configuration, Guest, Daily

The `monitor` command monitors and displays current health and status information for the cluster nodes and elements.

Synopsis

```
cscli monitor [-h] {nodes,elements,health} ...
```

Positional Arguments	Description
`health`	Current overall health information and status summary.
`nodes`	Current status for nodes.
`elements`	Current status for elements.

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

`monitor nodes` Subcommand

The `monitor nodes` command monitors individual nodes.

Synopsis

```
cscli monitor nodes [-h] [-y] [-v] [-n node_spec | -g genders_query] [-N
{down,unreachable,up,pending}]
```

Description	Positional Arguments
`-h | --help`	Displays the help message and exits.
`-y | --yaml`	Displays output data in YAML format.
`-v | --verbose`	Outputs extra data.
`-n` *`node_spec`* `| --node` *`node_spec`*`| --nodes` *`node_spec`*	Looks through passed hostname elements. Looks for pdsh style nodes host names (e.g. `node[100-110,120]`).
`-g` *`genders_query`*	Displays the node genders attributes query (e.g. mds=primary).
`-N {down,unreachable,up,pending} | --nodestatus {down,unreachable,up,pending}`	Displays node status.

`monitor elements` Subcommand

The `monitor elements` command monitors individual nodes.

Synopsis

```
cscli monitor elements [-h] [-y] [-v] [-n node_spec | -g genders_query] [-N
{down,unreachable,up,pending}] [-U {unknown,warning,ok,critical,pending}] [-S
element_filter]
```

Positional Arguments	Description
`-h | --help`	Displays the help message and exits.
`-y | --yaml`	Displays output data in YAML format.
`-v | --verbose`	Outputs extra data.
`-n node_spec| --node node_spec| --nodes node_spec`	Looks through passed hostname elements. Looks for **pdsh**-style nodes host names (e.g. `node[100-110,120]`).
`-g genders_query`	Displays the node genders attributes query (e.g. mds=primary).
`-N {down,unreachable,up,pending} | --nodestatus {down,unreachable,up,pending}` node status.	Displays node status.
`-U {unknown,warning,ok,critical,pending} | --elementstatus {unknown,warning,ok,critical,pending}`	Displays element status.
`-S element_filter | --search element_filter`	Searches by element name. The pattern is case-sensitive. Regular expressions are allowed.

IMPORTANT: If this command is used with no options, you may get thousands of elements on a large system.

Some `monitor` command examples and outputs are shown in *monitor Command Output Examples*.

15.16 `monitor` Command Examples

Following are examples of the `monitor` command, with outputs are including OK, WARNING and CRITICAL. See *monitor Command for System Health*.

```
[root@snx11000n000 ~]# cscli monitor health
Nodes:
up: 8   down: 0         unreachable: 0  pending: 0      total: 8
Elements:
ok: 78  warning: 0      critical: 0     unknown: 0      pending: 0      total: 78
```

No output means no errors :

```
[root@snx11000n000 ~]# cscli monitor elements -U unknown -U pending -U warning -U critical
[root@snx11000n000 ~]# cscli monitor elements -U unknown
[root@snx11000n000 ~]# cscli monitor elements -U pending
[root@snx11000n000 ~]# cscli monitor elements -U critical
[root@snx11000n000 ~]# cscli monitor elements -U warning
[root@snx11000n000 ~]# cscli monitor nodes
snx11000n000:  UP  for 28d 15h 33m 26s  checked 2014-02-06 10:00:36  "PING OK - Packet loss = 0%, RTA = 0.03 ms"
snx11000n001:  UP  for 15d 18h 14m 44s  checked 2014-02-06 10:02:56  "PING OK - Packet loss = 0%, RTA = 0.16 ms"
snx11000n002:  UP  for 15d 18h  6m 54s  checked 2014-02-06 10:05:36  "PING OK - Packet loss = 0%, RTA = 0.18 ms"
snx11000n003:  UP  for 15d 18h  8m 54s  checked 2014-02-06 10:03:36  "PING OK - Packet loss = 0%, RTA = 0.18 ms"
```

```
snx11000n003-Enclosure-R1C1-21U:  UP  for 28d 15h 35m  8s  checked 2014-02-06 09:55:36  "OK"
snx11000n004:  UP  for 5d 17h 14m 44s  checked 2014-02-06 10:01:26  "PING OK - Packet loss = 0%, RTA = 0.16 ms"
snx11000n005:  UP  for 5d 18h 30m 14s  checked 2014-02-06 10:03:26  "PING OK - Packet loss = 0%, RTA = 0.19 ms"
snx11000n005-Enclosure-R1C1-5U:  UP  for 28d 15h 34m 12s  checked 2014-02-06 10:02:36  "OK"
[root@snx11000n000 ~]# cscli monitor elements
```

Subset of output:

```
snx11000n000  "Arrays and Disk Status":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "All arrays are
operating normally"
snx11000n000  "Current Load":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "OK - load average: 0.08, 0.03,
0.02"
snx11000n000  "Current Users":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "USERS OK - 1 users currently
logged in"
snx11000n000  "Free Space":  OK  for 21d 18h 45m 53s  checked 2014-02-06 10:00:52  "DISK OK - free space: / 181915 MB
(98% inode=99%): /mnt/mgmt 778774 MB (99% inode=99%):"
snx11000n000  "Network statistics":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "NET OK - (Rx/Tx) eth0=(8.4B/
5.6B), eth1=(535.5B/349.9B), eth2=(0.0B/0.0B), eth3=(0.0B/0.0B), eth4=(0.0B/0.0B), ib0=(11.4B/0.0B), lo=(9.4B/9.4B),
meth0=(8.4B/5.6B), meth1=(0.0B/0.0B)"
snx11000n000  "RAM usage":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "OK - 11.6% (3807704 kB) used."
snx11000n000  "Swap Usage":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "SWAP OK - 100% free (31999 MB out
of 31999 MB)"
snx11000n000  "Total Processes":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "PROCS OK: 407 processes with
STATE = RSZDT"
snx11000n000  "crmd cpu usage":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "OK - Process: crmd, User: 496,
CPU: 0.0%, RAM: 0.0%, Start: Jan21, CPU Time: 127 min"
snx11000n000  "crmd memory usage":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "OK - Process: crmd, User:
496, CPU: 0.0%, RAM: 0.0%, Start: Jan21, CPU Time: 127 min"
snx11000n000  "heartbeat cpu usage":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "OK - Process: heartbeat,
User: root, CPU: 0.0%, RAM: 0.0%, Start: Jan21, CPU Time: 695 min"
snx11000n000  "heartbeat memory usage":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "OK - Process:
heartbeat, User: root, CPU: 0.0%, RAM: 0.0%, Start: Jan21, CPU Time: 695 min"
snx11000n000  "stonithd cpu usage":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "OK - Process: stonithd,
User: root, CPU: 0.0%, RAM: 0.0%, Start: Jan21, CPU Time: 60 min"
snx11000n000  "stonithd memory usage":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "OK - Process: stonithd,
User: root, CPU: 0.0%, RAM: 0.0%, Start: Jan21, CPU Time: 60 min"
snx11000n004  "Arrays and Disk Status":  OK  for 2d 14h 28m 10s  checked 2014-02-06 10:03:42  "All arrays are operating
normally"
snx11000n004  "Current Load":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42  "OK - load average: 0.01, 0.01,
0.01"
snx11000n004  "Current Users":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42  "USERS OK - 0 users currently
logged in"
snx11000n004  "Free Space":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42  "DISK OK - free space: /tmp 15966 MB
(99% inode=99%):"
snx11000n004  "Lustre Health":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42  "OK:Lustre is ok"
snx11000n004  "Network statistics":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42  "NET OK - (Rx/Tx)
eth0=(16.9B/5.8B), ib0=(169.5B/60.9B), ib1=(0.0B/0.0B), lo=(140.2B/140.2B), meth0=(16.9B/5.8B), meth1=(0.0B/0.0B),
xyvnic0=(71.5B/75.2B)"
snx11000n004  "RAM usage":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42  "OK - 12.9% (4203984 kB) used."
snx11000n004  "Swap Usage":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42  "SWAP OK - 100% free (16386 MB out
of 16386 MB)"
snx11000n004  "Total Processes":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42  "PROCS OK: 1239 processes with
STATE = RSZDT"
root@snx11000n000 ~]# cscli monitor elements -v
```

Subset of output:

```
snx11000n000  "Arrays and Disk Status":  OK  for 28d 15h 34m 45s  checked 2014-02-06 10:05:52  "All arrays are
operating normally
Array: md64, status: Ok, t10: disabled
Total number of disk slots available: 24
Total number of disks found: 24
slot:   2, wwn: 5000c50043b1e71f, cap:         450098159616, dev:   sdl, parts:        0, status: Hot Spare, t10:
11110100000
slot:  21, wwn: 5000c500479061af, cap:         450098159616, dev:   sdv, parts:        0, status: Hot Spare, t10:
11110100000
MD RAID to Lustre mapping
Array /dev/md/snx11000n003:md64 doesn't have associated WIB array
Degraded Array information:
All arrays are in clean state on node snx11000n000"
Performance Data:  None
Current Attempt:  1/3 (HARD state)
Check Type:  passive
Check Latency / Duration:  None / 0.0
Next Scheduled Active Check:  None
Last State Change:  2014-01-08 18:32:24
Last Update:  2014-02-06 10:07:06
--------------------------------------------------------------------------
snx11000n000  "Current Load":  OK  for 28d 15h 34m 45s  checked 2014-02-06 10:05:52  "OK - load average: 0.01, 0.02,
0.02"
Performance Data:  load1=0.013;1000000.000;1000000.000;0; load5=0.023;1000000.000;1000000.000;0;
load15=0.020;1000000.000;1000000.000;0;
Current Attempt:  1/3 (HARD state)
Check Type:  passive
Check Latency / Duration:  None / 0.0
```

```
Next Scheduled Active Check:  None
Last State Change:  2014-01-08 18:32:24
Last Update:  2014-02-06 10:07:06
--------------------------------------------------------------------------------
snx11000n000  "Current Users":  OK  for 28d 15h 34m 45s  checked 2014-02-06 10:05:52  "USERS OK - 1 users currently
logged in"
Performance Data:  users=1;10;50;0
Current Attempt:  1/3 (HARD state)
Check Type:  passive
Check Latency / Duration:  None / 0.0
Next Scheduled Active Check:  None
Last State Change:  2014-01-08 18:32:24
Last Update:  2014-02-06 10:07:06
 [root@snx11000n000 ~]# cscli monitor elements -S enclosures
snx11000n003-Enclosure-R1C1-21U  "FRU Fan Status":  OK  for 28d 15h 27m 54s  checked 2014-02-06 09:55:52  "All FRU's
are operating normally"
snx11000n003-Enclosure-R1C1-21U  "FRU Power Supply Status":  OK  for 28d 15h 27m 54s  checked 2014-02-06 09:55:52  "All
FRU's are operating normally"
snx11000n003-Enclosure-R1C1-21U  "FRU SBB Module Status":  OK  for 28d 15h 27m 54s  checked 2014-02-06 09:55:52  "All
FRU's are operating normally"
snx11000n003-Enclosure-R1C1-21U  "Fan Statistics":  OK  for 15d 17h 19m  4s  checked 2014-02-06 09:58:06  "Summary: 4
Fan Sensors available. All Sensors readings are within normal operating levels"
snx11000n003-Enclosure-R1C1-21U  "Power Statistics":  OK  for 15d 17h 19m  4s  checked 2014-02-06 09:58:06  "Summary:
Total System Power 168W"
snx11000n003-Enclosure-R1C1-21U  "Thermal Statistics":  OK  for 15d 17h 19m  4s  checked 2014-02-06 09:58:06  "Summary:
6 Thermal Sensors available. All Sensors readings are within normal operating levels"
snx11000n003-Enclosure-R1C1-21U  "Voltage Statistics":  OK  for 15d 17h 19m  4s  checked 2014-02-06 09:58:06  "Summary:
4 Voltage Sensors available. All Sensors readings are within normal operating levels"
snx11000n005-Enclosure-R1C1-5U  "FRU Fan Status":  OK  for 28d 15h 27m 54s  checked 2014-02-06 09:55:52  "All FRU's are
operating normally"
snx11000n005-Enclosure-R1C1-5U  "FRU Power Supply Status":  OK  for 20d 23h  0m 23s  checked 2014-02-06 09:55:52  "All
FRU's are operating normally"
snx11000n005-Enclosure-R1C1-5U  "FRU SBB Module Status":  OK  for 28d 15h 27m 54s  checked 2014-02-06 09:55:52  "All
FRU's are operating normally"
snx11000n005-Enclosure-R1C1-5U  "Fan Statistics":  OK  for 28d 15h 28m 40s  checked 2014-02-06 09:57:42  "Summary: 10
Fan Sensors available. All Sensors readings are within normal operating levels"
snx11000n005-Enclosure-R1C1-5U  "Power Statistics":  OK  for 28d 15h 28m 40s  checked 2014-02-06 09:57:42  "Summary:
Total System Power 1068W"
snx11000n005-Enclosure-R1C1-5U  "Thermal Statistics":  OK  for 28d 15h 28m 40s  checked 2014-02-06 09:57:42  "Summary:
13 Thermal Sensors available. All Sensors readings are within normal operating levels"
snx11000n005-Enclosure-R1C1-5U  "Voltage Statistics":  OK  for 28d 15h 28m 40s  checked 2014-02-06 09:57:42  "Summary:
2 Voltage Sensors available. All Sensors readings are within normal operating levels"
[root@snx11000n000 ~]#
[root@snx11000n000 ~]# cscli monitor nodes -n snx11000n004
snx11000n004:  UP  for 5d 17h 17m 26s  checked 2014-02-06 10:06:36  "PING OK - Packet loss = 0%, RTA = 0.17 ms"
[root@snx11000n000 ~]# cscli monitor elements  -n snx11000n004
snx11000n004  "Arrays and Disk Status":  OK  for 2d 14h 30m 42s  checked 2014-02-06 10:03:42  "All arrays are operating
normally"
snx11000n004  "Current Load":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42  "OK - load average: 0.01, 0.01,
0.01"
snx11000n004  "Current Users":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42  "USERS OK - 0 users currently
logged in"
snx11000n004  "Free Space":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42  "DISK OK - free space: /tmp 15966 MB
(99% inode=99%):"
snx11000n004  "Lustre Health":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42  "OK:Lustre is ok"
snx11000n004  "Network statistics":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42  "NET OK - (Rx/Tx)
eth0=(16.9B/5.8B), ib0=(169.5B/60.9B), ib1=(0.0B/0.0B), lo=(140.2B/140.2B), meth0=(16.9B/5.8B), meth1=(0.0B/0.0B),
xyvnic0=(71.5B/75.2B)"
snx11000n004  "RAM usage":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42  "OK - 12.9% (4203984 kB) used."
snx11000n004  "Swap Usage":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42  "SWAP OK - 100% free (16386 MB out
of 16386 MB)"
snx11000n004  "Total Processes":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42  "PROCS OK: 1239 processes with
STATE = RSZDT"
[root@snx11000n000 ~]# cscli monitor elements  -g oss
snx11000n004  "Arrays and Disk Status":  OK  for 2d 14h 31m 42s  checked 2014-02-06 10:08:43  "All arrays are operating
normally"
snx11000n004  "Current Load":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43  "OK - load average: 0.01, 0.01,
0.01"
snx11000n004  "Current Users":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43  "USERS OK - 0 users currently
logged in"
snx11000n004  "Free Space":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43  "DISK OK - free space: /tmp 15966 MB
(99% inode=99%):"
snx11000n004  "Lustre Health":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43  "OK:Lustre is ok"
snx11000n004  "Network statistics":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43  "NET OK - (Rx/Tx)
eth0=(16.9B/5.8B), ib0=(169.5B/60.9B), ib1=(0.0B/0.0B), lo=(140.3B/140.3B), meth0=(16.9B/5.8B), meth1=(0.0B/0.0B),
xyvnic0=(71.5B/75.2B)"
snx11000n004  "RAM usage":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43  "OK - 12.9% (4204568 kB) used."
snx11000n004  "Swap Usage":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43  "SWAP OK - 100% free (16386 MB out
of 16386 MB)"
snx11000n004  "Total Processes":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43  "PROCS OK: 1239 processes with
STATE = RSZDT"
snx11000n005  "Arrays and Disk Status":  OK  for 17d 14h 22m 41s  checked 2014-02-06 10:07:41  "All arrays are
operating normally"
snx11000n005  "Current Load":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41  "OK - load average: 0.01, 0.02,
0.02"
snx11000n005  "Current Users":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41  "USERS OK - 0 users currently
logged in"
```

```
snx11000n005  "Free Space":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41  "DISK OK - free space: /tmp 15966 MB
(99% inode=99%):"
snx11000n005  "Lustre Health":  OK  for 28d 15h 38m  7s  checked 2014-02-06 10:07:42  "OK:Lustre is ok"
snx11000n005  "Network statistics":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41  "NET OK - (Rx/Tx)
eth0=(25.8B/16.5B), ib0=(197.3B/64.8B), ib1=(0.0B/0.0B), lo=(6.0B/6.0B), meth0=(25.8B/16.5B), meth1=(0.0B/0.0B),
xyvnic0=(70.9B/76.1B)"
snx11000n005  "RAM usage":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41  "OK - 12.8% (4192544 kB) used."
snx11000n005  "Swap Usage":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41  "SWAP OK - 100% free (16386 MB out
of 16386 MB)"
snx11000n005  "Total Processes":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41  "PROCS OK: 1241 processes with
STATE = RSZDT"
[root@snx11000n000 ~]#
[root@snx11000n000 ~]# cscli monitor elements -S arrays
snx11000n000  "Arrays and Disk Status":  OK  for 28d 15h 38m 14s  checked 2014-02-06 10:05:52  "All arrays are
operating normally"
snx11000n001  "Arrays and Disk Status":  OK  for 28d 15h 39m 56s  checked 2014-02-06 10:08:07  "All arrays are
operating normally"
snx11000n002  "Arrays and Disk Status":  OK  for 28d 15h 36m 38s  checked 2014-02-06 10:07:56  "All arrays are
operating normally"
snx11000n003  "Arrays and Disk Status":  OK  for 28d 15h 36m 36s  checked 2014-02-06 10:06:24  "All arrays are
operating normally"
snx11000n004  "Arrays and Disk Status":  OK  for 2d 14h 32m 35s  checked 2014-02-06 10:08:43  "All arrays are operating
normally"
snx11000n005  "Arrays and Disk Status":  OK  for 17d 14h 23m 34s  checked 2014-02-06 10:07:41  "All arrays are
operating normally"
[root@snx11000n000 ~]# cscli monitor elements -S arrays –v
```

Subset of output:

```
snx11000n000  "Arrays and Disk Status":  OK  for 28d 15h 39m 16s  checked 2014-02-06 10:10:52  "All arrays are
operating normally
Array: md64, status: Ok, t10: disabled
Total number of disk slots available: 24
Total number of disks found: 24
slot:   2, wwn: 5000c50043b1e71f, cap:         450098159616, dev:   sdl, parts:        0, status: Hot Spare, t10:
11110100000
slot:  21, wwn: 5000c500479061af, cap:         450098159616, dev:   sdv, parts:        0, status: Hot Spare, t10:
11110100000
MD RAID to Lustre mapping
Array /dev/md/snx11000n003:md64 doesn't have associated WIB array
Degraded Array information:
All arrays are in clean state on node snx11000n000"
Performance Data:  None
Current Attempt:  1/3 (HARD state)
Check Type:  passive
Check Latency / Duration:  None / 0.0
Next Scheduled Active Check:  None
Last State Change:  2014-01-08 18:32:24
Last Update:  2014-02-06 10:11:36
--------------------------------------------------------------------------------
snx11000n001  "Arrays and Disk Status":  OK  for 28d 15h 40m 58s  checked 2014-02-06 10:08:07  "All arrays are
operating normally
Array: md67, status: Ok, t10: disabled
Array: md127, status: Ok, t10: disabled
Total number of disk slots available: 24
Total number of disks found: 24
slot:   2, wwn: 5000c50043b1e71f, cap:         450098159616, dev:   sdv, parts:        0, status: Hot Spare, t10:
11110100000
slot:  21, wwn: 5000c500479061af, cap:         450098159616, dev:   sdc, parts:        0, status: Hot Spare, t10:
11110100000
MD RAID to Lustre mapping
Array /dev/md/snx11000n003:md67 doesn't have associated WIB array
Degraded Array information:
All arrays are in clean state on node snx11000n001"
Performance Data:  None
Current Attempt:  1/3 (HARD state)
Check Type:  passive
Check Latency / Duration:  None / 0.0
Next Scheduled Active Check:  None
Last State Change:  2014-01-08 18:30:42
Last Update:  2014-02-06 10:11:36
--------------------------------------------------------------------------------
snx11000n002  "Arrays and Disk Status":  OK  for 28d 15h 37m 40s  checked 2014-02-06 10:07:56  "All arrays are
operating normally
Array: md65, status: Ok, t10: disabled
Total number of disk slots available: 24
Total number of disks found: 24
slot:   2, wwn: 5000c50043b1e71f, cap:         450098159616, dev:   sdv, parts:        0, status: Hot Spare, dev1:
sdaj, t10: 11110100000
slot:  21, wwn: 5000c500479061af, cap:         450098159616, dev:   sdc, parts:        0, status: Hot Spare, dev1:
sdat, t10: 11110100000
MD RAID to Lustre mapping
Array /dev/md/snx11000n003:md65 doesn't have associated WIB array
Target:     MGS
Degraded Array information:
All arrays are in clean state on node snx11000n002"
```

```
Performance Data:  None
Current Attempt:  1/3 (HARD state)
Check Type:  passive
Check Latency / Duration:  None / 0.0
Next Scheduled Active Check:  None
Last State Change:  2014-01-08 18:34:00
Last Update:  2014-02-06 10:11:36
----------------------------------------------------------------------
[root@snx11000n000 ~]# cscli monitor elements -S disk
snx11000n000  "Arrays and Disk Status":  OK  for 28d 15h 43m 32s  checked 2014-02-06 10:10:52  "All arrays are
operating normally"
snx11000n001  "Arrays and Disk Status":  OK  for 28d 15h 45m 14s  checked 2014-02-06 10:13:07  "All arrays are
operating normally"
snx11000n002  "Arrays and Disk Status":  OK  for 28d 15h 41m 56s  checked 2014-02-06 10:12:56  "All arrays are
operating normally"
snx11000n003  "Arrays and Disk Status":  OK  for 28d 15h 41m 54s  checked 2014-02-06 10:11:24  "All arrays are
operating normally"
snx11000n004  "Arrays and Disk Status":  OK  for 2d 14h 37m 53s  checked 2014-02-06 10:13:42  "All arrays are operating
normally"
snx11000n005  "Arrays and Disk Status":  OK  for 17d 14h 28m 52s  checked 2014-02-06 10:12:41  "All arrays are
operating normally"
root@snx11000n000 ~]# cscli monitor elements -S fan
snx11000n003-Enclosure-R1C1-21U  "FRU Fan Status":  OK  for 28d 15h 43m 55s  checked 2014-02-06 10:10:52  "All FRU's
are operating normally"
snx11000n003-Enclosure-R1C1-21U  "Fan Statistics":  OK  for 15d 17h 35m  5s  checked 2014-02-06 10:13:07  "Summary: 4
Fan Sensors available. All Sensors readings are within normal operating levels"
snx11000n005-Enclosure-R1C1-5U  "FRU Fan Status":  OK  for 28d 15h 43m 55s  checked 2014-02-06 10:10:52  "All FRU's are
operating normally"
snx11000n005-Enclosure-R1C1-5U  "Fan Statistics":  OK  for 28d 15h 44m 41s  checked 2014-02-06 10:12:42  "Summary: 10
Fan Sensors available. All Sensors readings are within normal operating levels"
[root@snx11000n000 ~]# cscli monitor elements -S power
snx11000n003-Enclosure-R1C1-21U  "FRU Power Supply Status":  OK  for 28d 15h 44m  8s  checked 2014-02-06 10:15:53  "All
FRU's are operating normally"
snx11000n003-Enclosure-R1C1-21U  "Power Statistics":  OK  for 15d 17h 35m 18s  checked 2014-02-06 10:16:25  "Summary:
Total System Power 178W"
snx11000n005-Enclosure-R1C1-5U  "FRU Power Supply Status":  OK  for 20d 23h 16m 37s  checked 2014-02-06 10:15:53  "All
FRU's are operating normally"
snx11000n005-Enclosure-R1C1-5U  "Power Statistics":  OK  for 28d 15h 44m 54s  checked 2014-02-06 10:12:42  "Summary:
Total System Power 1061W"
[root@snx11000n000 ~]# cscli monitor elements -S sbb
snx11000n003-Enclosure-R1C1-21U  "FRU SBB Module Status":  OK  for 28d 15h 44m 23s  checked 2014-02-06 10:15:53  "All
FRU's are operating normally"
snx11000n005-Enclosure-R1C1-5U  "FRU SBB Module Status":  OK  for 28d 15h 44m 23s  checked 2014-02-06 10:15:53  "All
FRU's are operating normally"
[root@snx11000n000 ~]# cscli monitor elements -S volt
snx11000n003-Enclosure-R1C1-21U  "Voltage Statistics":  OK  for 15d 17h 35m 53s  checked 2014-02-06 10:16:24  "Summary:
4 Voltage Sensors available. All Sensors readings are within normal operating levels"
snx11000n005-Enclosure-R1C1-5U  "Voltage Statistics":  OK  for 28d 15h 45m 29s  checked 2014-02-06 10:12:42  "Summary:
2 Voltage Sensors available. All Sensors readings are within normal operating levels"
[root@snx11000n000 ~]# cscli monitor elements -S disk
snx11000n000  "Arrays and Disk Status":  OK  for 28d 15h 45m  0s  checked 2014-02-06 10:15:53  "All arrays are
operating normally"
snx11000n001  "Arrays and Disk Status":  OK  for 28d 15h 46m 42s  checked 2014-02-06 10:13:07  "All arrays are
operating normally"
snx11000n002  "Arrays and Disk Status":  OK  for 28d 15h 43m 24s  checked 2014-02-06 10:12:56  "All arrays are
operating normally"
snx11000n003  "Arrays and Disk Status":  OK  for 28d 15h 43m 22s  checked 2014-02-06 10:16:24  "All arrays are
operating normally"
snx11000n004  "Arrays and Disk Status":  OK  for 2d 14h 39m 21s  checked 2014-02-06 10:13:42  "All arrays are operating
normally"
snx11000n005  "Arrays and Disk Status":  OK  for 17d 14h 30m 20s  checked 2014-02-06 10:12:41  "All arrays are
operating normally"
[root@snx11000n000 ~]# cscli monitor elements -S arrays
snx11000n000  "Arrays and Disk Status":  OK  for 28d 15h 45m 10s  checked 2014-02-06 10:15:53  "All arrays are
operating normally"
snx11000n001  "Arrays and Disk Status":  OK  for 28d 15h 46m 52s  checked 2014-02-06 10:13:07  "All arrays are
operating normally"
snx11000n002  "Arrays and Disk Status":  OK  for 28d 15h 43m 34s  checked 2014-02-06 10:12:56  "All arrays are
operating normally"
snx11000n003  "Arrays and Disk Status":  OK  for 28d 15h 43m 32s  checked 2014-02-06 10:16:24  "All arrays are
operating normally"
snx11000n004  "Arrays and Disk Status":  OK  for 2d 14h 39m 31s  checked 2014-02-06 10:13:42  "All arrays are operating
normally"
snx11000n005  "Arrays and Disk Status":  OK  for 17d 14h 30m 30s  checked 2014-02-06 10:12:41  "All arrays are
operating normally"
[root@snx11000n000 ~]#
```

15.17 `netfilter_level` Command

Modes: Site configuration, Daily

The Sonexion `netfilter_level` command manages the netfilter level on the Sonexion system.

IMPORTANT: Exercise caution before using the `--force` parameter.

Synopsis

```
cscli netfilter_level [-h] [-s] [-l level] [--force]
```

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-s |--show`	Shows the current netfilter level.
`-l level |--level level`	Sets the netfilter level (`off`, `lustre`, `on`).
`--force`	Forces the netfilter level to be set to off.

15.18 Network Setup Commands

The Sonexion commands with names ending in `network_setup` manage network parameters for the Lustre file system. This command includes functions to show, set, apply, and reset Lustre network parameters.

Show Network Parameters

Modes: Site configuration, Guest

The `show_network_setup` command displays the Lustre network configuration. If the Lustre network is not yet configured, no parameters are shown.

Synopsis

```
cscli show_network_setup [-h] [-c cluster_name]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-c cluster_name |--cluster cluster_name`	Specifies the cluster name.

Set Network Parameters

Modes: Site configuration

The `set_network` command specifies new Lustre network parameters and adds them to the database.

Synopsis

```
cscli set_network [-h] -k netmask -r ipranges [-d dns] [-t ntp] [-c cluster_name]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-k` *`netmask`*`|--netmask` *`netmask`*	Specifies the network mask value of the IP address.
`-r` *`ipranges`*`|--range` *`ipranges`*	Specifies the IP address range.
`-d` *`dns`* `|--dns` *`dns`*	Specifies the DNS server IP address (optional).
`-t` *`ntp`* `|--ntp` *`ntp`*	Specifies the NTP server's IP address (optional).
`-c` *`cluster_name`* `|--cluster` *`cluster_name`*	Specifies the cluster name.

Reset Network Parameters

Modes: Site configuration

The `reset_network_setup` command resets the Lustre network parameters by removing old values from the database and replacing them with default values.

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

Synopsis

```
cscli reset_network_setup [-h] [-y] [-c cluster_name]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-y |--yes`	Confirms the action to reset the network parameters.
`-c` *`cluster_name`*`|--cluster` *`cluster_name`*	Specifies the cluster name.

Apply Network Parameters

Modes: Site configuration

The `apply_network_setup` command applies new Lustre network parameters to the database.

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

Synopsis

```
cscli apply_network_setup [-h] [--yes] [-c cluster_name]
```

Optional Arguments	Description
`-h|--help`	Displays the help message and exits.
`|--yes`	Confirms the action that network setup parameters were applied.
`-c` *`cluster_name`*`|--cluster` *`cluster_name`*	Specifies the cluster name.

15.19 Node Control Commands

The Sonexion Lustre node control commands are used on MDS/MGS and OSS nodes in a clustered file system. The commands include functions to mount and unmount the Lustre nodes, show nodes in the file system. Additional functions include powering nodes on and off, managing node failover and failback, managing node auto-discovery and controlling CNG nodes.

Node Auto-Discovery Command

Modes: Site configuration, Daily

The `autodiscovery_mode` command manages node auto-discovery in the Sonexion system.

Synopsis

```
cscli autodiscovery_mode [-h] [-s] [--mode {enabled,disabled}]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-s |--status`	Indicates the status of the auto-discovery mode.
`--mode {enabled,disabled}`	Switches to the specified mode. Enables or Disables the auto-discovery mode.

Node Failback and Failover Commands

Modes: Daily

The `failover` and `failback` commands manage node failback and failover in the Sonexion system.

Synopsis

```
cscli failback [-h] (-F filter_sid | -n node_spec) |-c cluster_name |--cluster
cluster_name
```

```
cscli failover [-h] (-F filter_sid | -n node_spec) |-c cluster_name |--cluster
cluster_name
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-f filter_sid |--filter filter_sid`	The filter identifier for the specified node. Failover/ failback actions run on the nodes by filtering this filter.
`-n node_spec |--nodes node_spec`	Specifies the nodes on which the failover/failback operations are performed. Node hostnames should be passed in pdsh style. If this parameter is passed, the `--filter` parameter is ignored.
`-c cluster_name|--cluster cluster_name`	This parameter is deprecated. It is supported only for backward compatibility.

Lustre Target Mount and Unmount Commands

Modes: Site configuration, Daily

The `mount` and `unmount` commands control file system access to the Lustre targets (MDS/MGS and OSSs). The mount action enables file system access to the node. The unmount action disables file system access to the node.If one or more nodes are specified, the mount/unmount action is performed only on the selected nodes in the file system.

If no server nodes are specified, then the mount/unmount action is performed on all server nodes in the file system.

NOTE: Exercise caution before using the `--force` parameter.

Synopsis

```
cscli mount [-h] -f fs_name [-n node_spec] |-c cluster_name |--cluster cluster_name
```

```
cscli unmount [-h] -f fs_name [-n node_spec] |-c cluster_name |--cluster
cluster_name [-evict] [-force]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-f fs_name |--fs-name=`*`fs_name`*	Specifies the name of the file system.
`-n node_spec |--nodes=`*`node_spec`*	Specifies the node(s) on which the mount/unmount action is performed. Node hostnames should be passed in pdsh style.
`-c` *`cluster_name`*`|--cluster` *`cluster_name`*	This parameter is deprecated. It is supported only for backward compatibility.
`|--evict`	This parameter evicts clients before they unmount.
`|--force`	This parameter is the force mode to evict lustre clients.

Node Power Manage Command

Modes: Daily

The `power_manage` command manages the power on the Sonexion system. These commands power-cycle nodes on and off and also control HA resource hand-offs.

IMPORTANT: Exercise caution before using the `--force` parameter.

Synopsis

```
cscli power_manage [-h] (--filter filter_sid | -n node_spec) (--power-on | --power-
off | --reboot | --cycle | --reset | --hand-over) [--force] -c cluster_name, --
cluster cluster_name
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-f` *`filter_sid`* `|--filter` *`filter_sid`*	The filter identifier for the specified node. Failover and failback actions run on the nodes by filtering this filter. If `--filter` is specified, then `--nodes` is ignored.
`-n` *`node_spec`*	Specifies the nodes on which failover/failback operations are performed. Node hostnames should be passed in pdsh style.

Optional Arguments	Description
`--power-on`	Powers on the specified nodes.
`--power-off`	Powers off the specified nodes.
`--reboot`	Reboots the specified nodes.
`--cycle`	Power-cycles the specified nodes.
`--reset`	Resets the specified nodes.
`--hand-over`	Hands over resources.
`--force`	An optional flag that indicates the node operation should be performed in force mode; should only be used with --power-off.
`-c` *`cluster_name`*`|--cluster` *`cluster_name`*	This parameter is deprecated. It is supported only for backward compatibility.

Show Node Information

Modes: Guest, Daily

The `show_nodes` command displays information about specified system nodes. Nodes in the ADU (MDS nodes) will also display if they are configured.

Synopsis

```
cscli show_nodes [-h] [-F filter_sid] [-r] |-c cluster_name |--cluster cluster_name
```

Option	Description
`-h |--help`	Displays the help message and exits.
`-F` *`filter_sid`* `|--filter` *`filter_sid`*	Specifies the node filter.
`-r |--refresh`	Specifies the refresh mode (press **q** for quit).
`-c` *`cluster_name`*`|--cluster` *`cluster_name`*	This parameter is deprecated. It is supported only for backward compatibility.

15.20 Node Position and Management Commands

The Sonexion rack position commands manage the location of components (hosting system nodes) in a Sonexion rack. The MMU hosts the primary and secondary MGMT, MGS and MDS nodes. Each SSU hosts OSS nodes (two OSSes per SSU).

Get Node Position in a Sonexion Rack

Modes: Daily

The `get_rack_position` command indicates the location of server nodes in a Sonexion rack.

Synopsis

```
cscli get_rack_position [-h] [-l] [-r racks] [-a] [--yaml]
```

Option	Description
`-h |--help`	Displays the help message and exits.
`--yaml`	Prints node rack position information in YAML file format.
`-l |--list`	Displays the names of all the racks.
`-r racks|--racks racks`	Displays the hostnames in all positions within the specified racks. *racks* is a comma-separated list of rack names.
`-a |--all`	Displays information about all racks.

Set Node Position in a Sonexion Rack

Modes: Daily

The `set_rack_position` command sets the location of server nodes in the Sonexion rack. It changes the position of one node in the rack (or moves the node to another rack).

IMPORTANT: Exercise caution before using the `--force` parameter.

Synopsis

```
cscli set_rack_position [-h] (-y yaml_path | -r rack_name -n node_name -p position)
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`--force`	Creates a new rack if not found.
`-y yaml path |--yaml yaml_path`	Loads rack position information from *yaml* file format.
`-r rack_name |--rack rack_name`	Specifies the rack containing the node(s). This can be set manually.
`-n node_spec |--node node_spec`	Specifies the node(s) hostname. This can be set manually.
`-p position |--position=position`	Specifies the node position in rack units. This can be set manually.

Show Available Software Versions

Modes: Daily

The `show_update_versions` command lists software versions available in the Sonexion Management (MGMT) Server repository.

Synopsis

```
cscli show_update_versions [-h]
```

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

Show Nodes at Specified Software Version

Modes: Daily

The `show_version_nodes` command lists all system nodes at the specified software version.

Synopsis

```
cscli show_version_nodes [-h] [-q] -v sw_version
```

Option	Description
`-h |--help`	Displays the help message and exits.
`-q |--query`	Controls the format of the command output. If this flag is specified, nodes in output should display in genders style. For example, `snx11000n[002-005,097-098]`.
`-v sw_version|--version sw_version`	Specifies the Sonexion software version.

Show Node Versions

Modes: Daily

The `show_node_versions` command displays the Sonexion software version running on specified nodes.

Synopsis

```
cscli show_node_versions [-h] [-q] [-n node_spec] [-g genders_query]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-q, --query`	Controls output format. If this flag is specified, nodes in the output should be in genders style. For example, `snx11000n[002-005,097-098]`
`-n node_spec, --nodes node_spec`	Specifies nodes to indicate the Sonexion software version.
`-g genders_query`	Specifies a gender's style query.

15.21 RAID Management Commands

Enable RAID Checking Command

Modes: Daily

The Sonexion `raid_check` command enables RAID check on RAID devices.

Synopsis

```
cscli raid_check -h (-a | -n node_list) [-i] [-c {on,off}] [--now] [-s a_time]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-a |--all`	Looks through all nodes elements.
`-n node_list |--node node_list`	Looks through passed hostname elements. Looks for **pdsh**-style nodes host names.
`-i |--info`	Prints the current RAID check status for selected nodes.
`-c {on,off} |--cron {on,off}`	Enables/Disables the cron job for the RAID check.
`--now`	Performs the RAID check now.
`-s a_time |--set a_time`	Specifies a string to set a time to run the RAID check.

Manage RAID Rebuild Rate Command

Modes: Site configuration

The `rebuild_rate` command manages the RAID recovery rate management for data arrays. It sets the minimum rate in KB/s at which the system will perform recovery tasks after drive failure(s). The `rebuild_rate` command displays the numerical values when the rebuild is in progress. The minimum rates are now displayed on all nodes via `cscli rebuild_rate`.

Synopsis

```
cscli rebuild_rate [-h] [-n nodes] [--reset] [-1 single_rate] [-m multiple_rate] [-
b rebalance_rate]
```

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-n nodes |--node nodes`	Specifies pdsh-style node hostnames (e.g. `node[100-110,120]`). Sets global rates by omitting this parameter.
`--reset`	Reset rates to global values.
`-1 single_rate |--after-first-failure single_rate`	After a single drive failure, use this rate to recover array redundancy. For RAID-6, this is the rebuild rate. For GridRAID, this is the reconstruction rate
`-b rebalance_rate |--rebalance rebalance_rate`	For GridRAID, use this rate to rebalance data, parity, and spare space after reconstruction is completed for a single drive failure.

Optional Arguments	Description
	For RAID-6, this parameter is ignored.
`-m` *`multiple_rate`*`|--after-multiple-failures` *`multiple_rate`*	After multiple drive failure, increase to this rate. For RAID-6, this is the rebuild rate. For GridRAID, this is the reconstruction rate and the rebalance rate.

CLI Parameter	Diskmonitor
`-1`	Degraded parameter.
`-b`	Sets the diskmonitor with a new ‘recover’ parameter.
`-m`	Sets the "boost" rate for a `-m` parameter.

```
[root@snx11000n000 ~]# cscli rebuild_rate -n snx11000 [000-002,004-007]
KB/Sec:
--------------------------------------------------------------------------------
Node           Rebuild rate single     Rebalance rate      Rebuild rate multiple
--------------------------------------------------------------------------------
snx11000n000          50000                  50000                  80000
snx11000n001          50000                  50000                  80000
snx11000n002          50000                  50000                  80000
snx11000n004          50000                  50000                  80000
snx11000n005          50000                  50000                  80000
snx11000n006          50000                  50000                  80000
snx11000n007          50000                  50000                  80000
global                50000                  50000                  80000
```

15.22 Rack Management Command

Modes: Daily

The Sonexion `rack` command has options to `list`, `show`, `create`, `delete`, `move`, and `rename` racks.

Synopsis

```
cscli rack [-h] {rename,show,create,move,list,delete} ...
```

where:

Positional Arguments	Description
`-list`	Lists racks
`-show`	Shows racks
`-create`	Creates racks
`-delete`	Deletes racks

Positional Arguments	Description
`-move`	Moves racks
`-rename`	Renames racks

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

15.23 `service_console` Commands

Modes: Daily

The Sonexion `service_console` command lets a site administrator manage the system service console and block or allow its use for a service technician.

The service console is an alternate system management interface that is limited to maintenance functions, and is for use by authorized service personnel only. Even though it bypasses the normal administrative login, the service console cannot be used to access the root account, or to access data on the Lustre file system.

While it is not necessary to disable the service console for any system that is physically secure, administrators in high-security environments might prefer to disable it. In that case, they will need to re-enable the service console each and every time a service technician arrives on site to perform system maintenance, including replacement of failed disk drives and other parts.

Configuration consists of two main areas

- Configuring the SMTP relay, this allows the system to send emails to users and service emails.
- Configuring user email addresses to allow service email to go to system users.

This following commands and available subcommands are used to manage the service console.

Synopsis

```
cscli service_console -h
```

Usage:

```
cscli service_console [options]
```

where:

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

Sub-commands	Description
`enable`	Enables the service console
`disable`	Disables the service console

Sub-commands	Description
show	Show service console enabled status
notifications	View RAS notifications
show	Shows current RAS notifications
configure	Configure RAS notifications

Examples:

```
$ cscli service_console enable
Service Console enabled.
```

```
$ cscli service_console disable
Service Console disabled.
```

```
$ cscli service_console show
Service console enabled.
```

Show service console notifications (service events).

```
$ cscli service_console notifications show
Current outstanding service call events:

Item 1 : "Disk drive needs replacement"

Service Code: 002005001
Time the event was first detected: Wed, 17 Jun 2015 17:18:14 EDT

    Details of failed component
    Disk Serial #: S0M122HN0000B40298QD
    Disk Model: ST600MM0006 Drive
    Manufacturer: SEAGATE
    Firmware version of drive at time of failure: XLGE T10     Enabled: Yes

Location of failed component

    Rack Name: Rack1
    Enclosure Model: 2U24
    Enclosure Location: 36U
    Disk located in slot: 15

Item 2: "Power supply issue detected"

Service Code: 002005003
Time the event was first detected: Wed, 02 Dec 2015 12:58:57 PST

Details of failed component

    Power supply chassis type: OneStor Power One 764W_AC_PCM (Original Silver DFM).
    Power supply part number: 0945768-10
    Power supply product version: 0314
    Power supply serial number: PMW0945768J1BDY

Location of failed component

    Rack Name: R1C1
```

```
    Enclosure Model: 2U24
    Enclosure Location: 24U
    Power supply located in bay: 1
```

Show service console configuration options.

```
$ cscli service_console configure
Commands:
    smtp            Configure SMTP settings
    snmp            Configure SNMP settings
    email           Configure email for sending RAS notifications
    system          Configure system settings
    rest_api        Configure REST API settings
```

Service Console Configuration Commands

These commands are used to configure the SMTP relay, SNMP, email addresses for RAS notifications, system and REST API settings.

Synopsis

```
cscli service_console configure [options]
```

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

Subcommands	Description
`smtp`	Configures SMTP settings
`snmp`	Configures SNMP settings
`email`	Configure email for sending RAS notifications
`system`	Configure system settings
`rest_api`	Configure REST API settings

SMTP Commands

This command provides two SMTP options, configuring the SMTP relay and showing the SMTP configuration.

Synopsis

```
cscli service_console configure smtp [options]
```

```
cscli service_console configure smtp relay [options]
```

```
cscli service_console configure smtp show
```

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

Subcommands	Description
`relay`	Configures SMTP relay for sending RAS notifications
`-H |--host`	SMTP host (required)
`-P |--port`	SMTP port [default: 25]
`show`	Displays current SMTP configuration

Examples:

Configure the SMTP relay.

```
$ cscli service_console configure smtp relay -H
mailrelayus.xyus.xyratex.com
OK.
```

Show the current SMTP configuration.

```
$ $ cscli service_console configure smtp show
SMTP relay: mailrelayus.xyus.xyratex.com:25
```

SNMP Commands

This command provides options to configure SNMP (enable or disable) and show the current SNMP setting.

```
$ cscli service_console configure snmp [options]
```

Synopsis

```
$ cscli service_console configure snmp [options]
```

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

Subcommands	Description
`enable`	Enables SNMP monitoring
`disable`	Disables SNMP monitoring
`show`	Shows current SNMP status

Examples:

Enable SNMP monitoring.

```
$ cscli service_console configure snmp enable
SNMP has successfully been enabled.
```

Disable SNMP monitoring.

```
$ cscli service_console configure snmp disable
SNMP has successfully been disabled.
```

Show the status of SNMP monitoring (enabled).

```
$ cscli service_console configure snmp show
SNMP Enabled:
    Yes
SNMP Community:
    Public
SNMP Version:
    2c
```

Show the status of SNMP monitoring (disabled).

```
$ cscli service_console configure snmp show
SNMP Enabled:
    No
SNMP Community:
    Public
SNMP Version:
    2c
```

Email Notification Commands

This command configures email notifications, including enabling or disabling notification functionality, adds or deletes email addresses to the user list, shows user list entries and sends test email to determine if user notifications can be sent successfully.

Synopsis

```
cscli service_console configure email [options]
```

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

Subcommands	Description
`show`	Shows all notification email addresses
`add`	Adds a notification email address
`-A |--address`	Email address (required)
delete	Deletes a notification email address
`-A |--address`	Email address (required)
`enable`	Enables email notifications
`disable`	Disables email notifications
`send_test_email`	Sends test email to all enabled services

Examples:

Show email addresses configured to receive RAS notifications (no configured addresses).

```
$ cscli service_console configure email show
No email addresses are configured to receive service notifications.
```

Show email addresses configured to receive RAS notifications (several configured addresses).

```
$ cscli service_console configure email show
The following email addresses are configured to receive service notifications:
sam_jones@xyzcorp.com
amy_cooper@xyzcorp.com
```

Add an email address to the user list (success).

```
$ cscli service_console configure email add –A sam_jones@xyzcorp.com
Successfully added 'sam_jones@xyzcorp.com' to receive notification emails.
```

Add an email address to the user list (fails because it is currently associated with the user list).

```
$ cscli service_console configure email add –A sam_jones@xyzcorp.com
Unable to add: 'sam_jones@xyzcorp.com' is already configured for receiving
notification emails.
```

Delete an email address from the user list (success).

```
$ cscli service_console configure email delete –A sam_jones@xyzcorp.com
Successfully removed 'sam_jones@xyzcorp.com' from receiving notification emails.
```

Delete an email address from the user list (fails because it is not associated with the user list).

```
$ cscli service_console configure email add –A sam_jones@xyzcorp.com
Unable to delete: 'sam_jones@xyzcorp.com' is not configured for receiving
notification emails.
```

Enable email notifications.

```
$ cscli service_console configure email enable
OK.
```

Disable email notifications.

```
$ cscli service_console configure email disable
OK.
```

System Commands

This command provides options to set the Sonexion system identifier (name) and show the current identifier.

Synopsis

```
$ cscli service_console configure system [options]
```

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

Subcommands	Description
identifier	Sets the system identifier
show	Shows the current system identifier (name)

Examples:

Set the system identifier.

```
$ cscli service_console configure system identifier –name=snx-002
System identifier name has been set to: snx-002
```

Show the current system identifier (name specified).

```
$ cscli service_console configure system show
System settings:
    System identifier name: snx-002
```

Show the current system identifier (no name set).

```
$ cscli service_console configure system show
```

REST API Commands

This command manages REST API access settings with the option to show current status.

Synopsis

```
cscli service_console configure rest-api [options]
```

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

Subcommands	Description
enable	Enables the REST API
disable	Disables the REST API
show	Shows current status of the REST API

Enable REST API access

```
$ cscli service_console configure rest-api enable
REST API has been enabled
```

Disable REST API access.

```
$ cscli service_console configure rest-api disable
REST API has been disabled
```

Show REST API status (enabled).

```
$ cscli service_console configure rest-api show
REST API access: enabled
```

Show REST API status (disabled).

```
$ cscli service_console configure rest-api show
REST API access: disabled
```

15.24 `service_console configure` Command

Modes: Daily

The `service_console configure` command includes subcommands to configure Service Command help.

Synopsis

```
cscli service_console configure -h
```

Usage:

```
cscli service_console [options] configure [options]
```

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

Optional Arguments	Description
`smtp`	Configures SMTP settings
`email`	Configure e-mail for sending RAS notifications

`configure smtp` Subcommand

This command configures SMTP.

Synopsis

```
cscli service_console configure smtp -h
```

Usage:

```
cscli service_console [options] configure [options] smtp [options]
```

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

Sub-commands	Description
`relay`	Configures SMTP relay for sending RAS notifications
`show`	Displays the SMTP configuration

`configure smtp_relay` Subcommand

This command configures SMTP relay.

Synopsis

```
cscli service_console configure smtp relay -h
```

Usage:

```
cscli service_console [options] configure [options] smtp [options] relay [options]
```

Optional Arguments	Description
`-H |--host`	SMTP host (required)
`-P |--port`	SMTP port [default: 25]
`-h |--help`	Displays the help message and exits

`configure email` Subcommand

This command configures email.

Synopsis

```
cscli service_console configure email -h
```

Usage:

```
cscli service_console [options] configure [options] email [options]
```

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

Sub-commands	Description
`show`	Shows all notification e-mail addresses
`add`	Adds a notification e-mail address
`delete`	Deletes a notification e-mail address
`enable`	Enables e-mail notifications
`disable`	Disables e-mail notifications
`send_test_email`	Sends test email to all enabled services

`configure email_add` Subcommand

This command configures email.

Synopsis

```
$ cscli service_console configure email add -h
```

Usage:

```
cscli service_console [options] configure [options] email [options] add [options]
```

Optional Arguments	Description
-A |--address	E-mail address (required)
-h |--help	Displays the help message and exits

configure email_delete Subcommand

This command configures email.

Synopsis

```
$ cscli service_console configure email delete -h
```

Usage:

```
cscli service_console [options] configure [options] email [options] delete [options]
```

Optional Arguments	Description
-A |--address	E-mail address (required)
-h |--help	Displays the help message and exits

Additional context:

```
$ cscli service_console configure email show
No email addresses are configured to receive service notifications.
$ cscli service_console configure email add -A john.smith@seagate.com
Successfully added 'john.smith@seagate.com' to receive notification emails.
$ cscli service_console configure email add -A bob.smith@seagate.com
Successfully added 'bob.smith@seagate.com' to receive notification emails.
$ cscli service_console configure email add -A junk@example.com
Successfully added 'junk@example.com' to receive notification emails.
$ cscli service_console configure email add -A junk@example.com
Unable to add: 'junk@example.com' is already configured for receiving notification
emails.
$ cscli service_console configure email delete -A junk@example.com
Successfully removed 'junk@example.com' from receiving notification emails.
$ cscli service_console configure email delete -A junk@example.com
Unable to delete: 'junk@example.com' is not configured for receiving notification
emails.
$ cscli service_console configure email show
The following email addresses are configured to receive service notifications:
john.smith@xyzcorp.com
bob.smith@xyzcorp.com
$ cscli service_console configure email disable
```

```
$ cscli service_console configure email enable
```

Error example:

```
$ cscli service_console configure email send_test_email
Failed to send test e-mail to configured user(s): an error has occurred.
```

15.25 System Software Update Commands

The following commands prepare a software upgrade package for installation and apply it to Sonexion system nodes.

Update Software on a System Node

Modes: Daily

The `update_node` command updates software on the specified node(s).

Synopsis

```
cscli update_node [-h] –n node_spec
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-n node_spec|--node-spec node_spec`	Specifies hostnames of the nodes on which to update software.

Set a Node Version

Modes: Daily

The `set_node_version` command is used to change the image of diskless node(s). It is used to boot a given set of diskless appliance nodes into the specified version of Sonexion appliance for a system upgrade. This command is available for the "admin" account only.

Synopsis

```
cscli set_node_version [-h] -n nodes -v version
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-n nodes |--node nodes`	Displays **pdsh**-style nodes hostnames.
`-v version |--version version`	Version of the image to use.

15.26 `support_bundle` Command

The Sonexion `support_bundle` command manages support bundles and support bundle settings.

When a support file is collected, it contains extra information about RAID configuration, CNG configuration (if CNG hardware is configured), and local Lustre users (if any are defined). Additional log files are available in the support bundle for MDRAID examine output and lustre users/groups or local users/groups for CNG nodes.

Synopsis

```
cscli support_bundle [-h] {collect,set,export,show}
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-c |--collect-bundle`	Collects the support bundle.
`|--show`	Support bundle `show` command.
`|--export`	Export support bundles as the **Tar.GZ** archive (into the current folder).
`|--set`	Support bundle `set` command.

`support_bundle collect` Subcommand

The `support_bundle collect` subcommand displays help, **pdsh**-style node names and the time window in minutes.

Synopsis

```
cscli support_bundle collect [-h] [-n nodes] [-t minutes]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-n` *`nodes`*`|--nodes` *`nodes`*	Displays **pdsh**-style node names. The default is all nodes.
`-t` *`minutes`*`|--time-window` *`minutes`*	Displays the time window in minutes. The default is 45 minutes.

`support_bundle show` Subcommand

The `support_bundle show` subcommand displays help, triggers that initiate automatic bundle collection, a list of support bundles collected and the purge limit.

Synopsis

```
cscli support_bundle show [-h] (--triggers | --bundles | --purge-limit)
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-t |--triggers`	Displays triggers that initiate automatic bundle collection.
`-b |--bundles`	Displays a list of support bundles collected.
`-p |--purge-limit`	Displays the purge limit. Free file system space limit in percents, after reaching which Trinity will purge old support bundle files.

`support_bundle export` Subcommand

The `support_bundle export` subcommand displays help, and the bundle ID.

Synopsis

```
cscli support_bundle export [-h] bundle_id
```

Positional Arguments	Description
bundle_id	Displays the support bundle ID.

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

`support_bundle set` Subcommand

The `support_bundle set` subcommand displays help, purge limit for the support bundle and triggers that initiate automatic bundle collection.

Synopsis

```
cscli support_bundle set [-h](-p purge_limit | --trigger {lbug,Failover})[--on | --
enable | --off | --disable]
```

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-p purge_limit|--purge-limit` *purge_limit*	Sets the purge-limit in percentage for support bundles.
`-t |--trigger {`*lbug*`,Failover}, {`*lbug*`,Failover}`	Displays trigger that initiates automatic bundle collection.
`--on`	Turns the command on.
`--enable`	Enables the command.
`--off`	Turns the command off.
`--disable`	Disables the command.

16 GEM CLI Commands

Generic Enclosure Management (GEM) software controls and monitors hardware infrastructure for Sonexion 900, 1600, and 2000 storage systems. While these commands are not typically used during normal operation, they can be used to help resolve issues.

The GEM software controls and monitors the hardware infrastructure and overall system environmental conditions. GEM manages system health, provides power control to power-cycle major subsystems, monitors fans, thermals, power consumption, etc., and offers extensive event capture and logging mechanisms to support file system failover and enable post-failure analysis of hardware components.

Each enclosure and FRU within has an EEPROM containing Vital Product Data (VPD) that is used by GEM to identify and control system components. GEM can pars three main VPD types:

- Midplane
- Power/Cooling Module (PCM)
- Canister

Unified Sytem Management (USM) combines the GEM firmware with other component firmware, such as x86 subsystems, (BIOS, BMC, FPGAs) to control and monitor the system.

Serial port settings

Use the following settings for using HyperTerminal or other serial communications GUI to work with the CLI: Baud rate (bits/sec):

Baud rate (bits per second):	115200
Data bits:	8
Parity:	None
Stop bits:	1
Flow control:	None

The above settings apply to manually typed commands. If multiple commands are sent via a text file, then the baud rate needs to be reduced for all characters to be processed.

Set the baud rate in the running firmware by issuing:

```
rmon baud 0
```

Change the serial communications GUI settings to: Baud rate (bits/sec):

Baud rate (bits/sec):	9600
Data bits:	8

Parity:	None
Stop bits:	1
Flow control:	None

To return to the higher baud rate, issue:

```
rmon baud 4
```

The complete set of supported values is:

0 = 9600

1 = 19200

2 = 38400

3 = 57600

4 = 115200

Supported number bases

Numeric parameters passed into CLIs can be in different bases. Decimal is the default. Octal or hexadecimal can be supplied by using a leading code:

Decimal – Plain number

Octal – Leading ‘0’

Hexadecimal – Leading ‘0x’

For example, the decimal number 14 would be represented in the following ways:

Decimal – 14

Octal – 016

Hexadecimal – 0xE

16.1 ddump

Command name:	`ddump`
Command synopsis:	Returns a system-wide diagnostic dump
Command description:	Calls all commands of the command type `diagnostic` that do not demand an argument; Creates a single-shot diagnostic dump.
Command arguments:	None
Command type:	Diagnostic
Access level:	General

16.2 getboardid

Command name:	`getboardid`
Command synopsis:	Reports the local board slot ID and HA mode
Command description:	Reports the local board slot ID and HA mode in human-readable and machine-readable form.
Command arguments:	hex: Returns the slot ID (byte 1) and HA mode (byte 2) in hexadecimal form. If the canister is the master, then the HA mode is set to 0x0. If the canister is the slave, then the mode is 0x00.
Command type:	Debug
Access level:	General

16.3 getmetisstatus

Command name:	`getmetisstatus`
Command synopsis:	Reports Metis status for the enclosure. (Supplies reserve power to protect in-flight storage data, enabling it to be securely stored on persistent media).
Command description:	Invoking this command returns Metis status in human-readable or machine-readable form.
Command arguments:	Argument 1 [hex]: If the "hex" argument is present, the Metis status is reported in machine-readable form. If "hex" is not specified, the status is reported in human-readable form.
Command type:	Diagnostic
Access level:	Engineering

16.4 getvpd

Command name:	`getvpd`
Command synopsis:	Retrieves VPD information from all enclosure FRUs
Command description:	The `getvpd` command displays the following enclosure VPD data: ● Enclosure Vendor ● Enclosure Product ID

	• Enclosure WWN • Enclosure Serial Number • Enclosure Part Number • Canister VPD Version • Canister Vendor • Canister Product ID • Canister SAS Address • Canister Serial Number • Canister Part Number • Midplane VPD Version • Midplane Product ID • Midplane Serial Number • Midplane Part Number • PCM VPD Version • PCM Vendor • PCM Product ID • PCM Serial Number • PCM Part Number
Command arguments:	`getvpd` – No additional arguments
Command type:	Debug
Access level:	General

16.5 help

Command name:	`help`
Command synopsis:	Displays helpful information about the GEM commands
Command description:	Provides a mechanism to discover the available commands and display the command usage information. By default (i.e. no argument supplied), the command only lists the synopsis for those commands with the access level 'general'. The argument all lists the synopsis for all commands, regardless of access level. The argument testing lists the synopsis for all commands that have the 'testing' access level. If the argument matches a command (for example help ddump) then detailed help for the specified command displays instead.
Command arguments:	One optional argument - see description above.
Command type:	Control

Access level:	General

16.6 ipmi_power

Command name:	`ipmi_power`
Command synopsis:	Performs safe canister-level power control using chassis commands to the BMC.
Command description:	This command allows the user to request a canister-level shutdown through the BMC. The benefit of using this command is to cleanly shut down the x86 subsystem using ACPI.
Command arguments:	2 | "soft" – Orchestrated shutdown of x86 complex. 3 | "off" – Immediate shutdown of x86 complex. 4 | "cycle" – Canister power cycle. 5 | "reset" – Canister reset. 6 | "on" – Wake x86 complex from standby/soft-off.
Command type:	Control
Access level:	General Access

16.7 ipmi_setosboot

Command name:	`ipmi_setosboot`
Command synopsis:	Sets a value in the IPMI OS boot sensor indicating that the x86 subsystem has successfully booted. The OS boot sensor value is cleared to zero (0) on x86 resets and BMC firmware upgrades / reboots.
Command description:	This command is intended for use by an application on the local x86 subsystem to set the OS boot sensor to confirm that the system has finished booting and the OS is in full control. This command MUST be invoked by the customer OS on startup. If it is not set and GEM detects an AC loss event, then the module is automatically shut down. This shutdown ensures that the system batteries are not flattened by a module booting at full power. Without a parameter, the command reads the current sensor value. With a parameter of 1, the command sets the sensor to indicate that the system has booted (0x40) and then reads back the sensor for confirmation.
Command arguments:	`ipmi_setosboot` [*`setting`*]
Command type:	Control

Access level:	Engineering

16.8 logdump

Command name:	`logdump`
Command synopsis:	Displays logged messages
Command description:	Provides a mechanism to output logging information.
Command arguments:	6 optional arguments: Argument 1 specifies the area of memory from which to retrieve log messages from. 'r' = RAM, 'n' = non-volatile. Argument 2 specifies the order of the log messages. "old" = oldest first, "new" = newest first. Argument 3 limits the number of logged messages displayed to *n*. Set to zero (0) or omit the argument to display all logged messages. Argument 4 controls the generation of a *`timestamp`* field in the log dump messages. Set to 1 for enable; 0 for disable. Argument 5 controls the generation of a *`subsystem name`* field in the log dump messages. Set to 1 for enable; 0 for disable. Argument 6 controls the generation of a *`service name`* field in the log dump messages. Set to 1 for enable; 0 for disable. The default (for omitted command arguments) displays all logged messages from RAM, newest first, with all message fields enabled.
Command type:	Diagnostic
Access level:	General

16.9 report_faults

Command name:	`report_faults`
Command synopsis:	Reports all system-wide faults
Command description:	Outputs all known faults, collected from each GEM service.
Command arguments:	None
Command type:	Diagnostic
Access level:	General

16.10 settime

Command name:	`settime`
Command synopsis:	Sets GEM logging time in days, hours, minutes and seconds

Command description:	`settime` *`days hh mm ss`*
	For example:
	`settime 10 9 8 7`
	sets the logging time to 10 days, 9 hours, 8 minutes and 7 seconds. The new logging time appears in the log timestamps as:
	`10+09:08:07.123 M0 >`
	Using `settime` without arguments prints the current logging time to the CLI.

Command arguments:	*`days hh mm ss`*
Command type:	Control
Access level:	General

16.11 ver

Command name:	`ver`
Command synopsis:	Displays version information
Command description:	Displays version numbers and information for the components in the local canister, midplane and PCMs.
Command arguments:	None
Command type:	Diagnostic
Access level:	General