Sonexion™ Administrator’s Guide

Software Version 1.4.0

S-2537-14a

U.S. GOVERNMENT RESTRICTED RIGHTS NOTICE

TRADEMARKS

Direct requests for copies of publications to:
Mail: Cray Inc. Logistics
PO Box 6000
Chippewa Falls, WI 54729-0080
USA
E-mail: spares@cray.com
Fax: +1 715 726 4602
Direct comments about this publication to:
Mail: Cray Inc.
Technical Training and Documentation
P.O. Box 6000
Chippewa Falls, WI 54729-0080
USA
E-mail: ttd_online@cray.com
Fax: +1 715 726 49

Record of Revision

Publication Number	Description
HR5-6093-0	August 2012 Original Printing
HR5-6093-A	September 2012 Revises Appendix A, “CSSM CLI User Documentation” to reflect changes in CSSM 1.2.
HR5-6093-B	April 2013 Revises Appendix A, “CSSM CLI User Documentation” to reflect changes in CSSM 1.2.1.
S-2537-131	March 2014 Updates CSSM and CSCLI description for release 1.3.1. Adds upgrading RPMs. Publication number changed to indicate software document.
S-2537-131a	April 2014 Most of section 3 removed, referring to Field Installation Guide. LNET configuration revised.
S-2537-14	May 2014 Expanded discussion of support bundles, changes to CLI.
S-2537-14a	July 2014 Revisions to section 4. Old section 7 removed, not applicable to 1.4.0.

Contents

Tables

1. Introducing Sonexion

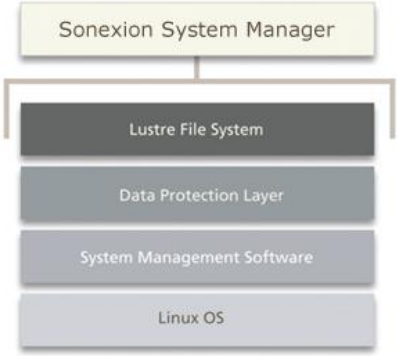
This Administration Guide provides step-by-step instructions on how to set up, use, and troubleshoot the Cray Sonexion storage system.This manual is intended for Site Service Providers who maintain Cray Sonexion storage systems.

1.1 Software Architecture

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

- Cray Sonexion System Manager (CSSM)
- Lustre file system
- Data protection layer (Redundant Array of Independent Disks, RAID)
- Unified System Management (USM)
- Linux OS

Sonexion runs Lustre 2.1.0 software in a standard Linux environment (Scientific Linux 6.2 operating system). The file system is fully integrated with CSSM, USM, and RAID layers in the stack.



1.1.1 Cray Sonexion System Manager (CSSM)

Cray Sonexion System Manager (CSSM) provides a single-pane-of-glass view of the Sonexion solution infrastructure. It includes a browser-based GUI that simplifies cluster installation and configuration, and provides consolidated management and control of the entire storage cluster. CSSM also provides distributed component services to manage and monitor system hardware and software.

CSSM includes wizards to guide you through configuration and node provisioning. Once the cluster is running, you can 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, CSSM 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.

1.1.2 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. See "New software features", page 14.

1.1.3 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:

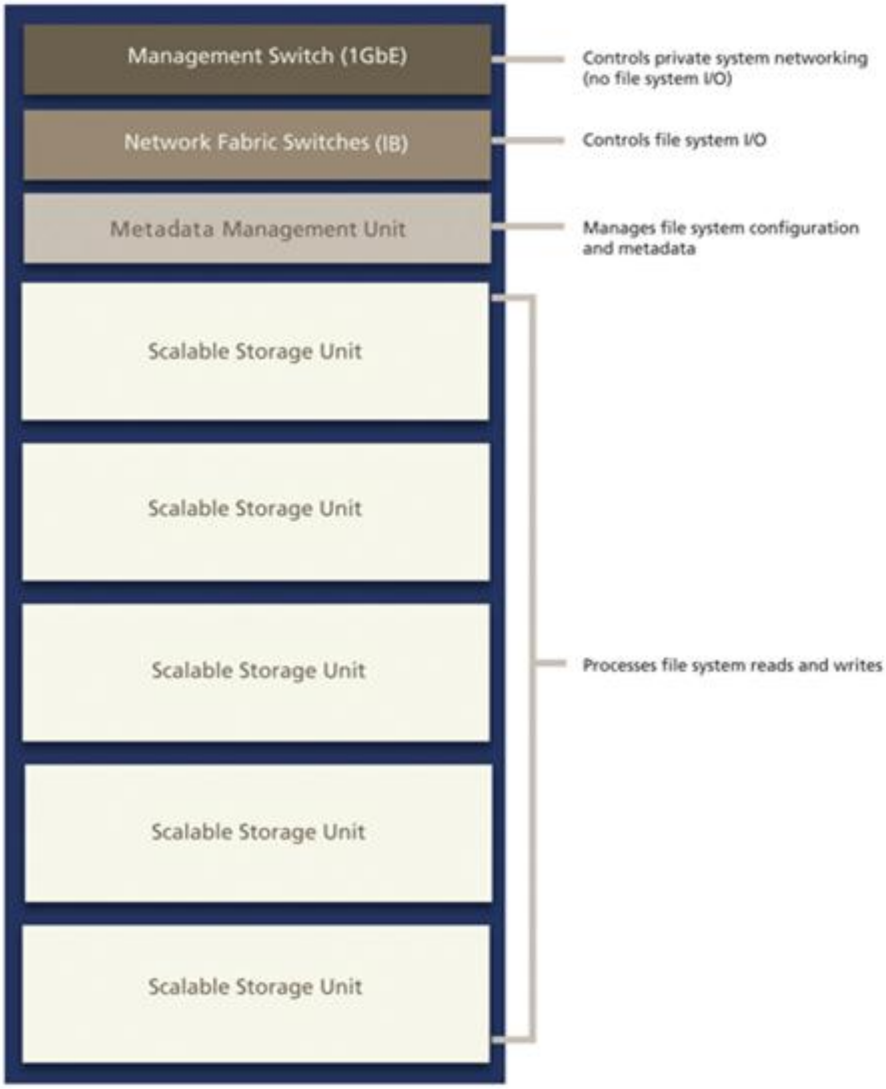
- Manages system health, providing random-access services (RAS) that cover all major components such as disks, fans, power-supply units (PSUs), Serial Attached SCSI (SAS) fabrics, PCI (Peripheral Component Interconnect) buses, memories, and CPUs, and provides alerts, logging, diagnostics, and recovery 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 keeps 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

1.2 Hardware Architecture

The Sonexion 1600 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.



1.2.1 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).

1.2.2 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.

1.2.3 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.

1.2.4 Management switches

The Management switches consists of a dedicated local network on dual 1GbE switches that is used for configuration management and health monitoring of all components in the Sonexion solution. 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 OSSs, enabling them to be power-cycled by CSSM.

2. What's New?

The following is a list of new features and improvements for this release of CSSM version 1.4.0.

2.1 New software features

- **GridRAID**

  GridRAID, the Sonexion implementation of Parity Declustered RAID (PDRAID), is a RAID organization that combines the logical structure of 8+2 RAID 6 data protection with the pseudorandom distribution of these RAID 6 parity groups, along with reserved spare space across a large number of physical storage devices.

  This allows accelerated restoration of redundancy, referred to as "GridRAID reconstruct" (phase 1), and a modified "GridRAID rebalance" (phase 2) that restores data for a failed drive onto a physical replacement drive. A comparison:

  - MDRAID (previous method): each OSS typically operates using 40 drives configured as four 8+2 RAID 6 arrays (4 OSTs per OSS and 8 OSTs total per SSU), plus 2 global hot spares for a total of 82 drives.
  - GridRAID (new method): 82 drives are configured as two 41-drive GridRAID arrays with no dedicated hot spare drives. Two drives' worth of spare space is distributed across all 41 drives in the array.

- **New and improved monitoring dashboard**

  The new Dashboard replaces the dashboard functionality from previous releases and provides a high level view into the entire storage cluster providing top level knowledge into the overall behavior and health of the system. The dashboard displays the following widgets that report high-level system health and issues with individual nodes to aid in troubleshooting and resolving problems quickly:

- Node Status
- File System Throughput
- Inventory
- Top System Statistics

- **SSU+n systems containing one SSU enclosure and up to three ESU enclosures**

  - The SSU+n feature, where the maximum value for n=3, whereby up to 3 Expansuion Storage Units (ESUs) can be added to each SSU, for this release provides additional capacity at reduced cost.
  - This feature is available only for newly installed systems. The ability to add ESUs to existing systems will be available in a future release.

- **GUI guest account**

  A built-in guest account for read-only access to CSSM is available in this release.

- **NIS GUI support**

  Added GUI support for configuring NIS as an option for Lustre users.

- **USM 2.16 firmware update**
- **Lustre features**

  - 4MB IO - Support for larger than 1MB sequential I/O RPCs. This requires a Lustre 2.4 Client.
  - Extended Attribute Cache for Lustre (XATTR Cache) - The patch implements an extended attribute cache for a Lustre client. It is organized as a write-through cache: reads are performed from cache, updates are sent synchronously to the MDS. An additional inode bit is added to protect the cache.

2.2 New hardware features

- The SSU+n configuration where the maximum value of n is 3, refers to an SSU with 3 ESUs attached. The SSU+n configuration consists of the following:
- One Scalable Storage Unit (SSU) and up to three ESU (Expansion Storage Unit)
- Maximum of 32 OSTs configured on the SSU for MDRAID at 8 OSTs per enclosure (8 OSTs x 4 enclosures = 32 OST’s managed by the SSU)
- Maximum of 8 OSTs configured on the SSU for GridRAID at 2 OSTs per enclosure (2 OSTs x 4 enclosures =8 OST’s managed by the SSU)
- Each SSU and its companion ESU must be located vertically adjacent in the same rack

3. Custom LNET configuration

Use the procedure in this chapter to configure a custom LNET configuration on the Sonexion system while in “daily mode” (see section 7.1.2, page 71).

For a new system, first follow the setup procedures described in Cray publication HR5-6124, *Sonexion Field Installation Guide*. Then execute the following installation.

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

   ```
   $ sudo su -
   ```

3. Stop the Lustre file system by running the command:

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

4. For version 1.2:
   a. Log in to the MGS node via SSH.

   b. If you do not know the MGS group, run this command:

   ```
   crm_mon -1 -r
   ```

   The group with **md65** in its name is the MGS group.

   c. Stop the MGS service, by entering:

   ```
   # stop_xyraid mgs_group
   ```

   d. Log out of the MGS node and back into primary MGMT server node.

5. To change the **o2ib** index, use the following steps:

   a. Start the MySQL client and connect to the **t0db** database, by entering:

   ```
   # mysql  t0db
   ```

   b. Display the **mgsNID**, **nidFormat**, and **nidIndex** entries by entering:

   ```
   mysql> select * from property where name in ('nidFormat',
      'mgsNID', 'nidIndex');
   ```

   Following is a sample output for versions 1.3.1 and 1.4.0:

```
mysql> select * from property where name in ('nidFormat', 'mgsNID',
   'nidIndex');
+-----+--------------------+-----------+---------------------+-----------+
| id  | context            | name      | value               | attr_type |
+-----+--------------------+-----------+---------------------+-----------+
|  22 | snx11033n:beConfig | nidFormat | l%s@o2ib%d          | str       |
| 106 | snx11033n:beConfig | nidIndex  | 3                   | int       |
| 109 | snx11033n:beConfig | mgsNID    | lsnx11033n002@o2ib3 | str       |
+-----+--------------------+-----------+---------------------+-----------+
```

   ```
   3 rows in set (0.00 sec)
   ```

   c. Set the **o2ib** index by modifying the **nidIndex** entry by specifying:

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

   Sample:

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

   d. Set the **mgsNID** entry to match the **o2ib** index by entering:

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

   Sample output:

   ```
   mysql> update property set value='lsnx11033n002@o2ib2' where
      name='mgsNID';
   Query OK, 1 row affected (0.04 sec)
   Rows matched: 1  Changed: 1  Warnings: 0
   ```

e. Verify the changes by repeating step 5b.

Sample output:

```
mysql> select * from property where name in ('nidFormat', 'mgsNID',
   'nidIndex');
+-----+--------------------+-----------+---------------------+-----------+
| id  | context            | name      | value               | attr_type |
+-----+--------------------+-----------+---------------------+-----------+
|  22 | snx11033n:beConfig | nidFormat | l%s@o2ib%d          | str       |
| 106 | snx11033n:beConfig | nidIndex  | 2                   | int       |
| 109 | snx11033n:beConfig | mgsNID    | lsnx11033n002@o2ib2 | str       |
+-----+--------------------+-----------+---------------------+-----------+

          3 rows in set (0.00 sec)
```

f. Close the MySQL session by specifying:

```
mysql> quit
```

g. Run puppet:

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

6. Run the 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
```

7. Verify that the customized LNET configuration has been applied.

   a. List the node NIDs by entering:

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

   b. List the nodes and targets, by entering:

   ```
   # cscli fs_info
   ```

8. For version 1.2 **only**, run:

```
# pdsh -g lustre mdraid-deactivate
# pdsh -g lustre manage_all_xyraid
```

9. Start the Lustre file system by entering:

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

Wait for the targets to mount on all system nodes.

This completes the procedure.

4. Change Network Settings

This chapter contains procedures for changing several different network settings on Sonexion. The following topics are included:

- Changing the DNS resolver configuration
- Changing the externally facing IP addresses
- Changing the LDAP settings in Daily Mode
- Configuring NIS Support in Daily Mode
- Changing NIS Settings in Daily Mode

4.1 Change the DNS Resolver Configuration

This procedure is used to change the DNS resolver, the system service which translates URLs into IP addresses.

1. SSH into the primary MGMT node. Run:

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

2. Sudo to root:

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

3. Update the DNS settings in the **t0db** database.Run:

```
[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. Run:

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

4.2 Change Externally Facing IP Addresses

Some customers may wish to change one or both of the externally facing 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 1.4.0 and later, that interface is **eth1**, which is used in the following examples.

To change this configuration, follow the steps given below:

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

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

3. Sudo to root:

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

4. Edit the Ethernet configuration file. Run:

   ```
   [admin@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 it 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 valid IP address). Change this file as desired.

5. Toggle the Ethernet interface. Run:

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

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

4.3 Change the LDAP settings in Daily Mode

LDAP settings are stored in the t0db database table ldap_setup. The following columns are present in this table and are used to configure LDAP on Sonexion nodes:

Table 1. 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_dn	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.

1. SSH into the primary MGMT node. Run:

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

2. Sudo to root:

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

3. Print the existing configuration as follows:

   a. Enter:

      ```
      [MGMT0]# cat /etc/nslcd.conf
      ```

      Sample output:

      ```
      [root@snx11000n000 ~]# 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. Enter:

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

Sample output:

```
[root@snx11000n000 ~]# mysql t0db -e "select * from ldap_setup;"

+---------------+--------------+------+----------------+----------------+-----------+-----------+---------+----------+------------+
| ldap_setup_id | server_name  | port | base_dn        | user_dns       | group_dns | hosts_dns | bind_dn | password | cluster_id |
+---------------+--------------+------+----------------+----------------+-----------+-----------+---------+----------+------------+
|             1 | 172.30.12.19 |  389 | dc=datacenter, | ou=People,     |           |           | NULL    | NULL     |          1 |
|               |              |      | dc=cray,dc=com | dc=datacenter, |           |           |         |          |          1 |
|               |              |      |                | dc=cray,dc=com |           |           |         |          |          1 |
+---------------+--------------+------+----------------+----------------+-----------+-----------+---------+----------+------------+
```

4. Change the base DN:

```
[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).

4.3.1 LDAP over TLS Configuration Settings in Daily Mode

Set up an LDAP/TLS server. Perform the following to get LDAP over TLS working on release 1.3.1:

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 your 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 your 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/1.3.1-18/appliance.x86_64/etc/openldap/cacerts/
```

4.4 Configure NIS Support in Daily Mode

This section provides instructions for configuring the Network Information Services (NIS) client for Sonexion. This procedure applies to releases in 1.3.1 and 1.4.0. Sonexion 1.3.1 supports LDAP and NIS but was only intended to support configuration of NIS during the Customer Wizard phase of the installation.

4.4.1 Requirements

NIS must not have been previously been configured during Customer Wizard mode, otherwise see section 4.5, “Change NIS Settings in Daily Mode”, page 24.

You will need to know the NIS Domain name, and the IP address of NIS servers reachable (pingable) from the Management Nodes.

4.4.2 NIS Installation

1. Enable support for NIS:

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

2. Configure in 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")'
```

If you wish 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. The above changes must be followed by:

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

5. Once this has been done, check that `/etc/puppet/data/CSSM/nis.yaml` has been updated on 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. Check `/etc/yp.conf` on MMU nodes contain 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. Check whether `ypbind` is running on all MMU nodes

8. Final checks to be run on all MMU nodes:

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

4.5 Change NIS Settings in Daily Mode

This section provides instructions to change the NIS settings in daily mode, assuming that NIS has already been configured. This procedure applies to releases 1.3.1 and 1.4.0.

4.5.1 Requirements

NIS has previously been configured, ether during customer Wizard or following instructions above in section 4.4, "Configure NIS Support in Daily Mode", page 23.

4.5.2 NIS Configuration

1. SSH into the primary MGMT node. Run:

```
$ 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]# myql t0db -e 'update property set value = "xx.xx.xx.xx
      yy.yy.yy.yy" where name = "nis_server"'
   ```

   You can specify more one or more IP addresses for NIS master and its reachable NIS slaves.

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

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

6. The above changes must be followed by:

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

7. Verify the new NIS server settings by repeating step 3.

5. Support Bundles

This section discusses the creation and use of *support bundles*, which are collections of event logs from field systems. Support bundles are used to debug many Sonexion problems and are collected by Cray personnel. Sonexion provides a mechanism for collecting support bundles, which may be initiated manually or triggered automatically by certain events (for example, Lustre bugs and failover events). These support bundles should be provided to Cray personnel in the course of requesting technical support.

When you encounter a problem, refer to the following topics to assist with resolving the issue. Use the following tabs in the CSSM GUI:

- 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.

5.1 Support file overview

The CSSM **Support** tab provides support functionality that lets you collect diagnostic information, including logs and configuration settings, on an automatic or manual basis. When a Lustre error occurs, the system automatically collects diagnostic information. Alternately, 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.

This section describes the data collection process and contains procedures to work with the diagnostic information.

5.1.1 Collecting Sonexion data in support files

When a Lustre error or a system event (such as failover) occurs, Sonexion automatically triggers a process to collect a 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 collecting 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 will not start a second data collection process related to the later error.

5.1.2 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

5.1.3 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.

You may also manually start a support bundle collections 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.

5.2 Manually collect support files

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

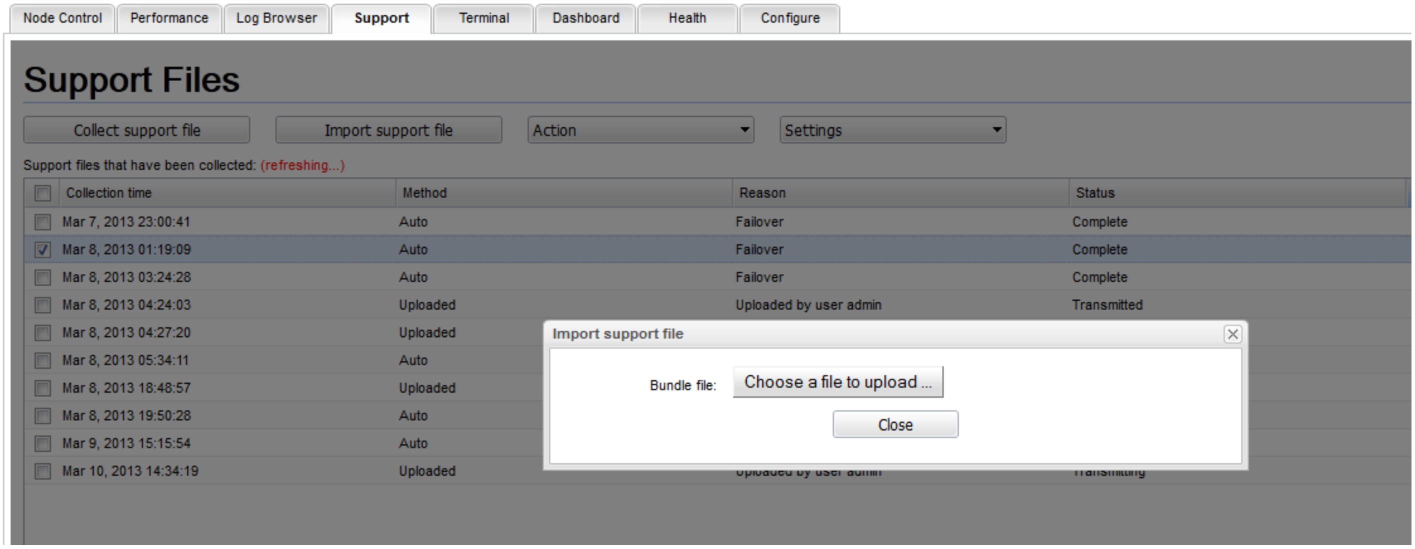
To manually collect Sonexion support files:

1. In CSSM, click the **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.



Support Files

Support files that have been collected: (refreshing...)

□	Collection time	Method	Reason	Status
□	Mar 7, 2013 23:00:41	Auto	Failover	Complete
☑	Mar 8, 2013 01:19:09	Auto	Failover	Complete
□	Mar 8, 2013 03:24:28	Auto	Failover	Complete
□	Mar 8, 2013 04:24:03	Uploaded	Uploaded by user admin	Transmitted
□	Mar 8, 2013 04:27:20	Uploaded		
□	Mar 8, 2013 05:34:11	Auto		
□	Mar 8, 2013 18:48:57	Uploaded		
□	Mar 8, 2013 19:50:28	Auto		
□	Mar 9, 2013 15:15:54	Auto		
□	Mar 10, 2013 14:34:19	Uploaded	Uploaded by user admin	Transmitting

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 will select 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 and the support file is created, which is a zip file containing hundreds of different log files.

5.2.1 Import a support file

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

To import a support file:

1. In CSSM, click the **Support** tab.

   The **Support Files** screen displays.

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

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

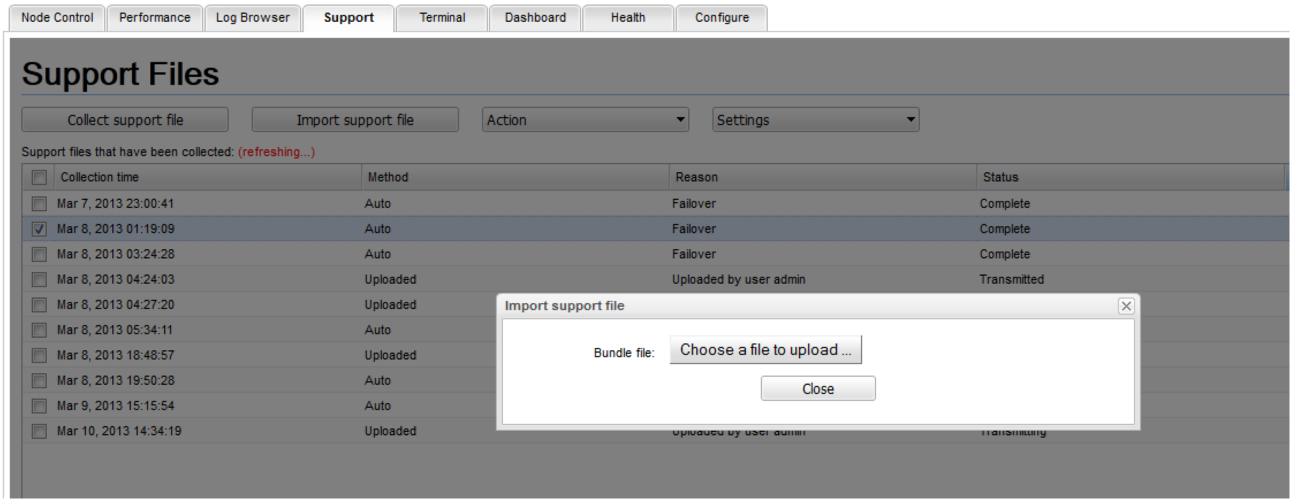
3. Select a support file to upload.

   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.



Support Files

Support files that have been collected: (refreshing...)

□	Collection time	Method	Reason	Status
□	Mar 7, 2013 23:00:41	Auto	Failover	Complete
☑	Mar 8, 2013 01:19:09	Auto	Failover	Complete
□	Mar 8, 2013 03:24:28	Auto	Failover	Complete
□	Mar 8, 2013 04:24:03	Uploaded	Uploaded by user admin	Transmitted
□	Mar 8, 2013 04:27:20	Uploaded		
□	Mar 8, 2013 05:34:11	Auto		
□	Mar 8, 2013 18:48:57	Uploaded		
□	Mar 8, 2013 19:50:28	Auto		
□	Mar 9, 2013 15:15:54	Auto		
□	Mar 10, 2013 14:34:19	Uploaded	Uploaded by user admin	Transmitting

Import support file

Bundle file:

5.2.2 Download a support file

Use the download a copy feature to save a local copy of the selected support file.

To download a support file:

1. In CSSM, click the **Support** tab.

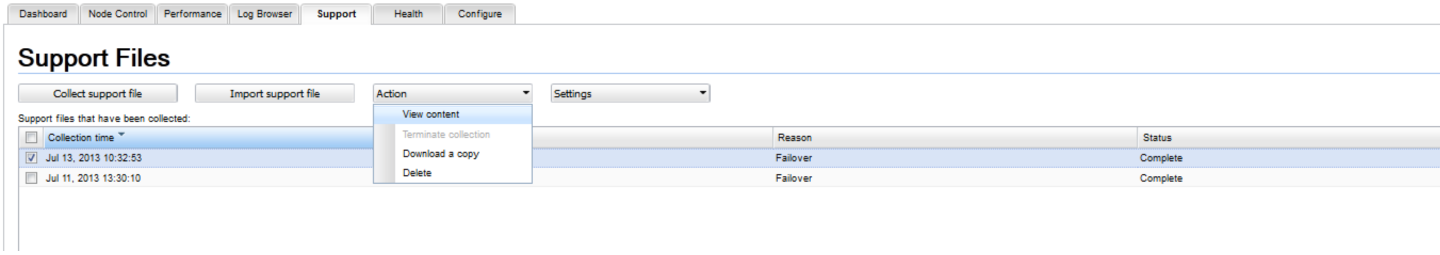
   The **Support Files** screen displays.

2. Select a support file to view.

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

   b. Select **Download a copy**.

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



Support Files

Support files that have been collected:

□	Collection time	Reason	Status
☑	Jul 13, 2013 10:32:53	Failover	Complete
□	Jul 11, 2013 13:30:10	Failover	Complete

5.2.3 Delete a support file

Use the delete file feature to delete a selected support file.

To delete a support file:

1. In CSSM, click the **Support** tab.

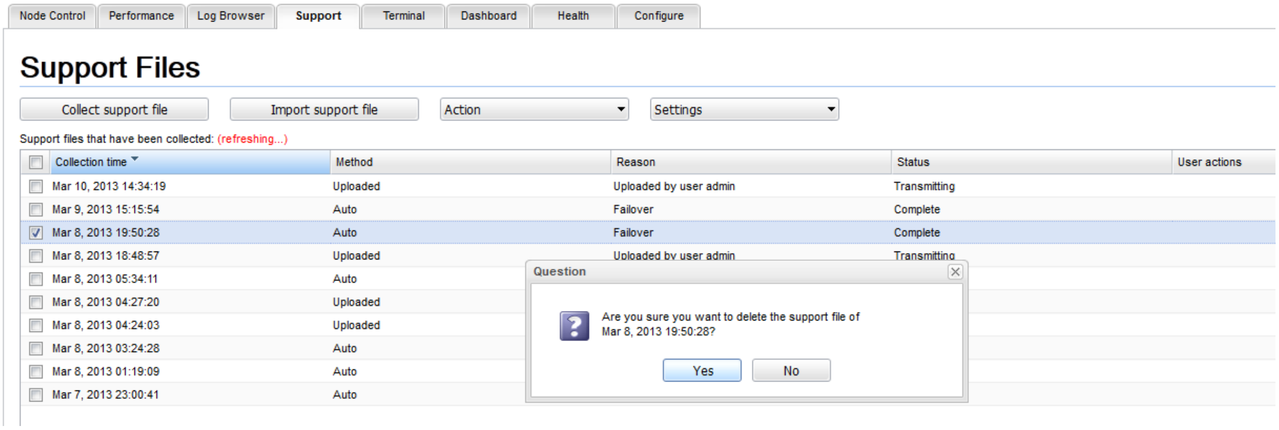
   The Support Files screen displays.

2. Click to select a support file.

   a. In the row containing the support file you want 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.



Support Files

Support files that have been collected: (refreshing...)

□	Collection time	Method	Reason	Status	User actions
□	Mar 10, 2013 14:34:19	Uploaded	Uploaded by user admin	Transmitting	
□	Mar 9, 2013 15:15:54	Auto	Failover	Complete	
☑	Mar 8, 2013 19:50:28	Auto	Failover	Complete	
□	Mar 8, 2013 18:48:57	Uploaded	Uploaded by user admin	Transmitting	
□	Mar 8, 2013 05:34:11	Auto			
□	Mar 8, 2013 04:27:20	Uploaded			
□	Mar 8, 2013 04:24:03	Uploaded			
□	Mar 8, 2013 03:24:28	Auto			
□	Mar 8, 2013 01:19:09	Auto			
□	Mar 7, 2013 23:00:41	Auto			

Are you sure you want to delete the support file of Mar 8, 2013 19:50:28?

5.3 Viewing support files

When a support file exists, either automatically or from using methods described in the preceding subsections, you can view the contents on the **Support** tab. Click the **Actions** pull-down menu and select the **View content** option.

Fields on the contents screen are described below.

- **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. See page 33.
- **Web Logs -** Lists all web logs for the cluster when the support file was created, for more details see page 33.
- **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. See page35.

To view Sonexion support files:

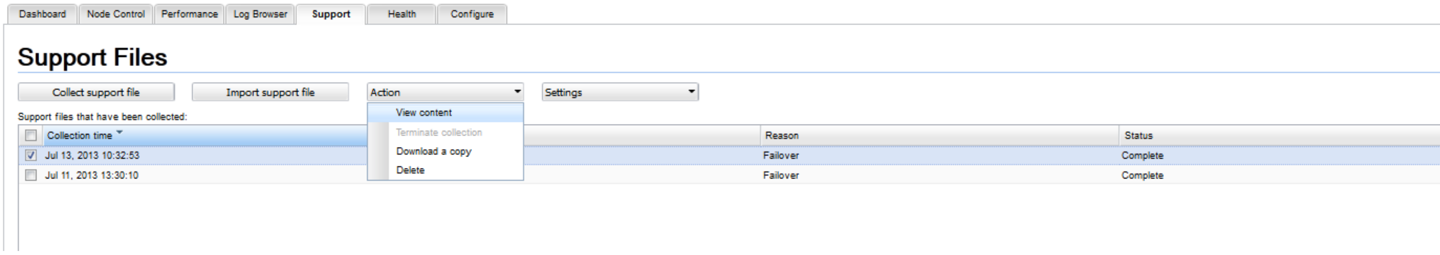
1. In CSSM, click the **Support** tab.

The **Support Files** screen displays and lists 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 you want to view (in the **User actions** column) click the **Actions** button.

   b. Select **View Content**.



Support Files

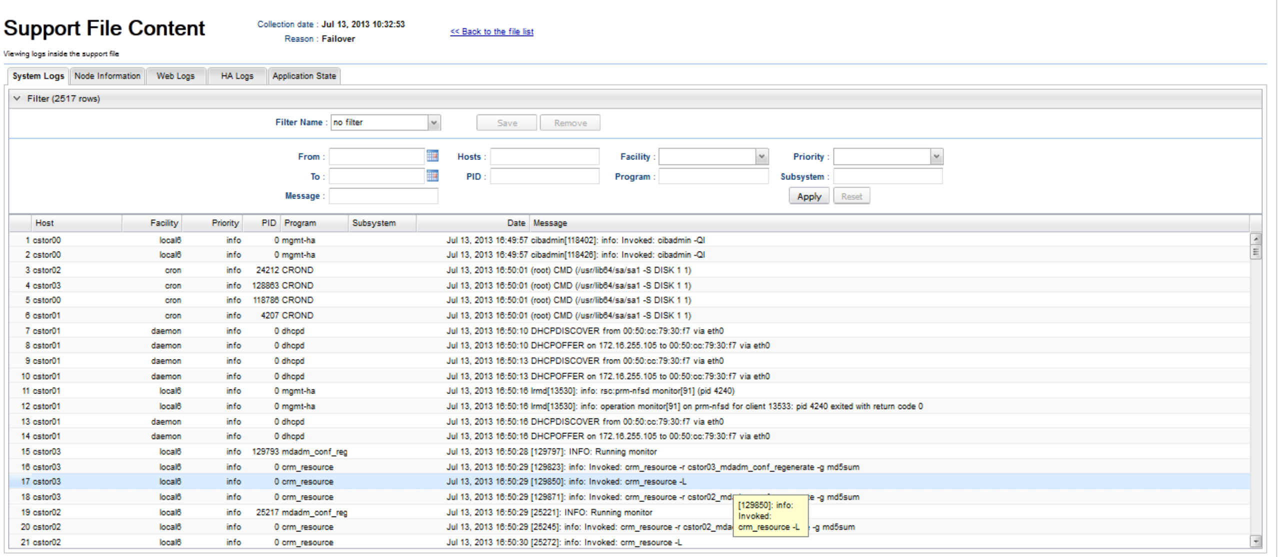
Support files that have been collected:

□	Collection time	Reason	Status
☑	Jul 13, 2013 10:32:53	Failover	Complete
□	Jul 11, 2013 13:30:10	Failover	Complete

The **Support Files** content screen opens and displays tabs that are discussed in the following paragreaphs.

System Logs

The **System Logs** tab lists all the system logs for the cluster when the support file was created.



Support File Content

Collection date : Jul 13, 2013 10:32:53
Reason : Failover

Viewing logs inside the support file

Filter (2517 rows)

Filter Name : no filter

From :
To :
Message :
Hosts :
PID :
Facility :
Program :
Priority :
Subsystem :

	Host	Facility	Priority	PID	Program	Subsystem	Date	Message
1	cstor00	local6	info	0	mgmt-ha		Jul 13, 2013 16:49:57	cibadmin[118402]: info: Invoked: cibadmin -Ql
2	cstor00	local6	info	0	mgmt-ha		Jul 13, 2013 16:49:57	cibadmin[118426]: info: Invoked: cibadmin -Ql
3	cstor02	cron	info	24212	CROND		Jul 13, 2013 16:50:01	(root) CMD (/usr/lib64/sa/sa1 -S DISK 1 1)
4	cstor03	cron	info	128863	CROND		Jul 13, 2013 16:50:01	(root) CMD (/usr/lib64/sa/sa1 -S DISK 1 1)
5	cstor00	cron	info	118786	CROND		Jul 13, 2013 16:50:01	(root) CMD (/usr/lib64/sa/sa1 -S DISK 1 1)
6	cstor01	cron	info	4207	CROND		Jul 13, 2013 16:50:01	(root) CMD (/usr/lib64/sa/sa1 -S DISK 1 1)
7	cstor01	daemon	info	0	dhcpd		Jul 13, 2013 16:50:10	DHCPDISCOVER from 00:50:cc:79:30:f7 via eth0
8	cstor01	daemon	info	0	dhcpd		Jul 13, 2013 16:50:10	DHCPOFFER on 172.16.255.105 to 00:50:cc:79:30:f7 via eth0
9	cstor01	daemon	info	0	dhcpd		Jul 13, 2013 16:50:13	DHCPDISCOVER from 00:50:cc:79:30:f7 via eth0
10	cstor01	daemon	info	0	dhcpd		Jul 13, 2013 16:50:13	DHCPOFFER on 172.16.255.105 to 00:50:cc:79:30:f7 via eth0
11	cstor01	local6	info	0	mgmt-ha		Jul 13, 2013 16:50:16	lrmd[13530]: info: rsc:prm-nfsd monitor[91] (pid 4240)
12	cstor01	local6	info	0	mgmt-ha		Jul 13, 2013 16:50:16	lrmd[13530]: info: operation monitor[91] on prm-nfsd for client 13533: pid 4240 exited with return code 0
13	cstor01	daemon	info	0	dhcpd		Jul 13, 2013 16:50:16	DHCPDISCOVER from 00:50:cc:79:30:f7 via eth0
14	cstor01	daemon	info	0	dhcpd		Jul 13, 2013 16:50:16	DHCPOFFER on 172.16.255.105 to 00:50:cc:79:30:f7 via eth0
15	cstor03	local6	info	129793	mdadm_conf_reg		Jul 13, 2013 16:50:28	[129797]: INFO: Running monitor
16	cstor03	local6	info	0	crm_resource		Jul 13, 2013 16:50:29	[129823]: info: Invoked: crm_resource -r cstor03_mdadm_conf_regenerate -g md5sum
17	cstor03	local6	info	0	crm_resource		Jul 13, 2013 16:50:29	[129850]: info: Invoked: crm_resource -L
18	cstor03	local6	info	0	crm_resource		Jul 13, 2013 16:50:29	[129871]: info: Invoked: crm_resource -r cstor02_md e -g md5sum
19	cstor02	local6	info	25217	mdadm_conf_reg		Jul 13, 2013 16:50:29	[25221]: INFO: Running monitor
20	cstor02	local6	info	0	crm_resource		Jul 13, 2013 16:50:29	[25245]: info: Invoked: crm_resource -r cstor02_mda -g md5sum
21	cstor02	local6	info	0	crm_resource		Jul 13, 2013 16:50:30	[25272]: info: Invoked: crm_resource -L

[129850]: info:
Invoked:
crm_resource -L

For each syslog, the following information is provided:

- **Host -** The hostname field consists of the host name (as configured on the host itself) or the IP address.
- **Facility -** This identifies who submitted the message. There are a small number of facilities defined. The kernel, the mail subsystem, FTP server, are just some examples of recognized facilities.
- **Priority -** The source or facility that generates the syslog message also specifies the severity of the message.
- **PID -** Process ID.
- **Program -** Name of program or module that produced the message.
- **Subsystem -** Filters Lustre, LustreError and some other classes of messages.

- **Date/Time -** The timestamp is the local time, in MMM DD HH:MM:SS format, of the device when the message was generated. This is also the default sort field, with the most recent collections at the top.
- **Message -** This is the text of the syslog message, along with some additional information about the process that generated the message.

Node information

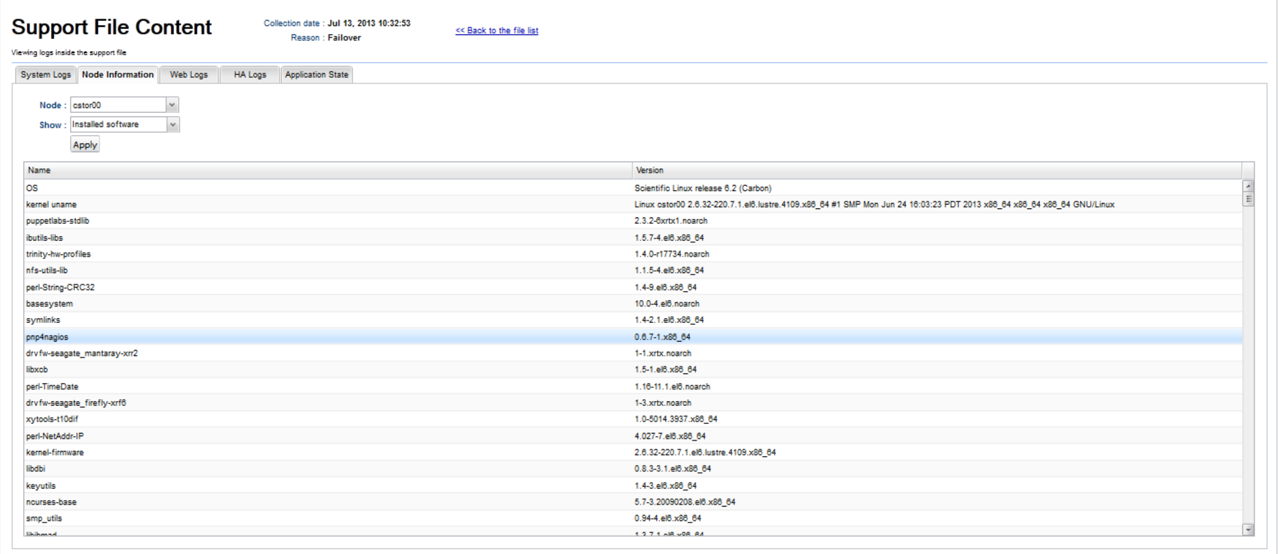
The **Node** log displays version information for the selected node.

- **Node**: List of nodes available in this collection.

  Click the down arrow and choose the node you wish to display, then click the **Apply** button.

- **Show**: Display the installed software, resources, and processes.

  Click the down arrow and choose which module you wish to display, then click the **Apply** button.



Support File Content

Collection date : Jul 13, 2013 10:32:53

Reason : Failover

<< Back to the file list

Viewing logs inside the support file

Node : cstor00

Show : Installed software

Name	Version
OS	Scientific Linux release 6.2 (Carbon)
kernel uname	Linux cstor00 2.6.32-220.7.1.el6.lustre.4109.x86_64 #1 SMP Mon Jun 24 16:03:23 PDT 2013 x86_64 x86_64 x86_64 GNU/Linux
puppetlabs-stdlib	2.3.2-6xrtx1.noarch
ibutils-libs	1.5.7-4.el6.x86_64
trinity-hw-profiles	1.4.0-r17734.noarch
nfs-utils-lib	1.1.5-4.el6.x86_64
perl-String-CRC32	1.4-9.el6.x86_64
basesystem	10.0-4.el6.noarch
symlinks	1.4-2.1.el6.x86_64
pnp4nagios	0.6.7-1.x86_64
drvfw-seagate_mantaray-xrr2	1-1.xrtx.noarch
libxcb	1.5-1.el6.x86_64
perl-TimeDate	1.16-11.1.el6.noarch
drvfw-seagate_firefly-xrf6	1-3.xrtx.noarch
xytools-t10dif	1.0-5014.3937.x86_64
perl-NetAddr-IP	4.027-7.el6.x86_64
kernel-firmware	2.6.32-220.7.1.el6.lustre.4109.x86_64
libdbi	0.8.3-3.1.el6.x86_64
keyutils	1.4-3.el6.x86_64
ncurses-base	5.7-3.20090208.el6.x86_64
smp_utils	0.94-4.el6.x86_64
libibmad	1.3.7-1.el6.x86_64

Web logs

Lists all web logs for the cluster when the support file was created. When a specific log is selected (using the Log File menu), the log entries are displayed in the lower pane. The following web logs are available:

- **wsgi_access.log**

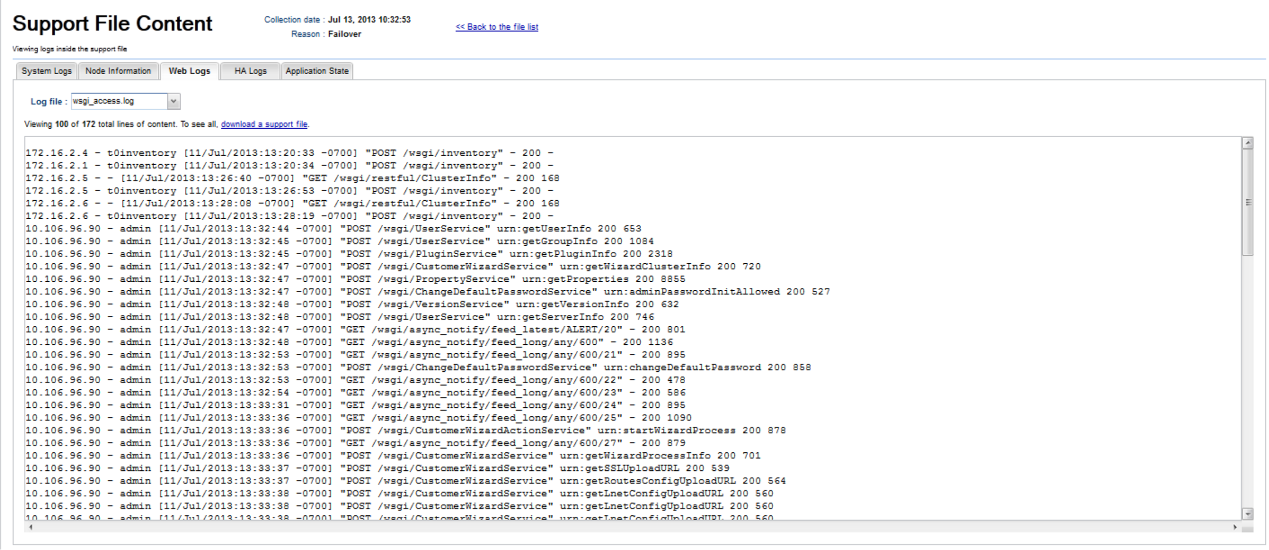
  These are the logs for the web server gateway interface.

- **access.log**

  This is a copy of the access log.

- **error.log**

  The error log contains information indicating when mysqld was started and stopped and also any critical errors that occur while the server is running.



Collection date : **Jul 13, 2013 10:32:53**
Reason : **Failover**

Log file : wsgi_access.log

Viewing **100** of **172** total lines of content. To see all, download a support file.

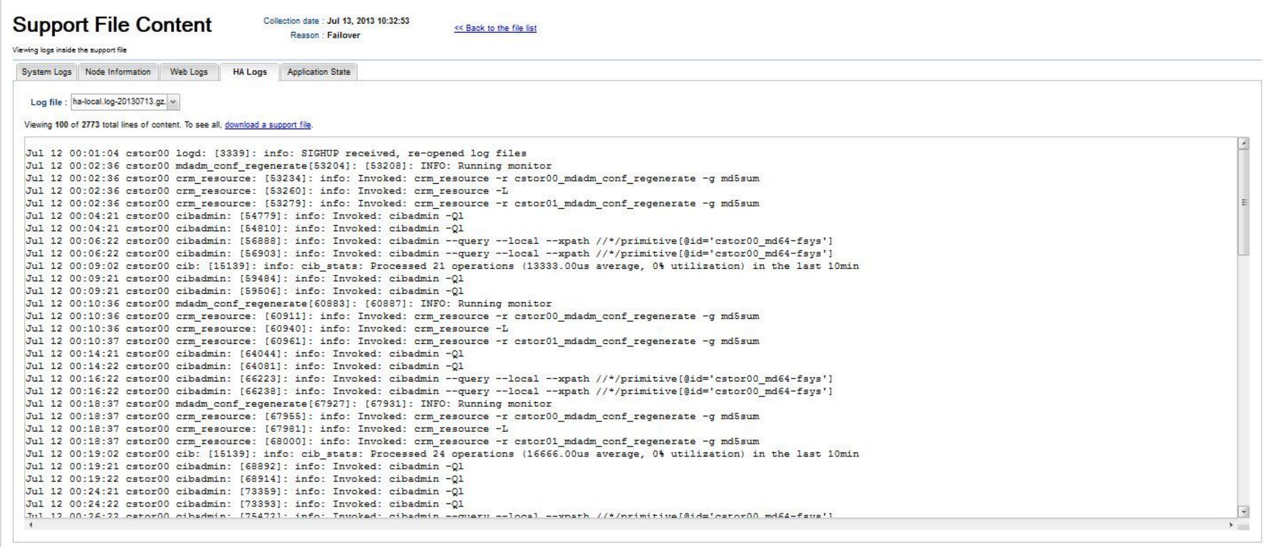
```
172.16.2.4 - t0inventory [11/Jul/2013:13:20:33 -0700] "POST /wsgi/inventory" - 200 -
172.16.2.1 - t0inventory [11/Jul/2013:13:20:34 -0700] "POST /wsgi/inventory" - 200 -
172.16.2.5 - - [11/Jul/2013:13:26:40 -0700] "GET /wsgi/restful/ClusterInfo" - 200 168
172.16.2.5 - t0inventory [11/Jul/2013:13:26:53 -0700] "POST /wsgi/inventory" - 200 -
172.16.2.6 - - [11/Jul/2013:13:28:08 -0700] "GET /wsgi/restful/ClusterInfo" - 200 168
172.16.2.6 - t0inventory [11/Jul/2013:13:28:19 -0700] "POST /wsgi/inventory" - 200 -
10.106.96.90 - admin [11/Jul/2013:13:32:44 -0700] "POST /wsgi/UserService" urn:getUserInfo 200 653
10.106.96.90 - admin [11/Jul/2013:13:32:45 -0700] "POST /wsgi/UserService" urn:getGroupInfo 200 1084
10.106.96.90 - admin [11/Jul/2013:13:32:45 -0700] "POST /wsgi/PluginService" urn:getPluginInfo 200 2318
10.106.96.90 - admin [11/Jul/2013:13:32:47 -0700] "POST /wsgi/CustomerWizardService" urn:getWizardClusterInfo 200 720
10.106.96.90 - admin [11/Jul/2013:13:32:47 -0700] "POST /wsgi/PropertyService" urn:getProperties 200 8855
10.106.96.90 - admin [11/Jul/2013:13:32:47 -0700] "POST /wsgi/ChangeDefaultPasswordService" urn:adminPasswordInitAllowed 200 527
10.106.96.90 - admin [11/Jul/2013:13:32:48 -0700] "POST /wsgi/VersionService" urn:getVersionInfo 200 632
10.106.96.90 - admin [11/Jul/2013:13:32:48 -0700] "POST /wsgi/UserService" urn:getServerInfo 200 746
10.106.96.90 - admin [11/Jul/2013:13:32:47 -0700] "GET /wsgi/async_notify/feed_latest/ALERT/20" - 200 801
10.106.96.90 - admin [11/Jul/2013:13:32:48 -0700] "GET /wsgi/async_notify/feed_long/any/600" - 200 1136
10.106.96.90 - admin [11/Jul/2013:13:32:53 -0700] "GET /wsgi/async_notify/feed_long/any/600/21" - 200 895
10.106.96.90 - admin [11/Jul/2013:13:32:53 -0700] "POST /wsgi/ChangeDefaultPasswordService" urn:changeDefaultPassword 200 858
10.106.96.90 - admin [11/Jul/2013:13:32:53 -0700] "GET /wsgi/async_notify/feed_long/any/600/22" - 200 478
10.106.96.90 - admin [11/Jul/2013:13:32:54 -0700] "GET /wsgi/async_notify/feed_long/any/600/23" - 200 586
10.106.96.90 - admin [11/Jul/2013:13:33:31 -0700] "GET /wsgi/async_notify/feed_long/any/600/24" - 200 895
10.106.96.90 - admin [11/Jul/2013:13:33:36 -0700] "GET /wsgi/async_notify/feed_long/any/600/25" - 200 1090
10.106.96.90 - admin [11/Jul/2013:13:33:36 -0700] "POST /wsgi/CustomerWizardActionService" urn:startWizardProcess 200 878
10.106.96.90 - admin [11/Jul/2013:13:33:36 -0700] "GET /wsgi/async_notify/feed_long/any/600/27" - 200 879
10.106.96.90 - admin [11/Jul/2013:13:33:36 -0700] "POST /wsgi/CustomerWizardService" urn:getWizardProcessInfo 200 701
10.106.96.90 - admin [11/Jul/2013:13:33:37 -0700] "POST /wsgi/CustomerWizardService" urn:getSSLUploadURL 200 539
10.106.96.90 - admin [11/Jul/2013:13:33:37 -0700] "POST /wsgi/CustomerWizardService" urn:getRoutesConfigUploadURL 200 564
10.106.96.90 - admin [11/Jul/2013:13:33:38 -0700] "POST /wsgi/CustomerWizardService" urn:getLnetConfigUploadURL 200 560
10.106.96.90 - admin [11/Jul/2013:13:33:38 -0700] "POST /wsgi/CustomerWizardService" urn:getLnetConfigUploadURL 200 560
10.106.96.90 - admin [11/Jul/2013:13:33:38 -0700] "POST /wsgi/CustomerWizardService" urn:getLnetConfigUploadURL 200 560
```

HA logs

Heartbeat is a daemon that provides cluster infrastructure (communication and membership) services to its clients. This allows clients to know about the presence (or absence) of peer processes on other machines and easily exchange messages with them.

Each pair of nodes has a shared HA configuration, which means that, if something goes wrong on any node in the pair the other node takes over all of services such as the lustre targets, Linux services, etc.

The logging feature used by HA writes down HA changes (resource transition, service restart, nodes being unresponsive) into log files named **ha-local.log** (for MGMT nodes) or **ha.log** (for all other nodes). All of these logs are stored in the **/var/log** dir and are linked to a shared storage providing the benefit that if one of the MGMT nodes' goes down the logs will not get lost. They can be accessed from the other node.



Support File Content

Collection date : Jul 13, 2013 10:32:53
Reason : Failover

Viewing logs inside the support file

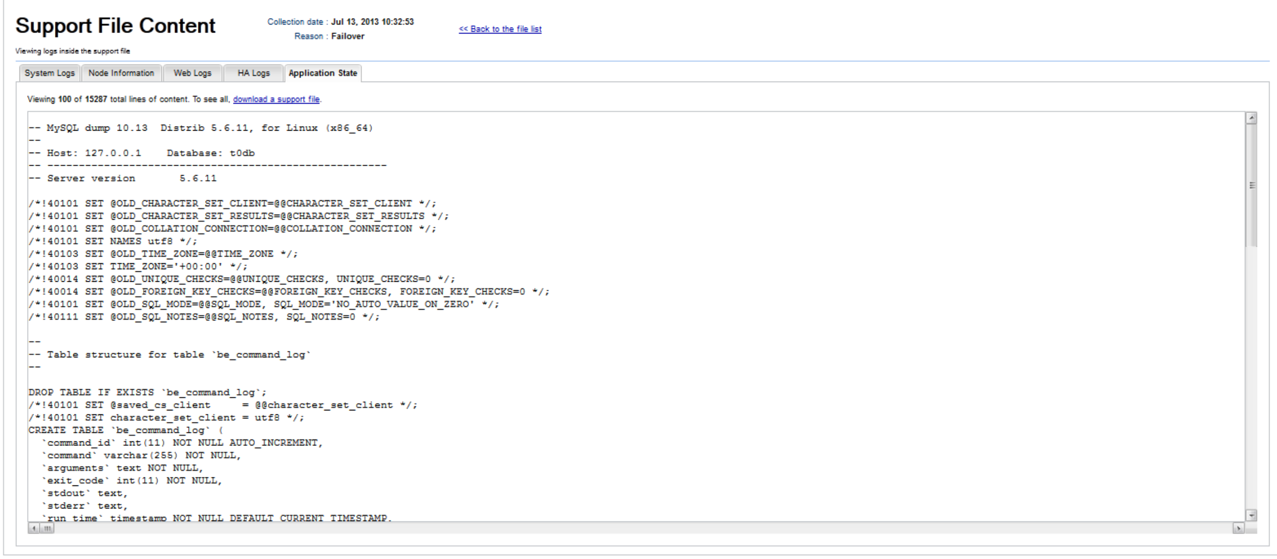
Log file : ha-local.log-20130713.gz.

Viewing 100 of 2773 total lines of content. To see all, download a support file.

```
Jul 12 00:01:04 cstor00 logd: [3339]: info: SIGHUP received, re-opened log files
Jul 12 00:02:36 cstor00 mdadm_conf_regenerate[53204]: [53208]: INFO: Running monitor
Jul 12 00:02:36 cstor00 crm_resource: [53234]: info: Invoked: crm_resource -r cstor00_mdadm_conf_regenerate -g md5sum
Jul 12 00:02:36 cstor00 crm_resource: [53260]: info: Invoked: crm_resource -L
Jul 12 00:02:36 cstor00 crm_resource: [53279]: info: Invoked: crm_resource -r cstor01_mdadm_conf_regenerate -g md5sum
Jul 12 00:04:21 cstor00 cibadmin: [54779]: info: Invoked: cibadmin -Ql
Jul 12 00:04:21 cstor00 cibadmin: [54810]: info: Invoked: cibadmin -Ql
Jul 12 00:06:22 cstor00 cibadmin: [56888]: info: Invoked: cibadmin --query --local --xpath //*/primitive[@id='cstor00_md64-fsys']
Jul 12 00:06:22 cstor00 cibadmin: [56903]: info: Invoked: cibadmin --query --local --xpath //*/primitive[@id='cstor00_md64-fsys']
Jul 12 00:09:02 cstor00 cib: [15139]: info: cib_stats: Processed 21 operations (13333.00us average, 0% utilization) in the last 10min
Jul 12 00:09:21 cstor00 cibadmin: [59484]: info: Invoked: cibadmin -Ql
Jul 12 00:09:21 cstor00 cibadmin: [59506]: info: Invoked: cibadmin -Ql
Jul 12 00:10:36 cstor00 mdadm_conf_regenerate[60883]: [60887]: INFO: Running monitor
Jul 12 00:10:36 cstor00 crm_resource: [60911]: info: Invoked: crm_resource -r cstor00_mdadm_conf_regenerate -g md5sum
Jul 12 00:10:36 cstor00 crm_resource: [60940]: info: Invoked: crm_resource -L
Jul 12 00:10:37 cstor00 crm_resource: [60961]: info: Invoked: crm_resource -r cstor01_mdadm_conf_regenerate -g md5sum
Jul 12 00:14:21 cstor00 cibadmin: [64044]: info: Invoked: cibadmin -Ql
Jul 12 00:14:22 cstor00 cibadmin: [64081]: info: Invoked: cibadmin -Ql
Jul 12 00:16:22 cstor00 cibadmin: [66223]: info: Invoked: cibadmin --query --local --xpath //*/primitive[@id='cstor00_md64-fsys']
Jul 12 00:16:22 cstor00 cibadmin: [66238]: info: Invoked: cibadmin --query --local --xpath //*/primitive[@id='cstor00_md64-fsys']
Jul 12 00:18:37 cstor00 mdadm_conf_regenerate[67927]: [67931]: INFO: Running monitor
Jul 12 00:18:37 cstor00 crm_resource: [67955]: info: Invoked: crm_resource -r cstor00_mdadm_conf_regenerate -g md5sum
Jul 12 00:18:37 cstor00 crm_resource: [67981]: info: Invoked: crm_resource -L
Jul 12 00:18:37 cstor00 crm_resource: [68000]: info: Invoked: crm_resource -r cstor01_mdadm_conf_regenerate -g md5sum
Jul 12 00:19:02 cstor00 cib: [15139]: info: cib_stats: Processed 24 operations (16666.00us average, 0% utilization) in the last 10min
Jul 12 00:19:21 cstor00 cibadmin: [68892]: info: Invoked: cibadmin -Ql
Jul 12 00:19:22 cstor00 cibadmin: [68914]: info: Invoked: cibadmin -Ql
Jul 12 00:24:21 cstor00 cibadmin: [73359]: info: Invoked: cibadmin -Ql
Jul 12 00:24:22 cstor00 cibadmin: [73393]: info: Invoked: cibadmin -Ql
Jul 12 00:26:22 cstor00 cibadmin: [75472]: info: Invoked: cibadmin --query --local --xpath //*/primitive[@id='cstor00_md64-fsys']
```

Application state

Application State shows the internal contents of the database which is used to track the states of the management application. This information is presented only for transparency, so that you can review in full the information that is being transmitted to the support staff. It is not intended for you to understand this information and you should not use it, as it may change format from version to version. However, it is information that will help the support staff to understand the context of the issue you are reporting, and may help them to debug the issue or identify unusual circumstances in which the issue appeared.



Support File Content

Collection date : Jul 13, 2013 10:32:53
Reason : Failover

Viewing logs inside the support file

Viewing 100 of 15287 total lines of content. To see all, download a support file.

```
-- MySQL dump 10.13  Distrib 5.6.11, for Linux (x86_64)
--
-- Host: 127.0.0.1    Database: t0db
-- ------------------------------------------------------
-- Server version       5.6.11

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `be_command_log`
--

DROP TABLE IF EXISTS `be_command_log`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `be_command_log` (
  `command_id` int(11) NOT NULL AUTO_INCREMENT,
  `command` varchar(255) NOT NULL,
  `arguments` text NOT NULL,
  `exit_code` int(11) NOT NULL,
  `stdout` text,
  `stderr` text,
  `run_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
```

5.4 Use CSCLI for support bundles

Support bundles can be created using the `cscli support_bundles` command, documented on page 119.

To collect a support bundle manually using CLI commands:

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

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

2. Change to root user. Run:

   ```
   $ sudo su -
   ```

3. Collect the support bundle.

   – To collect the bundle using the default 45 minute time period, run:

   ```
   [root@n000]# cscli support_bundle -c
   ```

   Sample output:

   ```
   [root@snx11000n000 ~]# cscli support_bundle -c
   Collecting support bundle: id:4, nodes:all, time-window:45
      minute(s)
   ```

   – To collect the bundle with a different time period, run:

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

   Sample command and output:

   ```
   [root@snx11000n000 ~]# cscli support_bundle -c -t 90
   Collecting support bundle: id:4, nodes:all, time-window:90
      minute(s)
   ```

4. To check the status of the data collection, enter the following:

   ```
   [root@snx11000n000 ~]# cscli support_bundle -e 22
   support_bundle: Error: Collection of support bundle with id 22 is in
      progress
   ```

5. To export the support bundle, enter the following:

   ```
   [root@snx11000n000 ~]# cscli support_bundle -e 22
   Support bundle with id 22 saved in file support_bundle_2013-08-
      08_10-54-07_310920.tgz
   ```

5.5 Interpret Sonexion support bundles

This section contains an overview of the support bundle contents. 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.

5.5.1 System-wide logs

- **lbug_syslog.csv**

  This file contains syslog messages, in comma-separated value (CSV) format.

  NOTE: The following files are not intended for use by Sonexion end users, but they may be valuable to Cray personnel and OEMs to better understand system states and behavior.

- **logs/access.log**

  This log contains Apache HTTP access data.

- **logs/data_tables.sql**

  This log 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**

  This file contains the Apache error log.

- **logs/wsgi_access.log**

  This mod_wsgi access log contains records of web service calls made from the CSSM.

5.5.2 Node-specific Logs:

- **nodes/***nodename***/conman.log**

  This 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**

  This 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**

  This log contains a dump of kernel messages collected from the node.

- **nodes/***nodename***/fru_dump.yaml**

  This file 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**

  This 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**

  This log contains state data of the MDRAID arrays, i.e., content of the `/proc/mdstat` file.

- **nodes/***nodename***/processes.csv**

  This file 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**

  This 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**

  This file contains a list of all installed packages with version information (`rpm -qa output`).

- **nodes/***nodename***/states.csv**

  This file contains miscellaneous state data, including power, memory, uptime, CPU load, and Lustre targets.

6. Troubleshooting

This section provides troubleshooting information for the Sonexion system and describes installation and post-installation issues and workarounds. This document also outlines Lustre performance and tuning considerations, CSSM, Networking, RAID, and High Availability (HA).

> **NOTE**: Procedures shown in this section apply to a range of Sonexion releases. Each procedure specifies the releases to which it applies.

6.1 Lustre performance considerations and tuning

6.1.1 Prerequisites

- **Pre-flight check**: Make sure all firmware on tested hardware is at latest stable version and there are no known kernel performance issues related to the hardware.
- **Catalog problem areas**: Single (slow) disk drives can slow down storage arrays. CPUs can slow down storage arrays. Buggy interconnect drivers can reduce bandwidth and increase roundtrip times.

6.1.2 Hardware performance testing

1. Start from the bottom and work your way up (this is critical).
2. For a full picture of hardware capabilities:
   - Test individual components

   - Test components collectively

3. Single disks:

   - Use `dd` and `sgp_dd` (`sgpdd-survey`) - great tools
   - Test various block sizes
   - Test while other disks are being tested at the same time

4. Arrays

   - Use `dd` and `sgpdd-survey`
   - Test various block sizes

5. OSTs and MDTs

   - OST testing: Use `obdfilter-survey` to directly and effectively test OSTs. Recommended filter parameters are as follows:

     - MDRAID: nobjlo=1 nobjhi=1 thrlo=256 thrhi=256 size=65536 obdfilter-survey

     - GridRAID: nobjlo=1 nobjhi=1 thrlo=512 thrhi=512 size=131072 obdfilter-survey

   - MDT testing: Mounting the block device as **ldiskfs** lets you perform MDTEST testing, as well as other methods directly against the MDT.

     NOTE: Remember to remove the test files once you are finished.

6. Interconnects

   - Lustre LNET self-test is a great tool to test one or more nodes on your network
   - LNET is protocol-neutral and runs at or near full wire speed

6.1.3 Benchmarking interconnect

- Establish a test baseline

- Are your results consistent with earlier tests?

- Test different nodes on different switching equipment

- Test across switching equipment

- Test multiple nodes? Are the test results expected?

- Adjust tunable parameters (`max_rpcs_in_flight`, `max_dirty_mb`, etc.)

6.1.4 Benchmarking - RAID tuning

- Making Lustre aware of the RAID layout can dramatically improve performance (especially on RAID6 solutions)

- Consider a RAID6 (6+2) configuration consisting of 64kB strides (made of 4kB blocks): (64kB / stride) / 4kB/blocks = 16 blocks per stride (64kB * 6 stripes) / 4kB/blocks = 96 blocks

Specify:

```
--mkfsoptions="-E stride=16,stripe_width=96"
```

Use the `mkfs.lustre` option when initially formatting the file system (this could be specified in installation YAML file).

6.1.5 Benchmarking - direct MDT and OST testing

- MDT testing:
  - MDTEST utility generates lots of small I/O activity against the MDT.
  - To improve metadata performance testing, mount the same Lustre file system multiple times on the clients.
  - Run multiple MDTEST iterations over a specific time period to establish minimum / maximum performance characteristics of the MDT.
- OST testing:
  - `obdfilter-survey` provides detailed data of OST read/write performance through the Lustre block device interface driver (however it may, under some configurations result in severe issues with cluster health, i.e. nodes may get in a panic state).
  - OST pools feature isolates individual OST sets on selected OSSs. OST pools are especially useful when:
    - Only limited clients are available
    - To determine optimal ratios of OSTs to OSSs
    - To locate interconnect bottlenecks between OSSs

6.1.6 Benchmarking - single client testing

- Measure performance characteristics on a single-client basis first
- Test various workloads
- Utilize different striping schemas
- Utilize small and large block size reads / writes to establish where ideal performance numbers can be achieved
- Use tools such as IOR and IOzone to simulate different types of loads

6.1.7 Benchmarking - multi-client testing

- Performance expectations for multi-client testing should be based on the results of earlier single client testing

- Determine the number of clients needed to fully saturate a single OSS
- Use OST pools to control the number of clients
- Use MPI IO to collect accurate performance data
- Use IOzone (preferred over IOR) for multi-client testing
- Different tools generate different results which should not be compared to one another

  For example, do not compare IOR results against IOzone results (apples to oranges)

6.2 Management software issues

6.2.1 Warning while unmounting Lustre: "Database assertion: created a new connection but pool_size …"

Release 1.2.0

Problem description

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.

6.2.2 Invalid puppet certificate on diskless node boot-up

Releases 1.2.0, 1.2.1, 1.2.3, 1.3.1

Problem description

If attempts to login to 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

1. SSH into the primary MGMT node.
2. Sudo to root, by entering:

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

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

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

4. SSH into the OSS node.

   If the attempt to SSH into the node succeeds, go to Step 5.

   If the attempt to SSH into the node fails, run the command:

   1.2.x:

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

   1.3.1 or 1.4.0:

   ```
   [root@n000]# ssh nfsserv find
      /mnt/nfsdata/var/lib/puppet/ssl_persistent/ -name hostname.* -delete
   ```

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

5. Sudo to root, by entering:

   ```
   [OSS node]$ sudo su -
   ```

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

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

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

   ```
   [OSS node]# puppetd –tv
   ```

8. Exit back out to the management node, by entering:

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

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

   For 1.2.x, run:

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

   For 1.3.1 or 1.4.0, run:

```
[root@n000]# ssh nfsserv rsync –zaHv --numeric-id
   hostname:/var/lib/puppet/ssl/ /mnt/nfsdata/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
    ```

The above command checks that the certificate file for each diskless node is the same in that nodes `/var/lib/puppet/ssl/certs` directory and the **`/var/lib/puppet/ssl_persistent/certs`** directory.

11. Verify that the current puppet certificate is valid. Run:

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

12. Using the Legacy HotFix Checker, determine if the HotFix 1.2.0-TRT-2 (the time synchronization hotfix) is installed. If it is not, then install it.

6.2.3 Need to change LDAP settings after GUI/wizard is complete

All releases

Problem description

Provide ability to change the LDAP settings post installation.

Workaround

Run this command from the MGS node:

```
[MGS]# /opt/xyratex/bin/beLDAPConfig.sh -H "host" -b "BaseDN" -p "UserDN"
   -g "GroupDN"
```

6.2.4 Unclean shutdown of management node causes database corruption

Release 1.2.0

Problem description

If the management node hosting the MySQL server (usually node 0, the primary MGMT server) is shut down uncleanly, the LMT database (named `filesystem_`*filesystem_name*) can become corrupt. This can manifest in several ways, including out-of-date information in the performance tab and problems accessing the management database **t0db**. There will usually be errors in the file `/var/log/mysql.log` indicating which tables are corrupt:

```
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 be repaired
```

Workaround

To repair the corrupted tables, execute the following procedure, saving output using the script command. Note that for very large tables, repair operations can create temporary files that are larger than the available filesystem space. It is recommended that the available space be monitored during this procedure.

1. As root, check all tables in the **filesystem_***filesystem_name* database:

   ```
   [root@snx11000n000 filesystem_ snx11000]# mysqlcheck filesystem_fs1
   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: 25CS-1600
   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

2. Repair all tables:

   ```
   [root@snx11000n000 filesystem_fs1]# mysqlcheck -s -r filesystem_fs1
   [root@snx11000n000 filesystem_fs1]#
   ```

3. Verify that repair worked and that all tables are OK:

   ```
   [root@snx11000n000 filesystem_fs1]# mysqlcheck filesystem_fs1
   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]

If there are still problems after this, contact Cray support.

6.2.5 Many nodes flapping

Release 1.2.0, 1.2.1

This log message indicates that the resources are changing state more often than bebundd expects. The purpose of this threshold is to prevent bebundd from collecting failover-initiated support bundle on each stop-start event. By itself, this error is benign. However, it may suggest other failover-related issues are occurring on the system.

6.3 Networking issues

6.3.1 Recovering from a top-of-rack (TOR) Ethernet switch failure

Release 1.2.0

Problem description

A failure on the top-of-rack switch makes some nodes inaccessible.

Workaround

If the failure occurred on the TOR switch to which the quad node MMU is connected, reboot the entire system. If it affected only expansion racks (and not the MMU), reboot the affected nodes. In either case, refer to the *Sonexion Power On / Power Off Procedures,* Cray publication HR5-6127.

6.3.2 Reseating a problematic high-speed network cable

Release 1.2.1

Problem description

On occasion, a node may lose its connection to the InfiniBand fabric.

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.

A faulty InfiniBand 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:

1. SSH into the primary MGMT node.
2. Sudo to root, by entering:

   ```
   [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 querying, by entering:

```
[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 repluged, verify failover on the node that was unplugged, by entering:

   ```
   [root@n000]# crm_mon -1
   ```

10. Replug the cable.
11. If the LED switch for that cable was previously on, verify that it comes back on after the cable has been replugged. Depending on how long is required for discovery, this may take up to a minute.
12. Enable HA’s INFINIBAND FABRIC querying, by entering:

    ```
    [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, by entering:

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

6.4 RAID/HA issues

6.4.1 RAIDs are not assembled correctly on the nodes

Release 1.2.0, 1.2.1, 1.2.2, 1.2.3, 1.3.1

Problem description

When an MDRAID device fails (for example, as a result of a chassis event temporarily removing several disks) the STONITH high-availability (HA) 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 -RAID resource but leaves the first three resources in the group assembled.

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, a chassis 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 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:

- 8 OSTs, each a RAID6 array consisting of 8 data disks and 2 parity disks
- 2 SSDs partitioned to create multiple independent RAID1 slices, used for MDRAID write intent bitmaps and external OST/ldiskfs file system journals
- 2 hot standby 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.

When an MDRAID device fails (for example, as a result of a chassis event 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 RAID resource but leaves the first three resources in the group assembled.

This procedure describes how to manually recover the RAID array.

Preparing to recover a failed RAID array

1. Log into the primary MGMT node via SSH.
2. Change to root user, by entering:

   ```
   [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). On the primary MGMT node, run:

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

   Here is sample output:

   ```
   [root@snx11000n000 ~]# pm -1 snx11000n004
   Command completed successfully
   ```

   If both OSS nodes are down, repeat Step 3 on the HA partner node.

4. Wait several minutes, and then log into the previously-downed OSS node via SSH to verify that it is back online, by entering:

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

   Here is sample output:

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

   If both OSS nodes were down, repeat Step 4 on the HA partner node.

5. Log into the OSS node that natively controls the MDRAID device, via SSH, by entering:

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

6. 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, by entering:

   ```
   [OSS node]# crm_mon -1
   ```

   IMPORTANT: When reviewing the `crm_mon` output, note the failed MDRAID device's resource group name. You will need this information when performing the procedure to recover the failed RAID array.

Here is sample output showing the resource groups and the failed actions messages:

```
 [root@snx11000n004 ~]# crm_mon -1
============
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
```

7. If the RAID fails to assemble with messages like this:

   ```
   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
   ```

   then the RAID recovery procedure has failed, go to Step 7. If the forceable reassembly does not produce these error messages, go to Step 8.

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. Use the **collect_superblock.sh** script to collect this data.

   a. Download the collect_superblock.sh script from Marlin or the XIC to /tmp on the primary MGMT node

   b. Run the script:

```
[root@n000]# /tmp/collect_superblock.sh /var/lib/mdraidscripts/mdacdm.conf
```

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

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

The procedure to prepare for recovering a failed RAID array is now complete. Proceed to the next section for the procedure to recover the RAID array.

Recovering a failed RAID array

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.

1. Stop the resource group containing the failed MDRAID device, by entering:

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

   Where *resource_group_name* is the resource group name, which is discussed in the note on page 49 and accompanying output.

   Here is sample output:

   ```
   [root@snx11000n005 ~]# stop_xyraid snx11000n004_md0-group
   [root@snx11000n005 ~]#
   ```

2. Unmanage the resource group, which will allow the resources to be started outside of the STONITH high-availability (HA) resource, by entering:

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

   Here is sample output:

   ```
   [root@snx11000n005 ~]# unmanage_xyraid snx11000n004_md0-group
   [root@snx11000n005 ~]#
   ```

3. Clean the resource group to remove the failed actions, by entering:

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

   Here is sample output:

   ```
   [root@snx11000n005 ~]# 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
   ```

4. If you determined in “Preparing to recover a failed RAID array”, page 49 , that the first three resources in the failed MDRAID device's resource group have failed over to the HA partner node, follow Steps a and b below:

a. Log into the OSS node that natively controls the MDRAID device, via SSH, by entering:

```
ssh OSS_nodename
```

b. Fail back resources to the OSS node, by entering:

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

Here is sample output:

```
[root@snx11000n004 ~]# failback_xyraid
[root@snx11000n004 ~]#
```

5. Determine if the --force argument is necessary to assemble the MDRAID device, by entering:

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

Here is sample output showing 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
   errorz
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 above assembly attempt was successful, proceed to Step 9.

6. Assemble the MDRAID device using the --force argument. If you are performing this procedure on a system running 1.2.x, run the following command:

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

On a system running 1.3.1:

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

Sample output of a successful 'forced' assembly of an MDRAID device:

```
[root@snx1100005 ~]# mdraid-activate -f i_am_sure_i_want_to_do_this,exit
   -d snx1100004_md0-group
mdadm: /dev/md/snx1100004:md129 has been started with 2 drives.
mdadm: failed to RUN_ARRAY /dev/md/snx1100004:md0: Input/output error
mdadm: Not enough devices to start the array.
mdadm: failed to RUN_ARRAY /dev/md/snx1100004: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/snx1100004:md0 for /dev/disk/by-
   id/wwn-0x5000c500212d3f2b
mdadm: Marking array /dev/md/snx1100004:md0 as 'clean'
mdadm: /dev/md/snx1100004:md0 has been started with 8 drives (out of 10).
assembled snx1100004:md0 in 1 tries
[root@snx1100005 ~]
```

On a system running 1.2.*x*:

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

Sample output of a successful 'forced' assembly of an MDRAID device:

```
[root@snx11000n004 ~]# mdraid-activate -df snx11000n004_md0-group
mdadm: /dev/md/snx11000n004:md129 has been started with 2 drives.
mdadm: forcing event count in /dev/disk/by-id/wwn-0x5000cca01c477720(3) from 3 upto 16
mdadm: forcing event count in /dev/disk/by-id/wwn-0x5000cca01c472818(8) from 3 upto 16
mdadm: forcing event count in /dev/disk/by-id/wwn-0x5000cca01b4a0ec4(9) from 3 upto 16
mdadm: clearing FAULTY flag for device 8 in /dev/md/snx11000n004:md0 for /dev/disk/by-
   id/wwn-0x5000cca01c477720
mdadm: clearing FAULTY flag for device 7 in /dev/md/snx11000n004:md0 for /dev/disk/by-
   id/wwn-0x5000cca01c472818
mdadm: clearing FAULTY flag for device 3 in /dev/md/snx11000n004:md0 for /dev/disk/by-
   id/wwn-0x5000cca01b4a0ec4
mdadm: Marking array /dev/md/snx11000n004:md0 as 'clean'
mdadm: /dev/md/snx11000n004:md0 has been started with 10 drives.
assembled snx11000n004:md0 in 1 tries
```

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

7. Run the `e2fsck` command on the MDRAID device.

   IMPORTANT: Do not run the **e2fsck** command on an OST larger than 16TB unless an appropriate up-to-date version of the **e2fsck** command is installed at your location. We strongly recommend using the version of the **e2fsck** command that is provided with Sonexion HotFix 1.2.0-MRP-1 or HotFix 1.2.1-MRP-1.

   – To check whether the OST size is larger than 16TB, run:

```
[OSS node]# mdadm --misc --detail /dev/md3 | grep Array
```

Example output of the command run on a very small OST:

```
[root@snx11000n204 ~]# mdadm --misc --detail /dev/md3 | grep Array
Array Size : 125829120 (120.00 GiB 128.85 GB)
```

- To check the version of **e2fsck** on your system, run:
- [OSS node]# **e2fsck -V**

  The version to run if the OST is larger than 16TB should be at least 1.42.6.x1. When you have the correct version of the e2fsck command, run:

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

  Here is sample output:

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

8. Stop the MDRAID device, by entering:

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

   Here is sample output:

   ```
   [root@snx11000n004 ~]# mdraid-deactivate snx11000n004_md0-group
   [root@snx11000n004 ~]#
   ```

9. Manage the MDRAID device's resource group, by entering:

   ```
   [OSS node]# manage_xyraid resource_group_name
   ```

   Here is sample output:

   ```
   [root@snx11000n004 ~]# manage-xyraid snx11000n004_md0-group
   [root@snx11000n004 ~]#
   ```

10. Start the MDRAID device's resource group, by entering:

    ```
    [OSS node]# start_xyraid resource_group_name
    ```

    Here is sample output:

    ```
    [root@snx11000n004 ~]# start-xyraid snx11000n004_md0-group
    [root@snx11000n004 ~]#
    ```

11. Use the `crm_mon` utility to verify that the MDRAID device's resource group started correctly, which can take several minutes, by entering:

    ```
    [OSS node]# crm_mon -1
    ```

    Here is sample output showing a healthy OST group:

```
[root@snx11000n004 ~]# crm_mon -1
============
Last updated: Wed Jan 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-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): 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_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
```

6.4.2 Starting Lustre on a given node without mounting the fsys resource

Release 1.2.0, 1.2.1

Problem description

The need may arise to start Lustre on a given node without starting the fsys resource (i.e. mounting the device), or to stop the fsys resource but leave the mdraid assembled.

Workaround

Run one of the following commands, as applicable:

To assemble the RAID arrays without mounting the filesystem, use the following commands:

1. To mount all resources on a node except `fsys`, run:

```
[node]# "crm_mon -1r | awk '/wibr|jnlr|wibs|raid/ {print $1}' |
   xargs -I {} start_xyraid {}"
```

2. To mount all resources except `fsys` on all OSSes, run:

```
[root@n000]# pdsh -g oss=primary "crm_mon -1r | awk '/wibr|jnlr|wibs|raid/
   {print \$1}' | xargs -I {} start_xyraid {}"
```

NOTE: The \ `in \$1` is required to escape the `$1`.

To stop only the `fsys` resource while leaving the mdraid assembled, use the following commands:

3. To stop only the `fsys` resource on a node, run:

```
[node]# "crm_mon -1r | awk '/fsys/ {print $1}' | xargs -I {}
   stop_xyraid {}"
```

4. To stop only the `fsys` resources on all OSS nodes, run:

   ```
   [root@n000]# pdsh -g oss=primary "crm_mon -1r | awk '/fsys/ {print
      $1}' | xargs -I {} stop_xyraid {}"
   ```

6.4.3 Lost one OST during forced failover. Release 1.2.0

Problem description

During a system test with a forced failover, an OST was lost.

Workaround

Trigger a `kdump` on the node. In the console log from the trigger, you should be able to see that the node successfully boots into the `kdump` kernel, but shuts down when its DHCP request is not answered. In this case, it will either be a bad cable or NIC. Run the following command to check the connection speed; notice how the connection speed of **eth0** is only 100Mbit/sec, when all the other nodes are at 1000Mbit/sec.

Sample `kdump`:

```
Sending discover...
Unable to get a DHCP address ret[   26.547454] md: stopping all md devices.
ry...
No lease, failing
eth0 failed to come up
[   27.665615] sd 10:0:72:0: [sdbt] Synchronizing SCSI cache
[   27.671577] sd 10:0:13:0: [sdp] Synchronizing SCSI cache
[   27.677282] sd 3:0:0:0: [sdc] Synchronizing SCSI cache
[   27.682684] sd 3:0:0:0: [sdc] Stopping disk
[^@   27.691845] e
[^@   27.767758] e
[   27.772799] [   27.787961] mpt2sas0: sending message unit reset !!
[   27.794746] mpt2sas0: message unit reset: SUCCESS
[   27.799677] mpt2sas 0000:12:00.0: PCI INT A disabled
[^@   31.344980] A[   31.458896] Disabling non-boot CPUs ...
[   31.462916] Power down.
```

1. Log in to the affected node using an admin account with the password set during the first-run (customer wizard) configuration, by entering:

   ```
   [MGMT]$ ssh admin@oss_nodename
   ```

2. Change to root user.

   ```
   [OSS]$ sudo su -
   ```

3. Check the connection speed, by entering:

   ```
   [OSS]# pdsh -g oss "ethtool eth0 | grep Speed"
   ```

4. The following is sample output:

```
[root@snx11000n000 ~]# pdsh -g oss "ethtool eth0 | grep Speed"

   snx11000n006: Speed: 1000Mb/s
```

```
snx11000n007: Speed: 1000Mb/s
snx11000n004: Speed: 1000Mb/s
snx11000n005: Speed: 100Mb/s
```

Replace the cable and run the check again. If the speed comes back, then it was the cable. If the speed did not return to normal, then replace the NIC card. Verify that the speed has now returned to normal.

6.4.4 Multiple HDDs spontaneously drop out of RAID arrays (heap overflows)

Release 1.2.0

Problem description

Errors in the SAS subsystem can trigger a reset of the SAS firmware or hardware. Even if the hardware/firmware recovers, the effect of this reset is seen at the Sonexion level as a large number of HDDs dropping out of RAID arrays. The most common cause of these resets is a heap overflow in the GEM firmware.

Workaround

First, determine if the drive drop-out were caused by a heap overflow. To determine if a drive drop-off was caused by a heap overflow, collect the GEM logs using `ddump`:

```
[MGMT0] conman nodename-gem
[gem] ddump
[gem] -ddump
```

The `ddump` command collects the local canister's GEM logs, and `-ddump` collects the partner node's GEM logs. The output from these commands is written to:

```
/var/log/conman/nodename-gem.log
```

Search the GEM log for the phrase “heap overflow detected”.

If these entries are not present, a heap overflow may not be responsible for the drive drop-offs. Please contact Cray support.

If these entries are present, then the problem was caused by a heap overflow. Now, determine if the software automatically recovered. Examine the kernel messages (which can be obtained using the dmesg command) for the following entries:

```
Restoring state sled x element x
Sled x element x version x.x.x.xx
```

If these kernel messages are not present, the problem can manifest itself in dm_report as empty drive slots. For example:

```
[root@n000]# cat dm_report.txt
Diskmonitor Inventory Report: Version: 1.0-2020.xrtx.2206  Host:
   snx110001  Time: Mon Mar 18 08:33:12 2013
```

```
encl:   0, wwn: 50050cc10c40036d, dev:     /dev/sg0, slots:  84,
   vendor: CRAY , product_id: UD-8435-CS-1600
slot:   0, status: Empty
slot:   1, status: Empty
slot:   2, status: Empty
slot:   3, status: Empty
slot:   4, status: Empty
slot:   5, status: Empty
slot:   6, status: Empty
slot:   7, status: Empty
slot:   8, status: Empty
slot:   9, status: Empty
slot:  10, status: Empty
slot:  11, status: Empty
slot:  12, status: Empty
slot:  13, status: Empty
slot:  14, wwn: 5000cca01c3f18f0, cap: 2000398933504, dev:
   sdce, parts: 0, status: Foreign Arrays
slot:  15, wwn: 5000cca01c3f1130, cap: 2000398933504, dev:
   sdcc, parts: 0, status: Foreign Arrays
slot:  16, wwn: 5000cca01c3f13b4, cap: 2000398933504, dev:
   sdcd, parts: 0, status: Foreign Arrays
slot:  17, wwn: 5000cca01c3e159c, cap: 2000398933504, dev:
   sdcf, parts: 0, status: Foreign Arrays
```

The expander is either hung or has become defective. Issue a reset to the expander and reboot GEM using the following commands:

1. SSH into the problematic OSS node's partner
2. Sudo to root
3. Unmanage the HA resources:

   ```
   [OSS]# unmanage xyraid all
   ```

4. From the MGMT node, conman into GEM on the problematic OSS node:

   ```
   [root@n000]# conman nodename-gem
   ```

   NOTE: Steps 5 and 6 must be performed within one second of each other:

5. Reboot all the expanders on this node:

   ```
   [GEM] gncli exp:local reboot
   ```

6. Reboot GEM:

   ```
   [GEM] reboot
   ```

7. Exit **conman**.

   Wait approximately one minute, then run the following command on the problematic node:

   ```
   [OSS node]# sg_map -i
   ```

This will check the status of the previously empty drive slots. If HDDs are present, then the reset cleared the expander. If drives are still not present, contact Cray support.

6.4.5 MD device fails to assemble with the message: “mdadm: cannot reread metadata … aborting”

Release 1.2.0

Problem Description

When attempting to mount Lustre, an MD device fails to assemble with the error: mdadm: cannot reread metadata from /dev/disk/by-id/*WWN* - aborting.

This problem is caused by a faulty hard drive. The following procedures will guide you to restore the MD device.

Workaround

Power down the bad drive and re-activate the raid array. At the earliest opportunity, replace the hard drive using Field Replacement of 5U84 DDIC (in the SSU).

1. SSH into the primary management node. Run

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

2. Determine which OSS node controls the failed hard drive.
3. SSH into that OSS node. Run:

   ```
   [MGMT0] ssh OSS_node
   ```

4. Determine the slot of the drive that is faulty with the drive's WWN (which is in the error message). Run:

   ```
   [OSS node]# dm_report | grep WWN
   ```

   It might be necessary to remove the last character from the WWN string.

5. Use the poweroffdrive command to power down the failed drive. Run:

   ```
   [OSS node]# echo "poweroffdrive Slot" | wbcli /dev/device
   ```

   For example, to power off /dev/sg0 located in slot 45, run:

   ```
   [root@snx11000n012 ~]# echo "poweroffdrive 45" | wbcli /dev/sg0
   Error: I/O timeout. ***
   GEMLITE>
   GEMLITE>[root@snx11000n012 ~]#
   ```

6. Verify that the drive slot is now empty with the command dm_report. Run:

   ```
   [OSS node]# dm_report
   ```

   Sample output

```
[root@snx11000n012 ~]# dm_report
```

```
Diskmonitor Inventory Report: Version: 1.0-2020.xrtx.2206 Host: snx11000n012 Time: Sat
   Aug 24 21:34:07 2013
encl: 0, wwn: 50050cc10c400107, dev: /dev/sg0, slots: 84, vendor: CRAY , product_id: UD-
   8435-CS-1600
...
slot: 42, wwn: 5000c50034ce95ef, cap: 2000398933504, dev: sdm, parts:
0, status: Ok
slot: 43, wwn: 5000c500348c8c83, cap: 2000398933504, dev: sdk, parts:
0, status: Foreign Arrays
slot: 44, wwn: 5000c500348cf957, cap: 2000398933504, dev: sdl, parts:
0, status: Foreign Arrays
slot: 45, status: Empty
slot: 46, wwn: 5000c50034a1632f, cap: 2000398933504, dev: sdi, parts:
0, status: Foreign Arrays
slot: 47, wwn: 5000c5003488751f, cap: 2000398933504, dev: sdj, parts:
0, status: Ok
```

...

If the drive slot is not “empty”, contact Cray support.

7. Clean the fail-counts on this HA resource. Run:

   ```
   [OSS node]# clean_xyraid HA_OST_resource_group
   ```

   After the fail counts have been cleared, the HA resource should start the problematic array. If HA fails again while starting this HA group, contact Cray support.

8. If the OST starts on the node that doesn't normally host it, fail it back with this command. Run:

   ```
   [OST node]# failback_xyraid
   ```

   on the OST's normal host.

9. From the MGMT node, mount Lustre: Run:

   ```
   [root@n000]# cscli mount -f filesystem_name
   ```

10. At the earliest opportunity, replace the failed hard drive using Field Replacement of 5U84 DDIC (in the SSU).

6.4.6 MD device fails to assemble with the message: “mdadm: cannot reread metadata from /dev/disk/by-id/*WWN* - aborting.”

Release 1.2.0

Problem Description

When attempting to mount Lustre, an MD device fails to assemble with the error: `mdadm: cannot reread metadata from /dev/disk/by-id/`*WWN* `- aborting`.

This problem is caused by a faulty hard drive. The following procedures will guide you to restore the MD device.

Workaround

Power down the bad drive and re-activate the raid array. At the earliest opportunity, replace the hard drive. (See publication HR5-6098, *Maintenance and Replacement Procedures for Cray Sonexion Storage Systems*, “5U84 Disk”.)

1. SSH into the primary management node. Run

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

2. Determine which OSS node controls the failed hard drive.

3. SSH into that OSS node, by entering:

   ```
   [MGMT0] ssh OSS_node
   ```

4. Determine the slot of the drive that is faulty with the drive's WWN (which is in the error message), by entering:

   ```
   [OSS node]# dm_report | grep WWN
   ```

   NOTE: It might be necessary to remove the last character from the WWN string.

5. Use the poweroffdrive command to power down the failed drive, by entering:

   ```
   [OSS node]# echo "poweroffdrive Slot" | wbcli /dev/device
   ```

   For example, to power off /dev/sg0 located in slot 45, run:

   ```
   [root@snx11000n012 ~]# echo "poweroffdrive 45" | wbcli /dev/sg0
   Error: I/O timeout. ***
   GEMLITE>
   GEMLITE>[root@snx11000n012 ~]#
   ```

6. Verify that the drive slot is now empty with the command dm_report, by entering:

   ```
   [OSS node]# dm_report
   ```

   Sample output:

   ```
   [root@snx11000n012 ~]# dm_report
   Diskmonitor Inventory Report: Version: 1.0-2020.xrtx.2206 Host:
      snx11000n012 Time: Sat Aug 24 21:34:07 2013
   encl: 0, wwn: 50050cc10c400107, dev: /dev/sg0, slots: 84, vendor:
      CRAY , product_id: UD-8435-CS-1600
   ...
   slot: 42, wwn: 5000c50034ce95ef, cap: 2000398933504, dev: sdm,
      parts: 0, status: Ok
   slot: 43, wwn: 5000c500348c8c83, cap: 2000398933504, dev: sdk,
      parts: 0, status: Foreign Arrays
   slot: 44, wwn: 5000c500348cf957, cap: 2000398933504, dev: sdl,
      parts: 0, status: Foreign Arrays
   slot: 45, status: Empty
   slot: 46, wwn: 5000c50034a1632f, cap: 2000398933504, dev: sdi,
      parts: 0, status: Foreign Arrays
   slot: 47, wwn: 5000c5003488751f, cap: 2000398933504, dev: sdj,
      parts: 0, status: Ok
   ```

```
. . .
```

If the drive slot is not 'empty', contact Cray support.

7. Clean the fail-counts on this HA resource, by entering:

   ```
   [OSS node]# clean_xyraid HA_OST_resource_group
   ```

   After the fail counts have been cleared, the HA resource should start the problematic array. If HA fails again while starting this HA group, contact Cray support.

8. If the OST starts on the node that doesn't normally host it, fail it back with this command, by entering:

   ```
   [OST node]# failback_xyraid
   ```

   on the OST's normal host.

9. From the MGMT node, mount Lustre, by entering:

   ```
   [root@n000]# cscli mount -f filesystem_name
   ```

10. At the earliest opportunity, replace the failed hard drive, (See publication HR5-6098, *Maintenance and Replacement Procedures for Cray Sonexion Storage Systems*, "5U84 Disk".)

6.5 Other Issues

6.5.1 Recovery for a faulty drive

Below is an example of a driver failure that requires manual intervention. Follow the main FRU process under normal drive failure.

1. Check the device on the SG map before recovering a faulty drive as the device may change by entering:

   ```
   node00$ sudo sg_map –i –x
   ```

   The preceding command should produce the following output:

   ```
   [sudo@node00 ~]# sg_map -i -x
   /dev/sg0 0 0 0 0 0 /dev/sda SEAGATE ST9450404SS XQB6
   /dev/sg1 0 0 1 0 0 /dev/sdb SEAGATE ST9450404SS XQB6
   /dev/sg2 0 0 2 0 0 /dev/sdc SEAGATE ST9450404SS XQB6
   /dev/sg3 0 0 3 0 0 /dev/sdd SEAGATE ST9450404SS XQB6
   /dev/sg4 0 0 4 0 0 /dev/sde SEAGATE ST9450404SS XQB6
   /dev/sg5 0 0 5 0 0 /dev/sdf SEAGATE ST9450404SS XQB6
   /dev/sg6 0 0 6 0 0 /dev/sdg SEAGATE ST9450404SS XQB6
   /dev/sg7 0 0 7 0 0 /dev/sdh SEAGATE ST9450404SS XQB6
   /dev/sg8 0 0 8 0 0 /dev/sdi SEAGATE ST9450404SS XQB6
   /dev/sg9 0 0 9 0 0 /dev/sdj SEAGATE ST9450404SS XQB6
   /dev/sg10 0 0 10 0 0 /dev/sdk SEAGATE ST9450404SS XQB6
   /dev/sg11 0 0 11 0 0 /dev/sdl SEAGATE ST9450404SS XQB6
   /dev/sg12 0 0 12 0 0 /dev/sdm SEAGATE ST9450404SS XQB6
   /dev/sg13 0 0 13 0 0 /dev/sdn SEAGATE ST9450404SS XQB6
   ```

```
/dev/sg14 0 0 14 0 0 /dev/sdo SEAGATE ST9450404SS XQB6
/dev/sg15 0 0 15 0 0 /dev/sdp SEAGATE ST9450404SS XQB6
/dev/sg16 0 0 16 0 0 /dev/sdq SEAGATE ST9450404SS XQB6
/dev/sg17 0 0 17 0 0 /dev/sdr SEAGATE ST9450404SS XQB6
/dev/sg18 0 0 18 0 0 /dev/sds SEAGATE ST9450404SS XQB6
/dev/sg19 0 0 19 0 0 /dev/sdt SEAGATE ST9450404SS XQB6
/dev/sg20 0 0 20 0 0 /dev/sdu SEAGATE ST9450404SS XQB6
/dev/sg21 0 0 21 0 0 /dev/sdv SEAGATE ST9450404SS XQB6
/dev/sg22 0 0 22 0 0 /dev/sdw HITACHI HUSSL4010ASS600 A182
/dev/sg23 0 0 23 0 0 /dev/sdx HITACHI HUSSL4010ASS600 A182
/dev/sg24 0 0 24 0 13 XYRATEX EB-2425-E6EBD 3022
/dev/sg25 1 0 0 0 0 /dev/sdy SEAGATE ST9450405SS 0002 internal drive
    0, left one
/dev/sg26 1 0 2 0 0 /dev/sdaa SEAGATE ST9450405SS 0002 internal
    drive , right one
```

2. Recover the faulty drive by entering:

   ```
   Node00$ sudo mdadm --manage /dev/md127 --remove faulty
   ```

   The preceding command produces the following output:

   ```
   [admin@node00 ~]$ sudo mdadm --manage /dev/md127 --re-add /dev/sdaa
    mdadm: re-added /dev/sdaa
   ```

6.5.2 Nodes are shown in “unknown” state in GUI

Release 1.2.0

Problem Description

If a node shows up as 'unknown' in the GUI, this indicates that the management nodes are unable to communicate with that node's IPMI interface.

Workaround

This problem is caused by an unresponsive BMC on the OSS node.

NOTE: This problem occurs less frequently in more recent USM releases. A USM upgrade may be advisable.

1. If possible, log into the node that has state 'unknown' as user root. Run :

   ```
   [Node]# ipmitool bmc reset cold
   ```

2. Wait approximately 2 minutes for the BMC to reboot, after which the node's status should no longer be 'unknown'.
3. If the node is still in the unknown state after resetting the BMC, check that this node's BMC network configuration is correct. On the node that's showing up as unknown, list the BMC's network parameters with the command:

   ```
   ipmitool lan print 1
   ```

   Example output:

```
[root@snx11000n01 ~]# ipmitool lan print 1
Set in Progress         : Set Complete
Auth Type Support       : NONE MD5 PASSWORD
Auth Type Enable        : Callback : NONE MD5 PASSWORD
                        : User     : NONE MD5 PASSWORD
                        : Operator : NONE MD5 PASSWORD
                        : Admin    : NONE MD5 PASSWORD
                        : OEM      :
IP Address Source       : Static Address
IP Address              : 172.16.0.101
Subnet Mask             : 255.255.0.0
MAC Address             : 00:1e:67:66:db:32
SNMP Community String   : public
IP Header               : TTL=0x00 Flags=0x00 Precedence=0x00
   TOS=0x00
BMC ARP Control         : ARP Responses Enabled, Gratuitous ARP
   Disabled
Gratituous ARP Intrvl   : 0.0 seconds
Default Gateway IP      : 172.16.0.101
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 1,2,3,4,6,7,8,9,11,12,13,15,16,17,18,0
Cipher Suite Priv Max   : caaaaXaaaaXaaaX
                                                   :     X=Cipher
   Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
[root@snx11000n000 ~]
```

4. Verify that the field 'IP Address Source' is set to 'Static Address', not 'dhcp'. If 'IP Address Source' is set to 'dhcp', fix this with the command, run:

   ```
   [root@n000]# ipmitool lan set 1 ipsrc static
   ```

5. Also verify that the field 'IP Address' is correct. The correct address can be obtained with the command, run:

   ```
   [root@n000]# host hostname-ipmi
   ```

   Example output:

   ```
   [root@snx11000n004 ~]# host snx11000n004-ipmi
   snx11000n004-ipmi has address 172.16.0.110
   [root@snx11000n004 ~]#
   ```

6. If this isn't set correctly in the BMC, fix it using a command in the following form:

   ```
   ipmitool lan 1 set ipaddr 172.16.0.101
   ```

   Example output:

   ```
   [root@snx11000n004 ~]# ipmitool lan set 1 ipaddr 172.16.0.110
   ```

```
Setting LAN IP Address to 172.16.0.110
[root@snx11000n004 ~]#
```

7. If the node still shows up as 'unknown' after correcting the BMC network settings, contact Cray support at my.Cray.com.

6.5.3 SSUs failed after AC power loss

Release 1.2.0

Problem Description

Several SSU's failed on an 18 SSU file system as a result of an AC power loss.

Workaround

When a power loss occurs, the Sonexion system will automatically fail over the HA components ensuring continuous operation. In the event of a total power loss, the entire system will shutdown. To understand what may have caused the situation, review the GEM logs for messages similar to the following:

```
2012-10-23 10:19:16.005; ENC_MGT; batt_manager; 01; Power Loss (AC Fail) detected
2012-10-23 10:19:16.005; ENC_MGT; drive_manager; 01; Enclosure power loss detected
2012-10-23 10:19:16.005; ENC_MGT; power_manager; 01; Enclosure Power Loss (AC Fail)
   detected
```

Once power is restored and the system booted, manually re-power (re-boot) the failed nodes from the primary MGMT node, by entering:

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

6.5.4 CS-1600 OSS will not power up, BMC out of memory

Release 1.2.1. The CS-1600 OSS will not power up because the BMC is out of memory.

Problem Description

It was observed in the field that some CS-1600 OSS units were failing to power up. Investigation determined that the BMC IPMI system event logs (sel logs) on those nodes had grown so large that they had consumed all the BMC memory.

An out-of-memory BMC is known to cause problematic behavior.

Workaround

To clear the `sel log` on an affected node, run this command:

```
[root@n000]# ipmitool -H nodename-ipmi -U admin -P admin sel clear
```

To clear all the sel logs on a machine, run this command:

```
[root@n000]# for addr in $(awk '/ipmi/ {print $1}' /etc/hosts); do echo
   $addr; ipmitool -H $addr -U admin -P admin sel clear; done
```

6.5.5 No response when physically connecting to serial port

Release 1.2.1

Problem Description

There is no response when physically connecting to the serial port and starting a hyperterminal session when using the following settings:

- Baud Rate: 115200
- Data bits: 8
- Parity: none
- Stop bits: 1
- Flow control: none
- Function Keys are set to VT100+

Workaround

Only one serial connection at a time is possible. This includes virtual serial connections. If the serial port is not responding to a physical connection, it is very likely that the controller is connected somewhere else using a Serial-On-LAN (SOL) connection.

1. First, forcibly disconnect any serial connections (SOL sessions), use the following command:

   ```
   [root@n000]# ipmitool -H nodename-ipmi -U admin -P admin bmc reset cold
   ```

2. Then, attempt to connect to the physical serial port using the above hyperterminal settings.

6.5.6 OSS/MDS nodes go down during FS testing

All releases

Problem Description

This problem is likely caused by a Lustre crash and a kernel panic. To verify this problem, connect to the OSS node using a serial cable (refer to section 7.6 for the hyperterminal settings) or conman and press "&L". If there is a stacktrace with an LBUG error, this is the issue.

Workaround

Power-cycle the node.

6.5.7 Non-responsive server

Release 1.2.0

Problem Description

A server node failed and is not responding to power manage commands to reboot.

Workaround

To revive the node, using `conman`, run the `ipmi` command to start the node.

1. Log in via console manager, by entering:

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

2. Issue the following command, by entering:

```
[gem]# -ipmi_power 4
```

7. CSCLI User Documentation

This chapter provides reference information for Sonexion's CLI command interface.

CLI commands are organized by mode; that is, certain commands are available according to the mode (state) of the Sonexion system. Two modes are relevant to customers – Customer Wizard 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.

- Customer Wizard Mode
- Daily Mode

7.1 CSCLI Overview

7.1.1 Customer Wizard mode

Use Wizard mode to configure the Sonexion system for customer use (after factory provisioning and before daily operations mode). Customer Wizard (custWizard) 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 Customer Wizard and Daily Modes.

7.1.2 Daily mode

Use Daily Mode mode when the Sonexion system is fully operational and available to manage the Lustre file system and cluster nodes.

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 Sonexion software on Lustre nodes.

> **Note**: The 1.4.0 release introduces a “Guest” account that lets non-privileged users run some commands to obtain information about system using read-only access to the system. Depending on the privileges, a subset of CSCLI commands are provided for a Guest account.

7.1.3 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, it reports an error 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, this stops the validation of upcoming arguments and raises an exception.

7.1.4 CLI command summary

Table 2 summarizes the CLI commands, with columns indicating the mode or modes that include each command.

Table 2. CLI Command Summary

Wizard Mode	Guest	Daily Mode	Command	Description
Network Setup Commands				
x			set_network	Specifies a Sonexion network setup.
x	x		show_network_setup	Shows a Sonexion network setup.
x			reset_network_setup	Resets the network setup of an existing Sonexion system.
x			apply_network_setup	Applies a network setup to a Sonexion system.
User setup commands				
x			get_lustre_users_ad	Shows the Lustre file system's AD settings.
x			get_lustre_users_ldap	Shows the Lustre file system's LDAP settings.

Wizard Mode	Guest	Daily Mode	Command	Description
x			get_lustre_users_nis	Shows configured NIS settings.
x			set_lustre_users_ad	Sets the Lustre file system's AD configuration.
x			set_lustre_users_ldap	Sets the Lustre file system's LDAP configuration.
x			clear_lustre_users_ad	Clears the Lustre file system's AD settings.
x			set_lustre_users_nis	Configures Filesystem NIS settings.
x			clear_lustre_users_ldap	Clears the Lustre file system's LDAP settings.
x			clear_lustre_users_nis	Clears the Lustre file system's NIS settings.
System alert commands				
x	x	x	alerts	Displays current and historical system health alerts.
x		x	alerts_config	Shows and updates the alerts configuration.
x		x	alerts_notifiy	Enables or disables alert notifications.
Node Control Commands				
x		x	autodiscovery_mode	Enables or disables auto-discovery mode on system nodes.
		x	failback	Fails back resources for the specified node.
		x	failover	Fails over resources to the specified node.
x		x	mount	Mounts the Lustre file system in the cluster.
x		x	unmount	Unmounts Lustre clients or targets on the file system.
		x	show_nodes	Displays node information.
Administrative Commands				
x	x	x	fs_info	Retrieves file system information.
x		x	cluster_mode	Toggles the system among 'daily mode, ‘custWizard' and 'pre-shipment' modes.
x	x	x	fru	Retrieves FRU (replacement) information.
x	x	x	list	Lists all supported commands.
x	x	x	syslog	Retrieves syslog entries.
x		x	batch	Runs a sequence of CSCLI commands in a batch file.
x		x	ip_routing	Manages IP routing.
x		x	set_admin_passwd	Changes administrator user password on an existing Sonexion system.

Wizard Mode	Guest	Daily Mode	Command	Description
Configuration Commands				
		x	configure_hosts	Configures host names for discovered nodes.
		x	configure_oss	Configures a new OSS node.
		x	show_new_nodes	Displays a table with new OSS nodes and their resources.
Filter Commands				
		x	create_filter	Creates customer filters for nodes.
		x	delete_filter	Deletes customer filters for nodes.
	x	x	show_filters	Shows customized and predefined node filters.
Updating System Software				
		x	prepare_update	Updates the specified node.
		x	split_ha_partners	Splits a set of nodes into two sets with each set containing no HA pairs.
		x	update_node	Updates the software version on the specified node.
	x	x	show_node_versions	Shows the current software version on the specified nodes.
		x	show_version_nodes	Shows all nodes at the specified software version.
		x	show_update_versions	Shows available software versions in the Sonexion Management Server repository.
Managing node position in a Sonexion rack				
		x	get_rack_position	Indicates the specified node's position in the Sonexion rack.
		x	set_rack_position	Changes a given node position in the Sonexion rack.
Monitoring System Health				
x	x	x	monitor	Monitors the current health of the cluster nodes and elements.
x		x	netfilter_level	Manages the netfilter level.
Enabling RAID Checks				
		x	raid_check	Enables RAID checks on RAID devices.
x		x	rebuild_rate	Manages the RAID rebuild rate.
		x	set_date	Manages the date setting on the Sonexion system
		x	set_timezone	Manages the timezone setting on the Sonexion system.
x		x	sm	Manages the InfiniBand Subnet Manager.

Wizard Mode	Guest	Daily Mode	Command	Description
x	x	x	support_bundle	Manages support bundles and support bundle settings.

7.1.5 Changes in release 1.4.0

Table 3 shows CSCLI commands that were removed from this release.

Table 3. Removed CLI Commands

Mode	Command	Description	Component
Wizard	power_manage	Manages file power.	Power
Daily	fs_export	Manages file system exports.	File System

7.2 Network setup commands

The `network_setup` command manages network parameters for the Lustre file system. This command includes functions to show, set, apply, and reset Lustre network parameters.

7.2.1 Show network parameters

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]
```

where:

Optional Arguments	Description
-h |--help	Shows the help message and exits.
-c *cluster_name* |--cluster *cluster_name*	Specifies the cluster name.

7.2.2 Set network parameters

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]
```

where:

Optional Arguments	Description
`-h |--help`	
`-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.

7.2.3 Reset network parameters

The `reset_network_setup` command resets the Lustre network parameters by removing old values from the database and replacing them with default values.

Synopsis

```
$ cscli reset_network_setup [-h] [-y] [-c cluster_name]
```

where:

Optional Arguments	Description
`-h|--help`	Shows 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.

7.2.4 Apply network parameters

The `apply_network_setup` command applies new Lustre network parameters to the database.

Synopsis

```
$ cscli apply_network_setup [-h] [--yes] [-c cluster_name]
```

where:

Optional Arguments	Description
`-h|--help`	Shows 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.

7.3 User setup commands

User setup commands include functions to configure the system's AD and LDAP settings and change the administrative user's password (used for CSSM login).

7.3.1 Get the file system's AD settings

The `get_lustre_users_ad` command retrieves the file system's AD settings.

Synopsis

```
$ cscli get_lustre_users_ad [-h] [-f fs_name] [--yaml-format]
```

where:

Optional Arguments	Description
`-h|--help`	Shows the help message and exits.
`-f` *fs_name* `|--fs` *fs_name*	Specifies the file system name.
`--yaml-format`	Shows the *ad* configuration in YAML file format.

When executed without parameters the `get_lustre_users_ad` command shows the current status:

```
$ /opt/xyratex/bin/cscli get_lustre_users_ad
AD Configuration
Cluster             : lmtest3
Filesystem          : testfs
    LDAP Uri        :
    Base DN         :
    Bind DN         :
```

7.3.2 Get the file system's LDAP settings

The `get_lustre_users_ldap` command retrieves the file system's *LDAP* settings.

Synopsis

```
$ cscli get_lustre_users_ldap [-h] [--yaml-format]
```

where:

Optional Arguments	Description
`-h|--help`	Shows the help message and exits.
`--yaml-format`	Shows the LDAP configuration in YAML file format.

7.3.3 Get the file system's NIS settings

The `get_lustre_users_nis` command retrieves and displays the file system's NIS settings.

Synopsis

```
$ cscli get_lustre_users_nis [-h] [-f fs_name] [--yaml-format]
   where:
```

Optional Arguments	Description
`-h|--help`	Shows the help message and exits.
`-f` *fs_name* `|--f` *fs_name*	Shows the *nis* file system name.
`--yaml-format`	Shows the *nis* configuration in YAML file format.

7.3.4 Set the file system's AD settings

The `set_lustre_users_ad` command specifies the file system's AD settings.

Synopsis

```
$ cscli set_lustre_users_ad [-h] -f fs_name [-l ldap_uri] [-b base_dn]
   [-i bind_dn] [-p password]
```

where:

Optional Arguments	Description
`-f` *fs_name* `|--fs` *fs_name*	Specifies the file system name.

`-l` *ldap_uri* `| --ldap-uri` *ldap_uri*	Specifies the LDAP URI. For example: `LDAP://127.0.0.1:389`
`-b` *base_dn* `|--base-dn` *base_dn*	Specifies the LDAP base DN.
`-i` *bind_dn* `|--bind-dn` *bind_dn*	Specifies the LDAP bind DN.
`-p` *password* `|--password` *password*	Specifies the LDAP bind password.

7.3.5 Set the file system's LDAP settings

The `set_lustre_users_ldap` command specifies the file system's LDAP settings.

Synopsis

```
$ cscli set_lustre_users_ldap [-h] [-N] [-l ldap_uri] [-b base_dn] [-u user_dn]
    [-G group_dn] [-s hosts_dn] [-i bind_dn] [-p password]
```

where:

Optional Arguments	Description
`-h|--help`	Shows the help message and exits.
`-N|--noauth`	Disables the LDAP configuration on the system.
`-l` *ldap_uri* `| --ldap-uri` *ldap_uri*	Specifies the LDAP URI. For example: `LDAP://127.0.0.1:389`
`-b` *base_dn* `|--base-dn` *base_dn*	Specifies the LDAP base DN.
`-u` *user_dn* `|--user-dn` *user_dn*	Specifies the LDAP user DN.
`-G` *group_on* `|--group-dn` *group_dn*	Specifies the LDAP group DN.
`-s` *hosts_dn* `|--hosts-dn` *hosts_dn*	Specifies the LDAP hosts DN.
`-i` *bind_dn* `|--bind-dn` *bind_dn*	Specifies the LDAP bind DN.
`-p` *password* `|--password` *password*	Specifies the LDAP bind password.

7.3.6 Configure the file system's NIS Settings

The `set_lustre_users_nis` command configures the file system's NIS settings.

Synopsis

```
$ cscli set_lustre_users_nis [-h] -f fs_name [-s nis_server] [-d nis_domain]
```

where:

Optional Arguments	Description
`-h|--help`	Shows the help message and exits.
`-f` *fs_name* `|--fs` *fs_name*	Shows the file system's name.
`-s` *nis_server* `|--nis_server` *nis_server*	Shows the NIS server. For example: "10.0.0.10 10.0.0.11" or "server1 server2" .
`-d` *nis_domain* `|--nis_domain` *nis_domain*	Shows the *NIS* domain. For example: nisdomain.

7.3.7 Clear the file system's AD settings

The `clear_lustre_users_ad` command clears the file system's AD settings.

Synopsis

```
$ cscli clear_lustre_users_ad [-h] [-f fs_name] [--all]
```

where:

Optional Arguments	Description
`-h|--help`	Shows the help message and exits.
`-f` *fs_name* `|--fs` *fs_name*	Specifies the file system name.
`--all`	Cleans all file systems' configurations.

7.3.8 Clear the file system's LDAP settings

The `clear_lustre_users_ldap` command clears the file system's LDAP settings.

Synopsis

```
$ cscli clear_lustre_users_ldap [--yes] [-h]
```

where:

Optional Arguments	Description
`-h|--help`	Shows the help message and exits.
`|--yes`	Confirms the action to clear the file system's *ldap* settings.

7.3.9 Clear the file system's NIS settings

The `clear_lustre_users_nis` command clears the file system's NIS settings.

Synopsis

```
$ cscli clear_lustre_users_nis [-h] [-f fs_name] [--all]
```

where:

Optional Arguments	Description
`-h|--help`	Shows the help message and exits.
`-f` *fs_name* `|--fs` *fs_name*	Confirms the file system's name.
`--all`	Clears all the file system's configuration

7.4 System alert commands

Alert commands include functions to view and update the alerts configuration, turn on/off alert notifications, and display current and historical system alerts.

7.4.1 Display current and historic system alerts

The `alerts` command 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}
```

where:

Positional Arguments	Description
`nodes`	Shows alert history for nodes.
`elements`	Shows alert history for elements.

`nodes_active`	Shows current alerts for nodes.
`elements_active`	Shows current alerts for elements.
`thresholds`	Shows editable alert thresholds and their current settings.

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

Subcommand **alerts elements_active**

```
$ cscli alerts elements_active [-h] [-y] [-v] [-x]
   [-n node_spec | -g genders_query] [-S element_filter]
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-y |--yaml`	Outputs data in YAML format.
`-v |--verbose`	Outputs extra data.
`-x |--unhandled`	Shows 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 (for example, `node[100-110,120]`).
`-g` *genders_query* `|--genders` *genders_query*	Specifies node genders attributes query (for example, `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.

Subcommand **alerts nodes**

```
$ cscli alerts nodes [-h] [-y] [-s start_time] [-e end_time] [-m limit]
   [-n node_name] [-N {down,unreachable,up}]
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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.

Subcommand **alerts elements**

```
$ cscli alerts elements [-h] [-y] [-s start_time] [-e end_time] [-m limit]
   [-n node_name] [-U {unknown,warning,ok,critical}]
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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, then `--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.

Subcommand **alerts nodes_active**

```
$ cscli alerts nodes_active [-h] [-y] [-v] [-x]
   [-n node_spec | -g genders_query]
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-y |--yaml`	Outputs data in YAML format.
`-v |--verbose`	Outputs extra data.
`-x |--unhandled`	Shows 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. For example: `node[100-110,120]`)
`-g` *genders_query* `|--genders` *genders_query*	Specifies node genders attributes query (e.g. mds=primary).

Subcommand **alerts threshold**

```
$ cscli alerts thresholds [-h] [-y]
```

Threshold fields are:

`name`	Short identifier of the threshold
`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:

`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

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-y |--yaml`	Outputs data in YAML format.

7.4.2 Manage the alerts configuration

The `alerts_config` command enables you to view and update 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}
```

where:

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 notification subscribers.
`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.

Subcommand **email_off**

The `email_off` command turns off notifications for subscribers.

```
$ cscli alerts_config email_off [-h] -u email
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-u` *email* `|--user` *email*	Displays subscriber email. Notifies subscribers they have new mail in /var/spool/mail or admin.

Subcommand **thresholds**

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.

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.

If you would like to make a group of changes to the thresholds, you may 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.

```
$ cscli alerts_config thresholds [-h] -t threshold_name -g gender_name
    [-W warning_threshold_value][-C critical_threshold_value][-A]
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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.
-A |--apply-config	Applies the threshold configuration.

Subcommand **email_update**

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

Notification Periods

The Notification period are:

- 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]
```

where:

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

`-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`".

Subcommand **email_server_update**

The `email_server_update` command configures the SMTP server to send alerts to external email addresses.

```
$ cscli alerts_config email_server_update [-h] -s smtp_server_address
   [--port port] [-S email_from] [-d domain] [-u smtp_user] [-p smtp_password]
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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 the `--domain` is set, the default value for the sender is *cluster_name@domain*. If the `--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.

Subcommand **email_delete**

The `email_delete` deletes notifications to subscribers.

```
$ cscli alerts_config email_delete [-h] -u email
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-u` *email* `|--user` *email*	Displays subscriber email. Notifies that you have new mail in /var/spool/mail or admin.

Subcommand **email_add**

The `email_add` command adds a new notification subscriber.

```
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:

The possible levels 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)

where:

Optional Arguments	Description
`-h |--help`	Shows 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.

`-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).`

Subcommand **email_on**

The `email_on` command turns on notifications for subscribers.

```
$ cscli alerts_config email_on [-h] -u email
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-u` *email* `|--user` *email*	Displays subscriber email. Notifies that you have new mail in /var/spool/mail or admin.

Subcommand **email_server**

The `email_server` command displays the relay *smtp* server configuration.

```
$ cscli alerts_config email_server [-h]
```

where:

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

Subcommand **emails**

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:

The possible Levels 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 or element is flapping between statuses
  - When a node or element is in scheduled downtime

Synopsis

```
$ cscli alerts_config emails [-h] [-y] [-v] [-u email]
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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 that you have new mail in /var/spool/mail or admin.

7.4.3 Manage the alerts notification

The `alerts_notify` command turns alert notifications on or off.

Synopsis

```
$ cscli alerts_notify [-h] {on,off} ...
```

where:

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.

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]
```

where:

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 the element name.

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]
```

where:

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 what you are doing.

7.5 Node control commands

The node control commands are used to control individual Lustre nodes (MDS/MGS and OSSs) 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 exporter nodes.

7.5.1 Manage node auto-discovery

This command manages node auto-discovery in the Sonexion system.

Synopsis

```
$ cscli autodiscovery_mode [-h] [-s] [--mode {enabled,disabled}]
```

where:

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.

7.5.2 Manage node failback and failover

These commands manage node failback and failover in the Sonexion system.

Synopsis

```
$ cscli failback [-h] (-F filter_sid | -n node_spec)
$ cscli failover [-h] (-F filter_sid | -n node_spec)
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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.

7.5.3 Mount and unmount Lustre targets

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, then 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.

Synopsis

```
$ cscli mount [-h] -f fs_name [-n node_spec]
$ cscli unmount [-h] -f fs_name [-n node_spec]
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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.

7.5.4 Manage node power

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.

Synopsis

```
$ cscli power_manage [-h] (--filter filter_sid| -n node_spec)
   (--power-on|--power-off|--reboot|--cycle|--reset|--hand-over) [--force]
```

where:

Optional Arguments	Description
-h |--help	Shows 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. 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.
--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.

7.5.5 Show node information

This command displays information about specified system nodes.

Synopsis

```
$ cscli show_nodes [-h] [-F filter_sid] [-r] | --refresh
```

where:

Option	Description
`-h |--help`	Shows the help message an d exits.
`-F` *filter_sid* `|--filter` *filter_sid*	Specifies the node filter.
`-r |--refresh`	Specifies the refresh mode (press 'q' for quit).

7.6 Administrative commands

Administrative commands include functions to get file system and cluster node information, retrieve syslog entries, show FRU information and list available commands.

7.6.1 Show file system information

The `fs_info` command shows all file system information.

Synopsis

```
$ cscli fs_info [-h] [-f fs_name
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-f` *fs_name* `|--fs` *fs_name*	Shows the file system name.

7.6.2 Retrieve FRU information

The `fru` command lists the defined 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]]
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-a|--all`	Shows FRUs (including status) grouped by type, for all nodes in the system.
`-n` *node_spec* `|--nodes` *node_spec*	Shows FRUs (including status) grouped by element type, for a specified node(s) in the system.
`-t {ArrayDevice,BMC,Cooling, Enclosure,PSU,Battery, Enclosure_Electronics }`	Shows frus (including status) for the specified element type. Examples of element types: array device, BMC, PSU, battery.
`-i` *index* `|--index` *index*	Shows FRUs (including status) for specified elements within a list of elements of the same type.
`-l [`*history*`] | --history [`*history*`]`	Shows FRU history (default is 10 lines of history).

7.6.3 Change the Sonexion mode

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]
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-s |--status`	Shows the status of the cluster.
`--mode {daily,custwiz, pre-shipment}`	Switches to the specified mode. Switches to either daily mode, customer wizard mode or pre-shipment mode. **CAUTION**: Use of the pre-shipment option will delete any current configuration settings.
`--db-only`	Update only the database. Does not sync nodes via puppet. Valid only with the '`--mode` ' argument.

7.6.4 List commands

The `list` command shows a list of available commands in the current Sonexion mode.

Synopsis

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

where:

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

7.6.5 Display log information

The `syslog` command displays Lustre log entries.

Synopsis

```
$ cscli syslog [-h] [-m max] [-F] [-d duration] [-s start_time] [-e end_time] [-r]
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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).

7.6.6 Set administrator password

The `set_admin_passwd` command changes and sets an administrator password.

Synopsis

```
$ cscli set_admin_passwd [-h] [-p password]
```

where:

Optional Arguments	Description
`-h|--help`	Shows the help message and exits.
`-p|--`*password*	Specify the new administrator password string.

7.6.7 Run batch file

The `batch` command runs a sequence of CSCLI commands in a batch file.

Synopsis

```
$ cscli batch [-h] -b batch_file
```

where:

Optional Arguments	Description
`-h |--help`	Displays the help message and exits.
`-b` *batch_file* `|--batch-file` *batch_file*	Specifies the command batch file.

7.6.8 Manage IP routing

The `ip_routing` command manages IP routing to and from the system database.

Synopsis

```
$ cscli ip_routing [arguments]
```

where [*arguments*] are:

```
  --show|-s [--loadable]
```

or

```
  --load path_to_file
```

or

```
  --add | -a  --dest destination_ip --prefix prefix_len --router router_ip
```

or

```
  --update | -u --route-id route_id [--destdestination_ip]
 [--prefixprefix_len] [--routerrouter_ip]
```

or

```
  --delete | -d --route-id route_id
```

or

```
  --clear | -c
```

or

```
--apply | -a
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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`).

7.7 Configuration commands

The configuration commands specifies the *mac* address and hostname for a given node and configures *oss* nodes

7.7.1 Configure hosts

The `configure_hosts` command configures the MAC address and host names for the discovered node.

Synopsis

```
$ cscli configure_hosts [-h] -m mac_address --hostname hostname [-f]
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-m` *mac_address* `|--mac` *mac_address*	Shows the *mac_address* and node *mac* address.
`--hostname` *hostname*	Shows the new node hostname.
`-f |--force`	Forces the mode (to skip hostname validation).

7.7.2 Configure new OSS nodes

This command configures new OSS nodes in the Sonexion system.

Synopsis

```
$ cscli configure_oss [-h] -n node_spec (-A | -b bind_arrays)
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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. For example: `md[0-3].`

7.7.3 Show information about new OSS nodes

This command displays a table of new OSS nodes and their resources.

Synopsis

```
$ cscli show_new_nodes [-h] [-v]
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-v, --verbose`	Specifies the verbose mode.

7.8 Filter commands

The filter commands create and delete a filter.

7.8.1 Create a filter

The `create_filter` command creates a customer nodes filter.

Synopsis

```
$ cscli create_filter [-h] -i filter_sid -F filter_name -e filter_expr
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-i` *filter_sid* `|--id` *filter_sid*	Shows the symbol identifier of the filter.
`-F` *filter_name* `|--name` *filter_name*	Shows the filter name.
`-e` *filter_expr* `|--expression` *filter_expr*	Shows the filter expression. Examples: "`host1,host2`", "`host[1-3]`", "`mds=primary`".

7.8.2 Show filters

The `show_filters` command shows all filters.

Synopsis

```
   $ cscli show_filters [-h] [-P] [-C]
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-P |--predefined`	Shows only predefined filters.

`-C |--custom`	Shows only custom filters.

7.8.3 Delete a filter

The `delete_filter` command deletes a customer nodes filter.

Synopsis

`$ cscli delete_filter [-h] -i` *filter_sid*

where:

Optional Arguments	Description
`-h	--help`	Shows the help message and exits.
`-i` *filter_sid* `|--id` *filter_sid*	Shows the symbol identifier of the filter.

7.9 Updating system software

These commands prepare a software upgrade package for installation and apply it to system nodes.

7.9.1 Prepare a software update

The `prepare_update` command runs the software update preparation process.

Synopsis

```
$ cscli prepare_update [-h] [--run]
```

where:

Optional Arguments	Description
`-h	--help`	Shows the help message and exits.
`--run`	Prepares the software upgrade package for installation.

7.9.2 Update software on a system node

The `update_node` command updates software on the specified node(s).

Synopsis

`$ cscli update_node [-h] -n` *node_spec*

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-n` *node_spec* `|--node-spec` *node_spec*	Specifies hostnames of the nodes on which to update software.

7.9.3 Split HA partners

The `split_ha_partners` command Split a set of nodes into 2 sets, where each set contains no HA pairs.

Synopsis

```
$ cscli split_ha_partners [-h] -g genders_query
```

where:

Option	Description
`-h |--help`	Displays the help message and exits.
`-g` *genders_query* `|--genders` *genders_query*	Specifies a genders style when splitting *ha* pairs of *oss* nodes.

7.9.4 Show nodes at specified software version

The `show_version_nodes` command lists all system nodes at the specified software version.

Synopsis

```
$ cscli show_version_nodes [-h] [-q] -v sw_version
```

where:

Option	Description
`-h |--help`	Shows 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. Example: `mycluster[02-05,97-98]`.
`-v` *sw_version* `|--version` *sw_version*	Specifies the Sonexion software version.

7.9.5 Show software versions on specified nodes

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]
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-q, --query`	Controls output format. If this flag is specified, nodes in the output should be in genders style. Example: `mycluster[02-05,97-98]`
`-n` *node_spec*, `--nodes` *node_spec*	Specifies nodes to indicate the Sonexion software version.
`-g` *genders_query*	Specifies a gender's style query.

7.9.6 Showing available software versions

The `show_update_versions` command lists software versions available in the Sonexion Management (*MGMT*) Server repository.

Synopsis

```
   $ cscli show_update_versions [-h]
```

where:

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

7.10 Managing node position in a Sonexion rack

The 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 OSSs per SSU).

7.10.1 Get node position in a Sonexion rack

The `get_rack_position` command indicates the location of server nodes in a Sonexion rack.

Synopsis

```
   $ cscli get_rack_position [-h] -r rack_name [--yaml]
```

where:

Option	Description
`-h |--help`	Displays the help message and exits.
`-r` *rack_name*, `--rack` *rack_name* `|--rack` *rack_name*	Specifies the rack containing the node(s).
`--yaml`	Prints node rack position information in *YAML* file format.

7.10.2 Set node position in a Sonexion rack

The `set_rack_position` command sets the location of server nodes in the Sonexion rack or moves a node to another rack.

Synopsis

```
$ cscli set_rack_position [-h] -r rack_name [--yaml] -n node_spec –p position
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-r` *rack_name* `|--rack` *rack_name*	Specifies the rack containing the node(s).
`-y` *yaml path* `|--yaml` *yaml_path*	Loads rack position information in *yaml* file format.
`-n` *node_spec* `|--node` *node_spec*	Specifies the node(s) hostname.
`-p` *position* `|--position=`*position*	Specifies the node position in rack units (Us).

7.10.3 Monitor system health

The `$ cscli monitor` command monitors and displays current health and status information for the cluster nodes and elements.

Synopsis

```
$ cscli monitor [-h] {nodes,elements,health} ...
```

where:

Positional Arguments	Description
`health`	Current overall health information - status summary.
`nodes`	Current status for nodes.
`elements`	Current status for elements.

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

The `$ cscli monitor nodes` command monitors individual nodes.

Synopsis

```
$ cscli monitor nodes [-h] [-y] [-v] [-n node_spec | -g genders_query]
   [-N {down,unreachable,up,pending}]
```

where:

Positional Arguments	Description
`-h |--help`	Shows 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. Example: `node[100-110,120]`).
`-g` *genders_query*	Displays the node genders attributes query (for example, `mds=primary`).

`-N {down,unreachable,up,pending}`
`|--nodestatus`
`{down,unreachable,up,pending}`
`node status`
	Displays node status.

The `$ cscli 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]
```

where:

Positional Arguments	Description
`-h |--help`	Shows 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 (for example, `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.

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

7.10.4 **cscli monitor** command examples

`cscli monitor` command examples and outputs are given below including OK, WARNING and CRITICAL.

Examples

```
[root@cstor01n000 ~]# 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

command , but no output means no errors
[root@cstor01n000 ~]# cscli monitor elements -U unknown -U pending -U warning -U
   critical
[root@cstor01n000 ~]# cscli monitor elements -U unknown
[root@cstor01n000 ~]# cscli monitor elements -U pending
[root@cstor01n000 ~]# cscli monitor elements -U critical
[root@cstor01n000 ~]# cscli monitor elements -U warning

[root@cstor01n000 ~]# cscli monitor nodes
cstor01n000:  UP  for 28d 15h 33m 26s  checked 2014-02-06 10:00:36  "PING OK - Packet
   loss = 0%, RTA = 0.03 ms"
cstor01n001:  UP  for 15d 18h 14m 44s  checked 2014-02-06 10:02:56  "PING OK - Packet
   loss = 0%, RTA = 0.16 ms"
cstor01n002:  UP  for 15d 18h  6m 54s  checked 2014-02-06 10:05:36  "PING OK - Packet
   loss = 0%, RTA = 0.18 ms"
cstor01n003:  UP  for 15d 18h  8m 54s  checked 2014-02-06 10:03:36  "PING OK - Packet
   loss = 0%, RTA = 0.18 ms"
cstor01n003-Enclosure-R1C1-21U:  UP  for 28d 15h 35m  8s  checked 2014-02-06 09:55:36
   "OK"
cstor01n004:  UP  for 5d 17h 14m 44s  checked 2014-02-06 10:01:26  "PING OK - Packet
   loss = 0%, RTA = 0.16 ms"
cstor01n005:  UP  for 5d 18h 30m 14s  checked 2014-02-06 10:03:26  "PING OK - Packet
   loss = 0%, RTA = 0.19 ms"
cstor01n005-Enclosure-R1C1-5U:  UP  for 28d 15h 34m 12s  checked 2014-02-06 10:02:36
   "OK"

[root@cstor01n000 ~]# cscli monitor elements
```

Subset of output:

```
cstor01n000  "Arrays and Disk Status":  OK  for 28d 15h 33m 49s  checked 2014-02-06
   10:00:52  "All arrays are operating normally"
cstor01n000  "Current Load":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "OK
   - load average: 0.08, 0.03, 0.02"
cstor01n000  "Current Users":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52
   "USERS OK - 1 users currently logged in"
cstor01n000  "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%):"
cstor01n000  "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), eth1=(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)"
cstor01n000  "RAM usage":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52  "OK -
   11.6% (3807704 kB) used."
cstor01n000  "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)"
```

```
cstor01n000  "Total Processes":  OK  for 28d 15h 33m 49s  checked 2014-02-06 10:00:52
   "PROCS OK: 407 processes with STATE = RSZDT"
cstor01n000  "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"
cstor01n000  "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"
cstor01n000  "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"
cstor01n000  "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"
cstor01n000  "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"
cstor01n000  "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"
cstor01n004  "Arrays and Disk Status":  OK  for 2d 14h 28m 10s  checked 2014-02-06
   10:03:42  "All arrays are operating normally"
cstor01n004  "Current Load":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42  "OK
   - load average: 0.01, 0.01, 0.01"
cstor01n004  "Current Users":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42
   "USERS OK - 0 users currently logged in"
cstor01n004  "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%):"
cstor01n004  "Lustre Health":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42
   "OK:Lustre is ok"
cstor01n004  "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)"
cstor01n004  "RAM usage":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42  "OK -
   12.9% (4203984 kB) used."
cstor01n004  "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)"
cstor01n004  "Total Processes":  OK  for 28d 15h 32m 14s  checked 2014-02-06 10:03:42
   "PROCS OK: 1239 processes with STATE = RSZDT"
```

```
root@cstor01n000 ~]# cscli monitor elements -v
```

Subset of output:

```
cstor01n000  "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/cstor01n003:md64 doesn't have associated WIB array

Degraded Array information:

All arrays are in clean state on node cstor01n000"

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
------------------------------------------------------------------------
cstor01n000  "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
------------------------------------------------------------------------
cstor01n000  "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@cstor01n000 ~]# cscli monitor elements -S enclosures
cstor01n003-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"
cstor01n003-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"
cstor01n003-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"
cstor01n003-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"
cstor01n003-Enclosure-R1C1-21U  "Power Statistics":  OK  for 15d 17h 19m  4s  checked
   2014-02-06 09:58:06  "Summary: Total System Power 168W"
cstor01n003-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"
cstor01n003-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"
```

```
cstor01n005-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"
cstor01n005-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"
cstor01n005-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"
cstor01n005-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"
cstor01n005-Enclosure-R1C1-5U  "Power Statistics":  OK  for 28d 15h 28m 40s  checked
   2014-02-06 09:57:42  "Summary: Total System Power 1068W"
cstor01n005-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"
cstor01n005-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@cstor01n000 ~]#
[root@cstor01n000 ~]# cscli monitor nodes -n cstor01n004
cstor01n004:  UP  for 5d 17h 17m 26s  checked 2014-02-06 10:06:36  "PING OK - Packet
   loss = 0%, RTA = 0.17 ms"
[root@cstor01n000 ~]# cscli monitor elements  -n cstor01n004
cstor01n004  "Arrays and Disk Status":  OK  for 2d 14h 30m 42s  checked 2014-02-06
   10:03:42  "All arrays are operating normally"
cstor01n004  "Current Load":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42  "OK
   - load average: 0.01, 0.01, 0.01"
cstor01n004  "Current Users":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42
   "USERS OK - 0 users currently logged in"
cstor01n004  "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%):"
cstor01n004  "Lustre Health":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42
   "OK:Lustre is ok"
cstor01n004  "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)"
cstor01n004  "RAM usage":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42  "OK -
   12.9% (4203984 kB) used."
cstor01n004  "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)"
cstor01n004  "Total Processes":  OK  for 28d 15h 34m 46s  checked 2014-02-06 10:03:42
   "PROCS OK: 1239 processes with STATE = RSZDT"

[root@cstor01n000 ~]# cscli monitor elements  -g oss
cstor01n004  "Arrays and Disk Status":  OK  for 2d 14h 31m 42s  checked 2014-02-06
   10:08:43  "All arrays are operating normally"
cstor01n004  "Current Load":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43  "OK
   - load average: 0.01, 0.01, 0.01"
cstor01n004  "Current Users":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43
   "USERS OK - 0 users currently logged in"
cstor01n004  "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%):"
cstor01n004  "Lustre Health":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43
   "OK:Lustre is ok"
```

```
cstor01n004  "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)"
cstor01n004  "RAM usage":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43  "OK -
   12.9% (4204568 kB) used."
cstor01n004  "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)"
cstor01n004  "Total Processes":  OK  for 28d 15h 35m 46s  checked 2014-02-06 10:08:43
   "PROCS OK: 1239 processes with STATE = RSZDT"
cstor01n005  "Arrays and Disk Status":  OK  for 17d 14h 22m 41s  checked 2014-02-06
   10:07:41  "All arrays are operating normally"
cstor01n005  "Current Load":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41  "OK
   - load average: 0.01, 0.02, 0.02"
cstor01n005  "Current Users":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41
   "USERS OK - 0 users currently logged in"
cstor01n005  "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%):"
cstor01n005  "Lustre Health":  OK  for 28d 15h 38m  7s  checked 2014-02-06 10:07:42
   "OK:Lustre is ok"
cstor01n005  "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)"
cstor01n005  "RAM usage":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41  "OK -
   12.8% (4192544 kB) used."
cstor01n005  "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)"
cstor01n005  "Total Processes":  OK  for 28d 15h 38m  8s  checked 2014-02-06 10:07:41
   "PROCS OK: 1241 processes with STATE = RSZDT"
[root@cstor01n000 ~]#

[root@cstor01n000 ~]# cscli monitor elements -S arrays
cstor01n000  "Arrays and Disk Status":  OK  for 28d 15h 38m 14s  checked 2014-02-06
   10:05:52  "All arrays are operating normally"
cstor01n001  "Arrays and Disk Status":  OK  for 28d 15h 39m 56s  checked 2014-02-06
   10:08:07  "All arrays are operating normally"
cstor01n002  "Arrays and Disk Status":  OK  for 28d 15h 36m 38s  checked 2014-02-06
   10:07:56  "All arrays are operating normally"
cstor01n003  "Arrays and Disk Status":  OK  for 28d 15h 36m 36s  checked 2014-02-06
   10:06:24  "All arrays are operating normally"
cstor01n004  "Arrays and Disk Status":  OK  for 2d 14h 32m 35s  checked 2014-02-06
   10:08:43  "All arrays are operating normally"
cstor01n005  "Arrays and Disk Status":  OK  for 17d 14h 23m 34s  checked 2014-02-06
   10:07:41  "All arrays are operating normally"

[root@cstor01n000 ~]# cscli monitor elements -S arrays –v
```

Subset of output:

```
cstor01n000  "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/cstor01n003:md64 doesn't have associated WIB array

Degraded Array information:

All arrays are in clean state on node cstor01n000"

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
------------------------------------------------------------------------
cstor01n001  "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/cstor01n003:md67 doesn't have associated WIB array

Degraded Array information:

All arrays are in clean state on node cstor01n001"

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
------------------------------------------------------------------------
cstor01n002  "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/cstor01n003:md65 doesn't have associated WIB array
Target:     MGS


Degraded Array information:
All arrays are in clean state on node cstor01n002"
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@cstor01n000 ~]# cscli monitor elements -S disk
cstor01n000  "Arrays and Disk Status":  OK  for 28d 15h 43m 32s  checked 2014-02-06
   10:10:52  "All arrays are operating normally"
cstor01n001  "Arrays and Disk Status":  OK  for 28d 15h 45m 14s  checked 2014-02-06
   10:13:07  "All arrays are operating normally"
cstor01n002  "Arrays and Disk Status":  OK  for 28d 15h 41m 56s  checked 2014-02-06
   10:12:56  "All arrays are operating normally"
cstor01n003  "Arrays and Disk Status":  OK  for 28d 15h 41m 54s  checked 2014-02-06
   10:11:24  "All arrays are operating normally"
cstor01n004  "Arrays and Disk Status":  OK  for 2d 14h 37m 53s  checked 2014-02-06
   10:13:42  "All arrays are operating normally"
cstor01n005  "Arrays and Disk Status":  OK  for 17d 14h 28m 52s  checked 2014-02-06
   10:12:41  "All arrays are operating normally"

root@cstor01n000 ~]# cscli monitor elements -S fan
cstor01n003-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"
cstor01n003-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"
cstor01n005-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"
cstor01n005-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@cstor01n000 ~]# cscli monitor elements -S power
cstor01n003-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"
cstor01n003-Enclosure-R1C1-21U  "Power Statistics":  OK  for 15d 17h 35m 18s  checked
   2014-02-06 10:16:25  "Summary: Total System Power 178W"
cstor01n005-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"
cstor01n005-Enclosure-R1C1-5U  "Power Statistics":  OK  for 28d 15h 44m 54s  checked
   2014-02-06 10:12:42  "Summary: Total System Power 1061W"

[root@cstor01n000 ~]# cscli monitor elements -S sbb
cstor01n003-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"
cstor01n005-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@cstor01n000 ~]# cscli monitor elements -S volt
cstor01n003-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"
cstor01n005-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@cstor01n000 ~]# cscli monitor elements -S disk
cstor01n000  "Arrays and Disk Status":  OK  for 28d 15h 45m  0s  checked 2014-02-06
   10:15:53  "All arrays are operating normally"
cstor01n001  "Arrays and Disk Status":  OK  for 28d 15h 46m 42s  checked 2014-02-06
   10:13:07  "All arrays are operating normally"
cstor01n002  "Arrays and Disk Status":  OK  for 28d 15h 43m 24s  checked 2014-02-06
   10:12:56  "All arrays are operating normally"
cstor01n003  "Arrays and Disk Status":  OK  for 28d 15h 43m 22s  checked 2014-02-06
   10:16:24  "All arrays are operating normally"
cstor01n004  "Arrays and Disk Status":  OK  for 2d 14h 39m 21s  checked 2014-02-06
   10:13:42  "All arrays are operating normally"
cstor01n005  "Arrays and Disk Status":  OK  for 17d 14h 30m 20s  checked 2014-02-06
   10:12:41  "All arrays are operating normally"
[root@cstor01n000 ~]# cscli monitor elements -S arrays
cstor01n000  "Arrays and Disk Status":  OK  for 28d 15h 45m 10s  checked 2014-02-06
   10:15:53  "All arrays are operating normally"
cstor01n001  "Arrays and Disk Status":  OK  for 28d 15h 46m 52s  checked 2014-02-06
   10:13:07  "All arrays are operating normally"
cstor01n002  "Arrays and Disk Status":  OK  for 28d 15h 43m 34s  checked 2014-02-06
   10:12:56  "All arrays are operating normally"
cstor01n003  "Arrays and Disk Status":  OK  for 28d 15h 43m 32s  checked 2014-02-06
   10:16:24  "All arrays are operating normally"
cstor01n004  "Arrays and Disk Status":  OK  for 2d 14h 39m 31s  checked 2014-02-06
   10:13:42  "All arrays are operating normally"
cstor01n005  "Arrays and Disk Status":  OK  for 17d 14h 30m 30s  checked 2014-02-06
   10:12:41  "All arrays are operating normally"
[root@cstor01n000 ~]#
```

7.10.5 Manage the netfilter level

The `netfilter_level` command manages the netfilter level on the Sonexion system.

Synopsis

```
$ cscli netfilter_level [-h] [-s] [-l level] [--force]
```

where:

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`.

7.10.6 Enable RAID checks

The `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]
```

where:

Optional Arguments	Description
`-h` |`--help`	Shows 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.

7.10.7 Manage the RAID rebuild rate

The rebuild_rate command manages the RAID rebuild rate on the Sonexion system.

Synopsis

```
$ cscli rebuild_rate [-h] [-n nodes] [--reset] [-1 single_rate] [-m multiple_rate]
```

where:

Optional Arguments	Description
-h |--help	Shows the help message and exits.
-n *nodes* |--node *nodes*	Specifies pdsh-style node hostnames. For example, node[100-110,120. Global RAID rebuild rates are installed without this argument.
--reset	Resets the RAID rebuild rate.
-1 *single_rate* |--after-first-failure *single_rate*	Specifies the RAID rebuild rate for a single drive failure.
-m *multiple_rate* |--after-multiple-failures *multiple_rate*	Specifies the RAID rebuild rate for multiple drive failures.

7.10.8 Manage the administrative password

The set_admin_passwd command sets the Sonexion system administrator's user password.

Synopsis

```
$ cscli set_admin_passwd [-h] -p password
```

where:

Option	Description
-h |--help	Prints the help message and exits.
-p |--password	Sets the system administrator's password.

7.10.9 Manage the system date

The set_date command manages the date on the Sonexion system.

Synopsis

```
$ cscli set_date [-h] [-s new_date] [--force-ntp]
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-s` *new_date* `|--set` *new_date*	Specifies the new date in this format: *mmddhhmmccyy.ss*.
`--force-ntp`	Forces NTP configuration.

7.10.10 Manage the system timezone

The `set_timezone` command manages the timezone on the Sonexion system.

Synopsis

```
$ cscli set_timezone [-h] [-s new_timezone] [-l]
```

where:

Optional Arguments	Description
`-h |--help`	Shows 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.

7.10.11 Manage the InfiniBand Subnet Manager

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}]
```

where:

Optional Arguments	Description
`-h |--help`	Shows the help message and exits.
`-e |--enable`	Enables the IB storage manager used with the Sonexion system.
`-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.

7.10.12 Manage support bundles

The `support_bundle` command manages support bundles and support bundle settings. See examples in section 5.4, page 36.

Synopsis

```
$ cscli support_bundle [-h] [-c] [-n nodes] [-t minutes] [-e bundle_id]
   [--disable-trigger trigger] [--get-purge-limit] [--set-purge-limit percents]
```

where:

Option	Description
`-h |--help`	Displays the help message and exits.
`-c |--collect-bundle`	Collects the support bundle.
`-n` *nodes* `|--nodes` *nodes*	Shows a comma-separated list of nodes. Default value is all nodes.
`-t` *minutes* `|--time-window` *minutes*	Specifies the time window to collect data for the support bundle (in minutes). Default value is 45 minutes.
`-e` *bundle_id* `|--export-bundle` *bundle_id*	Identifies an export-specified support bundle.
`|--show-triggers`	Shows the triggers that initiate automatic collection of support bundles.

`|--enable-trigger` *trigger*	Enables a specific trigger.
`|--disable-trigger` *trigger*	Disables a specific trigger.
`|--get-purge-limit`	Shows the purge limit as a percentage of free file system space. If the purge limit is reached, the Sonexion system purges old support bundle files.
`|--set-purge-limit` *percents*	*Sets* the purge limit as a percentage of free file system space.

8. GEM CLI Commands

This appendix details the supported command line interface (CLI) provided by the GEM software.

8.1 Serial port settings

Use the following settings for using HyperTerminal or other serial communications GUI to work with the CLI:

Baud rate (bits/sec):	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):	9600
Data bits:	8
Parity:	None
Stop bits:	1

Flow control:	None

NOTE: 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

8.2 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

8.3 Supported commands

The following CLI commands are supported by the GEM software.

8.3.1 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, i.e. a simple single-shot diagnostic dump.
Command arguments:	None
Command type:	Diagnostic
Access level:	General

8.3.2 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

8.3.3 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

8.3.4 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

8.3.5 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: 1 optional argument - see description above.

Command type: Control

Access level: General

8.3.6 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: ipmi_power [type]

Type:

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

8.3.7 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

8.3.8 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

8.3.9 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

8.3.10 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 the "settime" command on its own, without any arguments, prints the current logging time to the CLI.
Command arguments:	days hh mm ss
Command type:	Control
Access level:	General

8.3.11 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