Installing and Configuring Cray Linux Environment™ (CLE) Software

S–2444–4003

RECORD OF REVISION

S–2444–4003 Published March 2012 Supports the 4.0.UP03 release of the Cray Linux Environment (CLE) operating system running on Cray XE and Cray XK systems.

S–2444–4002 Published December 2011 Supports the 4.0.UP02 release of the Cray Linux Environment (CLE) operating system running on Cray XE and Cray XK systems.

S–2444–4001 Published September 2011 Supports the 4.0.UP01 release of the Cray Linux Environment (CLE) operating system running on Cray XE systems.

S–2444–40 Published June 2011 Supports the 4.0 release of the Cray Linux Environment (CLE) operating system running on Cray XE systems.

S–2444–3103 Published March 2011 Supports the 3.1.UP03 release of the Cray Linux Environment (CLE) operating system running on Cray XT and Cray XE systems.

S–2444–3102 Published December 2010 Supports the 3.1.UP02 release of the Cray Linux Environment (CLE) operating system running on Cray XT and Cray XE systems.

3.1 Published June 2010 Supports the 3.1 release of the Cray Linux Environment (CLE) operating system running on Cray XT and Cray XE systems.

3.0 Published March 2010 Supports the 3.0 release of the Cray Linux Environment (CLE) operating system running on Cray XT6 systems.

2.2 Published July 2009 Supports general availability (GA) release of the Cray Linux Environment (CLE) 2.2 operating system running on Cray XT systems.

2.1 Published November 2008 Supports general availability (GA) release of the Cray Linux Environment (CLE) 2.1 operating system running on Cray XT systems.

2.0 Published October 2007 Supports general availability (GA) release of Cray XT systems running the Cray XT Programming Environment 2.0, UNICOS/lc 2.0, and System Management Workstation (SMW) 3.0.1 releases.

1.5 Published November 2006 Supports general availability (GA) release of Cray XT series systems running the Cray XT series Programming Environment 1.5, UNICOS/lc 1.5, and System Management Workstation (SMW) 1.5 releases.

1.4 Published May 2006 Supports Cray XT3 systems running Cray XT3 Programming Environment 1.4, System Management Workstation (SMW) 1.4, and UNICOS/lc 1.4 releases.

1.3 Published November 2005 Supports Cray XT3 systems running Cray XT3 Programming Environment 1.3, System Management Workstation (SMW) 1.3, and UNICOS/lc 1.3 releases.

1.2 Published September 2005 Supports Cray XT3 systems running Cray XT3 Programming Environment 1.2, System Management Workstation (SMW) 1.2, and UNICOS/lc 1.2 releases.

1.1 Published June 2005 Supports Cray XT3 systems running Cray XT3 Programming Environment 1.1, System Management Workstation (SMW) 1.1, and UNICOS/lc 1.1 releases.

Contents

Page

Page

Page

Part II: Update and Upgrade Installations

Procedures

Examples

Tables

Introduction [1]

This guide contains procedures for installation and configuration of the Cray Linux Environment (CLE) 4.0 operating system release for Cray systems and is intended for system administrators who are familiar with operating systems derived from UNIX.

CLE is a Linux-based operating system that runs on Cray systems. The CLE 4.0 release includes Cray's customized version of the SUSE Linux Enterprise Server (SLES) 11 SP1 operating system. All software is installed by means of scripts and RPM Package Manager (RPM) files.

Throughout this document, any reference to *Cray systems* includes Cray XE6, Cray XK6, Cray XE6m, Cray XE5, and Cray XE5m systems unless otherwise noted.

CLE software installations fall into one of the following categories:

Initial — An initial software installation involves installing and configuring the entire system and is generally performed for new hardware. If an initial installation is performed on an existing system set, the previous configuration is lost.

Update — An update installation involves applying an update package for a release that is already running on your system. For example, installing CLE 4.0.UP03 on a system that is already running an earlier version of CLE 4.0 is considered an update installation.

Upgrade — An upgrade installation involves moving to the next release. For example, installing CLE 4.0 on a system that is running CLE 3.1 is considered an upgrade.

This guide describes procedures for the following types of installations.

- **Initial** or new software installations. Follow Part I, *New System Installations*.
- **Upgrade** installations. Follow Part II, *Update and Upgrade Installations* to upgrade an existing system running CLE 3.0 to run the CLE 4.0 release.
- **Update** installations. Follow Part II, *Update and Upgrade Installations* to apply a CLE 4.0 update package (for example, CLE 4.0.UP03) to a system that is already running the 4.0 release level of CLE.

The procedures in this document require that you have already installed the appropriate System Management Workstation (SMW) software release on the SMW. See Before You Start the CLE Software Installation on page 15.

An Adobe PDF version of this guide is available on the CrayDoc CD or on the CrayPort website at http://crayport.cray.com.

1.1 Other Related Publications

The following documents contain additional information that may be helpful:

- *Cray Linux Environment (CLE) Software Release Overview* (S–2425)
- *Cray Linux Environment (CLE) Software Release Overview Supplement* (S–2497)
- *Installing Cray System Management Workstation (SMW) Software* (S–2480)
- *Managing System Software for Cray XE and Cray XK Systems* (S–2393)
- *Managing Lustre for the Cray Linux Environment (CLE)* (S–0010)
- *Network Resiliency for Cray XE and Cray XK Systems* (S–0032)
- *Introduction to Cray Data Virtualization Service* (S–0005)
- *Repurposing Compute Nodes as Service Nodes on Cray XE and Cray XT Systems* (S–0029)
- *Using Cray Management Services (CMS)* (S–2484)
- *Using and Configuring System Environment Data Collections (SEDC)* (S–2491)
- *Cray Application Developer's Environment Installation Guide* (S–2465)

1.2 Distribution Media

The CLE 4.0 release distribution media includes two DVDs required to install the CLE 4.0 release on a Cray XE system. The first is labeled `Cray CLE 4.0.UP`*nn* `Software` and contains software specific to Cray systems. The second is labeled `Cray-CLEbase11-`*yyyymmdd* and contains the CLE 4.0 base operating system which is based on SLES 11 SP1. All software is installed by means of scripts and RPM Package Manager (RPM) files.

Part I: New System Installations

Preparing to Install a New System [2]

Follow these procedures to perform an initial software installation of the Cray Linux Environment (CLE) 4.0 software release for a new Cray XE or Cray XK system.

> **Note:** In the following chapters, some examples are left-justified to better fit the page. Left justification has no special significance.

2.1 Before You Start the CLE Software Installation

Perform the following tasks before you install the CLE 4.0 software release.

- **Review release package documentation.** Read the *CLE 4.0 Release Errata*, *Limitations for CLE 4.0* and *README* documents provided with the release for any installation-related requirements and corrections to this installation guide.

  Additional installation information may also be included in *Cray Linux Environment (CLE) Software Release Overview* and *Cray Linux Environment (CLE) Software Release Overview Supplement*.

- **Confirm the SMW software release level.** You must install the SMW 6.0.UP03 release or later on your SMW before installing the CLE 4.0.UP03 release. If a specific SMW update package is required for your installation, that information is documented in the *README* file provided with the CLE 4.0.UP03 release. The procedures in this guide assume that the SMW software has been successfully installed and the SMW is operational; type the following command to determine the HSS/SMW version:

  ```
  crayadm@smw:~> cat /opt/cray/hss/default/etc/smw-release
  6.0.UP03
  ```

2.2 Passwords

The following default account names and passwords are used throughout the CLE software installation process. Cray recommends that you change all default passwords; see Changing the Default System Passwords on page 71.

Table 1. Default System Passwords

Account Name	Password
root	initial0
crayadm	crayadm

For procedures on handling SMW and RAID accounts and passwords, see *Installing Cray System Management Workstation (SMW) Software*.

Access to MySQL databases requires a user name and password. The MySQL accounts and privileges are shown in Table 2.

Table 2. MySQL Database Accounts and Privileges

Account	Default Password	Privilege
root	None; you must create a password.	All available privileges.
basic	basic	Read access to most tables; most applications use this account.
sys_mgmt	sys_mgmt	Most privileged non-root account; all privileges required to manipulate CLE tables.
mazama	mazama	Create, delete, update, and insert permissions on mazama and mz* databases. Used by CMS/Mazama. For more information, see *Using Cray Management Services (CMS)* (S–2484).

For steps to change MySQL account passwords, see Procedure 23 on page 84.

Configuring the Boot RAID [3]

This chapter describes how to configure, format, zone, and partition the boot RAID (redundant array of independent disks) system.

> **Note:** Cray ships systems with much of this configuration completed. You may not have to perform all of the steps described in this chapter unless you are making changes to the configuration.

Cray provides support for system boot RAID from two different vendors, Data Direct Networks (DDN) and NetApp Corporation. You may also have a QLogic SANbox Fibre Channel switch from QLogic Corporation.

Installing Cray System Management Workstation (SMW) Software (S–2480) contains device specific instructions for configuring boot RAID LUNs (Logical Units) and volume groups.

> **Note:** The DDN RAID uses LUNs; the NetApp, Inc. Engenio RAID uses volumes.

If you use NetApp, Inc. Engenio devices for your boot RAID, you must have installed SANtricity Storage Manager software from NetApp, Inc. Corporation. For more information about third party software applications required to configure your boot RAID, see *Installing Cray System Management Workstation (SMW) Software*.

After Configuring the Boot RAID LUNs or Volume Groups on page 18, follow the procedures for Partitioning the LUNs on page 20.

3.1 Prerequisites and Assumptions for Configuring the Boot RAID

In typical system installations, the RAID provides the storage for both the boot node root file systems and the shared root file system. Although these file systems are managed from the boot node during normal operation, you must use the SMW to perform an initial installation of the Cray Linux Environment (CLE) base operating system, based on SUSE Linux Enterprise Server (SLES) 11 SP1, and Cray CLE software packages onto the boot RAID disks.

In typical system installations, RAID units provide user and scratch space and can be configured to support a variety of file systems. Different RAID controller models support Fibre Channel (FC), Serial ATA (SATA), and Serial Attached SCSI (SAS) disk options.

The following assumptions are relevant throughout this chapter:

- The SMW has an Ethernet connection to the Hardware Supervisory System (HSS) network.
- The boot node(s) have Ethernet connections to the SMW.
- The SMW has a switched FC or SAS connection to the boot RAID.
- The boot node(s) have a switched FC or SAS connection to the boot RAID.
- The service database (SDB) node(s) have a switched FC or SAS connection to the boot RAID.
- If a dedicated `syslog` node is configured, it has a switched FC or SAS connection to the boot RAID.

3.2 Configuring the Boot RAID LUNs or Volume Groups

Follow the procedures in *Installing Cray System Management Workstation (SMW) Software* to configure your boot RAID. You must configure the boot RAID with at least six LUNs to support the various system management file systems. The recommended configuration listed in Table 3 describes nine LUNs. You can specify units as GB or MB.

If you have DDN devices, follow the boot RAID configuration procedures in *Installing Cray System Management Workstation (SMW) Software* to `telnet` to the RAID controller and use the `lun add` and `lun delete` commands to configure LUNs following the recommendations in Table 3.

If you have NetApp, Inc. Engenio devices, follow the boot RAID configuration procedures in *Installing Cray System Management Workstation (SMW) Software* to use the SANtricity Storage Manager software to create the boot RAID volume group and configure the volumes following the recommendations in Table 3.

Table 3. Recommended Boot RAID LUN Values

LUN	Label	Group?	Capacity in MB	No. of Tiers	Tier ID	Block Size	Format?
0	bootroot0	n	40000	1	1	4096	y
1	bootroot1	n	40000	1	2	4096	y
2	shroot0	n	280000	1	3	4096	y
3	sdb0	n	80000	1	2	4096	y
4	syslog	n	80000	1	1	4096	y
5	shroot1	n	280000	1	4	4096	y

LUN	Label	Group?	Capacity in MB	No. of Tiers	Tier ID	Block Size	Format?
6	bootroot2	n	40000	1	3	4096	y
7	shroot2	n	280000	1	1	4096	y
8	sdb1	n	80000	1	4	4096	y

3.3 Zoning the LUNs

After you configure and format the LUNs, you must grant host access to the LUNs by using a process called *zoning*. Zoning maps a host port on the RAID controller to the LUNs that the host accesses. If you have a QLogic switch, zoning maps the host ports on the switch. Although it is possible to enable all hosts to have access to all LUNs, Cray recommends that each host be granted access only to the LUNs it requires.

Note: If a LUN is to be shared between failover host pairs, each host must be given access to the LUN. The SMW host port should be given access to all LUNs.

3.3.1 Zoning the LUNs for DDN Devices

If you have DDN devices, follow the procedure to zone LUNs for DDN in *Installing Cray System Management Workstation (SMW) Software*. Use the `zoning` command to edit each port number and map the LUNs; follow the recommendations in Table 4.

Table 4. Recommended DDN Zoning

Port	External LUN, Internal LUN		
1	000,000	001,001	002,002
2	003,003	004,004	005,005
3	006,006	007,007	008,008
4			

If you have created or modified your LUN configuration, you must reboot the SMW to enable it to recognize the new LUN configuration and zoning information. Verify that all of your changes have been recognized by invoking the `lsscsi` command. Procedure 1 on page 22 provides example output for the `lsscsi` command. For more information, see the `lsscsi`(8) man page on the SMW.

Warning: Failure to reboot the SMW at this point might produce unexpected results. If your SMW does not properly recognize the boot RAID configuration, the system installation procedures could overwrite existing data.

After you finish creating, formatting, and zoning the Volume Groups and LUNs on the boot RAID, you must partition them. On the SMW, follow Partitioning the LUNs on page 20.

3.3.2 Zoning the QLogic FC Switch

If you have a QLogic Fibre Channel Switch, follow the procedures described in *Installing Cray System Management Workstation (SMW) Software* to zone the LUNs on your QLogic SANBox switch. Use the QuickTools utility to create a Zone Set and define the ports in the zone; follow the recommendations in Table 5. These recommendations presuppose that the disk device has four host ports connected to ports 0-3 for the QLogic SANbox switch.

> **Note:** QuickTools is an application, embedded in your QLogic switch, which is accessible from the SMW by using a web browser.

Zoning for a QLogic switch is implemented by creating a *zoneset*, adding one or more zones to the zone set, and selecting the ports to use in the zone.

Follow this procedure after the SANBox is configured and on the HSS network.

Table 5. Recommended QLogic Zoning

Zone	Port	SANBox Connection
Boot	0	Boot RAID
Boot	4	Boot Node
Boot	5	SDB Node
Boot	10	SMW
Boot	6	Syslog node (if dedicated)

If you have created or modified your LUN configuration, you must reboot the SMW to enable it to recognize the new LUN configuration and zoning information. Verify that all of your changes have been recognized by invoking the `lsscsi` command. Procedure 1 on page 22 provides example output for the `lsscsi` command. For more information, see the `lsscsi`(8) man page on the SMW.

> **Warning:** Failure to reboot the SMW at this point might produce unexpected results. If your SMW does not properly recognize the boot RAID configuration, the system installation procedures could overwrite existing data.

3.4 Partitioning the LUNs

After you finish creating, formatting, and zoning the LUNs on the boot RAID, you must partition them by invoking the `fdisk` command on the SMW.

Table 6 contains an example of a partition layout. The SMW Device names in column 4 are consistent with SMW hardware that is Restriction of Hazardous Substances (RoHS) compliant, with three external disk drives.

Table 6. Example of Boot LUN Partitions (for an SMW with three disks)

LUN	Part Num	Part Type	SMW Device	Size	Type	Description
0	1	Primary	sdc1	30GB	Linux	Boot node 0 root file system
0	2	Primary	sdc2	10GB	Swap	Boot node 0 swap
1	1	Primary	sdd1	30GB	Linux	Boot node 1 root file system
1	2	Primary	sdd2	10GB	Swap	Boot node 1 swap
2	1	Extended	sde1	ALL		Primary Partition
2	5	Logical	sde5	30GB	Linux	Reserved for future use
2	6	Logical	sde6	180GB	Linux	NFS Shared Root
2	7	Logical	sde7	10GB	Linux	RAW Partition for Boot Image #0
2	8	Logical	sde8	10GB	Linux	RAW Partition for Boot Image #1
2	9	Logical	sde9	50GB	Linux	Reserved for future use (optional persistent `/var`)
3	1	Primary	sdf1	40GB	Linux	Service Database (SDB)
3	2	Primary	sdf2	40GB	Linux	UFS
4	1	Primary	sdg1	40GB	Linux	Syslog
4	2	Primary	sdg2	40GB	Linux	Reserved for future use
5	1	Extended	sdh1	ALL	Linux	Alternative primary Partition
5	5	Logical	sdh5	30GB	Linux	Reserved for future use
5	6	Logical	sdh6	180GB	Linux	Backup NFS Shared Root
5	7	Logical	sdh7	10GB	Linux	Alternative RAW partition for boot image
5	8	Logical	sdh8	10GB	Linux	Alternative RAW Partition for boot image
5	9	Logical	sdh9	50GB	Linux	Reserved for future use
6	1	Primary	sdi1	30GB	Linux	Extra boot node 2 root file system
6	2	Primary	sdi2	10GB	Swap	Extra boot node 2 swap
7	1	Primary	sdj1	180GB	Linux	Reserved for future use
8	1	Primary	sdk1	40GB	Linux	Backup SDB
8	2	Primary	sdk2	40GB	Linux	Backup UFS

Procedure 1. Partitioning the LUNs

1. Log on to the SMW as `root`.

   ```
   crayadm@smw:~> su - root
   ```

2. Use the `lsscsi` command to verify that the LUNs were recognized. Your first SMW device is the first non-ATA device listed. On an SMW with two internal SATA drives, the output should resemble the following example. Note that two of the disks are ATA, not DDN or NetApp, Inc. (formerly LSI) Engenio disks.

   ```
   smw:~ # lsscsi
   [0:0:0:0]    cd/dvd  LITE-ON  DVDRW SHW-160P6S PS0A  /dev/sr0
   [2:0:0:0]    disk    ATA      ST3320620AS      3.AA  /dev/sda
   [4:0:0:0]    disk    ATA      ST3320620AS      3.AA  /dev/sdb
   [6:0:0:0]    disk    LSI      INF-01-00        0736  /dev/sdc
   [6:0:0:1]    disk    LSI      INF-01-00        0736  /dev/sdd
   [6:0:0:2]    disk    LSI      INF-01-00        0736  /dev/sde
   [6:0:0:3]    disk    LSI      INF-01-00        0736  /dev/sdf
   [6:0:0:4]    disk    LSI      INF-01-00        0736  /dev/sdg
   [6:0:0:5]    disk    LSI      INF-01-00        0736  /dev/sdh
   [6:0:0:6]    disk    LSI      INF-01-00        0736  /dev/sdi
   [6:0:0:7]    disk    LSI      INF-01-00        0736  /dev/sdj
   [6:0:0:8]    disk    LSI      INF-01-00        0736  /dev/sdk
   ...
   ```

   > **Note:** The `cd/dvd` entry may be different on your system.

3. Create the partitions shown in Table 6 by using the `fdisk` command. If you are not familiar with `fdisk`, see Appendix E, Configuring Primary and Extended File Partitions on page 167 and the `fdisk`(8) man page.

   ```
   smw:~# fdisk /dev/sdc
   ```

   Repeat this command for `/dev/sdd` through `/dev/sdk`; use the values in Table 6 for each `fdisk` session.

   > **Note:** Changes you make to the partition table are not effective until you type **`w`** to write and exit.

4. Invoke the `fdisk` command with the `-l` option to verify that the LUNs (volumes) are configured according to Table 6. Your LUN sizes may be slightly different; for example, 43G instead of 40G, as listed in the table.

   Note: The following output represents the example; your output is specific to your actual LUN configuration.

```
smw:~ # fdisk -l
Disk /dev/sda: 320.0 GB, 320072933376 bytes
255 heads, 63 sectors/track, 38913 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1         523     4200966   82  Linux swap / Solaris
/dev/sda2   *         524       38913   308367675   83  Linux

Disk /dev/sdb: 320.0 GB, 320072933376 bytes
255 heads, 63 sectors/track, 38913 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         523     4200966   82  Linux swap / Solaris
/dev/sdb2   *         524       38913   308367675   83  Linux

Disk /dev/sdc: 41.9 GB, 41943040000 bytes
64 heads, 32 sectors/track, 40000 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdc doesn't contain a valid partition table

Disk /dev/sdd: 41.9 GB, 41943040000 bytes
64 heads, 32 sectors/track, 40000 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdd doesn't contain a valid partition table

Disk /dev/sde: 293.6 GB, 293601280000 bytes
255 heads, 63 sectors/track, 35694 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sde doesn't contain a valid partition table

Disk /dev/sdf: 83.8 GB, 83886080000 bytes
255 heads, 63 sectors/track, 10198 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdf doesn't contain a valid partition table

Disk /dev/sdg: 83.8 GB, 83886080000 bytes
255 heads, 63 sectors/track, 10198 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdg doesn't contain a valid partition table

Disk /dev/sdh: 293.6 GB, 293601280000 bytes
255 heads, 63 sectors/track, 35694 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Disk /dev/sdh doesn't contain a valid partition table

Disk /dev/sdi: 41.9 GB, 41943040000 bytes
64 heads, 32 sectors/track, 40000 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes
Disk /dev/sdi doesn't contain a valid partition table

Disk /dev/sdj: 293.6 GB, 293601280000 bytes
255 heads, 63 sectors/track, 35694 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdj doesn't contain a valid partition table

Disk /dev/sdk: 83.8 GB, 8388ZZ6080000 bytes
255 heads, 63 sectors/track, 10198 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdk doesn't contain a valid partition table
```

About Installation Configuration Files [4]

This chapter contains essential information about parameters that you must set before you install the Cray Linux Environment (CLE) software on a Cray system. Review this information before installing CLE and again for every CLE software update or upgrade installation.

The CLE software installation process uses an installation script called `CLEinstall`. The `CLEinstall` program, in turn, references two configuration files to determine site-specific configuration parameters used during installation. These configuration files are `CLEinstall.conf` and `/etc/sysset.conf`. Prior to invoking the `CLEinstall` installation program, you must carefully examine these two configuration files and make site-specific changes.

⚠ **Caution:** Improper configuration of the `CLEinstall.conf` and `/etc/sysset.conf` files can result in a failed installation.

`CLEinstall.conf`: Based on the settings you define in `CLEinstall.conf`, the `CLEinstall` program updates other configuration files, thus eliminating many manual configuration steps. The `CLEinstall.conf` file is created during the installation process by copying the `CLEinstall.conf` template from the distribution media. This chapter groups the `CLEinstall.conf` settings into three categories: parameters that must be defined for your specific configuration, parameters with default or standard settings that do not need to be changed in most cases, and additional parameters that are required to configure optional functionality or subsystems.

`sysset.conf`: You can install `bootroot` and `sharedroot` to an alternative location while your Cray system is running. This enables you to do the configuration steps in the alternative root location and then move over to the alternative location after it is configured, thus reducing the need for dedicated system time for installation and configuration. Use the `/etc/sysset.conf` file to identify sets of disk partitions on the boot RAID as alternative *system sets*. Each system set provides a complete collection of all file systems and boot images, thus making it possible to switch easily between two or more versions of the system software. For example, by using system sets, it is possible to keep a stable "production" system available for your users while simultaneously having a "test" system available for new software installation, configuration, and testing.

Note: If you have existing `CLEinstall.conf` and `/etc/sysset.conf` files, save copies before you make any changes.

4.1 About CLEinstall.conf Parameters that Must Be Defined

A template CLEinstall.conf is delivered on the Cray CLE 4.0.UP*nn* Software DVD. Use this sample file to prepare your installation configuration settings before you begin the installation. Carefully examine each installation parameter and the associated comments in the file to determine the changes that are required for your planned configuration.

In Procedure 4 on page 55, you are directed to edit your CLEinstall.conf file. Make site-specific changes at that point in the installation process.

These parameters **must** be changed or verified for your configuration. For more information, see the comments in the CLEinstall.conf file or the CLEinstall.conf(5) man page.

Mount points on the SMW

Set bootroot_dir and sharedroot_dir to choose the boot root and shared root file system mount points on the SMW.

Hostname settings

Set xthostname and node_class_login_hostname to the hostname for your Cray system.

Node settings

Set node_* parameters to identify which nodes are the sdb, ufs, syslog, login and boot node(s).

Node class settings

Set node_class* parameters to assign nodes to a node class for /etc/opt/cray/sdb/node_classes.

Note: You must keep the node_class* parameters current with the system configuration. Refer to Maintaining Node Class Settings and Hostname Aliases on page 27 for more information.

SSH on boot node settings

Set ssh_* parameters to configure boot node root secure shell (ssh) keys.

ALPS settings

Set `alps_*` parameters for various Application Level Placement Scheduler (ALPS) configuration options.

GPU Settings

Set `GPU=`**`yes`** **only** if your machine has Cray XK6 blades with GPUs installed. Setting this parameter to **`yes`** will install the required RPMs and code for GPU systems. If your machine has Cray XK6 blades without GPUs, or it does not have Cray XK6 blades installed, set this parameter to `no`.

Note: For Cray XK6 systems with GPUs, you must configure and enable the alternative compute node root run time environment for dynamic shared objects and libraries (DSL). Cray XK6 blades require access to shared libraries to support GPUs.

4.1.1 Maintaining Node Class Settings and Hostname Aliases

For an initial CLE software installation, the `CLEinstall` program creates the `/etc/opt/cray/sdb/node_classes` file and adds Cray system hostname and alias entries to the `/etc/hosts` file. Additionally, each time you update or upgrade your CLE software, `CLEinstall` verifies the content of `/etc/opt/cray/sdb/node_classes` and modifies `/etc/hosts` to match the configuration specified in your `CLEinstall.conf` file.

Unless you confirm that your hardware changed, the `CLEinstall` program fails if `/etc/opt/cray/sdb/node_classes` does not agree with `node_class[`*idx*`]` parameters in `CLEinstall.conf`. Therefore, you must keep the following parameters current with your Cray system configuration:

`node_class_login=`*login*

Specifies the node class label for the login nodes.

`node_class_default=`*service*

Specifies the default node class label for service nodes. A service node can only be in one class; typical classes might be `service`, `login`, `network`, `sdb`, `ost`, `mds`, or `lustre`. Classes can have any name provided the names are used consistently by using the `xtspec` command. Node IDs that are not designated as part of a class default to `node_class_default`.

node_class[*idx*]=*class NID* [*NID*] ...

Specifies the name of the class for index *idx* and the integer node IDs (NIDs) that belong to the class. CLEinstall uses node_class[*idx*] parameters along with other parameters in CLEinstall.conf to create, update or verify /etc/opt/cray/sdb/node_classes and /etc/hosts files.

You must configure a login class with at least one NID. A NID can be a member of only one node_class.

Example 1. Setting the node_class[*idx*] parameters

```
node_class[0]=login 8 30
node_class[1]=network 9 13 27 143
node_class[2]=sdb 5
node_class[3]=lustre 12 18 26
```

For each class defined, host name aliases in /etc/hosts are assigned based on the class name and **order** of NIDs specified for this parameter.

Example 2. Host alias assignments based on the node_class[*idx*] parameters

If you define the following node_class class entry:

```
node_class[1]=network 9 13 27 143 19
```

Host name aliases for the network class are assigned as follows:

```
nid00009 - network1
nid00013 - network2
nid00027 - network3
nid00143 - network4
nid00019 - network5
```

CLEinstall uses the information you specify for these parameters to update the /etc/hosts file as follows:

- A copy of the original file is saved as /etc/hosts.*$$*.preinstall.
- The Cray system entries (IP address, node ID, and physical name) are moved to the end of the file.
- Any Cray hostname aliases specified in CLEinstall.conf are added for the appropriate nodes.
- A copy of the modified file is saved as /etc/hosts.*$$*.postinstall.

4.2 About CLEinstall.conf Parameters with Standard Settings

The standard or default values for settings in the following categories are appropriate in most cases. Verify that these default values are acceptable for your site. For more information, see the comments in the `CLEinstall.conf` file or the `CLEinstall.conf`(5) man page.

- Shared root setting
- Boot node network settings
- Persistent `var` settings
- UFS (home directory) for login nodes
- `syslog` settings
- Partition setting
- SDB database settings
- Writeable `/tmp` for CNL setting
- Node Health Check (NHC) on boot

Additional parameters that you should review are described in greater detail in the following sections.

4.2.1 Changing the Default High-speed Network (HSN) Settings

By default, the HSN IP address is `10.128.0.0` for Cray XE systems with Gemini based system interconnection network. You can modify these parameters to configure another valid address; for example `10.33.0.0`. In most cases, the default value is acceptable.

`HSN_byte1=`
`HSN_byte2=` Specifies the HSN IP address. The default on Cray XE systems is `HSN_byte1=10`, `HSN_byte2=128`.

If you change `HSN_byte1` and `HSN_byte2` from the default, `CLEinstall` implements this change by modifying the following files:

`/etc/sysconfig/xt` on the boot root and shared root

`/etc/hosts` on the boot root and shared root

`/etc/sysconfig/alps` on the boot root and shared root

`/etc/opt/cray/rca/fomd.conf` on the boot root and shared root

`/etc/opt/cray/hosts/service_alias.conf` on the boot root and shared root

`/opt/xt-images/templates/default/etc/hosts` for CNL and SNL images

`/opt/xt-images/templates/default/etc/krsip.conf` for CNL images with RSIP

In addition, the CNL parameters file and the SNL parameters file in the bootimage are updated to include `bootnodeip`, `sdbnodeip`, `ippob1` and `ippob2`.

Caution: Due to site-specific local modifications, you may need to update additional files when you change your HSN IP address; for example, `/etc/hosts.allow`, `/etc/hosts.deny`, `/etc/exports` or `/etc/security/access.conf`.

Note: Cray recommends that you select values for `HSN_byte1` and `HSN_byte2` that do not overlap subnets listed as default IP addresses in *Installing Cray System Management Workstation (SMW) Software*.

`bootimage_bootifnetmask`

This netmask must be consistent with the modified `HSN_byte1` and `HSN_byte2` parameters.

```
persistent_var_IPaddr
home_directory_server_IPaddr
bootnode_failover_IPaddr
bootimage_bootnodeip
alps_directory_server_IPaddr
```

The `HSN_byte1` and `HSN_byte2` parameters and the netmask must be consistent with the first two bytes of these IP addresses that are defined in `CLEinstall.conf`.

Change `home_directory_server_IPaddr` only if `home_directory_ufs=no`; change `alps_directory_server_IPaddr` only if `alps_directory_server_hostname` is not the `ufs` node hostname.

4.2.2 Changing Parameters to Tune Virtual Memory or NFS

You may choose to modify these parameters based on your system configuration.

`sysctl_conf_vm_min_free_kbytes`

Specifies the `vm_min_free_kbytes` parameter of the Linux kernel. Linux virtual memory must keep a minimum number of kilobytes free. The virtual memory uses this number to compute a `pages_min` value for each `lowmem` zone in the system. Based on this value, each `lowmem` zone is allocated a number of reserved free pages, in proportion to its size.

The default value of `vm_min_free_kbytes` in the `/etc/sysctl.conf` file is 102,400 KB of free memory. For some configurations, the default value may be too low, and memory exhaustion may occur even though free memory is available. If this happens, adjust the `vm_min_free_kbytes` parameter to increase the value to 5% or 6% of total memory.

`nfs_mountd_num_threads`

Controls an NFS `mountd` tuning parameter that is added to `/etc/sysconfig/nfs` and used by `/etc/init.d/nfsserver` to configure the number of `mountd` threads on the boot node. By default, NFS `mountd` behavior is unchanged (a single thread). For systems with more than 50 service I/O nodes, Cray recommends that you configure multiple threads by setting this parameter to `4`. If you have a larger Cray system (greater than 50 service I/O nodes), contact your Cray service representative for assistance changing the default setting.

`use_kernel_nfsd_number`

Specifies the number of NFSD threads. By default, this variable in `/etc/sysconfig/nfs` is set to `16`.

A large site may wish to change both `nfs_mountd_num_threads` and `use_kernel_nfsd_number`. Contact your Cray service representative for assistance changing the default setting.

4.2.3 Changing the Default `bootimage` Settings

You can change several parameters related to the boot image configuration. In most cases, the default values are acceptable. For information about additional bootimage parameters, see the `CLEinstall.conf`(5) man page.

`bootimage_temp_directory=`*/home/crayadm/boot*

Specifies the parent directory on the SMW for temporary directories used to extract a boot image and adjust the boot image parameters file.

`bootimage_bootnodeip=`*10.131.255.254*

Specifies the virtual IP address for the boot node. The default is `10.131.255.254` for Cray XE systems with Gemini network interconnect. In most cases, the default value is acceptable. If you change the default, you must also modify default value for `bootnode_failover_IPaddr` and `persistent_var_IPaddr` to match the address specified by `bootimage_bootnodeip`.

`bootimage_bootifnetmask=`*255.252.0.0*

Specifies the network mask for the boot node virtual IP address. The default is `255.252.0.0` for Cray XE systems with Gemini network interconnect. In most cases, the default value is acceptable.

`bootimage_xtrel`

Set this parameter to **yes** to add `xtrel=$XTrelease` to the boot image SNL parameters file. This option is used for release switching; for more information see the `xtrelswitch`(8) command. The default value is `no`.

4.2.4 Node Health on Boot

Node Health Checker (NHC) automatically checks the health of compute nodes on boot using the Node Health Checker. The `NHC_on_boot` variable controls this feature and is set to `NHC_on_boot=yes` by default.

The `NHC_on_boot` variable affects the NHC configuration file in the compute node image and is used only when NHC is run on boot. Every NHC invocation after the compute node boot, either by ALPS or manually, uses the configuration file on the shared root.

If your site does not have a site customized file in:

```
/opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf
```

for node health on boot, then the a sample one is copied into place there. You should modify the file in the template directory for your site.

4.3 About `CLEinstall.conf` Parameters for Additional Features and Subsystems

You **must** modify additional settings if you configure optional functionality or subsystems. To configure and enable a particular functionality, follow the referenced section:

Lustre File System Support and Tuning on page 34

Configuring Boot-node Failover on page 35

Configuring SDB Node Failover on page 37

Including DVS in the Compute Node Boot Image on page 39

Configuring Dynamic Shared Objects and Libraries (DSL) and the Compute Node Root Runtime Environment (CNRTE) on page 39

Configuring Realm-Specific IP Addressing (RSIP) on page 41

Configuring Cluster Compatibility Mode (CCM) on page 42

Configuring Graphics Processing Units on page 45

Configuring `ntpclient` for Clock Synchronization on page 45

Including Security Auditing in the Compute Node Boot Image on page 45

Configuring Checkpoint/Restart (CPR) on page 46

Configuring Comprehensive System Accounting (CSA) on page 47

Configuring the Parallel Command (`pcmd`) Tool for Unprivileged Users on page 48

If you want to partition a Cray XE system into *logical machine*s, see Appendix G, Creating Partitions on a Cray XE System on page 175

For more information, see the comments in the `CLEinstall.conf` file or the `CLEinstall.conf`(5) man page.

4.3.1 Lustre File System Support and Tuning

Optional: The Lustre file system is optional; however, applications that run on CNL compute nodes require either Lustre file systems or DVS in order to perform I/O operations.

Several `CLEinstall.conf` parameters are available to configure your system for Lustre file systems and set up basic Lustre file system tuning. In most cases, the default values are acceptable. In addition to setting these parameters, refer to Chapter 6, Configuring Lustre File Systems on page 119, as you complete the installation or upgrade process.

`lustre=`**`yes`** If you are using Lustre file systems, set this parameter to **`yes`** to add Lustre-specific options to `modprobe.conf` kernel configuration files.

The shared root default view of `/etc/modprobe.conf.local` is updated for service nodes as follows, where `<*>` is replaced by a value that is defined by `lustre_*` parameters in `CLEinstall.conf`.

```
options lnet networks=gni
options max_nodes=<*> credits=<*> \
peer_hash_table_size=<*> \
options ost oss_num_threads=<*>
options libcfs libcfs_panic_on_lbug=1
```

`/opt/xt-images/templates/default/etc/modprobe.conf` is updated for compute nodes as follows:

```
options lnet networks=gni
options max_nodes=<*> \
options libcfs libcfs_panic_on_lbug=1
```

The `CLEinstall` program sets the networks option to *gni* for Cray XE systems with the Gemini network interconnect.

The `libcfs_panic_on_lbug` option is not configured in `CLEinstall.conf`. For more information, see *Managing Lustre for the Cray Linux Environment (CLE)*.

`lustre_elevator=noop`

Specifies a value for `elevator` in the SNL boot image parameters file; sets the default scheduler for a Lustre object storage server (OSS). Currently, the `noop` scheduler is recommended for Lustre on high-performance storage.

`lustre_clients=`

Specifies a value for `max_nodes` in `/etc/modprobe.conf.local` for service nodes; used to calculate buffer allocation for connection to Lustre clients. Cray recommends that you set this parameter to the total number of compute nodes and login nodes configured on your Cray system, rounded up to the nearest 100.

`lustre_servers=`

Specifies a value for `max_nodes` in `/opt/xt-images/templates/default/etc/modprobe.conf` for compute nodes; used to calculate buffer allocation for connection to Lustre servers. Cray recommends that you set this parameter to the total number of Lustre servers configured on your Cray system, rounded up to the nearest 100.

`lustre_credits=2048`

Specifies a value for `credits` in `/etc/modprobe.conf.local` for service nodes; defines the number of outstanding transactions allowed for a Lustre server. Cray recommends that you set this parameter to `2048`.

`lustre_peer_hash_table_size=509`

Specifies a value for `peer_hash_table_size` in `/etc/modprobe.conf.local` for service nodes; defines the size of the hash table for the client peers and enables `lnet` to search large numbers of peers more efficiently. Cray recommends that you set this parameter to `509`.

`lustre_oss_num_threads=256`

Specifies a value for `lustre_oss_num_threads` in `/etc/modprobe.conf.local` for service nodes; defines the number of threads a Lustre OSS uses. Cray recommends that you set this parameter to `256` threads.

4.3.2 Configuring Boot-node Failover

Optional: Boot-node Failover is an optional CLE feature.

You can configure your system to automatically failover to a backup (alternate) boot node when the primary boot node fails.

Set these parameters to configure `CLEinstall` to automatically complete several configuration steps for boot-node failover.

In addition, you must specify the primary and backup nodes in the boot configuration and configure the STONITH capability on the blade or module of the primary boot node. You are directed to complete these steps after Creating Boot Images on page 62 for new system installations, or after Creating Boot Images on page 144 for update package installations.

The following `CLEinstall.conf` parameters configure boot-node failover.

`node_boot_alternate=`

Specifies the backup or alternate boot node. The alternate boot node requires an Ethernet connection to the SMW and a QLogic Host Bus Adapter (HBA) card to communicate with the boot RAID. The alternate boot node **must not** reside on the same blade as the primary boot node.

`bootnode_failover=`**`yes`**

Set this parameter to **`yes`** to configure boot-node failover.

⚠ **Caution:** The STONITH capability is required to implement boot-node failover. Because STONITH is a per blade setting and not a per node setting, you must ensure that your primary boot node is not assigned to a blade that hosts services with conflicting STONITH requirements, such as Lustre.

`bootnode_failover_IPaddr=`*10.131.255.254*
`bootimage_bootnodeip=`*10.131.255.254*
`persistent_var_IPaddr=`*10.131.255.254*

Specifies the virtual IP address for boot-node failover. These must all match. The default is `10.131.255.254` for Cray XE systems with Gemini network interconnect. In most cases, the default value is acceptable. You must modify default value for the other two parameters to match the address specified by `bootnode_failover_IPaddr`.

`bootnode_failover_netmask=`*255.252.0.0*

Specifies the network mask for the boot-node failover virtual IP address. The default is `255.252.0.0` for Cray XE systems with Gemini network interconnect. In most cases, the default value is acceptable.

`bootnode_failover_interface=`*ipogif0:1*

> Specifies the virtual network interface for boot-node failover. The default value is `ipogif0:1` for Cray XE systems with Gemini network interconnect. In most cases, the default value is acceptable.

For additional information, including manual boot node failover configuration steps, see *Managing System Software for Cray XE and Cray XK Systems*.

4.3.3 Configuring SDB Node Failover

Optional: SDB Node Failover is an optional CLE feature.

You can configure your system to automatically failover to a backup (alternate) SDB node when the primary SDB node fails.

Use the parameters described in this section to configure `CLEinstall` to automatically complete several configuration steps for SDB node failover.

In addition, you must configure STONITH for the primary SDB node, specify the primary and backup nodes in the boot configuration, and optionally create a site-specific `sdbfailover.conf` file for the backup SDB node. You are directed to complete these steps after Creating Boot Images on page 62 for new system installations, or after Creating Boot Images on page 144 for update package installations.

After booting and testing your system, follow Procedure 42 on page 116 to configure your system to start SDB services automatically on the backup SDB node in the event of a SDB node failover.

The following `CLEinstall.conf` parameters configure SDB node failover.

`node_sdb_alternate=`

> Specifies the backup or alternate SDB node. The alternate SDB node requires a QLogic Host Bus Adapter (HBA) card to communicate with the RAID. This node is dedicated and cannot be used for other service I/O functions. The alternate SDB node **must** reside on a separate blade from the primary SDB node.

`sdbnode_failover=`**`yes`**

Set this parameter to **`yes`** to configure SDB node failover.

Caution: The STONITH capability is required to implement SDB node failover. Because STONITH is a per blade setting and not a per node setting, you must ensure that your primary SDB node is not assigned to a blade that hosts services with conflicting STONITH requirements, such as Lustre.

`sdbnode_failover_IPaddr=`*10.131.255.253*

Specifies the virtual IP address for SDB node failover. The default is `10.131.255.253` for Cray XE systems with Gemini network interconnect. In most cases, the default value is acceptable.

`sdbnode_failover_netmask=`*255.252.0.0*

Specifies the network mask for the SDB node failover virtual IP address. The default is `255.252.0.0` for Cray XE systems with Gemini network interconnect. In most cases, the default value is acceptable.

`sdbnode_failover_interface=`*ipogif0:1*

Specifies the virtual network interface for SDB node failover. This parameter must be defined even if you are not configuring SDB node failover. The default value is `ipogif0:1` for Cray XE systems with Gemini network interconnect. In most cases, the default value is acceptable.

When these parameters are used to configure SDB node failover, the `CLEinstall` program will verify and turn on `chkconfig` services and associated configuration files for `sdbfailover`.

Note: The backup SDB node uses the `/etc` files that are class or node specialized for the primary SDB node and not for the backup node itself; the `/etc` files for the backup node are identical to those that existed on the primary SDB node.

For additional information about SDB node failover, see *Managing System Software for Cray XE and Cray XK Systems*.

4.3.4 Including DVS in the Compute Node Boot Image

Optional: Cray DVS is an optional CLE feature.

The following `CLEinstall.conf` parameter configures `CLEinstall` to include the DVS RPM in the compute node boot image. In addition to setting this parameter, refer to Configuring Cray DVS on page 100, as you complete the installation or upgrade process.

`CNL_dvs=`**`yes`**

Set this parameter to **`yes`** to include the DVS RPM in the compute node boot image.

Optionally, you can edit the `/var/opt/cray/install/shell_bootimage_`*LABEL*`.sh` script and specify `CNL_DVS=`**`y`** before you update the CNL boot image.

For additional information about DVS, see *Introduction to Cray Data Virtualization Service*.

4.3.5 Configuring Dynamic Shared Objects and Libraries (DSL) and the Compute Node Root Runtime Environment (CNRTE)

Optional: Dynamic shared objects and libraries (DSL) and the compute node root runtime environment (CNRTE) are optional.

When the CLE compute node root runtime environment (CNRTE) is configured, users can link and load dynamic shared objects in their applications. To configure and install the compute node root runtime environment, you must configure the shared root as a DVS-projected file system.

To configure DSL and the compute node root runtime environment for your Cray system, follow these steps as you complete your CLE installation or upgrade.

1. Select the service or compute nodes to configure as compute node root servers. Any compute nodes used for CNRTE will no longer be part of the compute node pool. Do not use the same nodes configured as Lustre server nodes.

2. When you edit `CLEinstall.conf` Procedure 4 on page 55), modify DSL-specific parameters according to your configuration.

3. If you are configuring compute nodes as compute node root servers, Cray recommends that you configure the nodes as repurposed compute nodes. Before you run `CLEinstall`, follow Repurposing Compute Nodes as Service Nodes on page 56.

   Optionally, you can configure DSL with compute nodes that are not repurposed nodes. After you boot the service nodes, manually boot any compute nodes configured as a compute node root server as described in Booting Compute Node Root Servers on page 89 and when you refer to Configuring a Boot Automation File on page 114, follow additional steps to start the compute node root servers.

   > **Note:** This capability is deprecated and may not be supported in future releases. To configure compute nodes for CNRTE, Cray recommends that you repurpose the nodes as service nodes.

When you set the following parameters in the `CLEinstall.conf` file, the `CLEinstall` program automatically configures your system for the compute node root runtime environment.

`DSL=`**`yes`**
: Set this parameter to **`yes`** to enable dynamic shared objects and libraries and the compute node root runtime environment (CNRTE). The default is `no`.

> **Note:** Setting this option to `yes` will automatically enable DVS.

`DSL_nodes=`
: Specifies the nodes that will act as DVS compute node root servers. These nodes can be a combination of service or compute nodes. Set to integer node IDs (NIDs) separated by a space.

`DSL_mountpoint=`***/dsl***

Specifies the mount point on the DVS servers for the compute nodes; it is the projection of the shared root file system. The compute nodes will mount this path as `/`. In most cases, the default value is acceptable.

`DSL_attrcache_timeout=`***14400***

Specifies the attribute cache time out for compute node root servers; it is the number of seconds before DVS attributes are considered invalid and are retrieved from the server again. In most cases, the default value is acceptable.

The `CLEinstall` program creates a default `cnos` specialization class. This class allows an administrator to specialize files specifically for compute nodes; it is used with dynamic shared objects and libraries (DSL). If the `cnos` specialization class exists and DSL is enabled, those specialized `/etc` files are automatically mounted on the compute node roots.

For additional information about DSL, see *Managing System Software for Cray XE and Cray XK Systems*. For additional information about DVS, see *Introduction to Cray Data Virtualization Service*.

4.3.6 Configuring Realm-Specific IP Addressing (RSIP)

Optional: Realm-Specific IP Addressing (RSIP) is an optional CLE feature.

Realm-Specific IP Addressing (RSIP) allows CLE compute and service nodes to share IP addresses configured on the external Gigabit and 10 Gigabit Ethernet interfaces of network nodes. By sharing the external addresses, you may rely on your system's use of private address space and avoid the need to configure compute nodes with addresses within your site's IP address space. The external hosts see only the external IP addresses of the Cray system.

Note: RSIP on Cray systems supports IPv4 TCP and UDP transport protocols but not IP Security and IPv6 protocols.

Select the nodes to configure as RSIP servers. RSIP servers must run on service nodes that have a local external IP interface such as a 10GbE network interface card (NIC). Cray requires that you configure RSIP servers as dedicated network nodes.

Warning: Do not run RSIP servers on service nodes that provide Lustre services, login services, or batch services.

The following `CLEinstall.conf` parameters configure RSIP.

`rsip_nodes=`

Specifies the RSIP servers. Populate with space separated integer NIDs of the nodes you have identified as RSIP servers.

`rsip_interfaces=`

Specifies the IP interface for each RSIP server node. Populate with a space separated list of interfaces that correlate with the `rsip_nodes` parameter.

`rsip_servicenode_clients=`

Set this parameter to a space separated integer list of service nodes you would like to use for RSIP clients.

⚡ **Warning:** Do not configure service nodes with external network connections as RSIP clients. Configuring a network node as an RSIP client will disrupt network functionality. Service nodes with external network connections will route all non-local traffic into the RSIP tunnel and IP may not function as desired.

`CNL_rsip=`**`yes`**

Set this parameter to **`yes`** to include the RSIP RPM in the compute node boot image.

Optionally, you can edit the `/var/opt/cray/install/shell_bootimage_`*LABEL*`.sh` script and specify `CNL_RSIP=`**`y`** before you update the CNL boot image.

For example, to configure `nid00016` and `nid00020` as RSIP servers both using an external interface named `eth0`; `nid00064` as an RSIP server using an external interface named `eth1`; and `nid00000` as a service node RSIP client, set the following parameters.

```
rsip_nodes=16 20 64
rsip_interfaces=eth0 eth0 eth1
rsip_servicenode_clients=0
CNL_rsip=yes
```

For additional information, see the `rsipd`(8), `xtrsipcfg`(8), and `rsipd.conf`(5) man pages and *Managing System Software for Cray XE and Cray XK Systems*. Enhancements to the default RSIP configuration require a detailed analysis of site-specific configuration requirements. Contact your Cray representative for assistance in changing the default RSIP configuration.

4.3.7 Configuring Cluster Compatibility Mode (CCM)

Optional: Cluster Compatibility Mode (CCM) is an optional CLE feature.

Cluster Compatibility Mode (CCM) provides the services needed to run most cluster-based independent software vendor (ISV) applications out-of-the-box, however some configuration changes may be appropriate based on program specifications. CCM is built on top of the compute node root runtime environment (CNRTE), the infrastructure used to provide dynamic library support in Cray systems.

CCM is tightly coupled to the batch workload management system; it uses the batch system to logically designate part of the Cray system as an emulated cluster for the duration of a job. Users can execute cluster applications alongside workload-managed jobs running in a traditional batch or interactive queue.

Specific third-party batch system software releases are required for CCM support. For more information, access the **3rd Party Batch SW** link on the CrayPort website at http://crayport.cray.com. For more information about CCM, including steps required to create CCM batch queues on Cray systems, see *Managing System Software for Cray XE and Cray XK Systems*.

Requirements:

- CNRTE must be installed; see Configuring Dynamic Shared Objects and Libraries (DSL) and the Compute Node Root Runtime Environment (CNRTE) on page 39.
- (Optional) RSIP must be installed if you have applications that need access to a license server; see Configuring Realm-Specific IP Addressing (RSIP) on page 41.

To configure CCM for your Cray system, follow these steps as you complete your CLE installation or upgrade.

1. When you edit `CLEinstall.conf` (Procedure 4 on page 55), modify CCM-specific parameters according to your configuration.
2. After you boot the service nodes during your installation or upgrade, follow the steps in Completing CCM Configuration on page 103 to mount user home directories on the compute nodes and modifying CCM and platform-MPI configuration files.
3. After your CLE system installation is complete, install a third-party batch system software (for example, PBS or TORQUE) at a level that supports CCM; see Installing a Batch System on page 155.
4. Configure CCM batch queues on your Cray system; see *Managing System Software for Cray XE and Cray XK Systems*.

When you set the following parameters in the `CLEinstall.conf` file, `CLEinstall` automatically installs CCM.

`CCM=`**`yes`**
: Set this parameter to **`yes`** to enable Cluster Compatibility Mode (CCM) and install the appropriate RPMs.

`CCM_ENABLERSH=`**`yes`**
: Set this parameter to **`yes`** to enable services or daemons that most ISV applications need to run. Examples of these services are `xinetd`, `portmap`, `rsh`, and `rlogin`. If you set `CCM_ENABLERSH` to `no` some ISV applications will not work. If you do not specify this parameter, `rsh` is enabled by default.

`CCM_QUEUES=`***ccm_queue1*,*ccm_queue2***

Specifies one or more batch queues used in the workload management system. List queue names, separated by commas. The default value is `ccm_queue`.

> **Note:** After your batch system software is installed, you must manually create the queues you specify here. For steps required to create CCM batch queues, see *Managing System Software for Cray XE and Cray XK Systems*.

`CCM_ENABLENIS=`**`no`**

Set this parameter to **`yes`** to start `ypservices` on the compute node. If Network Information Service (NIS) is not properly configured, network calls may time out, significantly slowing down CCM startup. The default is to disable NIS.

`CCM_WLM=`**`pbs`**

Specifies the batch processing system; valid values are `pbs`, `torque`, and `lsf`.

> **Note:** If `CCM_WLM=lsf` is specified and any non-null values are set for `CRAY_QSTAT_PATH` and `CRAY_BATCH_VAR`, a message is displayed by `CLEinstall` stating that the settings of `CRAY_QSTAT_PATH` and `CRAY_BATCH_VAR` will be ignored and the variables will be set to `""` in `/etc/opt/cray/ccm/ccm.conf` on the shared root.

`CRAY_QSTAT_PATH=`***/opt/pbs/default/bin***

Specifies the path to the batch system software `qstat` command. The default value is the path for PBS Professional; the path for Moab and TORQUE is included as a comment in the configuration file.

This variable is ignored if `CCM_WLM=`**`lsf`**.

`CRAY_BATCH_VAR=`***/var/spool/PBS***

Specifies the path to the batch system software `/var` directory.

This variable is ignored if `CCM_WLM=`**`lsf`**.

You may also set these parameters after installation by editing `/etc/opt/cray/ccm/ccm.conf`. This file also contains exclusively post-install options.

4.3.8 Configuring Graphics Processing Units

`GPU=`**`yes`**

Set this value to **`yes` only** if your machine has Cray XK6 blades with GPUs installed. Setting this parameter to **`yes`** will install the required RPMs and code for GPU systems. If your machine has Cray XK6 blades without GPUs, or it does not have Cray XK6 blades installed, set this parameter to `no`.

Note: For Cray XK6 systems with GPUs, you must configure and enable the alternative compute node root run time environment for dynamic shared objects and libraries (DSL). Cray XK6 blades require access to shared libraries to support GPUs.

4.3.9 Including Security Auditing in the Compute Node Boot Image

Optional: Cray Audit is an optional CLE feature.

Cray Audit is an optional set of Cray specific extensions to Linux security auditing. Cray specific utilities simplify the administration of auditing across many nodes. For more information, see *Managing System Software for Cray XE and Cray XK Systems*.

Use the following `CLEinstall.conf` parameter to include security auditing RPMs in the compute node boot image. In addition to setting this parameter, refer to Completing Cray Audit Configuration on page 105, as you complete the installation or upgrade process.

`CNL_audit=`**`yes`**

Set this parameter to **`yes`** to include the security auditing RPM in the compute node boot image.

Optionally, you can edit the `/var/opt/cray/install/shell_bootimage_`*`LABEL`*`.sh` script and specify `CNL_AUDIT=`**`y`** before you update the CNL boot image.

4.3.10 Configuring `ntpclient` for Clock Synchronization

Optional: The `ntpclient` is an optional CLE feature.

A network time protocol (NTP) client, `ntpclient`, is available to install on compute nodes; it synchronizes the time of day on the compute node clock with the clock on the boot node.

Without this feature, compute node clocks drift apart over time. When `ntpclient` is installed, the clocks drift apart during a four hour calibration period and then converge on the time reported by the boot node. Note that the standard CLE configuration includes an NTP daemon (`ntpd`) on the boot node to synchronize with the clock on the SMW, and the service nodes run `ntpd` to synchronize with the boot node.

Use the following `CLEinstall.conf` parameter to enable `ntpclient` on the compute nodes.

`CNL_ntpclient=`**`yes`**

Set this parameter to **`yes`** to include the `ntpclient` RPMs in the compute node boot image.

Optionally, you can edit the `/var/opt/cray/install/shell_bootimage_`*LABEL*`.sh` script and specify `CNL_NTPCLIENT=`**`y`** before you update the CNL boot image.

4.3.11 Configuring Checkpoint/Restart (CPR)

Optional: Checkpoint/Restart (CPR) is an optional CLE feature.

CPR provides a way to suspend and snapshot the state of a running application. This snapshot can be used to restart the application at a later time. The CPR feature for CLE is built on the Berkeley Lab Checkpoint/Restart (BLCR) for Linux.

When you use the following `CLEinstall.conf` parameters to configure CPR, the `CLEinstall` program handles most of the details for installing the software that is required to support checkpoint/restart.

`cpr=`**`yes`** Set this parameter to **`yes`** to configure CPR.

`CNL_cpr=`**`yes`**

Set this parameter to **`yes`** to include the RPM for the CPR client in the compute node boot image.

Optionally, you can edit the `/var/opt/cray/install/shell_bootimage_`*LABEL*`.sh` script and specify `CNL_CPR=`**`y`** before you update the CNL boot image.

Because of the known file-per-node I/O access of checkpoint/restart, the checkpoint directory's file system setting should be optimized for this access pattern. For Lustre, it is optimal to set the stripe count to one. Perform this step later in the installation process when Configuring Striping on Lustre File Systems on page 129.

```
lfs setstripe checkpoint_dir -s 0 -i -1 -c 1
```

In addition to the software automatically installed by `CLEinstall`, specific third-party batch system software releases are required for checkpoint/restart support. For more information, access the **3rd Party Batch SW** link on the CrayPort website at http://crayport.cray.com.

For more information about CPR, including additional batch software configuration steps required to support checkpoint/restart on Cray systems, see *Managing System Software for Cray XE and Cray XK Systems*.

4.3.12 Configuring Comprehensive System Accounting (CSA)

> **Optional:** Comprehensive System Accounting (CSA) is an optional software package. The CSA package depends on the job software package, which is mandatory. The job package must be enabled using the `chkconfig` command while in the `xtopview` utility.

The CSA software package includes accounting utilities to perform standard types of system accounting processing on CSA generated accounting files. Create jobs on the system by using either a batch job entry system (when such a system is used to launch jobs) or by using the `pam_job` module for interactive sessions.

CSA software includes the `csa`, `projdb`, `job`, and `account` RPMs. CSA is turned off by default because you must make CSA configuration files system specific before you can use CSA accounting.

The `csa`(8) and `intro_csa`(8) man pages describe all of the features and capabilities of CSA.

Specific third-party software releases are required for batch system compatibility with CSA on Cray systems. For more information, access the **3rd Party Batch SW** link on the CrayPort website at http://crayport.cray.com.

Use the following `CLEinstall.conf` parameter to configure `CLEinstall` to install the CSA RPMs and associated man pages. In addition to setting this parameter, see *Managing System Software for Cray XE and Cray XK Systems* for information about completing the installation and configuration of CSA on a Cray system.

`CNL_csa=`**`yes`**

Set this parameter to **`yes`** to include the CSA RPMs in the compute node boot image.

Optionally, you can edit the `/var/opt/cray/install/shell_bootimage_`*LABEL*`.sh` script and specify `CNL_CSA=`**`y`** before you update the CNL boot image.

4.3.13 Configuring the Parallel Command (pcmd) Tool for Unprivileged Users

> **Optional:** Configuring the Parallel Command Tool for unprivileged users is an optional CLE feature. Sites that are uncomfortable having a `setuid root` program on their system may keep `pcmd` a root-only tool.

Parallel Command (`pcmd`) is a secure tool that runs commands on the compute nodes as the user who launched the command. A user may specify which nodes to run the command on. For more information, see the `pcmd`(1) man page.

By default, `pcmd` will be installed as a root only tool. In order for non-root users to be able to run the tool, `pcmd` should be installed as a `setuid root` program; this can be done at installation time by specifying `NHC_pcmd_suid=yes` in `CLEinstall.conf`.

4.4 About System Set Configuration in `/etc/sysset.conf`

The `/etc/sysset.conf` configuration file defines system sets. Each system set is defined by the following information for each device or boot RAID disk partition in the set: *function*, *SMWdevice*, *host*, *hostdevice*, *mountpoint*, and a *shared* flag. Each system set definition also contains a `LABEL` and a `DESCRIPTION`. The information regarding the disk partition is based on the zoning of the LUNs on the boot RAID.

Using this file, the system administrator can configure a group of disk devices and disk partitions on the boot RAID into a system set that can be used as a complete bootable system. By configuring system sets, a system administrator can easily switch between different software releases or configurations. For example, you can use (or create) separate production and test system sets to manage updates and upgrades of the CLE operating system.

In Creating Configuration Files on page 55, you are directed to create a `/etc/sysset.conf` file specifically for your system configuration. A sample or template file for `/etc/sysset.conf` is delivered on the `Cray CLE 4.0.UP`*nn* `Software` DVD. The template contains two example system sets (`BLUE` and `GREEN`). Modify these examples to match your system configuration. You must create the `/etc/sysset.conf` file before you invoke the installation program, at which time you specify the system set to install, upgrade, or update.

Follow these requirements, restrictions, and tips when you create a site-specific `sysset.conf` file. For more information, see the `sysset.conf`(5) man page.

- The `/etc/sysset.conf` file includes two sets of device names for the boot RAID; *SMWdevice* is the pathname to the disk partition on the SMW and *hostdevice* is the pathname on the Cray system (host).
- You **must** configure persistent device names for the boot RAID disk devices. Cray recommends that you use the `/dev/disk/by-id/` persistent device names. For more information, see About Persistent Boot RAID Device Names on page 50.
- Some partitions may be shared between two or more sets, such as `/syslog`.
- Some partitions must exist in **only** one set; for example, a matched triplet of boot root, shared root, and boot image.
- *SMWdevice* may be a path name to a device or a dash (`-`).
- *hostdevice* may be a path name to a device or a dash (`-`).
- Set *SMWdevice* and *hostdevice* to dash (`-`) for `BOOT_IMAGE`*n* if the boot image is a file and not a raw device.
- *hostdevice* may be a dash (`-`) with a real *SMWdevice* only when the *function* is `RESERVED`.
- `BOOT_IMAGE`*n* may be a raw disk device that has *SMWdevice* and *hostdevice* as path names to real devices. Specify *mountpoint* as a link to that device.
- `BOOT_IMAGE`*n* may be an archive (`cpio`) file in a directory. The directory must exist on both the SMW and the boot root, with the same name. Specify *mountpoint* as the path name to this type of boot image file.
- *mountpoint* may be a dash (`-`) if it is a Lustre device (`LUSTREMDS0` or `LUSTREOST0`).
- The `RESERVED` *function* can be used to indicate that a partition has a site-defined function and should not be overwritten by `CLEinstall` or `xthotbackup`.
- Some partitions may be marked `RESERVED` and yet belong to a system set.
- The `RESERVED` system set `LABEL` contains all orphaned disk partitions that are not in any other system set.
- If the SMW does not have access to the `SDB` and `SYSLOG` disk devices on the boot RAID, specify *SMWdevice* for these entries as a dash (`-`). Ensure *hostdevice* is set to the node that has access to these disk partitions. In this case, the `CLEinstall` program generates scripts to create these file systems and suggests when to run the scripts.

In Procedure 4 on page 55, you are directed to create and edit your `/etc/sysset.conf` file. Make all site-specific changes at that point in the installation process.

4.4.1 About Device Partitions in `/etc/sysset.conf`

Check the boot RAID configuration and QLogic switch zoning (for QLogic Fibre Channel switch or DDN device) or SANshare configuration (for NetApp, Inc. disks). These can be configured to allow all hosts to see all LUNs or to allow some hosts to see only a few LUNs.

Use the `fdisk` command on the SMW to confirm that your partitions are identified. Invoke `fdisk -l` to display a list of all detected partitions on the boot RAID disk devices. Compare the output to the list of *SMWdevice* partitions included in your `/etc/sysset.conf` file; identify any partitions without an assigned *function* and confirm that they are unused. You may include these remaining partitions in the system set labeled `RESERVED` in `/etc/sysset.conf`.

4.4.2 About Persistent Boot RAID Device Names

The `/etc/sysset.conf` file includes two sets of device names for the boot RAID; *SMWdevice* and *hostdevice*. Because SCSI device names (`/dev/sd*`) are not guaranteed to be numbered the same from boot to boot, you **must** configure persistent device names for these boot RAID disk devices. Cray recommends that you use the `by-id` persistent device names.

Caution: You must use `/dev/disk/by-id` when specifying the root file system. There is no support in the `initramfs` for `cray-scsidev-emulation` or custom `udev` rules.

To configure persistent `by-id` device names, modify the `SMWdevice` and `hostdevice` columns to match the `/dev/disk/by-id/` SCSI device names on your system.

Example 3. System set format from the sysset.conf template

```
# LABEL:
# DESCRIPTION:
# function        SMWdevice                  host      hostdevice                 mountpoint           shared
# BOOTNODE_ROOT   /dev/disk/by-id/IDa-part1  boot      /dev/disk/by-id/IDa-part1  /                      no
# BOOTNODE_SWAP   /dev/disk/by-id/IDa-part2  boot      /dev/disk/by-id/IDa-part2  swap                   no
# SHAREDROOT      /dev/disk/by-id/IDc-part6  boot      /dev/disk/by-id/IDc-part6  /rr                    no
# BOOT_IMAGE0     /dev/disk/by-id/IDc-part7  boot      /dev/disk/by-id/IDc-part7  /raw0                  no
# BOOT_IMAGE1     -                          boot      -                          /bootimagedir/xt.tst1  no
# BOOT_IMAGE2     -                          boot      -                          /bootimagedir/xt.tst2  no
# BOOT_IMAGE3     -                          boot      -                          /bootimagedir/xt.tst3  no
# SDB             /dev/disk/by-id/IDd-part1  sdb       /dev/disk/by-id/IDd-part1  /var/lib/mysql         no
# SYSLOG          /dev/disk/by-id/IDe-part1  syslog    /dev/disk/by-id/IDe-part1  /syslog                no
# UFS             /dev/disk/by-id/IDd-part2  ufs       /dev/disk/by-id/IDd-part2  /ufs                   no
# PERSISTENT_VAR /dev/disk/by-id/IDc-part9  boot      /dev/disk/by-id/IDc-part9  /snv                   no
# LUSTREMDS0      /dev/disk/by-id/IDc-part5  nid00008  /dev/disk/by-id/IDc-part5  -                      no
# LUSTREOST0      /dev/disk/by-id/IDh-part1  nid00011  /dev/disk/by-id/IDh-part1  -                      no
```

Example 4. Modifying /etc/sysset.conf for persistent by-id device names

When you create a site-specific /etc/sysset.conf file (Procedure 4 on page 55), modify each device path to use the persistent device names in /dev/disk/by-id.

For each partition identified in About Device Partitions in /etc/sysset.conf on page 50, determine the by-id persistent device name. For example, if you defined the boot node root and swap to be devices sdc1 and sdc2, invoke the following commands and note the volume identifier portion of the names.

```
crayadm@smw:~> ls -l /dev/disk/by-id/* | grep sdc
lrwxrwxrwx 1 root root  9 2010-02-23 14:35 /dev/disk/by-id/scsi-3600a0b800026e1
400000192a4b66eb21 -> ../../sdc
lrwxrwxrwx 1 root root 10 2010-02-23 14:35 /dev/disk/by-id/scsi-3600a0b800026e1
400000192a4b66eb21-part1 -> ../../sdc1
lrwxrwxrwx 1 root root 10 2010-02-23 14:35 /dev/disk/by-id/scsi-3600a0b800026e1
400000192a4b66eb21-part2 -> ../../sdc2
crayadm@smw:~>
```

Replace IDa-part* for both SMWdevice and hostdevice with the volume identifier and partition number. For example, change:

```
# BOOTNODE_ROOT  /dev/disk/by-id/IDa-part1  boot      /dev/disk/by-id/IDa-part1  /                 no
# BOOTNODE_SWAP  /dev/disk/by-id/IDa-part2  boot      /dev/disk/by-id/IDa-part2  swap              no
```

To

```
BOOTNODE_ROOT  /dev/disk/by-id/scsi-3600a0b800026e1400000192a4b66eb21-part1 boot \
/dev/disk/by-id/scsi-3600a0b800026e1400000192a4b66eb21-part1 /         no
BOOTNODE_SWAP  /dev/disk/by-id/scsi-3600a0b800026e1400000192a4b66eb21-part2 boot \
/dev/disk/by-id/scsi-3600a0b800026e1400000192a4b66eb21-part2 swap      no
```

Note: Ensure that each entry is on a single line. For formatting purposes, the example splits each entry into two lines.

Example 5. Modified system set with persistent device names

```
LABEL:MYCRAYPRD
DESCRIPTION: mycray production system set
# function     SMWdevice                                                       host \
               hostdevice                                                         mountpoint   shared
BOOTNODE_ROOT  /dev/disk/by-id/scsi-3600a0b800026e1400000192a4b66eb21-part1    boot \
               /dev/disk/by-id/scsi-3600a0b800026e1400000192a4b66eb21-part1       /             no
BOOTNODE_SWAP  /dev/disk/by-id/scsi-3600a0b800026e1400000192a4b66eb21-part2    boot \
               /dev/disk/by-id/scsi-3600a0b800026e1400000192a4b66eb21-part2       swap          no
SHAREDROOT     /dev/disk/by-id/scsi-3600a0b800026e1400000192c4b66eb70-part1    boot \
               /dev/disk/by-id/scsi-3600a0b800026e1400000192c4b66eb70-part1       /rr           no
BOOT_IMAGE0    /dev/disk/by-id/scsi-3600a0b800026e1400000192c4b66eb70-part2    boot \
               /dev/disk/by-id/scsi-3600a0b800026e1400000192c4b66eb70-part2       /raw0         no
BOOT_IMAGE1    /dev/disk/by-id/scsi-3600a0b800026e1400000192c4b66eb70-part3    boot \
               /dev/disk/by-id/scsi-3600a0b800026e1400000192c4b66eb70-part3       /raw1         no
PERSISTENT_VAR /dev/disk/by-id/scsi-3600a0b800026e1400000192c4b66eb70-part4    boot \
               /dev/disk/by-id/scsi-3600a0b800026e1400000192c4b66eb70-part4       /snv          no
SDB            /dev/disk/by-id/scsi-3600a0b800026e1400000192e4b66eb97-part1    sdb  \
               /dev/disk/by-id/scsi-3600a0b800026e1400000192e4b66eb97-part1       /var/lib/mysql no
UFS            /dev/disk/by-id/scsi-3600a0b800026e1400000192e4b66eb97-part2    ufs  \
               /dev/disk/by-id/scsi-3600a0b800026e1400000192e4b66eb97-part2       /ufs          no
SYSLOG         /dev/disk/by-id/scsi-3600a0b800026e140000019304b66ebbb-part1    syslog \
               /dev/disk/by-id/scsi-3600a0b800026e140000019304b66ebbb-part1       /syslog       no
```

Note: Ensure that each entry is on a single line. For formatting purposes, the example splits each entry into two lines.

Installing CLE on a New System [5]

This chapter contains the information and procedures that are required to perform an initial installation of the Cray Linux Environment (CLE) base operating system (based on SLES 11 SP1) and Cray CLE software packages on a new Cray system.

After you have configured, formatted, zoned, and partitioned the RAID, follow the steps in this chapter to install the system software on the boot RAID partitions. Perform this work on the SMW.

Warning: The procedures in this chapter re-install the operating system software on your Cray system. You will overwrite existing CLE system software on the SMW and on the designated system partitions. If you are running CLE 3.1 or CLE 4.0 software, see Part II, *Update and Upgrade Installations*.

Note: Some examples are left-aligned to fit better on the page. Left alignment has no special significance.

5.1 Installing CLE Software on the SMW

Two DVDs are required to install the CLE 4.0 release on a Cray system. The first is labeled `Cray CLE 4.0.UP`*nn* `Software` and contains software specific to Cray systems. Optionally, you may have an ISO image called `xe-sles11sp1-`*4.0.46e04*`.iso`, where *n.n.nn* indicates the CLE release build level, and *avv* indicates the installer version.

The second DVD is labeled `Cray-CLEbase11sp1-`*yyyymmdd* and contains the CLE 4.0 base operating system which is based on SLES 11 SP1.

Procedure 2. Copying the software to the SMW

1. If the Cray system is booted, use your site-specific procedures to shut down the system. For example, to shutdown using an automation file:

   ```
   crayadm@smw:~> xtbootsys -s last -a auto.xtshutdown
   ```

   For more information about using automation files, see the `xtbootsys`(8) man page.

   Although not the preferred method, alternatively execute these commands as `root` from the boot node to shutdown your system.

   ```
   boot:~ # xtshutdown -y
   boot:~ # shutdown -h now;exit
   ```

2. Log on to the SMW as `root`.

   ```
   crayadm@smw:~> su - root
   ```

3. Insert the `Cray CLE 4.0.UP`*nn* `Software` DVD into the SMW DVD drive and mount it.

   ```
   smw:~# mount /dev/cdrom /media/cdrom
   ```

   Or

   To mount the release media using the ISO image, execute the following command, where `xe-sles11sp1-`*4.0.46e04*`.iso` is the path name to the ISO image file.

   ```
   smw:~# mount -o loop,ro xe-sles11sp1-4.0.46e04.iso /media/cdrom
   ```

4. Copy all files to a directory on the SMW in `/home/crayadm/install.`*xtrel*, where `xtrel` is a site-determined name specific to the release being installed.

   ```
   smw:~# mkdir /home/crayadm/install.4.0.46
   smw:~# cp -pr /media/cdrom/* /home/crayadm/install.4.0.46
   ```

5. Unmount the `Cray CLE 4.0.UP`*nn* `Software` DVD and eject it.

   ```
   smw:~# umount /media/cdrom
   smw:~# eject
   ```

6. Insert the `Cray-CLEbase11sp1` DVD into the SMW DVD drive and mount it.

   ```
   smw:~# mount /dev/cdrom /media/cdrom
   ```

   Or

   To mount the base operating system media using the ISO image, execute the following command, where `Cray-CLEbase11sp1-`*yyyymmdd*`.iso` is the path name to the ISO image file.

```
smw:~# mount -o loop,ro Cray-CLEbase11sp1-yyyymmdd.iso /media/cdrom
```

Procedure 3. Running `CRAYCLEinstall.sh`

1. As `root`, execute the installation script to install the Cray CLE software on the SMW.

   ```
   smw:~# /home/crayadm/install.4.0.46/CRAYCLEinstall.sh \
   -m /home/crayadm/install.4.0.46 -v -i -w
   ```

2. At the prompt '`Do you wish to continue?`', type **`y`** and press `Enter`.

   The output of the installation script displays on the console.

   > **Note:** If this script fails, you can restart it with the same options. However, rerunning this script may generate numerous error messages as it attempts to install already-installed RPMs. You may safely ignore these messages.

5.2 Creating Configuration Files

⚠ **Caution:** Chapter 4, About Installation Configuration Files on page 25 contains essential information about specific parameters that you must set before you install CLE software on a Cray system. Read it carefully before continuing. Improper configuration of the `CLEinstall.conf` and `/etc/sysset.conf` files can result in a failed installation.

As noted in About System Set Configuration in `/etc/sysset.conf` on page 48, you can install the CLE 4.0 release software to a system that has never had the CLE 4.0 release installed on it, or you can install the release to an alternative root location.

If this is the first installation, you must create the `CLEinstall.conf` and `/etc/sysset.conf` configuration files. After you complete the first installation, any installations to the alternative root location can use the `/etc/sysset.conf` file that was created during the first installation.

Based on the settings you choose in the `CLEinstall.conf` file, the `CLEinstall` program updates other configuration files. The `/etc/sysset.conf` file describes the assignment of devices and disk partitions on the boot RAID and their file systems or functions. For a description of the contents of these files, see Chapter 4, About Installation Configuration Files on page 25 or the `sysset.conf`(5) and `CLEinstall.conf`(5) man pages.

Note: You need to log out and back in again to access man pages that were installed in Procedure 3 on page 54.

Procedure 4. Creating the installation configuration files

1. Edit the `/home/crayadm/install.`*xtrel*`/CLEinstall.conf` configuration file. Carefully follow Chapter 4, About Installation Configuration Files on page 25 and make modifications for your specific configuration.

```
smw:~# chmod 644 /home/crayadm/install.4.0.46/CLEinstall.conf
```

```
smw:~# vi /home/crayadm/install.4.0.46/CLEinstall.conf
```

   Tip: Use the `rtr --system-map` command to translate between node IDs (NIDs) and physical ID names.

2. Copy the `/home/crayadm/install.`*xtrel*`/sysset.conf` system set template file to `/etc/sysset.conf`.

   ⚠ **Caution:** If you already have an `/etc/sysset.conf` file from a previous installation or upgrade, skip this step and do **not** overwrite it.

```
smw:~# cp -p /home/crayadm/install.4.0.46/sysset.conf /etc/sysset.conf
```

3. Edit the `/etc/sysset.conf` file so that it describes the disk devices and disk partitions that have been previously created on the boot RAID; designate the function or file system for each disk device and disk partition.

   ```
   smw:~# chmod 644 /etc/sysset.conf
   smw:~# vi /etc/sysset.conf
   ```

   Caution: You **must** ensure that *SMWdevice* and *hostdevice* are configured with persistent device names, based on your configuration. For more information, see About Persistent Boot RAID Device Names on page 50 and the `sysset.conf`(5) man page.

   a. For each function, determine the persistent `by-id` device names for your system by using the following command. For a complete example, see About Persistent Boot RAID Device Names on page 50, Example 4.

      ```
      crayadm@smw:~> ls -l /dev/disk/by-id/* | grep sdc
      ```

   b. Modify the `SMWdevice` and `hostdevice` columns to match the `/dev/disk/by-id/` SCSI device names on your system.

4. Make all site-specific changes; for example, configure separate production and test system sets. Save the file. For more information, see About System Set Configuration in `/etc/sysset.conf` on page 48.

5.3 Repurposing Compute Nodes as Service Nodes

Optional: If you intend to use compute nodes for a service node role, for example `DSL_nodes`, you can repurpose the nodes as service nodes.

CLE and SMW software include functionality to change the role of a compute node and boot the hardware with service node images. By using this functionality, you can add additional service nodes for services that do not require external connectivity. When a compute node is configured with a service node role, that node is referred to as a *repurposed compute node*.

The Cray system hardware state data is maintained in an HSS database where each node is marked with a compute or service node role. By using the `xtcli mark_node` command, you can mark a node in a compute blade to have a role of `service`.

Because they are marked as service nodes within the HSS, repurposed compute nodes are initialized as service nodes by the `CLEinstall` program and are booted automatically when all service nodes are booted.

For additional information, see *Repurposing Compute Nodes as Service Nodes on Cray XE and Cray XT Systems*.

Procedure 5. Marking repurposed compute nodes as service nodes in the HSS

Repeat the following steps for each NID you want to repurpose, for example, compute nodes as `DSL_nodes`.

1. Mark the repurposed compute node as a service node by using the `xtcli mark_node` command. For example:

   ```
   crayadm@smw:~> xtcli mark_node service c0-0c0s7n0
   ```

2. Verify that the node is a service node by using the `xtcli status` command. For example:

```
crayadm@smw:~> xtcli status c0-0c0s7n0
Network topology: class 0
Network type: Gemini
           Nodeid: Service  Core Arch|  Comp state      [Flags]
--------------------------------------------------------------------------
       c0-0c0s7n0: service  MC24   OP|          on      [noflags|]
--------------------------------------------------------------------------
crayadm@smw:~>
```

5.4 Running the `CLEinstall` Installation Program

The `CLEinstall` installation program installs and performs basic configuration of the CLE software for your configuration by using information in the `CLEinstall.conf` and `sysset.conf` configuration files.

The `CLEinstall` program accepts the following options:

`--label=`*system_set_label*

This option is required. Specify the label of the system set to be used for this installation. The specified label must exist in the system set configuration file that is specified with the `--syssetfile` option. This label is case-sensitive.

`--install|--upgrade|--bootimage-only`

This option is required. For full installations, use the `--install` option. For upgrade or update installations, use the `--upgrade` option. The `--upgrade` option requires that you specify the release with `--XTrelease=`*release_number* and Cray recommends that you also use the `--CLEmedia` option to specify a release-specific directory for the CLE software media. The `--bootimage-only` option recreates the `shell_bootimage_`*LABEL*`.sh` script and performs no other installation or upgrade related tasks.

[`--syssetfile=`*system_set_configuration_file*]

Specify the system set configuration file. The default is `/etc/sysset.conf`.

[`--configfile=`*CLEinstall_configuration_file*]

Specify the installation configuration file. The default is `./CLEinstall.conf`.

[`--nodebug`] Turn off debugging output to a debug file. By default, debugging output is written to `/var/adm/cray/logs/CLEinstall.debug.`*pid*.

[`--Basemedia=`*directory*]

Specify which directory the CLE base operating system media is mounted on. The default is `/media/cdrom`.

[`--CLEmedia=`*directory*]

Specify the directory where the software media has been placed. The default is `/home/crayadm/install`. The `--CLEmedia` option is required if the media is not in the default location. Documented installation procedures place the software media in a release-specific directory; for example, `/home/crayadm/install.`*release_number*, therefore, Cray recommends that you always use this option.

[`--XTrelease=`*release_number*]

Specify the CLE release and build level. *release_number* is a string in the form *x.y.level*, where *level* is the unique build identifier; for example, 4.0.46.

Note: The `--XTrelease` option is required with the `--upgrade` option and is not valid with the `--install` option.

[`--xthwinvxmlfile=`*XT_hardware_inventory_XML_file*]

Specify the hardware inventory XML file to use in place of the output from the `xthwinv` command with the `-x` option.

By default, `CLEinstall` invokes the `xthwinv -x` command on the SMW to retrieve hardware component information and creates a file, `/etc/opt/cray/sdb/attr.xthwinv.xml`, on the boot root file system. When this option is specified, *hardware_inventory_XML_file* is copied to `/etc/opt/cray/sdb/attr.xthwinv.xml` and the `xthwinv -x` command is not invoked. The `/etc/opt/cray/sdb/attr.xthwinv.xml` file is used in conjunction with the `/etc/opt/cray/sdb/attr.defaults` file to populate the node attributes table of the Service Database (SDB).

Use this option when the Cray system hardware is unavailable, or when you configure a back-up SMW that is not connected to the Hardware Supervisory System (HSS) network, or when you configure an unavailable partition on a partitioned system.

The *hardware_inventory_XML_file* must contain output from the `xthwinv -x` command.

[`--bootparameters=`*file*]

Specify the service node boot parameters file to be used when making the service node boot image. The `CLEinstall` program modifies this file as needed to include parameters that are defined in the `CLEinstall.conf` or `sysset.conf` configuration files. If this option is not specified for a new system installation, the default parameters file is used. If this option is not specified for an upgrade installation, the parameters file within an existing boot image is used.

[`--CNLbootparameters=`*file*]

Specify the CNL compute node boot parameters file to be used when making the CNL boot image. The `CLEinstall` program modifies this file as needed to include parameters that are defined in the `CLEinstall.conf` or `sysset.conf` configuration files. If this option is not specified for a new system installation, the default parameters file is used. If this option is not specified for an upgrade installation, the parameters file within an existing boot image is used.

[`--Lustreversion=`*version_number*]

Specify the version of Lustre to be installed. For example, `1.8.6`. If this option is not specified, `CLEinstall` will use the default version of Lustre.

[`--noforcefsck`]

Prevent `CLEinstall` from forcing a file system check. If this option is specified, `CLEinstall` invokes the `fsck` command without the `-f` option. This option is not recommended for normal use; however, you may specify this option when you restart `CLEinstall` after resolving an error.

[`--version`] Display the version of the `CLEinstall` program.

[`--help`] Display help message.

This information is also available in the `CLEinstall`(8) man page.

Procedure 6. Running `CLEinstall`

1. Invoke the `CLEinstall` program on the SMW. `CLEinstall` is located in the directory you created in Procedure 2 on page 53.

```
smw:~# /home/crayadm/install.4.0.46/CLEinstall --install --label=system_set_label \
--configfile=/home/crayadm/install.4.0.46/CLEinstall.conf \
--CLEmedia=/home/crayadm/install.4.0.46
```

2. Examine the initial messages and note the process ID (PID) of the `CLEinstall` process. Log files are created in `/var/adm/cray/logs` and named by using this PID. For example:

```
09:20:21 Installation output will be captured in /var/adm/cray/logs/\
CLEinstall.stdout.27670
09:20:21 Installation errors (stderr) will be captured in /var/adm/cray/logs/\
CLEinstall.stderr.27670
09:20:21 Installation debugging messages will be captured in /var/adm/cray/logs/\
CLEinstall.debug.27670
```

3. `CLEinstall` validates `sysset.conf` and `CLEinstall.conf` configuration settings and then confirms the expected status of your boot node and file systems.

   Confirm that the installation is proceeding as expected, respond to warnings and prompts, and resolve any issues. For example:

   - If you are installing to a system set that is not running, and you did not shut down your Cray system, respond to the following warning and prompt:

     ```
     WARNING:  At least one blade of p0 seems to be booted.
     Please confirm that the system set you are intending
     to update is not booted.
     Do you wish to proceed?[n]:
     ```

Warning: If the boot node has a file system mounted and `CLEinstall` on the SMW creates a new file system on that disk partition, the running system will be corrupted.

   - If you have configured file systems that are shared between two system sets, respond to the following prompt to confirm creation of new file systems:

```
09:21:24 INFO: The PERSISTENT_VAR disk function for the LABEL system set is marked shared.
09:21:24 INFO: The /dev/sdr1 disk partition will be mounted on the SMW for PERSISTENT_VAR
disk function.  Confirm that it is not mounted on any nodes in a running XT
system before continuing.
Do you wish to proceed?[n]:y
```

- If the `node_class[`*idx*`]` parameters do not match the existing `/etc/opt/cray/sdb/node_classes` file, `CLEinstall` will abort and require you to correct the node class configuration in `CLEinstall.conf` and/or the `node_classes` file on the `bootroot` and/or `sharedroot`. Correct the file, unmount the file systems and re-run `CLEinstall`:

```
09:21:41 WARNING: valid service node 56 of class server_dvs from \
/bootroot0/etc/opt/cray/sdb/node_classes \
is not in CLEinstall.conf and is not the default class service.
09:21:41 INFO: There is one WARNING about a discrepancy between CLEinstall.conf \
and /bootroot0/etc/opt/cray/sdb/node_classes.
09:21:41 FATAL: Correct the node_class settings discrepancy between CLEinstall.conf \
and /bootroot0/etc/opt/cray/sdb/node_classes and restart CLEinstall
```

`CLEinstall` may resolve some issues after you indicate that you want to proceed; for example, disk devices are already mounted, boot image file or links already exist, HSS daemons are stopped on the SMW.

Caution: Some problems can be resolved only through manual intervention via another terminal window or by rebooting the SMW; for example, a process is using a mounted disk partition, preventing `CLEinstall` from unmounting the partition.

4. After you have resolved all issues, complete these steps.

   a. Monitor the debug output. Create another terminal window and invoke the `tail` command by using the path and PID value displayed in step 2 (after `CLEinstall` was invoked).

   ```
   smw~:# tail -f /var/adm/cray/logs/CLEinstall.debug.pid
   ```

   b. In the `CLEinstall` console window, locate the following warning and prompt and type **`y`**.

```
*** Preparing to INSTALL software on system set label system_set_label.
This will DESTROY any existing data on disk partitions in this system set.
Do you wish to proceed? [n]
```

5. The `CLEinstall` program now installs the release software. This process takes a long time; `CLEinstall` runs from 30 minutes to 1½ hours, depending on your specific system configuration.

   Monitor the output to ensure that your installation is proceeding without error.

   Note: Several error messages from the `tar` command are displayed as the persistent `/var` is updated for each service node. You may safely ignore these messages.

6. Confirm that the `CLEinstall` program has completed successfully.

   On completion, the `CLEinstall` program generates a list of suggested commands to be run as the next steps in the upgrade process. These commands are customized, based on the variables in the `CLEinstall.conf` and `sysset.conf` files, and include runtime variables such as PID numbers in filenames.

Complete the installation and configuration of your Cray system by using both the commands that the `CLEinstall` program provides and the information in the remaining sections of this chapter and in Chapter 6, Configuring Lustre File Systems on page 119.

As you complete these procedures, you can cut and paste the suggested commands from the output window or from the window created in step 4, which tailed the debug file. The log files created in `/var/adm/cray/logs` for `CLEinstall.stdout.`*pid* and `CLEinstall.debug.`*pid* also contain the suggested commands.

5.5 Creating Boot Images

The Cray CNL compute nodes and Cray service nodes use RAM disks for booting. Service nodes and CNL compute nodes use the same `initramfs` format and workspace environment. This space is created in `/opt/xt-images/`*machine-xtrelease-partition/nodetype*, where *machine* is the Cray hostname, *xtrelease* is the build level for the CLE release, *partition* describes either the full machine or a system partition, and *nodetype* is either `compute` or `service`.

Caution: Existing files in `/opt/xt-images/templates/default` are copied into the new bootimage work space. In most cases, you can use the older version of the files with your upgraded system. However, some file content may have changed with the new release; you must verify that site-specific modifications are compatible. For example, you can use existing copies of `/etc/hosts`, `/etc/passwd` and `/etc/modprobe.conf`, but if you changed `/init` for the template, the site-modified version that is copied and used for CLE 4.0 may cause a boot failure.

Follow the procedures in this section to prepare the work space in `/opt/xt-images`. For more information about configuring boot images for service and compute nodes, see the `xtclone`(8) and `xtpackage`(8) man pages.

Procedure 7. Modifying boot image parameters for service nodes

The `CLEinstall` program modifies a `parameters` file for service nodes located in the `bootimage_temp_directory`.

Example 6. `parameters` files for service nodes

If the `CLEinstall` pid is *21822* and `bootimage_temp_directory` is `/home/crayadm/boot`, the modified `parameters` file is:

```
/home/crayadm/boot/bootimage.default.21822/SNL0/parameters
```

The default `parameters` file is:

```
/home/crayadm/boot/bootimage.default.21822/SNL0/parameters-snl
```

If you had one, your old parameters file is:

```
/home/crayadm/boot/bootimage.24107/SNL0/parameters
```

Example 7. Sample `parameters` file

This file contains a single line, but has been formatted here for readability.

```
earlyprintk=ttyS0,115200
load_ramdisk=1
ramdisk_size=80000
console=ttyS0,115200n8
bootnodeip=10.131.255.254
bootproto=ipog
bootpath=/rr/current
rootfs=nfs-shared
root=/dev/disk/by-id/scsi-3600a0b800051215e000003a84b4ad820-part1
pci=lastbus=3
oops=panic
bootifnetmask=255.252.0.0
elevator=noop
ippob1=10
ippob2=128
iommu=off
pci=noacpi
bad_page=panic
sdbnodeip=10.131.255.253
```

1. Inspect the modified parameters file. In most cases, you do not need to change this file.

```
smw:~# cat /home/crayadm/boot/bootimage.default.21822/SNL0/parameters
```

2. If you need to change one or more of the variables that are not set from `CLEinstall.conf` or `sysset.conf`, edit the parameters file.

```
smw:~# vi /home/crayadm/boot/bootimage.default.21822/SNL0/parameters
```

Procedure 8. Preparing compute and service node boot images

Invoke the `shell_bootimage_`*LABEL*`.sh` script to prepare boot images for the system set with the specified *LABEL*. This script uses `xtclone` and `xtpackage` to prepare the work space in `/opt/xt-images`.

`shell_bootimage_`*LABEL*`.sh` accepts the following options:

`-v` Run in verbose mode.

`-T` Do not update the `default` template link.

`-h` Display help message.

`-c` Create and set the boot image for the next boot. The default is to display `xtbootimg` and `xtcli` commands that will generate the boot image. Use the `-c` option to invoke these commands automatically.

`-b` *bootimage*

Specify *bootimage* as the boot image disk device or file name. The default *bootimage* is determined by using values for the system set *LABEL* when `CLEinstall` was run. Use this option to override the default and manage multiple boot images.

`-C` *coldstart_dir*

Specify *coldstart_dir* as the path to the HSS coldstart applets directory. The default is `/opt/hss-coldstart+gemini/default/xt` for Cray XE systems. Use this option to override the default. For more information, see the `xtbounce`(8) man page.

Optionally, this script includes `CNL_*` parameters that you can use to modify the CNL boot image configuration you defined in `CLEinstall.conf`. Edit the script and set the associated parameter to **`y`** to load an optional RPM or change the `/tmp` configuration.

1. Run `shell_bootimage_`*LABEL*`.sh`, where *LABEL* is the system set label specified in `/etc/sysset.conf` for this boot image. For example, if the system set label is *BLUE*, log on to the SMW as `root` and type:

   ```
   smw:~# /var/opt/cray/install/shell_bootimage_BLUE.sh
   ```

   On completion, the script displays the `xtbootimg` and `xtcli` commands that are required to build and set the boot image for the next boot. If you specified the `-c` option, the script invokes these commands automatically and you should skip the remaining steps in this procedure.

2. Create a unified boot image for compute and service nodes by using the `xtbootimg` command suggested by the `shell_bootimage_`*LABEL*`.sh` script.

   In the following example, replace *bootimage* with the *mountpoint* for `BOOT_IMAGE0` in the system set that is defined in `/etc/sysset.conf`. Set *bootimage* to either a raw device; for example `/raw0` or a file name; for example `/bootimagedir/bootimage.new`.

   Caution: If *bootimage* is a file, verify that the file exists in the same path on both the SMW and the boot root.

   Type this command:

   ```
   smw:~# xtbootimg \
   -L /opt/xt-images/xthostname-XT_version/compute/CNL0.load \
   -L /opt/xt-images/xthostname-XT_version/service/SNL0.load \
   -C /opt/hss-coldstart+gemini/default/xt \
   -c bootimage
   ```

   a. At the prompt '`Do you want to overwrite`', type **`y`** to overwrite the existing boot image file.

   b. If *bootimage* is a file, copy the boot image file from the SMW to the same directory on the boot root. If *bootimage* is a raw device, skip this step. For example, if the *bootimage* file is `/bootimagedir/bootimage.new` and `bootroot_dir` is set to `/bootroot0`, type the following command:

```
smw:~# cp -p /bootimagedir/bootimage.new /bootroot0/bootimagedir/bootimage.new
```

3. Set the boot image for the next system boot by using the suggested `xtcli` command.

   The `shell_bootimage_`*LABEL*`.sh` program suggests an `xtcli` command to set the boot image based on the value of `BOOT_IMAGE0` for the system set that you are using. The `-i` *bootimage* option specifies the path to the boot image and is either a raw device; for example, `/raw0` or `/raw1`, or a file such as `/bootimagedir/bootimage.new`.

   Caution: The next boot, anywhere on the system, uses the boot image you set here.

   a. Display the currently active boot image. Record the output of this command.

      If the partition variable in `CLEinstall.conf` is `s0`, type:

      ```
      smw:~# xtcli boot_cfg show
      ```

      Or

      If the partition variable in `CLEinstall.conf` is a partition value such as `p0`, `p1`, and so on, type:

      ```
      smw:~# xtcli part_cfg show pN
      ```

b. Invoke `xtcli` with the `update` option to set the default boot configuration used by the boot manager.

If the partition variable in `CLEinstall.conf` is `s0`, type the following command to select the boot image to be used for the entire system.

```
smw:~# xtcli boot_cfg update -i bootimage
```

Or

If the partition variable in `CLEinstall.conf` is a partition value such as `p0`, `p1`, and so on, type the following command to select the boot image to be used for the designated partition.

```
smw:~# xtcli part_cfg update pN -i bootimage
```

Procedure 9. Enabling boot-node failover

Optional: Boot-node failover is an optional CLE feature.

If you have configured boot-node failover for the first time, follow these steps. If you did not configure boot-node failover, skip this procedure.

To enable boot-node failover, you must set `bootnode_failover` parameters in the `CLEinstall.conf` file before you run the `CLEinstall` program. For more information, see Configuring Boot-node Failover on page 35.

In this example, the primary boot node is *c0-0c0s0n1* (`node_boot_primary=1`) and the backup or alternate boot node is *c0-0c1s1n1* (`node_boot_alternate=61`).

Tip: Use the `rtr --system-map` command to translate between NIDs and physical ID names.

1. As `crayadm` on the SMW, halt the primary and alternate boot nodes.

   Warning: Verify that your system is shut down before you invoke the `xtcli halt` command.

   ```
   crayadm@smw:~> xtcli halt c0-0c0s0n1,c0-0c1s1n1
   ```

2. Specify the primary and backup boot nodes in the boot configuration.

   If the partition variable in `CLEinstall.conf` is `s0`, type the following command to select the boot node for the entire system.

   ```
   crayadm@smw:~> xtcli boot_cfg update -b c0-0c0s0n1,c0-0c1s1n1
   ```

   Or

   If the partition variable in `CLEinstall.conf` is a partition value such as `p0`, `p1`, and so on, type the following command to select the boot node for the designated partition.

   ```
   crayadm@smw:~> xtcli part_cfg update pN -b c0-0c0s0n1,c0-0c1s1n1
   ```

3. To use boot-node failover, you must enable the STONITH capability on the blade or module of the primary boot node. Use the `xtdaemonconfig` command to determine the current STONITH setting.

   ```
   crayadm@smw:~> xtdaemonconfig c0-0c0s0 | grep stonith
   c0-0c0s0: stonith=false
   crayadm@smw:~>
   ```

   Note: If you have a partitioned system, invoke `xtdaemonconfig` with the `--partition p`*n* option.

   ⚠ **Caution:** STONITH is a per blade setting, not a per node setting. You must ensure that your primary boot node is not assigned to a blade that hosts services with conflicting STONITH requirements, such as Lustre.

4. To enable STONITH on your primary boot node blade, type the following command:

   ```
   crayadm@smw:~> xtdaemonconfig c0-0c0s0 stonith=true
   c0-0c0s0: stonith=true
   crayadm@smw:~> xtcli halt c0-0c0s0n1,c0-0c1s1n1
   crayadm@smw:~> xtcli boot_cfg update -b c0-0c0s0n1,c0-0c1s1n1
   crayadm@smw:~>
   ```

   Note: If you have a partitioned system, invoke `xtdaemonconfig` with the `--partition p`*n* option.

Procedure 10. Enabling SDB node failover

Optional: SDB node failover is an optional CLE feature.

If you have configured SDB node failover for the first time, follow these steps. If you did not configure SDB node failover, skip this procedure.

Note: In addition to this procedure, refer to Procedure 42 on page 116 **after** you have completed the remaining configuration steps and have booted and tested your system.

To enable SDB node failover, you must set `sdbnode_failover` parameters in the `CLEinstall.conf` file before you run the `CLEinstall` program. For more information, see Configuring SDB Node Failover on page 37.

In this example, the primary SDB node is *c0-0c0s2n1* (`node_sdb_primary=5`) and the backup or alternate SDB node is *c0-0c1s3n1* (`node_sdb_alternate=57`).

Tip: Use the `rtr --system-map` command to translate between NIDs and physical ID names.

1. Invoke `xtdaemonconfig` to determine the current STONITH setting on the blade or module of the primary SDB node. For example:

   ```
   crayadm@smw:~> xtdaemonconfig c0-0c0s0 | grep stonith
   c0-0c0s0: stonith=false
   crayadm@smw:~>
   ```

   Note: If you have a partitioned system, invoke `xtdaemonconfig` with the `--partition p`*n* option.

   Caution: STONITH is a per blade setting, not a per node setting. You must ensure that your primary SDB node is not assigned to a blade that hosts services with conflicting STONITH requirements, such as Lustre.

2. Enable STONITH on your primary SDB node. For example:

   ```
   crayadm@smw:~> xtdaemonconfig c0-0c0s2 stonith=true
   c0-0c0s2: stonith=true
   The expected response was received.
   crayadm@smw:~>
   ```

   Note: If you have a partitioned system, invoke `xtdaemonconfig` with the `--partition p`*n* option.

3. Specify the primary and backup SDB nodes in the boot configuration.

   For example, if the partition variable in `CLEinstall.conf` is `s0`, type the following command to select the primary and backup SDB nodes.

   ```
   crayadm@smw:~> xtcli halt c0-0c0s2n1,c0-0c1s3n1
   crayadm@smw:~> xtcli boot_cfg update -d c0-0c0s2n1,c0-0c1s3n1
   ```

   Or

   If the partition variable in `CLEinstall.conf` is a partition value such as `p0`, `p1`, and so on, type the following command:

   ```
   crayadm@smw:~> xtcli part_cfg update pN -d c0-0c0s2n1,c0-0c1s3n1
   ```

Procedure 11. Running post-`CLEinstall` commands

1. Unmount and eject the release software DVD from the SMW DVD drive if it is still loaded.

   ```
   smw:~# umount /media/cdrom
   smw:~# eject
   ```

2. Run the `shell_post_install.sh` script on the SMW to unmount the boot root and shared root file systems and perform other cleanup as needed.

```
smw:~# /var/opt/cray/install/shell_post_install.sh /bootroot0 /sharedroot0
```

> **Warning:** Exercise care when you mount and unmount file systems. If you mount a file system on the SMW and boot node simultaneously, you may corrupt the file system.

3. Confirm that the `shell_post_install.sh` script successfully unmounted the boot root and shared root file systems.

   If a file system does not unmount successfully, the script displays information about open files and associated processes (by using the `lsof` and `fuser` commands). Attempt to terminate processes with open files and if necessary, reboot the SMW to resolve the problem.

5.6 Booting and Logging on to the Boot Node

At this point, you are ready to boot and log on to the boot node on the Cray system.

> **Note:** The remaining procedures require dedicated Cray system time.

> **Warning:** Before you start this procedure, verify that the boot root and shared root file systems are no longer mounted on the SMW. If you mount the file systems on the SMW and boot node simultaneously, you may corrupt the file systems.

Procedure 12. Booting and logging on to the boot node

1. Log on to the SMW as `crayadm`.

2. In a shell window, use the `xtbootsys` command to boot the boot node.

   ```
   crayadm@smw:~> xtbootsys
   ```

3. The `xtbootsys` command prompts you with a series of questions. Cray recommends that you answer yes by typing **Y** to each question.

The session pauses at:

```
0) boot bootnode ...
1) boot sdb ...
2) boot compute ...
3) boot service ...
4) boot all (not supported) ...
5) boot all_comp (not supported by cpio archive) ...
10) boot bootnode and wait ...
11) boot sdb and wait ...
12) boot compute and wait ...
13) boot service and wait ...
14) boot all and wait (not supported) ...
15) boot all_comp and wait (not supported by cpio archive) ...
17) boot using a loadfile ...
18) turn console flood control off ...
19) turn console flood control on ...
20) spawn off the network link recovery daemon (xtnlrd)...
q) quit.
Enter your boot choice:
```

Choose option **`10`** to boot the boot node and wait.

To confirm your selection, press the `Enter` key or type **`Y`** to each question.

```
Do you want to boot the boot node ? [Yn] Y
Do you want to send the ec_boot event ('no' means to only load memory) ? [Yn] Y
```

4. When the boot node has finished booting you are prompted to `Enter your boot choice` again. **Do not close** the `xtbootsys` terminal session and **do not respond** to the prompt at this point in the installation process.

   You will use this window later to boot the SDB, service, and compute nodes. If you lose the window, restart `xtbootsys` by using the `-s` option. For more information, see the `xtbootsys`(8) man page.

5. Open another shell window on the SMW and use the ssh command to log on to the boot node.

```
smw:~# ssh root@boot
boot:~ #
```

Note: The first time that the root and crayadm accounts on the SMW use the ssh command to log on to the boot node, the host key for the boot node is cached. For an initial installation of the boot root on an SMW that has had prior use, it is possible to get the following error message. If this situation is not corrected for the crayadm account, an attempt to boot by using xtbootsys and a boot automation file may result in a partial failure.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
87:65:39:4e:76:de:43:f0:47:f1:d3:12:ac:b7:b0:92.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending key in /root/.ssh/known_hosts:4
RSA host key for boot has changed and you have requested strict checking.
Host key verification failed.
```

If the preceding warning appears when you use the ssh command to log on to the boot node, edit /root/.ssh/known_hosts and /home/crayadm/.ssh/known_hosts to remove the previous SSH host key for "boot." The hostname and the IP address are first on the line for the SSH host keys in the known_hosts file. The warning lists the line that contains the ssh mismatched host key. In the previous example, the known_hosts file has an error in line 4.

5.7 Changing the Default System Passwords

For security, you should change the root and crayadm passwords at this time.

Procedure 13. Changing passwords on boot and service nodes

1. To change the passwords on the boot node, type the following commands. You are prompted to enter and confirm new root and administrative passwords.

   ```
   boot:~ # passwd root
   boot:~ # passwd crayadm
   ```

2. To change the passwords on the service nodes, type the following commands. Again, you are prompted to enter and confirm new root and administrative passwords.

   > **Note:** Because the SDB has not been started, use the `-x /etc/opt/cray/sdb/node_classes` option with `xtopview` to specify node/class relationships.

   ```
   boot:~ # xtopview -x /etc/opt/cray/sdb/node_classes
   default/:/ # passwd root
   default/:/ # passwd crayadm
   default/:/ # exit
   ```

   > **Note:** You are prompted to type **`c`** and enter a brief comment describing the changes you made. To complete your comment, type **`Ctrl-d`** or a period on a line by itself. Do this each time you exit `xtopview` to log a record of revisions into an RCS system.

Procedure 14. Changing the `root` password on compute nodes

Update the root password in the shadow password file on the SMW.

> **Note:** To make these changes for a system partition, rather than for the entire system, replace `/opt/xt-images/templates` with `/opt/xt-images/templates-p`*N*, where *N* is the partition number.

1. Copy the master shadow password file to the template directory.

   ```
   smw:~# cp /opt/xt-images/master/default/etc/shadow \
   /opt/xt-images/templates/default/etc/shadow
   ```

2. Edit the `shadow` file to include a new encrypted password for root.

   ```
   smw:~# vi /opt/xt-images/templates/default/etc/shadow
   ```

   > **Note:** To use the root password you created in Procedure 13 on page 72, copy the second field of the root entry in the `/etc/shadow` file on the boot node.

3. Update the boot image to include these changes; follow the steps in Procedure 8 on page 64.

   > **Note:** Several procedures in this chapter include a similar step. You can defer this step and update the boot image **once** before you Finish Booting the System on page 109.

5.8 Modifying SSH Keys for Compute Nodes

> **Optional:** The steps in this section are not required for installation of CLE software.

The `dropbear` RPM is provided with the CLE release. Using `dropbear` SSH software, you can supply and generate site-specific SSH keys for compute nodes in place of the keys provided by Cray.

Procedure 15. Using `dropbear` to generate site-specific SSH keys

Follow these steps to replace the RSA and DSA/DSS keys provided by the `CLEinstall` program.

1. Load the `dropbear` module.

   ```
   crayadm@smw:~> module load dropbear
   ```

2. Create a directory for the new keys on the SMW.

   ```
   crayadm@smw:~> mkdir dropbear_ssh_keys
   crayadm@smw:~> cd dropbear_ssh_keys
   ```

3. To generate a `dropbear` compatible RSA key, type:

```
crayadm@smw:~/dropbear_ssh_keys> dropbearkey -t rsa -f ssh_host_rsa_key.db
Will output 1024 bit rsa secret key to 'ssh_host_rsa_key.db'
Generating key, this may take a while...
Public key portion is:
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgwCQ9ohUgsrrBw5GNk7w2H5RcaBGajmUv8XN6fxg/YqrsL4t5
CIkNghI3DQDxoiuC/ZVIJCtdwZLQJe708eiZee/tg5y2g8JIb3stg+ol/9BLPDLMeX24FBhCweUpfGCO6Jfm4
Xg4wjKJIGrcmtDJAYoCRj0h9IrdDXXjpS7eI4M9XYZ
Fingerprint: md5 00:9f:8e:65:43:6d:7c:c3:f9:16:48:7d:d0:dd:40:b7
crayadm@smw:~/dropbear_ssh_keys>
```

   To generate a `dropbear` compatible DSS key, type:

```
crayadm@smw:~/dropbear_ssh_keys> dropbearkey -t dss -f ssh_host_dss_key.db
Will output 1024 bit dss secret key to 'ssh_host_dss_key.db'
Generating key, this may take a while...
Public key portion is:
ssh-dss AAAAB3NzaC1kc3MAAACBAMEkThlE9N8iczLpfg0wUtuPtPcpIs7Y4KbG3Wg1T4CAEXDnfMCKSyuCy
21TMAvVGCvYd80zPtL04yc1eUtD5RqEKy0h8jSBs0huEvhaJGHx9FzKfGhWi1ZOVX5vG3R+UCOXG+71wZp3LU
yOcv/U+GWhalTWpUDaRU81MPRLW7rnAAAAFQCEqnqW61bouSORQ52d+MRiwp27MwAAAIEAho69yAfGrNzxEI/
kjyDE5IaxjJpIBF262N9UsxleTX6F65OjNoL84fcKqlSL6NV5XJ5O00SKgTuVZjpXO913q9SEhkcI0Zy0vRQ8
H5x3osZZ+Bq20QWof+CtWTqCoWN2xvne0NtET4lg81qCt/KGRq1tY6WG+a01yrvunzQuafQAAACASXvs8h8AA
EK+3TEDj57rBRV4pz5JqWSlUaZStSQ2wJ3Oy1pIJIhKfqGWytv/nSoWnr8YbQbvH9k1BsyQU8sOc5IJyCFu7+
Exom1yrxq/oirfeSgg6xC2rodcs+jH/K8EKoVtTak3/jHQeZWijRok4xDxwHdZ7e3l2HgYbZLmA5Y=
Fingerprint: md5 cd:a0:0b:41:40:79:f9:4a:dd:f9:9b:71:3f:59:54:8b
crayadm@smw:~/dropbear_ssh_keys>
```

4. As `root`, copy the SSH keys to the boot image template.

   > **Note:** To make these changes for a system partition, rather than for the entire system, replace `/opt/xt-images/templates` with `/opt/xt-images/templates-p`*N*, where *N* is the partition number.

   ```
   crayadm@smw:~/dropbear_ssh_keys> su root
   ```

For the RSA key:

```
smw:/home/crayadm/dropbear_ssh_keys # cp -p ssh_host_rsa_key.db \
/opt/xt-images/templates/default/etc/ssh/ssh_host_rsa_key
```

For the DSA/DSS key:

```
smw:/home/crayadm/dropbear_ssh_keys # cp -p ssh_host_dss_key.db \
/opt/xt-images/templates/default/etc/ssh/ssh_host_dss_key
```

5. Update the boot image to include these changes; follow the steps in Procedure 8 on page 64.

5.9 Modifying /etc/hosts

Optional: The steps in this section are not required for installation of CLE software.

You may require site-specifc changes to the /etc/hosts to meet your networking requirements. Follow this procedure to edit the hosts file for the boot node, service nodes, and CNL compute nodes.

Procedure 16. Modifying the /etc/hosts file

Important: CLEinstall modifies Cray system entries in /etc/hosts each time you update or upgrade your CLE software. For additional information, see Maintaining Node Class Settings and Hostname Aliases on page 27.

1. Edit the /etc/hosts file on the boot node and make site-specific changes.

```
boot:~ # vi /etc/hosts
```

2. Copy the edited file to the shared root by using xtopview in the default view.

   Note: Because the SDB has not been started, use the -x /etc/opt/cray/sdb/node_classes option with xtopview to specify node/class relationships.

```
boot:~ # cp -p /etc/hosts /rr/current/software
boot:~ # xtopview -x /etc/opt/cray/sdb/node_classes
default/:/ # cp -p /software/hosts /etc/hosts
default/:/ # exit
```

3. Make your site-specific changes to the `/opt/xt-images/templates/default/etc/hosts` file on the SMW.

   Note: To make these changes for a system partition, rather than for the entire system, replace `/opt/xt-images/templates` with `/opt/xt-images/templates-p`*N*, where *N* is the partition number.

   ```
   smw:~# vi /opt/xt-images/templates/default/etc/hosts
   ```

   a. Update the boot image to include these changes; follow the steps in Procedure 8 on page 64.

      Note: You can defer this step and update the boot image **once** before you Finish Booting the System on page 109.

5.10 Configuring Login Nodes and Other Network Nodes

Follow these procedures to configure network access and other login class specific information for the login nodes. These procedures also apply to other service nodes, such as network nodes or nodes acting as RSIP servers, which use the shared root and have Ethernet interfaces.

Caution: Login nodes and other service nodes do not have swap space. If users consume too many resources, service nodes can run out of memory. When an out of memory condition occurs, the node can become unstable or may crash. System administrators should take steps to manage system resources on service nodes. For example, resource limits can be configured by using the `pam_limits` module and the `/etc/security/limits.conf` file. For more information, see the `limits.conf`(5) man page.

Procedure 17. Configuring network settings for all login and network nodes

The login and network nodes are the portals between the customer's network and the Cray system. Configure basic network information for each login and network node.

1. Use `xtopview` to access each node by either integer node ID or physical ID. For example, to access node 8, type the following:

   ```
   boot:~ # xtopview -x /etc/opt/cray/sdb/node_classes \
   -m "network settings" -n 8
   ```

   Note: Because the SDB has not been started, use the `-x /etc/opt/cray/sdb/node_classes` option to specify node/class relationships.

   Tip: Optionally specify the `-m` option with a brief site-specific comment describing the changes you are making. If this option is specified, `xtopview` does not prompt for comments. This option is suggested when multiple files are changed within a single `xtopview` session.

2. Create and specialize the /etc/sysconfig/network/ifcfg-eth0 file for the node.

   ```
   node/8:/ # touch /etc/sysconfig/network/ifcfg-eth0
   node/8:/ # xtspec -n 8 /etc/sysconfig/network/ifcfg-eth0
   ```

   For a description of specialization, see the shared_root(5) man page.

3. Edit /etc/sysconfig/network/ifcfg-eth0 for the node to include site dependent information. For example, if the site uses static IP addresses, the file might contain the following:

   ```
   BOOTPROTO='static'
   STARTMODE='auto'
   IPADDR='172.30.12.71/24'
   ```

   Where "/24" on the IPADDR line is the PREFIXLEN, or number of bits that form the network address; alternatively, you may specify PREFIXLEN on its own line, although any value appended to IPADDR takes precedence. Previous CLE release's ifcfg configuration files also may contain the parameter DEVICE. Refer to the ifcfg(5) man page for more information. Repeat step 1 through step 3 for each login and network node you have configured.

4. (Optional) Specialize and edit /etc/hosts.allow and /etc/hosts.deny to configure host access control. The information in these files is site dependent. For information about the contents of these files see the hosts_access(5) man page. For example, to specialize these files for a single login node, type the following commands.

   ```
   node/8:/ # xtspec -n 8 /etc/hosts.allow
   node/8:/ # vi /etc/hosts.allow
   node/8:/ # xtspec -n 8 /etc/hosts.deny
   node/8:/ # vi /etc/hosts.deny
   ```

5. (Optional) Specialize and edit /etc/HOSTNAME. This file is given a value from the node_class_login_hostname variable in CLEinstall.conf, but may be modified for site-specific considerations.

   ```
   node/8:/ # xtspec -n 8 /etc/HOSTNAME
   node/8:/ # vi /etc/HOSTNAME
   ```

6. Exit from xtopview.

   ```
   node/8:/ # exit
   ```

Procedure 18. Configuring class-specific login and network node information

After you have configured the basic network information, follow this procedure to configure class-specific information for login or network nodes. The following examples configure the `login` class. Repeat the steps in this procedure for each site-defined class that contains network or RSIP server nodes.

1. Use the `xtopview` command to access login nodes by class.

   Note: Because the SDB has not been started, use the `-x /etc/opt/cray/sdb/node_classes` option with `xtopview` to specify node/class relationships.

   ```
   boot:~ # xtopview -x /etc/opt/cray/sdb/node_classes \
   -m "login class settings" -c login
   ```

2. Specialize and modify the network configuration file using information from the SMW. Make changes consistent with your network.

   ```
   class/login:/ # xtspec -c login /etc/sysconfig/network/config
   class/login:/ # vi /etc/sysconfig/network/config
   ```

   Modify the following variables:

   ```
   NETCONFIG_DNS_STATIC_SEARCHLIST=""
   NETCONFIG_DNS_STATIC_SERVERS=""
   NETCONFIG_DNS_FORWARDER=""
   ```

3. Create and specialize the network routes file for the login class. Use information from the SMW to make changes consistent with your network.

   ```
   class/login:/ # touch /etc/sysconfig/network/routes
   class/login:/ # xtspec -c login /etc/sysconfig/network/routes
   class/login:/ # vi /etc/sysconfig/network/routes
   ```

4. Create and specialize the `/etc/resolv.conf` file for the login class. Invoke the `netconfig` command to populate the file.

   ```
   class/login:/ # touch /etc/resolv.conf
   class/login:/ # xtspec -c login /etc/resolv.conf
   class/login:/ # netconfig update -f
   ```

5. Specialize and edit `/etc/pam.d/sshd` for the login class. To configure PAM to prevent users with key-based authentication from logging in when `/etc/nologin` exists, add the following line from the example below:

   Important: This must be the **first** line in the file.

   ```
   class/login:/ # xtspec -c login /etc/pam.d/sshd
   class/login:/ # vi /etc/pam.d/sshd
   account required pam_nologin.so
   ```

6. (Optional) The following services are turned off by default. Depending on your site policies and requirements, you may need to turn them on by using the `chkconfig` command.

   `cron` (see Configuring `cron` Services on page 96)
   `boot.localnet`
   `flexlm`
   `postfix`

   Note: If `postfix` is configured and run on a service node, change the following setting in `/etc/sysconfig/mail` from:

   ```
   MAIL_CREATE_CONFIG="yes"
   ```

   to

   ```
   MAIL_CREATE_CONFIG="no"
   ```

   Doing so prevents the `master.cf` and `main.cf` `postfix` configuration files from being recreated during software updates or fixes.

7. Exit `xtopview`.

   ```
   class/login:/ # exit
   ```

5.11 Configuring OpenFabrics InfiniBand

InfiniBand is an efficient, low-cost transport between Cray's internal High-speed Network (HSN) and external I/O devices. It can replace or complement Gigabit Ethernet (GigE). OFED/IB driver support is included in the CLE release; OFED and InfiniBand RPMs are installed by default.

You must have the appropriate Host Channel Adapter (HCA) installed for OFED/IB to function correctly. Configure OFED/IB for the particular functionality that you desire. InfiniBand can be configured as follows:

- IB connects service nodes on a Cray system, acting as Lustre servers, to external storage devices. These nodes are commonly referred to as LNET servers. Follow Procedure 19 on page 79.
- IB can provide IP connectivity between devices on the fabric. To configure IP over InfiniBand (IPoIB), follow Procedure 20 on page 80.
- If you are using devices that require the SCSI RDMA Protocol (SRP), follow Procedure 21 on page 81.

  Important: Direct-attached InfiniBand file systems require SRP; Lustre file systems external to the Cray system do not require SRP.

For additional information, see *Managing System Software for Cray XE and Cray XK Systems*.

Procedure 19. Configuring InfiniBand on service nodes

InfiniBand includes the core OpenFabrics stack and a number of upper layer protocols (ULPs) that use this stack. Configure InfiniBand by modifying `/etc/sysconfig/infiniband` for each IB service node.

1. Use the `xtopview` command to access service nodes with IB HCAs.

   For example, if the service nodes with IB HCAs are part of a node class called `lnet`, type the following command:

   ```
   boot:~ # xtopview -x /etc/opt/cray/sdb/node_classes -c lnet
   ```

   Or

   Access each IB service node by specifying either a node ID or physical ID. For example, access node 27 by typing the following:

   ```
   boot:~ # xtopview -x /etc/opt/cray/sdb/node_classes -n 27
   ```

2. Specialize the `/etc/sysconfig/infiniband` file:

   ```
   node/27:/ # xtspec -n 27 /etc/sysconfig/infiniband
   ```

3. Add IB services to the service nodes by using standard Linux mechanisms, such as executing the `chkconfig` command while in the `xtopview` utility or executing `/etc/init.d/openibd start | stop | restart` (which starts or stops the InfiniBand services immediately). Use the `chkconfig` command to ensure that IB services are started at system boot.

   ```
   node/27:/ # chkconfig --force openibd on
   ```

4. While in the `xtopview` session, edit `/etc/sysconfig/infiniband` and make these changes.

   ```
   node/27:/ # vi /etc/sysconfig/infiniband
   ```

   a. By default, IB services do not start at system boot. Change the `ONBOOT` parameter to **`yes`** to enable IB services at boot.

      ```
      ONBOOT=yes
      ```

   b. By default at boot time, the Internet Protocol over InfiniBand (IPoIB) driver loads on all nodes where IB services are configured. Verify that the value for `IPOIB_LOAD` is set to **`yes`** to enable IPoIB services.

      ```
      IPOIB_LOAD=yes
      ```

      Important: LNET routers use IPoIB to select the paths that data will travel via RDMA.

c. The SCSI RDMA Protocol (SRP) driver loads by default on all nodes where IB services are configured to load at boot time. If your Cray system needs SRP services, verify that the value for SRP_LOAD is set to **yes** to enable SRP.

```
SRP_LOAD=yes
```

> **Important:** Direct-attached InfiniBand file systems require SRP; Lustre file systems external to the Cray system do not require SRP.

5. Exit xtopview.

```
node/27:/ # exit
boot:~ #
```

> **Note:** You are prompted to type **c** and enter a brief comment describing the changes you made. To complete your comment, type **Ctrl-d** or a period on a line by itself. Do this each time you exit xtopview to log a record of revisions into an RCS system.

6. Proper IPoIB operation requires additional configuration. See Procedure 20 on page 80.

Procedure 20. Configuring IP Over InfiniBand (IPoIB) on Cray systems

1. Use xtopview to access each service node with an IB HCA by specifying either a node ID or physical ID. For example, to access node 27, type the following:

```
boot:~ # xtopview -x /etc/opt/cray/sdb/node_classes -n 27
```

2. Specialize the /etc/sysconfig/network/ifcfg-ib0 file.

```
node/27:/ # xtspec -n 27 /etc/sysconfig/network/ifcfg-ib0
```

3. Modify the site-specific /etc/sysconfig/network/ifcfg-ib0 file on each service node with an IB HCA.

```
node/27:/ # vi /etc/sysconfig/network/ifcfg-ib0
```

For example, to use static IP address, *172.16.0.1*, change the BOOTPROTO line in the file.

```
BOOTPROTO='static'
```

Add the following lines to the file.

```
IPADDR='172.16.0.1'
NETMASK='255.128.0.0'
```

To configure the interface at system boot, change the STARTMODE line in the file.

```
STARTMODE='onboot'
```

4. (Optional) If you would like to configure IPoIB on both ports of a two port IB HCA, repeat step 2 and step 3 for `/etc/sysconfig/network/ifcfg-ib1`. Use a unique IP address from separate networks for each port.

Procedure 21. Configuring and enabling SRP on Cray Systems

1. Use the `xtopview` command to access service nodes with IB HCAs.

   For example, if the service nodes with IB HCAs are part of a node class called `ib`, type the following command:

   ```
   boot:~ # xtopview -x /etc/opt/cray/sdb/node_classes -c ib
   ```

2. Edit `/etc/sysconfig/infiniband`

   ```
   ib/:/ # vi /etc/sysconfig/infiniband
   ```

   and change the value of `SRP_DAEMON_ENABLE` to `yes`:

   ```
   SRP_DAEMON_ENABLE=yes
   ```

3. Edit `srp_daemon.conf` to increase the maximum sector size for SRP.

   ```
   ib/:/ # vi /etc/srp_daemon.conf
   ```

   ```
   a       max_sect=8192
   ```

4. Edit `/etc/modprobe.conf.local` to increase the maximum number of gather-scatter entries per SRP I/O transaction.

   ```
   ib/:/ # vi /etc/modprobe.conf.local
   ```

   ```
   options ib_srp srp_sg_tablesize=255
   ```

5. Exit from `xtopview`.

   ```
   ib/:/ # exit
   boot:~ #
   ```

5.12 Completing Configuration of the SDB

This procedure proceeds uninterrupted from the previous procedure. At this time, you have three shell sessions open: one running a `tail` command, one running an `xtbootsys` session, and one logged on to the boot node as `root`. The `xtbootsys` session should be paused at the following prompt:

```
0) boot bootnode ...
1) boot sdb ...
2) boot compute ...
3) boot service ...
4) boot all (not supported) ...
5) boot all_comp (not supported by cpio archive) ...
10) boot bootnode and wait ...
11) boot sdb and wait ...
12) boot compute and wait ...
13) boot service and wait ...
14) boot all and wait (not supported) ...
15) boot all_comp and wait (not supported by cpio archive) ...
17) boot using a loadfile ...
18) turn console flood control off ...
19) turn console flood control on ...
20) spawn off the network link recovery daemon (xtnlrd)...
q) quit.
Enter your boot choice:
```

Procedure 22. Booting and configuring the SDB node

Continue in the terminal session for `xtbootsys` that you started in Procedure 12 on page 69.

1. Select option **11** to boot the SDB and wait.

   To confirm your selection, press the `Enter` key or type **Y** to each question.

```
Do you want to boot the sdb node ? [Yn] Y
Do you want to send the ec_boot event ('no' means to only load memory) ? [Yn] Y
```

   Note: Until you start the SDB MySQL database in step 5, a number of error messages similar to "`cpadb_mysql_connect:  sdb connection failure`" may display in your console log file. You may safely ignore these messages.

2. When the SDB node has finished booting, you are prompted to `Enter your boot choice` again. **Do not close** the `xtbootsys` terminal session. You will use it later to boot the remaining service nodes.

3. In another terminal session, run the `shell_bootnode_first.sh` script on the boot node. This script creates `ssh` keys for root on the boot node and copies the `shell_sdbnode_first.sh` script to the SDB.

   ```
   boot:~ # /var/opt/cray/install/shell_bootnode_first.sh
   ```

   If `ssh_generate_root_sshkeys=yes` is set in the `CLEinstall.conf` file, this step generates `ssh` DSA and RSA keys for the root account on the boot node.

   The `shell_bootnode_first.sh` script copies the `shell_sdbnode_first.sh` script to the SDB node for the next step.

   You are prompted to choose the *passphrase* for the `ssh` keys of the root account on the boot node. Use the default file name and specify a null *passphrase*. A null *passphrase* is required to allow passwordless `pdsh` access from the boot node to the other service nodes. This functionality is required by several CLE system utilities, for example `xtshutdown` and Lustre startup.

   Press the `Enter` key to choose the defaults and a null *passphrase*.

   For the DSA key:

   ```
   Generating public/private dsa key pair.
   Enter file in which to save the key (/root/.ssh/id_dsa):
   Created directory '/root/.ssh'.
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /root/.ssh/id_dsa.
   Your public key has been saved in /root/.ssh/id_dsa.pub.
   ```

   For the RSA key:

   ```
   Generating public/private rsa key pair.
   Enter file in which to save the key (/root/.ssh/id_rsa):
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /root/.ssh/id_rsa.
   Your public key has been saved in /root/.ssh/id_rsa.pub.
   ```

4. Run the script to install and configure the MySQL database.

   Log on to the SDB node. Run the `shell_sdbnode_first.sh` script, and then log off the SDB node.

   Press the `Enter` key to enter a null password when you are prompted for a password.

   ```
   boot:~ # ssh root@sdb
   sdb:~ # /tmp/shell_sdbnode_first.sh
   Script output
   ...
   Enter password:

   Script output
   ...
   sdb:~ # exit
   boot:~ #
   ```

5. On the boot node, run the `shell_bootnode_second.sh` script. This script starts the SDB database and completes SDB configuration.

   ```
   boot:~ # /var/opt/cray/install/shell_bootnode_second.sh
   ```

Procedure 23. Changing default MySQL passwords on the SDB

For security, you should change the default passwords for MySQL database accounts.

1. If you have not set a site-specific MySQL password for root, type the following commands. Press the `Enter` key when prompted for a password.

```
boot:~ # ssh root@sdb
sdb:~ # mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.0.64-enterprise MySQL Enterprise Server (Commercial)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> set password for 'root'@'localhost' = password('newpassword');
Query OK, 0 rows affected (0.00 sec)
mysql> set password for 'root'@'%' = password('newpassword');
Query OK, 0 rows affected (0.00 sec)
mysql> set password for 'root'@'sdb' = password('newpassword');
Query OK, 0 rows affected (0.00 sec)
```

2. (Optional) Set a site-specific password for other MySQL database accounts.

   a. To change the password for the `sys_mgmt` account, type the following MySQL command. You must also update `.my.cnf` in step 4.

```
mysql> set password for 'sys_mgmt'@'%' = password('newpassword');
Query OK, 0 rows affected (0.00 sec)
```

   b. To change the password for the `basic` account, type the following MySQL command. You must also update `/etc/opt/cray/MySQL/my.cnf` in step 5.

   Note: Changing the password for the `basic` MySQL user account will not provide any added security. This read-only account is used by the system to allow all users to run `xtprocadmin`, `xtnodestat`, and other commands that require SDB access.

```
mysql> set password for 'basic'@'%' = password('newpassword');
Query OK, 0 rows affected (0.00 sec)
```

   c. To change the password for the `mazama` account, type the following MySQL commands. You must also update `/etc/sysconfig/mazama` in step 6.

```
mysql> set password for 'mazama'@'%' = password('newpassword');
Query OK, 0 rows affected (0.00 sec)
mysql> set password for 'mazama'@'localhost' = password('newpassword');
Query OK, 0 rows affected (0.00 sec)
```

Note: When making changes to the MySQL database, your connection may time out; however, it is automatically reconnected. If this happens, you will see messages similar to the following. These messages may be ignored.

```
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id:    21127
Current database: *** NONE ***

Query OK, 0 rows affected (0.00 sec)
```

3. Exit from MySQL and the SDB.

```
mysql> exit
Bye
sdb:~ # exit
boot:~ #
```

4. (Optional) If you set a site-specific password for `sys_mgmt` in step 2, update the `.my.cnf` file for root with the new password.

   a. Edit `.my.cnf` for root on the boot node.

   ```
   boot:~ # cd ~root
   boot:~ # vi .my.cnf
   [client]
   user=sys_mgmt
   password=newpassword
   ```

   b. Edit `.my.cnf` for root in the shared root.

   ```
   boot:~ # xtopview
   default/:/ # vi /root/.my.cnf
   [client]
   user=sys_mgmt
   password=newpassword
   default/:/ # exit
   boot:~ #
   ```

5. (Optional) If you set a site-specific password for `basic` in step 2, update the `/etc/opt/cray/MySQL/my.cnf` file with the new password.

   a. Edit `/etc/opt/cray/MySQL/my.cnf` on the boot node.

   ```
   boot:~ # vi /etc/opt/cray/MySQL/my.cnf
   # The following options will be passed to all MySQL clients
   [client]
   user=basic
   password=newpassword
   ```

   b. Edit `/etc/opt/cray/MySQL/my.cnf` in the shared root.

   ```
   boot:~ # xtopview
   default/:/ # vi /etc/opt/cray/MySQL/my.cnf
   # The following options will be passed to all MySQL clients
   [client]
   user=basic
   password=newpassword
   default/:/ # exit
   boot:~ #
   ```

6. (Optional) If you set a site-specific password for `mazama` in step 2, update the `/etc/sysconfig/mazama` file with the new password. In addition, update the `mazama` MySQL account on the SMW to match.

   a. Edit `/etc/sysconfig/mazama` on the boot node.

   ```
   boot:~ # vi /etc/sysconfig/mazama
   ## Type:                string
   ## Default:             mazama
   ## Config:              ""
   #
   # Default password for mazama user in the mazama database
   #
   passwd=newpassword
   ```

   b. Edit `/etc/sysconfig/mazama` in the shared root.

   ```
   boot:~ # xtopview
   default/:/ # vi /etc/sysconfig/mazama
   ## Type:                string
   ## Default:             mazama
   ## Config:              ""
   #
   # Default password for mazama user in the mazama database
   #
   passwd=newpassword
   default/:/ # exit
   boot:~ #
   ```

   c. To change the password for the MySQL accounts on the SMW, type the following MySQL commands.

```
boot:~ # exit
smw:~# mysql -u root -p
mysql> set password for 'mazama'@'%' = password('newpassword');
mysql> set password for 'mazama'@'localhost' = password('newpassword');
mysql> set password for 'mazama'@'smw' = password('newpassword');
mysql> exit
```

   d. Update `/etc/sysconfig/mazama` on the SMW.

   ```
   smw:~# vi /etc/sysconfig/mazama
   ## Type:                string
   ## Default:             mazama
   ## Config:              ""
   #
   # Default password for mazama user in the mazama database
   #
   passwd=newpassword
   ```

   Make the following additional change, unless you are using a remote MySQL server for CMS logs.

   ```
   ## Type:                string
   ## Default:             mazama
   # Default password for mazama user in the mazama Log database
   #
   log_passwd=newpassword
   ```

Procedure 24. Adding node-specific services

After the SDB is running, configure the services that run on specific nodes or classes of nodes. The list of supported Cray system services is located in `/etc/opt/cray/sdb/serv_cmd`. An example is provided in `/opt/cray/sdb/default/etc/serv_cmd.example`. You may add other optional services by using this procedure.

1. Invoke the `xtservconfig` command on the boot node to show the available services.

   ```
   boot:~ # xtservconfig avail
   ```

   You can also use this command to show the services already assigned.

   ```
   boot:~ # xtservconfig list
   ```

   > **Note:** Do not add `SYSLOG` or `CRON` by using `xtservconfig`. Follow Configuring `cron` Services on page 96 and Configuring System Message Logs on page 108 to configure these services later in the installation process.

2. Assign services to nodes as appropriate. For example, type the following command:

   ```
   boot:~ # xtservconfig -a add service-name
   ```

   Use the `-c` *class* option to assign a service to a class of nodes, or the `-n` *nid* option to assign a service to a specific node.

5.13 Configuring Additional Services

Boot the login nodes and all other service nodes and configure the following services. Do this **before** you boot the compute nodes. Note that some of these services are optional.

5.13.1 Booting the Remaining Service Nodes

Boot the login nodes and all other service nodes **before** you boot the compute nodes. After your system is booted, you can reboot it as needed.

Procedure 25. Booting the remaining service nodes

Continue in the terminal session for `xtbootsys` that you started in Procedure 12 on page 69.

1. Select option **`13`** to boot the service nodes and wait.

2. You are prompted to enter a list of the service nodes to be booted.

   To display service node information, type one of the following commands. Use the `s0` option for the entire system or the `p`*n* option for a partition; for example, partition 2.

   ```
   smw:~# xtcli status s0 | grep service
   smw:~# xtcli status p2 | grep service
   ```

3. Type **`all_serv`** to boot all remaining service nodes.

   Or

   Use the information displayed in step 2, and type a list of service nodes to be booted. For example:

   ```
   c0-0c0s0n0 c0-0c0s2n0 c0-0c0s4n0 c0-0c0s4n1
   ```

   Or

   If you have a partitioned system, type the partition value such as `p0`, `p1`, and so on, to boot the remaining service nodes in the partition.

4. To confirm your selection, press the `Enter` key or type **`Y`** to each question.

```
Do you want to boot service c0-0c0s0n0,c0-0c0s2n0,c0-0c0s4n0,c0-0c0s4n1 ? [Yn] Y
Do you want to send the ec_boot event ('no' means to only load memory) ? [Yn] Y
```

5. After the specified service nodes are booted, you are prompted to `Enter your boot choice` again. **Do not close** the `xtbootsys` terminal session. You will use it later to boot the compute nodes.

5.13.2 Booting Compute Node Root Servers

Optional: Dynamic shared objects and libraries (DSL) and the compute node root runtime environment (CNRTE) are optional.

If you have configured DVS servers for DSL (`DSL_nodes`) by using compute nodes as service nodes without using the `xtcli` command (*manually repurposed*), follow Procedure 26 on page 90 to manually boot the compute node root servers.

Note: The capability to manually repurpose compute nodes is deprecated and may not be supported in future releases. To configure compute nodes for CNRTE, Cray recommends that you follow Repurposing Compute Nodes as Service Nodes on page 56. If you followed Procedure 5 on page 57, the repurposed nodes should already be booted as service nodes.

Procedure 26. Booting compute nodes as compute node root servers

At this point in the installation process, all service nodes are booted. However, any compute nodes that have been manually repurposed as DSL servers must also be booted with the `SNL0` service node image.

1. Return to the terminal session for `xtbootsys` that you started in Procedure 12 on page 69.

2. Select **17** from the `xtbootsys` menu to boot by using a loadfile.

```
Enter your boot choice: 17
Enter a boot type string (or nothing to do nothing): SNL0
Enter a boot type option (or nothing to do nothing): compute
Enter a component list (or nothing to do nothing): c0-0c0s7n0,c0-0c0s7n1
Enter 'any' to wait for any console output,
   or 'linux' to wait for a linux style boot,
   or 'mtk', 'threadstorm', 'ts', or 'xmt' to wait for a MTK style boot,
   or anything else (or nothing) to not wait at all: linux
Enter an alternative CPIO archive name (or nothing):
Do you want to send the ec_boot event ('no' means to only load memory) ? [Yn]
```

3. After the specified nodes are booted, you are prompted to `Enter your boot choice` again. **Do not close** the `xtbootsys` terminal session. You will use it later to boot the remaining compute nodes.

Configuring a Boot Automation File on page 114 describes procedures to automatically start the compute note root servers following a reboot of your Cray system.

5.13.3 Populating the known_hosts File

If `ssh_generate_root_sshkeys=yes` is set in the `CLEinstall.conf` file, run the `shell_ssh.sh` script on the boot node to populate the `known_hosts` file for the root account by using the `ssh` host keys from the service nodes. Type the following command.

```
boot:~ # /var/opt/cray/install/shell_ssh.sh
```

This is done to verify that `xtshutdown` can contact all service nodes and initiate shutdown procedures by using `pdsh`.

5.13.4 Configuring Lustre File Systems

If you plan to configure Lustre file systems, follow the procedures in Chapter 6, Configuring Lustre File Systems on page 119 and then return here to continue the installation.

5.13.5 Creating New Login Accounts

> **Optional:** The steps in this section are not required for installation of CLE software.

To add additional accounts to the shared root for login nodes, use the `groupadd` and `useradd` commands from the default `xtopview` session. For example:

```
boot:~ # xtopview -m "adding user accounts" -c login
class/login:/ # groupadd options
class/login:/ # useradd options
class/login:/ # exit
boot:~ #
```

The `groupadd` and `useradd` commands create group and shadow password entries for new users. However, these commands do not create home directories; you must create home directories manually. Set the ownership and permissions to enable users to access their home directories. For information about managing user accounts on service nodes, see *Managing System Software for Cray XE and Cray XK Systems*.

5.13.6 Configuring the Login Failure Logging PAM

> **Optional:** Although the steps in this section are not required for installation of CLE software, Cray recommends that you configure login failure logging on all service nodes.

The `cray_pam` pluggable authentication module (PAM), when configured, provides information to the user at login time about any failed login attempts since their last successful login. To configure this feature, edit the following files on the boot node and then on the service nodes by using the shared root file system:

```
/etc/pam.d/common-auth
/etc/pam.d/common-account
/etc/pam.d/common-session
```

The default location of the `pam_tally` counter file is `/var/log/faillog`. The default location for the `cray_pam` temporary directory is `/var/opt/cray/faillog`. Change these defaults by editing `/etc/opt/cray/pam/faillog.conf` and by using the `file=` option for each `pam_tally` and `cray_pam` entry. You can find an example `faillog.conf` file in `/opt/cray/pam/`*xtrelease-xtversion*`/etc`.

Procedure 27. Configuring `cray_pam` to log failed login attempts

1. Edit the `/etc/pam.d/common-auth`, `/etc/pam.d/common-account`, and `/etc/pam.d/common-session` files on the boot node.

   Note: In these examples, the `pam_faillog.so` and `pam.tally.so` entries can include an optional `file=`*/path/to/pam_tally/counter/file* argument to specify an alternate location for the tally file.

   Example 8 shows these files after they have been modified to report failed login using an alternate location for the tally file.

   a. Edit the `/etc/pam.d/common-auth` file and add the following lines as the first and last entries:

   ```
   boot:~ # vi /etc/pam.d/common-auth
   auth required pam_faillog.so [file=alternatepath]  (as the FIRST entry)
   auth required pam_tally.so [file=alternatepath]  (as the LAST entry)
   ```

   Your modified `/etc/pam.d/common-auth` file should look like this:

```
#%PAM-1.0
#
# This file is autogenerated by pam-config. All changes
# will be overwritten.
#
# Authentication-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.). The default is to use the
# traditional Unix authentication mechanisms.
#
auth    required        pam_faillog.so
auth    required        pam_env.so
auth    required        pam_unix2.so
auth    required        pam_tally.so
```

   b. Edit the `/etc/pam.d/common-account` file and add the following line as the last entry:

   ```
   boot:~ # vi /etc/pam.d/common-account
   account required pam_tally.so [file=alternatepath]
   ```

Your modified `/etc/pam.d/common-account` file should look like this:

```
#%PAM-1.0
#
# This file is autogenerated by pam-config. All changes
# will be overwritten.
#
# Account-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authorization modules that define
# the central access policy for use on the system.  The default is to
# only deny service to users whose accounts are expired.
#
account required        pam_unix2.so
account required        pam_tally.so
```

c. Edit the `/etc/pam.d/common-session` file and add the following line as the last entry:

```
boot:~ # vi /etc/pam.d/common-session
session optional pam_faillog.so [file=alternatepath]
```

Your modified `/etc/pam.d/common-session` file should look like this:

```
#%PAM-1.0
#
# This file is autogenerated by pam-config. All changes
# will be overwritten.
#
# Session-related modules common to all services

#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define tasks to be performed
# at the start and end of sessions of *any* kind (both interactive and
# non-interactive).  The default is pam_unix2.
#
session required        pam_limits.so
session required        pam_unix2.so
session optional        pam_umask.so
session optional        pam_faillog.so
```

2. Copy the edited files to the shared root by using `xtopview` in the default view.

```
boot:~ # cp -p /etc/pam.d/common-auth /rr/current/software
boot:~ # cp -p /etc/pam.d/common-account /rr/current/software
boot:~ # cp -p /etc/pam.d/common-session /rr/current/software
boot:~ # xtopview -m "configure login failure logging PAM"
default/:/ # cp -p /software/common-auth /etc/pam.d/common-auth
default/:/ # cp -p /software/common-account /etc/pam.d/common-account
default/:/ # cp -p /software/common-session /etc/pam.d/common-session
```

3. Exit `xtopview`.

```
default/:/ # exit
boot:~ #
```

Example 8. Modified PAM configuration files configured to report failed login by using an alternate path

If you configure `pam_tally` to save tally information in an alternate location by using the `file=` option, each entry for `cray_pam` must also include the `file=` option to specify the alternate location.

Your modified `/etc/pam.d/common-auth` file should look like this:

```
#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.).  The default is to use the
# traditional Unix authentication mechanisms.
#
auth    required        pam_faillog.so file=/ufs/logs/tally.log
auth    required        pam_env.so
auth    required        pam_unix2.so
auth    required        pam_tally.so file=/ufs/logs/tally.log
```

Your modified `/etc/pam.d/common-account` file should look like this:

```
#
# /etc/pam.d/common-account - authorization settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authorization modules that define
# the central access policy for use on the system.  The default is to
# only deny service to users whose accounts are expired.
#
account required        pam_unix2.so
account required        pam_tally.so file=/ufs/logs/tally.log
```

Your modified `/etc/pam.d/common-session` file should look like this:

```
#
# /etc/pam.d/common-session - session-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define tasks to be performed
# at the start and end of sessions of *any* kind (both interactive and
# non-interactive).  The default is pam_unix2.
#
session required        pam_limits.so
session required        pam_unix2.so
session optional        pam_umask.so
session optional        pam_faillog.so file=/ufs/logs/tally.log
```

5.13.7 Configuring the Load Balancer

Optional: The load balancer service is optional on systems that run CLE.

The load balancer can distribute user logins to multiple login nodes, allowing users to connect by using the same Cray host name, for example *xthostname*.

Two main components are required to implement the load balancer, the `lbnamed` service (on the SMW and Cray login nodes) and the site-specific domain name service (DNS).

When an external system tries to resolve *xthostname*, a query is sent to the site-specific DNS. The DNS server recognizes *xthostname* as being part of the Cray domain and shuttles the request to `lbnamed` on the SMW. The `lbnamed` service returns the IP address of the least-loaded login node to the requesting client. The client connects to the Cray system login node by using that IP address.

The CLE software installation process installs `lbnamed` in `/opt/cray-xt-lbnamed` on the SMW and in `/opt/cray/lbcd` on all service nodes. Configure `lbnamed` by using the `lbnamed.conf` and `poller.conf` configuration files on the SMW. For more information about configuring `lbnamed`, see the `lbnamed.conf`(5) man page.

Procedure 28. Configuring `lbnamed` on the SMW

1. (Optional) If site-specific versions of `/etc/opt/cray-xt-lbnamed/lbnamed.conf` and `/etc/opt/cray-xt-lbnamed/poller.conf` do not already exist, copy the provided example files to these locations.

```
smw:~ # cd /etc/opt/cray-xt-lbnamed/
smw:/etc/opt/cray-xt-lbnamed/ # cp -p lbnamed.conf.example lbnamed.conf
smw:/etc/opt/cray-xt-lbnamed/ # cp -p poller.conf.example poller.conf
```

2. Edit the `lbnamed.conf` file on the SMW to define the `lbnamed` host name, domain name, and polling frequency.

   ```
   smw:/etc/opt/cray-xt-lbnamed/ # vi lbnamed.conf
   ```

   For example, if `lbnamed` is running on the host name `smw.mysite.com`, set the login node domain to the same domain specified for the `$hostname`. The Cray system *xthostname* is resolved within the domain specified as `$login_node_domain`.

   ```
   $poller_sleep = 30;
   $hostname = "mysite-lb";
   $lbnamed_domain = "smw.mysite.com";
   $login_node_domain = "mysite.com";
   $hostmaster = "rootmail.mysite.com";
   ```

3. Edit the `poller.conf` file on the SMW to configure the login node names.

   ```
   smw:/etc/opt/cray-xt-lbnamed/ # vi poller.conf
   #
   # groups
   # -------------------------
   # login     mycray1-mycray3

   mycray1 1 login
   mycray2 1 login
   mycray3 1 login
   ```

   Note: Because `lbnamed` runs on the SMW, `eth0` on the SMW must be connected to the same network from which users log on to the login nodes. Do not put the SMW on the public network.

Procedure 29. Installing the load balancer on an external "white box" server

Optional: Install `lbnamed` on an external "white box" server as an alternative to installing it on the SMW. **Cray does not test or support this configuration.**

A "white box" server is any workstation or server that supports the `lbnamed` service.

1. Shut down and disable `lbnamed`.

   ```
   smw:~# /etc/init.d/lbnamed stop
   smw:~# chkconfig lbnamed off
   ```

2. Locate the `cray-xt-lbnamed` RPM on the `Cray CLE 4.0.UP`*nn* `Software` media and install this RPM on the "white box." Do **not** install the `lbcd` RPM.

3. Follow the instructions in the `lbnamed.conf`(5) man page to configure `lbnamed`, taking care to substitute the name of the external server wherever `SMW` is indicated, then enable the service.

5.13.8 Configuring `cron` Services

Optional: Configuring `cron` services is optional on CLE systems.

The `cron` daemon is disabled, by default, on the shared root file system and the boot root. It is enabled, by default, on the SMW. Use standard Linux procedures to enable `cron` on the boot root, following Procedure 30 on page 97.

On the shared root, how you configure `cron` for CLE depends on whether you have set up persistent `/var`. If you have persistent `/var` follow Procedure 31 on page 97; if you have not set up persistent `/var`, follow Procedure 32 on page 98.

The `/etc/cron.*` directories include a large number of `cron` scripts. During new system installations and any updates or upgrades, the `CLEinstall` program disables execute permissions on these scripts and you must manually enable any scripts you want to use.

Procedure 30. Configuring cron for the SMW and the boot node

Note: By default, the cron daemon on the SMW is enabled and this procedure is required only on the boot node.

1. Log on to the target node as root and determine the current configuration status for cron.

   On the on the SMW:

   ```
   smw:~# chkconfig cron
   cron on
   ```

   On the boot node:

   ```
   boot:~ # chkconfig cron
   cron off
   ```

2. Use the chkconfig command to configure the cron daemon to start. For example, to enable cron on the boot node, type the following command:

   ```
   boot:~ # chkconfig --force cron on
   ```

The cron scripts shipped with the Cray customized version of SLES are located under /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, and /etc/cron.monthly. The system administrator can enable these scripts by using the chkconfig command. However, if you do not have a persistent /var, Cray recommends that you follow Procedure 32.

Procedure 31. Configuring cron for the shared root with persistent /var

Use this procedure for service nodes by using the shared root on systems that are set up with a persistent /var file system.

1. Invoke the chkconfig command in the default view to enable the cron daemon.

   ```
   boot:~ # xtopview -m "configuring cron"
   default/:/ # chkconfig --force cron on
   ```

2. Examine the /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, and /etc/cron.monthly directories and change the file access permissions to enable or disable distributed cron scripts to meet your needs. To enable a script, invoke chmod ug+x to make the script executable. By default, CLEinstall removes the execute permission bit to disable all distributed cron scripts.

   ⚠ **Caution:** Some distributed scripts impact performance negatively on a CLE system. To ensure that all scripts are disabled, type the following:

   ```
   default/:/ # find /etc/cron.hourly /etc/cron.daily \
   /etc/cron.weekly /etc/cron.monthly \
   -type f -follow -exec chmod ugo-x {} \;
   ```

3. Exit xtopview.

   ```
   default/:/ # exit
   boot:~ #
   ```

Procedure 32. Configuring cron for the shared root without persistent /var

Because CLE has a shared root, the standard cron initialization script /etc/init.d/cron activates the cron daemon on all service nodes. Therefore, the cron daemon is disabled by default and you must turn it on with the xtservconfig command to specify which nodes you want the daemon to run on.

1. Edit the /etc/group file in the default view to add users who do not have root permission to the "trusted" group. The operating system requires that all cron users who do not have root permission be in the "trusted" group.

   ```
   boot:~ # xtopview
   default/:/ # vi /etc/group
   default/:/ # exit
   ```

2. Create a /var/spool/cron directory in the /ufs file system on the ufs node which is shared among all the nodes of class login.

   ```
   boot:~ # ssh root@ufs
   ufs:~# mkdir /ufs/cron
   ufs:~# cp -a /var/spool/cron /ufs
   ufs:~# exit
   ```

3. Designate a single login node on which to run the scripts in this directory. Configure this node to start cron with the xtservconfig command rather than the /etc/init.d/cron script. This enables users, including root, to submit cron jobs from any node of class login. These jobs are executed only on the specified login node.

   a. Create or edit the following entry in the /etc/sysconfig/xt file in the shared root file system in the default view.

      ```
      boot:~ # xtopview
      default/:/ # vi /etc/sysconfig/xt
      CRON_SPOOL_BASE_DIR=/ufs/cron
      default/:/ # exit
      ```

   b. Start an xtopview shell to access all login nodes by class and configure the spool directory to be shared among all nodes of class login.

      ```
      boot:~ # xtopview -c login
      class/login:/ #
      ```

c. Edit the /etc/init.d/boot.xt-local file to add the following lines.

```
class/login:/ # vi /etc/init.d/boot.xt-local
MYCLASS_NID=`rca-helper -i`
MYCLASS=`xtnce $MYCLASS_NID | awk -F: '{ print $2 }'  | tr -d [:space:]`
CRONSPOOL=`xtgetconfig CRON_SPOOL_BASE_DIR`
if [ "$MYCLASS" = "login" -a -n "$CRONSPOOL" ];then
  mv /var/spool/cron /var/spool/cron.$$
  ln -sf $CRONSPOOL /var/spool/cron
fi
```

d. Examine the /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, and /etc/cron.monthly directories and change the file access permissions to enable or disable distributed cron scripts to meet your needs. To enable a script, invoke chmod ug+x to make the file executable. By default, CLEinstall removes the execute permission bit to disable all distributed cron scripts.

Caution: Some distributed scripts impact performance negatively on a CLE system. To ensure that all scripts are disabled, type the following:

```
class/login:/ # find /etc/cron.hourly /etc/cron.daily \
/etc/cron.weekly /etc/cron.monthly \
-type f -follow -exec chmod ugo-x {} \;
```

e. Exit from the login class view.

```
class/login:/ # exit
boot:~ #
```

f. Use the xtservconfig command to enable the cron service on a single login node; in this example, node 8.

```
boot:~ # xtopview -n 8
node/8:/ # xtservconfig -n 8 add CRON
node/8:/ # exit
```

The cron configuration becomes active on the next reboot. For more information, see the xtservconfig(8) man page.

5.13.9 Configuring IP Routes

Optional: Configuring IP routes for compute nodes is not required on a CLE system.

The /etc/routes file can be edited in the CNL template image to provide route entries for compute nodes. This provides a mechanism for administrators to configure routing access from CNL compute nodes to login and network nodes, using external IP destinations without having to traverse RSIP tunnels. Careful consideration should be given before using this capability for general purpose routing.

Procedure 33. Configuring IP routes

A new `/etc/routes` file is created in the CNL images; it is examined during startup. Non-comment, non-blank lines are passed to the `route add` command. The empty template file contains comments describing the syntax.

> **Note:** To make these changes for a system partition, rather than for the entire system, replace `/opt/xt-images/templates/default` with `/opt/xt-images/templates/default-p`*N*, where *N* is the partition number.

1. Edit `/opt/xt-images/templates/default/etc/routes` and make site-specific changes.
2. Update the boot image to include these changes; follow the steps in Procedure 8 on page 64.

   > **Note:** You can defer this step and update the boot image **once** before you Finish Booting the System on page 109.

5.13.10 Configuring Cray DVS

> **Optional:** Cray Data Virtualization Service (Cray DVS) is an optional software package.

Cray Data Virtualization Service (Cray DVS) is a parallel I/O forwarding service that enables the transparent use of multiple file systems in CLE systems with close-to-open coherence, much like NFS. DVS provides compute nodes transparent access to external file systems mounted on the service I/O nodes via the Cray high-speed network. Administration of Cray DVS is very similar to configuring and mounting any Linux file system.

> ⚠ **Caution:** DVS service nodes must be dedicated and not share service I/O nodes with other services (e.g., SDB, Lustre, login or MOM nodes).

Cray DVS supports multiple POSIX-compliant, VFS-based file systems. Two supported modes, *serial* and *cluster parallel*, provide functionality for different implementations of existing file systems. Since site conditions and systems requirements differ, please contact your Cray service representative about projecting your preferred file system over DVS.

Because DVS on Cray systems uses the Lustre networking driver (LNET) the following line must be in `/etc/modprobe.conf.local` on DVS servers and in `/etc/modprobe.conf` on DVS clients in those systems:

```
options lnet networks=gni
```

If you configured your system to use a different network identifier than the default (`gni` on Cray systems) you should use that identifier instead. For example, if your LND is configured to use `gni1` as a name, insert the following lines in `modprobe.conf`:

```
options dvsipc_lnet lnd_name=gni1
options lnet networks=gni1
```

Setting the `lnd_name` option for `dvsipc_lnet` is needed so DVS looks for the alternative network identifier since it assumes `gni` as the default. Setting the `networks` option for `lnet` is generally needed when the LNET network type identifier is different.

For more information, see *Introduction to Cray Data Virtualization Service* (S–0005) and the `dvs`(5) man page.

Procedure 34. Configuring the system to mount DVS file systems

After Cray DVS software has been successfully installed on both the service and compute nodes, you can `mount` a file system on the compute nodes that require access to the network file system that is mounted on DVS server nodes. When a client mounts the file system, all of the necessary information is specified on the `mount` command.

> **Note:** The node that is projecting the file system needs to mount it. Therefore, if the file system is external to the Cray, the DVS server must have external connectivity.
>
> At least one DVS server must be active when DVS is loaded on the client nodes to ensure that all DVS mount points are configured to enable higher-level software, such as the compute node root runtime environment (CNRTE), to function properly.

The following example configures a DVS server at `c0-0c0s4n3` (node 23 on a Cray XE system) to project the file system that is served via NFS from *nfs_serverhostname*. For more information about Cray DVS mount options, see the `dvs`(5) man page.

> **Note:** To make these changes for a system partition, rather than for the entire system, replace `/opt/xt-images/templates` with `/opt/xt-images/templates-p`*N*, where *N* is the partition number.

1. Enter `xtopview` with the node view for your DVS server and create the `/dvs-shared` directory that you will be projecting */nfs_mount* from.

   ```
   boot:~ # xtopview -n 23
   node/23:/ # mkdir /dvs-shared
   ```

2. Specialize the `/etc/fstab` file for the server and add a DVS entry to it.

   ```
   node/23:/ # xtspec -n 23 /etc/fstab
   node/23:/ # vi /etc/fstab
   nfs_serverhostname:/nfs_mount /dvs-shared nfs tcp,rw 0 0
   node/23:/ # exit
   ```

3. Log into the DVS server and mount the file system:

   ```
   boot:~ # ssh nid00023
   nid00023:/ # mount /dvs-shared
   nid00023:/ # exit
   ```

4. Create mount point directories in the compute image for each DVS mount in the `/etc/fstab` file. For example, type the following command from the SMW:

   ```
   smw:~ # mkdir -p /opt/xt-images/templates/default/dvs
   ```

5. (Optional) Create any symbolic links that are used in the compute node images. For example:

   ```
   smw:~ # cd /opt/xt-images/templates/default
   smw:/opt/xt-images/templates/default # ln -s dvs link_name
   ```

6. To allow the compute nodes to mount their DVS partitions, add an entry in the `/etc/fstab` file in the compute image and add entries to support the DVS mode you are configuring.

   ```
   smw:~# vi /opt/xt-images/templates/default/etc/fstab
   ```

   For serial mode, add a line similar to the following example which mounts `/dvs-shared` from DVS server `c0-0c0s4n3` to `/dvs` on the client node.

   ```
   /dvs-shared /dvs dvs path=/dvs,nodename=c0-0c0s4n3
   ```

   For cluster parallel mode, add a line similar to the following example which mounts `/dvs-shared` from multiple DVS servers to `/dvs` on the client node. Setting `maxnodes` to 1 indicates that each file hashes to only one server from the list.

```
/dvs-shared /dvs dvs path=/dvs,nodename=c0-0c2s1n0:c0-0c2s1n3:c0-0c2s2n0,maxnodes=1
```

   For stripe parallel mode, add a line similar to the following example which mounts `/dvs-shared` from the DVS servers to `/dvs` on the client nodes. Specifying a value for `maxnodes` greater than 1 or removing it altogether makes this stripe parallel access mode.

```
/dvs-shared /dvs dvs path=/dvs,nodename=c0-0c2s1n0:c0-0c2s1n3:c0-0c2s2n0
```

For atomic stripe parallel mode, add a line similar to the following example which mounts `/dvs-shared` from the DVS servers to `/dvs` on the client nodes. Specifying atomic makes this atomic stripe parallel mode as opposed to stripe parallel mode.

```
/dvs-shared /dvs dvs path=/dvs,nodename=c0-0c2s1n0:c0-0c2s1n3:c0-0c2s2n0,atomic
```

For loadbalance mode, add a line similar to the following example to project `/dvs-shared` from multiple DVS servers to `/dvs` on the client node. The `ro` and `cache` settings specify to mount the data read-only and cache it on the compute node. The `attrcache_timeout` option specifies the amount of time in seconds that file attributes remain valid on a DVS client after they are fetched from a DVS server. Failover is automatically enabled and does not have to be specified.

```
/dvs-shared /dvs dvs path=/dvs,nodename=c0-0c2s1n0:c0-0c2s1n3:c0-0c2s2n0,\
loadbalance,cache,ro,attrcache_timeout=14400
```

7. If you set `CNL_dvs=`**`yes`** in `CLEinstall.conf` before you ran the `CLEinstall` program, update the boot image (by preparing a new compute and service node boot image.)

   Otherwise, you must first edit the `/var/opt/cray/install/shell_bootimage_`*`LABEL`*`.sh` script and set `CNL_DVS=`**`y`** and then update the boot image.

   Note: It is important to keep `CLEinstall.conf` consistent with changes made to your configuration in order to avoid unexpected changes during upgrades or updates. Remember to set `CNL_dvs` equal to **`yes`** in `CLEinstall.conf`.

Note: You can defer updating the boot image and update it **once** before you Finish Booting the System on page 109.

5.13.11 Completing CCM Configuration

Optional: Cluster Compatibility Mode (CCM) is an optional CLE feature.

If you have configured CCM for your system by setting the CCM-specific parameters in `CLEinstall.conf`, complete the CCM configuration for CLE.

Note: After your CLE software installation is complete, you must complete the CCM batch system configuration. For more information, see Configuring Cluster Compatibility Mode (CCM) on page 42.

Procedure 35. Using DVS to mount home directories on the compute nodes for CCM

For each DVS server node you have configured, mount the path to the user home directories. Typically, these will be provided from a location external to the Cray system.

1. Specialize and add a line to the `/etc/fstab` file on the DVS server by using `xtopview` in the node view. For example, if your DVS server is `c0-0c0s2n3` (node 27 on a Cray XE system), type the following:

```
boot:~ # xtopview -m "mounting home dirs" -n 27
node/27:/ # xtspec -n 27 /etc/fstab
node/27:/ # vi /etc/fstab
nfs_home_server:/home        /home    nfs     tcp,rw  0 0
node/27:/ # exit
```

2. Log into each DVS server and mount the file system:

```
boot:~ # ssh nid00027
nid00027:~ # mount /home
nid00027:~ # exit
```

3. To allow the compute nodes to mount their DVS partitions, add an entry in the `/etc/fstab` file in the compute image for each DVS file system. For example:

```
smw:~ # vi /opt/xt-images/templates/default/etc/fstab
/home /home dvs path=/home,nodename=c0-0c0s2n3
```

4. For each DVS mount in the `/etc/fstab` file, create a mount point in the compute image.

```
smw:~ # mkdir -p /opt/xt-images/templates/default/home
```

5. Update the boot image to include these changes; follow the steps in Procedure 8 on page 64.

   Note: You can defer this step and update the boot image **once** before you finish booting the system.

Procedure 36. Modifying CCM and Platform-MPI system configurations

1. If you wish to enable additional features such as debugging and Linux NIS (Network Information Service) support, edit the CCM configuration file by using `xtopview` in the default view.

   ```
   boot:~ # xtopview -m "configuring ccm.conf"
   default/:/ # vi /etc/opt/cray/ccm/ccm.conf
   ```

   If you wish to configure additional CCM debugging, set `CCM_DEBUG=yes`.

   If you wish to enable NIS support, set `CCM_ENABLENIS=yes`.

2. (Optional) You may have a site configuration where the paths for the `qstat` command is not at a standard location. Change the values in the configuration file for `CRAY_QSTAT_PATH` and `CRAY_BATCH_VAR` accordingly for your site configuration.

3. Save and close `ccm.conf`.

4. Exit `xtopview`.

   ```
   default/:/ # exit
   boot:~ #
   ```

   > **Important:** If your applications will use Platform-MPI (previously known as *HP-MPI*), Cray recommends that users populate their `~/.hpmpi.conf` (or `~/.pmpi.conf`) file with these values.

```
MPI_REMSH=ssh
MPIRUN_OPTIONS="-cpu_bind=MAP_CPU:0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,\
22,23,24,25,26,27,28,29,30,31"
```

5.13.12 Completing Cray Audit Configuration

> **Optional:** Cray Audit is an optional CLE feature.

If you have included security auditing RPMs in the compute node boot image by setting `CNL_audit=yes` in `CLEinstall.conf`, complete the Cray Audit configuration.

Procedure 37. Configuring Cray Audit

By default, Linux security auditing is disabled and Cray Audit extensions are enabled. Follow these steps to define your site-specific auditing rules and enable standard Linux auditing.

> **Note:** To make these changes for a system partition, rather than for the entire system, replace `/opt/xt-images/templates` with `/opt/xt-images/templates-p`*N*, where *N* is the partition number. Also, replace `/opt/xt-images/`*xthostname*`-`*XT_version* with `/opt/xt-images/`*xthostname*`-`*XT_version*`-p`*N*.

1. Follow these steps to edit the auditing configuration files in the compute node image and enable auditing on CNL compute nodes.

   a. Copy the `auditd.conf` and `audit.rules` configuration files to the template directory so that modifications are retained when new boot images are created in the future.

```
smw:~# cp /opt/xt-images/xthostname-XT_version/compute/etc/auditd.conf \
/opt/xt-images/templates/default/etc/auditd.conf
smw:~# cp /opt/xt-images/xthostname-XT_version/compute/etc/audit.rules \
/opt/xt-images/templates/default/etc/audit.rules
```

   b. Edit `/opt/xt-images/templates/default/etc/auditd.conf` on the SMW and set the `log_file` parameter. For example, if the mount point for your Lustre file system is `mylusmnt` and you want to place audit logs in a directory called `auditdir`, type the following commands.

```
smw:~# vi /opt/xt-images/templates/default/etc/auditd.conf
log_file = /mylusmnt/auditdir/audit.log
```

Warning: If you run auditing on compute nodes without configuring the audit directory, audit records that are written to the local ram-disk could cause the ram-disk to fill.

   c. Edit the `/opt/xt-images/templates/default/etc/audit.rules` file on the SMW. Change this file to set site-specific auditing rules for the compute nodes. At a minimum, you should set the `-e` option to `1` (one) to enable auditing.

```
smw:~# vi /opt/xt-images/templates/default/etc/audit.rules
```

   Make your changes after the following line; for example:

```
# Feel free to add below this line.  See auditctl man page
-e 1
```

   d. Create the following symbolic link.

```
smw:~# mkdir -p -m 755 /opt/xt-images/templates/default/etc/init.d/rc3.d
smw:~# cd /opt/xt-images/templates/default/etc/init.d/rc3.d
smw:/opt/xt-images/templates/default/etc/init.d/rc3.d # ln -s ../auditd S12auditd
```

   e. If you set `CNL_audit=`**`yes`** in `CLEinstall.conf` before you ran the `CLEinstall` program, update the boot image by following the steps in Procedure 8 on page 64.

   Otherwise, you must first edit the `/var/opt/cray/install/shell_bootimage_`*LABEL*`.sh` script and set `CNL_AUDIT=`**`y`** and then update the boot image following the steps in Procedure 8 on page 64.

2. Follow these steps to enable and configure auditing on login nodes.

   a. Log on to the boot node and use the `xtopview` command to access all login nodes by class.

   ```
   smw:~# ssh root@boot
   boot:~ # xtopview -c login -m "configuring audit files"
   ```

   b. Specialize these files to the `login` class.

   ```
   class/login:/ # xtspec -c login /etc/auditd.conf
   class/login:/ # xtspec -c login /etc/audit.rules
   ```

   c. Edit `/etc/auditd.conf` and set the `log_file` parameter. For example, if your Lustre file system is called *filesystem* and you want to place audit logs in a directory called *auditdir*, type the following commands.

   ```
   class/login:/ # vi /etc/auditd.conf
   log_file = /filesystem/auditdir/audit.log
   ```

   d. Edit the `/etc/audit.rules` file to set site-specific auditing rules for the login nodes. At a minimum, you should set the `-e` option to `1` (one).

   ```
   class/login:/ # vi /etc/audit.rules
   ```

   Make your changes after the following line; for example:

   ```
   # Feel free to add below this line.  See auditctl man page
   -e 1
   ```

   e. Exit `xtopview`.

   ```
   class/login:/ # exit
   ```

3. You must configure auditing on the boot node to use standard Linux auditing. Follow these steps to turn off Cray audit extensions for the boot node. Configure the boot node to use the default `log_file` parameter in the `auditd.conf` file and set the `cluster` entry to `no`.

   a. While logged on to the boot node, edit the `/etc/auditd.conf` file.

   ```
   boot:~ # vi /etc/auditd.conf
   log_file = /var/log/audit/audit.log
   cluster = no
   ```

   b. Edit the `/etc/audit.rules` file to set site-specific auditing rules for the boot node. At a minimum, you should set the `-e` option to `1` (one).

   ```
   boot:~ # vi /etc/audit.rules
   ```

   Make your changes after the following line; for example:

   ```
   # Feel free to add below this line.  See auditctl man page
   -e 1
   ```

   c. Configure the audit daemon to start on the boot node.

   ```
   boot:~ # chkconfig --force auditd on
   ```

4. Create the log file directory. Log into a node that has the Lustre file system mounted and type the following commands:

   ```
   login:~# mkdir -p /filesystem/auditdir
   login:~# chmod 700 /filesystem/auditdir
   ```

5. Edit the boot automation file to configure your system to start the Cray audit daemon on login nodes by invoking `/etc/init.d/auditd start` on each login node.

Make these changes to your boot automation file later in the installation process. Configuring a Boot Automation File on page 114 describes specific steps.

After you enable auditing by using the previous procedures, you can change the audit configuration temporarily by using the `xtauditctl` command. For more information, see *Managing System Software for Cray XE and Cray XK Systems* (S–2393) and the `xtauditctl`(8) man page.

5.13.13 Configuring System Message Logs

Optional: The steps in this section are not required for installation of CLE software.

The CLE release uses the Linux `syslog-ng` daemon and associated `syslog-ng.conf` configuration file to log system messages. For more information, see the `syslog-ng`(8) and `syslog-ng.conf`(5) man pages.

Procedure 38. Configuring `syslog-ng` system message logs

Follow these steps to modify the default `syslog-ng` configuration.

1. Log on to the boot node and edit the `syslog-ng.conf` configuration file. For more information on this file and its configuration options, see the `syslog-ng.conf`(5) man page.

   ```
   smw:~# ssh root@boot
   boot:~ # vi /etc/syslog-ng/syslog-ng.conf
   ```

2. Restart the `syslog-ng` daemon on the boot node.

   ```
   boot:~ # /etc/init.d/syslog restart
   ```

3. Edit the configuration file on the syslog node and make the desired changes.

   ```
   boot:~ # xtopview -n 5
   node/5:/ # vi /etc/syslog-ng/syslog-ng.conf
   node/5:/ # exit
   ```

4. Restart the `syslog-ng` daemon on the syslog node.

   ```
   boot:~ # ssh syslog /etc/init.d/syslog restart
   ```

5. Edit the configuration file on other service nodes by using `xtopview` in the default view and make the desired changes.

   ```
   boot:~ # xtopview
   default/:/ # vi /etc/syslog-ng/syslog-ng.conf
   default/:/ # exit
   ```

6. Restart the `syslog-ng` daemon on the remaining service nodes. For each service node, type the following command.

   ```
   boot:~ # ssh nodename /etc/init.d/syslog restart
   ```

5.13.14 Configuring the Node Health Checker

The CLE installation and upgrade processes automatically install and enable the Node Health Checker (NHC) by default; you do not need to change installation parameters or issue any commands. However, you can edit the `/etc/opt/cray/nodehealth/nodehealth.conf` file to specify which NHC tests are to be run and to alter the behavior of NHC tests (including time-out values and actions for tests when they fail); configure time-out values for Suspect Mode and disable/enable Suspect Mode; or disable or enable NHC.

The NHC configuration file, `/etc/opt/cray/nodehealth/nodehealth.conf` is located in the shared root. After you modify the `nodehealth.conf` file, the changes are reflected immediately the next time NHC runs.

To disable NHC entirely, set the value of the `nhcon` global variable in the `nodehealth.conf` file to off (the default value is `on`).

5.14 Finish Booting the System

After all service nodes are booted, boot the compute nodes. After your system is fully booted, you can reboot it as needed. For information about customizing an automatic boot process, see Configuring a Boot Automation File on page 114.

> **Important:** If you deferred updating the boot image in any of the previous procedures, update the boot image now by following the steps in Procedure 8 on page 64.

Procedure 39. Booting CNL compute nodes

At this point in the installation process, all service and login nodes are booted.

1. Return to the terminal session for `xtbootsys` that you started in Procedure 12 on page 69.

2. Select **17** from the `xtbootsys` menu to boot by using a loadfile. A series of prompts are displayed. Type the responses indicated in the following example.

For the `component list` prompt, type **`p0`** to boot the entire system, or **`p`***N* (where *N* is the partition number) to boot a partition. At the final three prompts, press the `Enter` key.

```
Enter your boot choice: 17
Enter a boot type string (or nothing to do nothing): CNL0
Enter a boot type option (or nothing to do nothing): compute
Enter a component list (or nothing to do nothing): p0
Enter 'any' to wait for any console output,
   or 'linux' to wait for a linux style boot,
   or 'mtk', 'threadstorm', 'ts', or 'xmt' to wait for a MTK style boot,
   or anything else (or nothing) to not wait at all:
Enter an alternative CPIO archive name (or nothing):
Do you want to send the ec_boot event ('no' means to only load memory) ? [Yn]
```

3. After all the compute nodes are booted, return to the `xtbootsys` menu. Type **`q`** to exit the `xtbootsys` program.

5.15 Testing the System

To verify that the system is operational, follow these steps.

Procedure 40. Testing the system for basic functionality

1. If the system was shut down by using `xtshutdown`, remove the `/etc/nologin` file from all service nodes to permit a non-root account to log on.

```
smw:~# ssh root@boot
boot:~ # xtunspec -r /rr/current -d /etc/nologin
```

2. Log on to the login node as `crayadm`.

```
boot:~ # ssh crayadm@login
```

3. Use system-status commands, such as `xtnodestat`, `xtprocadmin`, and `apstat`.

The xtnodestat command displays the current allocation and status of the compute nodes, organized by physical cabinet. The last line of the output shows the number of available compute nodes.

```
crayadm@login:~> xtnodestat
Current Allocation Status at Fri May 21 07:11:48 2010

     C0-0     C1-0     C2-0     C3-0
  n3 ;;;;;;;; ;;;;;;;; ;;;;;;;; ;;;;;;;;
  n2 ;;;;;;;; ;;;;;;;; ;;;;;;;; ;;;;;;;;
  n1 ;;;;;;;; ;;;;;;;; ;;;;;;;; ;;;;;;;;
c2n0 ;;;;;;;; ;;;;;;;; ;;;;;;;; ;;;;;;;;
  n3 ;;;;;;;; ;;;;;;;; ;S;S;S;S ;;;;;;;;
  n2 ;;;;;;;; ;;;;;;;; ;S;S;S;S ;;;;;;;;
  n1 ;;;;;;;; ;;;;;;;; ;S;S;S;S ;;;;;;;;
c1n0 ;;;;;;;; ;;;;;;;; ;S;S;S;S ;;;;;;;;
  n3 S;S;S;S; ;;;;;;;; ;;;;;;;; ;;;;;;;;
  n2 S;S;S;S; ;;;;;;;; ;;;;;;;; ;;;;;;;;
  n1 S;S;S;S; ;;;;;;;; ;;;;;;;; ;;;;;;;;
c0n0 S;S;S;S; ;;;;;;;; ;;;;;;;; ;;;;;;;;
    s01234567 01234567 01234567 01234567

Legend:
   nonexistent node                 S  service node
;  free interactive compute node    -  free batch compute node
A  allocated, but idle compute node ?  suspect compute node
X  down compute node                Y  down or admindown service node
Z  admindown compute node

Available compute nodes:        352 interactive,          0 batch
```

The xtprocadmin command displays the current values of processor flags and node attributes.

```
crayadm@login:~> xtprocadmin
   NID    (HEX)    NODENAME     TYPE    STATUS         MODE
     0      0x0  c0-0c0s0n0  service      up interactive
     2      0x2  c0-0c0s1n0  service        up interactive
     4      0x4  c0-0c0s2n0  service        up interactive
     6      0x6  c0-0c0s3n0  service        up interactive
. . .
    93     0x5d  c0-0c2s1n3  service        up interactive
    94     0x5e  c0-0c2s0n2  service        up interactive
    95     0x5f  c0-0c2s0n3  service        up interactive
```

The apstat command displays the current status of all applications running on the system.

```
crayadm@login:~> apstat -v
Compute node summary
    arch config     up   resv    use  avail   down
      XT     51     51      0      0     51      0

No pending applications are present

No placed applications are present
```

4. Run a simple job on the compute nodes.

   At the conclusion of the installation process, the `CLEinstall` program provides suggestions for runtime commands and indicates how many compute nodes are available for use with the `aprun -n` option.

   Note: For `aprun` to work cleanly, the current working directory on the login node should also exist on the compute node. Change your current working directory to either `/tmp` or to a directory on a mounted Lustre file system.

   For example, type the following.

```
crayadm@login:~> cd /tmp
crayadm@login:~> aprun -b -n 16 -N 1 /bin/cat /proc/sys/kernel/hostname
```

   This command returns the hostname of each of the 16 computes nodes used to execute the program.

```
nid00010
nid00011
nid00012
nid00020
nid00016
nid00040
nid00052
nid00078
nid00084
nid00043
nid00046
nid00049
. . .
```

5. Test file system functionality. For example, if you have a Lustre file system named */mylusmnt/filesystem*, type the following.

```
crayadm@login:~> cd /mylusmnt/filesystem
crayadm@login:/mylustremnt/filesystem> echo lustretest > testfile
crayadm@login:/mylustremnt/filesystem> aprun -b -n 5 -N 1 /bin/cat ./testfile
lustretest
lustretest
lustretest
lustretest
lustretest
Application 109 resources: utime ~0s, stime ~0s
```

6. Test the optional features that you have configured on your system.

   a. To test RSIP functionality, log on to an RSIP client node (compute node) and ping the IP address of the SMW or other host external to the Cray system. For example, if `c0-0c0s7n2` is an RSIP client, type the following commands.

   ```
   crayadm@login:~> exit
   boot:~ # ssh root@c0-0c0s7n2
   root@c0-0c0s7n2's password:
   Welcome to the initramfs
   # ping 172.30.14.55
   172.30.14.55 is alive!
   # exit
   Connection to c0-0c0s7n2 closed.
   boot:~ # exit
   ```

   Note: RSIP clients on the compute nodes make connections to the RSIP server(s) during system boot. Initiation of these connections is staggered over a two minute window; during that time, connectivity over RSIP tunnels is unreliable. Avoid using RSIP services for three to four minutes following a system boot.

   b. To check the status of DVS, type the following command on the DVS server node.

   ```
   crayadm@login:~> ssh root@nid00019 /etc/init.d/dvs status
   DVS service: ..running
   ```

   To test DVS functionality, invoke the `mount` command on any compute node.

```
crayadm@login:~> ssh root@c0-0c0s7n2 mount | grep dvs
/dvs-shared on /dvs type dvs (rw,blksize=16384,nodename=c0-0c0s4n3,nocache,nodatasync,\
retry,userenv,clusterfs,maxnodes=1,nnodes=1)
```

   Create a test file on the DVS mounted file system. For example, type the following.

```
crayadm@login:~> cd /dvs
crayadm@login:/dvs> echo dvstest > testfile
crayadm@login:/dvs> aprun -b -n 5 -N 1 /bin/cat ./testfile
dvstest
dvstest
dvstest
dvstest
dvstest
Application 121 resources: utime ~0s, stime ~0s
```

7. Following a successful installation, the file `/etc/opt/cray/release/clerelease` is populated with the installed release level. For example,

   ```
   crayadm@login:~> cat /etc/opt/cray/release/clerelease
   4.0.UP00
   ```

If the preceding simple tests ran successfully, the system is operational.

5.16 Configuring a Boot Automation File

A sample boot automation file, `/opt/cray/etc/auto.generic.cnl`, is provided as a basis for further customizing the boot process. You are asked to shut down your system so that you can test the customized boot automation files.

For more information about boot automation, see the `xtbootsys`(8) man page.

Procedure 41. Configuring boot automation on the SMW

1. Use your site-specific procedures to shut down the system. For example, to shutdown using an automation file, type the following:

   ```
   crayadm@smw:~> xtbootsys -s last -a auto.xtshutdown
   ```

   Although not the preferred method, alternatively execute these commands as `root` from the boot node to shutdown your system.

   ```
   boot:~ # xtshutdown -y
   boot:~ # shutdown -h now;exit
   ```

2. Prepare a boot automation file. If no boot automation file exists, copy the template file.

```
crayadm@smw:~> cp -p /opt/cray/etc/auto.generic.cnl /opt/cray/etc/auto.xthostname
```

3. Edit the boot automation file.

   ```
   crayadm@smw:~> vi /opt/cray/etc/auto.xthostname
   ```

   Note: The boot automation file contains many of the following commands but the lines are commented out. Uncomment the pertinent lines and edit them as needed.

   a. To enable non-root logins following a system shutdown, add the following as the last command:

```
lappend actions { crms_exec_on_bootnode "root" "xtunspec -r /rr/current -d /etc/nologin" }
```

   b. If you have configured Lustre file systems for your system, add the following line to start Lustre servers and mount Lustre file systems on the clients.

   Note: Start Lustre on the service nodes before you boot the compute nodes.

```
lappend actions { crms_exec_on_bootnode "root" "/etc/init.d/lustre start"}
```

   Optionally, specify a Lustre file system name, for example:

```
lappend actions { crms_exec_on_bootnode "root" "/etc/init.d/lustre start filesystem"}
```

   For information about Lustre file systems, see Chapter 6, Configuring Lustre File Systems on page 119.

c. If you have configured your system to run Cray Audit, start the audit daemon on the login nodes after Lustre is started. Add a line for each login node on your system. For example, if `login01` and `login02` are login nodes:

```
lappend actions { crms_exec_via_bootnode "login01" "root" "/etc/init.d/auditd start" }
lappend actions { crms_exec_via_bootnode "login02" "root" "/etc/init.d/auditd start" }
```

d. If you have configured DVS servers (for DSL) by using compute nodes that were **manually** repurposed as service nodes, boot these nodes after the SDB and other service nodes are booted. For example, if `c0-0c0s7n0` and `c0-0c0s7n1` are compute node root servers:

```
lappend actions [list crms_boot_loadfile SNL0 compute "c0-0c0s7n0 c0-0c0s7n1" linux]
lappend actions { crms_sleep 5 }
```

Note: The capability to manually repurpose compute nodes is deprecated and may not be supported in future releases. To configure compute nodes for CNRTE, Cray recommends that you follow Procedure 5 on page 57 and *Repurposing Compute Nodes as Service Nodes on Cray XE and Cray XT Systems*.

e. If you have configured an RSIP service node client with `CLEinstall`, add this line to start the RSIP client. Invoke a `modprobe` of the RSIP service node client's `krsip` module with an IP argument pointing to the HSN IP address of an RSIP server node. For example, if the IP address of the RSIP server is `10.128.0.17` and the RSIP service node client is `nid00000`, add this line:

```
lappend actions { crms_exec_via_bootnode "nid00000" "root" "modprobe \
krsip ip=10.128.0.17" }
```

f. Make additional site-specific changes as needed and save the file.

4. Use the `xtbootsys` command to boot the Cray system.

Caution: You must shut down your Cray system before you invoke the `xtbootsys` command. If you are installing to an alternate system set, you must shut down the currently running system before you boot the new boot image.

Type the following command to boot the entire system.

```
crayadm@smw:~> xtbootsys -a auto.xthostname
```

Or

Type the following command to boot a partition.

```
crayadm@smw:~> xtbootsys --partition pN -a auto.xthostname
```

5. (Optional) Reboot your system and confirm that shutdown and boot procedures operate as expected.

The software installation of your Cray system is complete. Cray recommends that you use the `xthotbackup` utility to create a backup of your newly installed system. For more information, see the `xthotbackup`(8) man page.

5.16.1 Configuring Boot Automation for SDB Node Failover

Optional: If you have configured your system for SDB node failover and you have commands in your boot automation script that apply to the SDB, follow these steps to ensure the appropriate boot automation commands are invoked in the event of an SDB node failure.

SDB-specific commands in the boot automation script must be invoked for the backup SDB node in the event of a failover, however, the boot automation script does not apply to the backup SDB node in a failover situation.

Procedure 42. Configuring the SDB failover configuration file

1. Create or edit the `sdbfailover.conf` file in the shared root file system in the default view.

   ```
   boot001:~# xtopview
   default/:/ # vi /etc/opt/cray/sdb/sdbfailover.conf
   ```

   Make optional site-specific changes. For example, if the boot automation file started a batch scheduler (it was not started by using `chkconfig`) or set up a route to an external license server, you need to add the same commands to the `sdbfailover.conf` file so that they are invoked when the backup SDB node is started. For example:

   ```
   #
   # Commands to be run on the backup sdb node after it has failed over
   #
   /bin/netstat -r
   /sbin/route add default gw login
   /etc/init.d/torque_server start
   /etc/init.d/moab start
   ```

2. Exit `xtopview`.

   ```
   default/:/ # exit
   ```

5.17 Post Installation System Management

You should now have an operational Cray system running CLE software. For information about additional software you may need on your system, including programming environment and batch software, see Appendix A, Installing Additional Software on page 155. Appendix B, Installing RPMs on page 157 provides generic instructions for installing RPM Package Manager (RPM) packages.

For information about additional system administrative tasks to manage operation of your system, see *Managing System Software for Cray XE and Cray XK Systems*. It presents the following topics in greater detail:

- Managing the system
- Monitoring system activity
- Managing user access
- Modifying an installed system
- Managing services
- SMW and CLE System Administration Commands

Managing System Software for Cray XE and Cray XK Systems provides complete documentation for most CLE features. These features or subsystems may require site-specific configuration and administration.

- Application Level Placement Scheduler (ALPS)
- OpenFabrics Interconnect Drivers
- Node Health Checker (NHC)
- System Environmental Data Collector (SEDC)

If you configured the following optional features, additional configuration is required. See *Managing System Software for Cray XE and Cray XK Systems* for more information.

- Dynamic Shared Objects and Cluster Compatibility Mode (CCM)
- Cray Audit
- Comprehensive System Accounting (CSA)
- Checkpoint/Restart

Configuring Lustre File Systems [6]

Optional: The Lustre file system is optional; your storage RAID may be configured with other file systems.

Lustre is a scalable, high-performance POSIX-compliant file system that uses the `ldiskfs` file system from Oracle for back-end storage. The `ldiskfs` file system is an extension to the Linux `ext4` file system with Oracle enhancements for Lustre.

The Lustre file system consists of software subsystems, storage, and an associated network. The RPMs for the Lustre file system are installed during the Cray Linux Environment (CLE) software installation.

6.1 Lustre File System Documentation

Lustre file system concepts and administration are discussed in detail in *Managing Lustre for the Cray Linux Environment (CLE)*. The *Lustre Operations Manual* from Oracle is included as a PDF file in the release package. Additional Information about Lustre is available at the following websites:

http://wiki.lustre.org
http://wiki.lustre.org/index.php/Lustre_Documentation
http://www.oracle.com/us/products/servers-storage/storage/storage-software/031855.htm

Note: The Lustre information presented in this guide is based, in part, on documentation from Oracle. Lustre information contained in Cray publications, supersedes information located in Lustre publications from Oracle.

6.2 Lustre Software Components

The following software components of Lustre can be implemented on selected nodes of the Cray system.

- Client

  Clients are services or programs that access the file system. On Cray systems clients are typically associated with login or compute nodes.

- Object storage target (OST)

  An *object storage target (OST)* is the software interface to back-end storage volumes. There may be one or more OSTs. The OSTs handle file data and enforce security for client access. The client performs parallel I/O operations across multiple OSTs.

  You configure the characteristics of the OSTs during the Lustre setup.

- Object storage server (OSS)

  An *object storage server (OSS)* is a node that hosts the OSTs. Each OSS node, referenced by node ID (NID), has Fibre Channel or InfiniBand connections to a RAID controller. The OST is a logical device; the OSS is the physical node.

- Metadata server (MDS)

  The *metadata server (MDS)* owns and manages information about the files in the Lustre file system. It handles namespace operations such as file creation, but it does not contain any file data. It stores information about which file is located on which OSTs, how the blocks of files are striped across the OSTs, the date and time the file was modified, and so on. The MDS is consulted whenever a file is opened or closed. Because file namespace operations are done by the MDS, they do not impact operations that manipulate file data.

  You configure the characteristics of the MDS during the Lustre setup.

- Metadata target (MDT)

  The *metadata target (MDT)* is the software interface to back-end storage volumes for the MDS and stores metadata for the Lustre file system.

- Management server (MGS)

  The *management server (MGS)* controls the configuration information for all Lustre file systems running at a site. Clients and servers contact the MGS to retrieve or change configuration information. Cray installation and upgrade utilities automatically create a default Lustre configuration in which the MGS and the Meta Data Server (MDS) are co-located on a service node and share the same physical device for data storage.

6.3 Lustre File System Configuration on a Cray System

The CLE software includes Lustre control utilities from Cray. These utilities provide a layer of abstraction to the standard Lustre configuration and mount utilities by implementing a centralized configuration file and describing each Lustre file system in a site-specific file system definition file. The Cray Lustre control scripts (`generate_config.sh` and `lustre_control.sh`) use the information in this file system definition file to interface with Lustre's MountConf system and Management Server (MGS). When using the Lustre control configuration utilities, system administrators do not need to access the MGS directly.

If this is a new installation with Lustre file systems, follow the procedures in this section to configure your Lustre file systems. If you are upgrading to a new version of the CLE software, Lustre is already configured and no further action is required. Note, however, that you can use the information and procedures described in this chapter to manage future changes to an existing Lustre configuration.

6.3.1 Lustre Control Utilities and File System Definition Parameters

When you use the Lustre control utilities (see Creating Lustre File Systems on page 125), the first step is to create a Lustre file system definition file (*filesystem*`.fs_defs`) for each Lustre file system on your Cray system. A sample file system definition file is provided in `/etc/opt/cray/lustre-utils/sample.fs_defs` on the boot node.

This section describes the parameters that define your Lustre file system. The descriptions use the following conventions for node and device naming:

- *nodename* is a host or node name using the format *nidxxxxx*; for example, `nid00008`.
- *device* is a device path using the format `/dev/disk/by-id/`*ID-partN* where *ID* is the volume identifier and *partN* is the partition number (if applicable); for example:

  ```
  /dev/disk/by-id/scsi-3600a0b800026e1400000192e4b66eb97-part2
  ```

Note: If you change any of the parameters in your *filesystem*`.fs_defs` file, you will need to run the `lustre_control.sh write_conf` command to regenerate the Lustre configuration and apply your changes.

`FSNAME` Unique name for the Lustre file system defined by this *filesystem*`.fs_defs` file. Limited to 8 characters. Used internally by Lustre.

`LUSTRE_CSV` The path of the `*.csv` configuration file that is created by the `generate_config.sh` script. The default path is `/etc/opt/cray/lustre-utils/${FSNAME}.config.csv`.

`LUSNID[`*NID*`]` The hostname to NID table. Not required, but if it is not set, `xtprocadmin` is used to generate these values, which requires the Service Database (SDB) to be up when Lustre configuration scripts are run. Setting `LUSNID` will improve Lustre start and stop times on larger systems. Format: *nodename*. The index must be the node ID of the specified node; for example, `LUSNID[18]="nid00018"`.

`MDSHOST` The hostname of the MDS node. Format: *nodename*.

`MDSDEV` MDS physical device. Format: `${MDSHOST}:`*device*.

> **Note:** If you are configuring Lustre failover, use the failover format example in the comments. For more information, see Configuring Lustre Failover on page 129.

`MGSDEV` MGS physical device. In a default configuration, this parameter is commented out and the MGS and MDS are co-located on a service node and share the same physical device for data storage; use this parameter to designate a separate MGS device. Specify the node and physical device to start the MGS with this file system; specify the node only if another file system will start the MGS. Format: *nodename*`[:`*device*`]`.

> **Note:** The file system associated with the MGS must be started first and stopped last.

`OSTDEV[`*n*`]` Table of OST devices. The index [*n*] is the OST number. The index can start at either 0 or 1. Format: *nodename*:*device* where *nodename* can be `${LUSNID[`*n*`]}`, if `LUSNID` is defined.

> **Important:** The value set for `MaxStartups` in `/etc/ssh/sshd_config` must be greater than the number of OSTs per OSS in order for all OSTs to mount successfully. The default value for `MaxStartups` is `10`.

> **Note:** If you are configuring Lustre failover, use the failover format example in the comments. For more information, see Configuring Lustre Failover on page 129.

`MOUNTERS` Hostnames for the service nodes that mount this file system, usually login nodes. Format: "*host1 host2*" or `pdsh` syntax, for example `host[1-2],host5`.

`MOUNT_POINT`

Path for the client mount point on service nodes specified in the `MOUNTERS` parameter.

`NETTYPE` Type of underlying interconnect. Set to `gni` for systems with the Gemini based system interconnection network.

`STRIPE_SIZE`

Stripe size in bytes. Cray recommends a default value of 1048576 bytes (1MB.) For more information, see Configuring Striping on Lustre File Systems on page 129.

`STRIPE_COUNT`

Integer count of the default number of OSTs used for a file. Valid range is 1 to the number of OSTs. A value of `-1` specifies striping across all OSTs. Cray recommends a stripe count of 2 to 4 OSTs. For more information, see Configuring Striping on Lustre File Systems on page 129.

`QUOTAOPTS` Set to `quotaon=ug` to enable user and group quotas for the file system or to `quotaon=g` to enable only group quotas. For information on Lustre quotas, see the *Lustre Operations Manual*.

`AUTO_FAILOVER`

Set to `yes` to enable automatic failover when failover is configured; set to `no` to select manual failover. The default setting is `yes`.

`ENABLE_IMP_RECOVERY`

Set to `yes` to enable imperative recovery (explicit client notification during the failover process). The default setting is `no`.

`RECOVERY_TIME_HARD`

Specifies a hard recovery window timeout for failover. The server will incrementally extend its timeout up to a hard maximum of `RECOVERY_TIME_HARD` seconds. The default hard recovery timeout is set to 900 seconds (15 minutes).

`RECOVERY_TIME_SOFT`

Specifies a rolling recovery window timeout for failover. This value should be less than or equal to `RECOVERY_TIME_HARD`. Allows `RECOVERY_TIME_SOFT` seconds for clients to reconnect for recovery after a server crash. This timeout will incrementally extend if it is about to expire and the server is still handling new connections from recoverable clients. The default soft recovery timeout is set to 300 (5 minutes).

You can modify the following parameters, however, in most cases the default value is preferred. Only experienced Lustre administrators should change these options.

`EXT3_JRNL_SIZE`

Journal size, in megabytes, on underlying `ldiskfs` file systems. The default value is 400.

OST_MOUNTFSOPTIONS

Mount options for the OSTs. The default value is extents,mballoc,errors=remount-ro.

MDS_MOUNTFSOPTIONS

Mount options for the MDS. This parameter is not required for the MDS and is commented out by default.

CLIENT_MOUNTOPTIONS

Mount options for clients such as service nodes. Option flock is required and included by default.

MDS_MKFSOPTIONS

Options used when creating an MDS file system. These options are passed as --mkfsoptions to the mkfs.lustre utility when the file system is created. This parameter is commented out by default.

OST_MKFSOPTIONS

Options used when creating an OST file system. These options are passed as --mkfsoptions to the mkfs.lustre utility when the file system is created. This parameter is commented out by default.

LUSTRE_FILESYS_DATA

The path of the file that is used to load data into the filesystem SDB table. The default is $FSNAME.filesys.csv in the directory specified for LUSTRE_CSV. The generate_config.sh utility creates or overwrites this file with the proper failover information.

LUSTRE_FAILOVER_DATA

The path of the file that is used to load data into the lustre_failover SDB table. The default is ${FSNAME}.lustre_failover.csv in the directory specified for LUSTRE_CSV. The generate_config.sh utility creates or overwrites this file with the proper failover information.

LUSTRE_SERVICE_DATA

The path of the file that is used to load data into the lustre_service SDB table. The default is ${FSNAME}.lustre_serv.csv in the directory specified for LUSTRE_CSV. The generate_config.sh utility creates or overwrites this file with the proper failover information.

SERVERMNT Directory prefix for Lustre servers to use when mounting the OSTs. The default path is /tmp/lustre.

`TIMEOUT` Lustre timeout in seconds. The default value is 300.

`FSTYPE` Lustre file system type. The default value is `ldiskfs`.

`PDSH` Command syntax for `pdsh`. The default value is `"pdsh -f 256 -S"`.

`VERBOSE` Verbose output flag. The default setting is `yes`.

The `lustre.fs_defs`(5) man page also includes this information about file system definition parameters.

6.3.2 Creating Lustre File Systems

Use the Cray Lustre control utilities to configure your system to use Lustre file systems. Follow Procedure 43 on page 125 to create file system definition files, start and stop Lustre servers, and format and mount Lustre file systems.

Caution: You must use persistent device names in the Lustre file system definition file. Non-persistent device names (for example, `/dev/sdc`) can change when the system is rebooted. If non-persistent names are specified in the *filesystem*`.fs_defs` file, then Lustre may try to mount the wrong devices and fail to start when the you reboot the system. If you are upgrading from a release that did not require persistent device names, you must convert to persistent device names to avoid this problem. Use the `verify_config`, `update_config`, and `dump_target_devnames` options with `lustre_control.sh` to update existing *filesystem*`.fs_defs` files to use persistent names.

For more information about Lustre control utilities see the `lustre_control.sh`(8), `generate_config.sh`(8) and `lustre.fs_defs`(5) man pages.

Procedure 43. Creating, formatting, and starting Lustre file systems

Follow these steps to configure, create, and start Lustre file systems. This example uses the following Lustre configuration, where *IDn* represents the volume identifier on the disk, for example, `scsi-3600a0b800026e1400000192e4b66eb97`:

MDS is on `nid00012`, `/dev/disk/by-id/`*IDa*

OST0 is on `nid00018`, `/dev/disk/by-id/`*IDb*

OST1 is on `nid00026`, `/dev/disk/by-id/`*IDc*

OST2 is on `nid00018`, `/dev/disk/by-id/`*IDd*

OST3 is on `nid00026`, `/dev/disk/by-id/`*IDe*

Login nodes are `nid00008` and `nid00030`

1. Create a *filesystem*`.fs_defs` file for each Lustre file system you want to configure.

   Note: Lustre control utilities require that you name this file by using your file system name followed by `.fs_defs`. For example, if your file system is called `filesystem`, you must name the Lustre configuration definition file `filesystem.fs_defs`. The file system name used here must match the name used in step 3. This name does not need to match the `FSNAME` parameter set in the file itself. `FSNAME` is used internally by Lustre to uniquely identify each file system.

   For example, to create a file system definition file for a file system called `filesystem`, type the following command:

   ```
   boot:~ # cd /etc/opt/cray/lustre-utils
   boot:/etc/opt/cray/lustre-utils # cp -p \
   sample.fs_defs  filesystem.fs_defs
   ```

2. Edit each newly created *filesystem*`.fs_defs` file and modify the configuration parameters to define your Lustre configuration. For example:

   ```
   boot:/etc/opt/cray/lustre-utils # vi filesystem.fs_defs
   FSNAME="lus0"
   MDSHOST="nid00012"
   MDSDEV="${MDSHOST}:/dev/disk/by-id/IDa"
   OSTDEV[0]="nid00018:/dev/disk/by-id/IDb"
   OSTDEV[1]="nid00026:/dev/disk/by-id/IDc"
   OSTDEV[2]="nid00018:/dev/disk/by-id/IDd"
   OSTDEV[3]="nid00026:/dev/disk/by-id/IDe"
   MOUNTERS="nid00008,nid00004"
   MOUNT_POINT="/mnt/filesystem"
   STRIPE_SIZE=1048576
   STRIPE_COUNT=2
   ```

   Note: You must include quotes around values that contain spaces.

   For additional information about Lustre configuration parameters, see the `lustre.fs_defs`(5) man page.

3. Edit `xt.lustre.config` to enable the `/etc/init.d/lustre` startup script to start the Lustre file system at boot time. For every *filesystem*`.fs_defs` file, add the file system name to the `FILESYSTEMS=` line in `xt.lustre.config`. For example:

   ```
   boot:/etc/opt/cray/lustre-utils # vi xt.lustre.config
   FILESYSTEMS="filesystem"
   ```

   If you have more than one Lustre file system, include all configured file system names, separated by a space. For example:

   ```
   FILESYSTEMS="filesystem filesystem2"
   ```

4. Generate CSV files for your file systems. Type this command for each *filesystem*`.fs_defs` file you created in step 1.

```
boot:/etc/opt/cray/lustre-utils # ./generate_config.sh filesystem.fs_defs
no LUSNID defined in .fs_defs, gathering nids from xtprocadmin...
Created Lustre config at /etc/opt/cray/lustre-utils/filesystem.config.csv
No failover configuration specified.
```

5. Create the file system mount point on the shared root file system in the default view. Type these commands for each Lustre file system you have configured. For example, if `MOUNT_POINT` is set to `/mnt/filesystem`, you would type:

   ```
   boot:/etc/opt/cray/lustre-utils # xtopview
   default/:/ # mkdir -p /mnt/filesystem
   default/:/ # exit
   boot:/etc/opt/cray/lustre-utils #
   ```

6. Format the file systems and start the Lustre servers. Type these commands for each Lustre file system you have configured.

```
boot:/etc/opt/cray/lustre-utils # ./lustre_control.sh filesystem.fs_defs reformat
```

   Note: This process takes a while, possibly an hour or more.

7. Mount the service node clients. Type this command for each Lustre file system you have configured.

```
boot:/etc/opt/cray/lustre-utils # ./lustre_control.sh filesystem.fs_defs mount_clients
```

8. Verify that the Lustre clients have the Lustre file systems mounted.

```
boot:/etc/opt/cray/lustre-utils # ssh login mount | grep lustre
12@gni:/lus0 on /mnt/filesystem type lustre (rw,flock)
```

9. Stop Lustre clients and servers. Type these commands for each Lustre file system you have configured.

```
boot:/etc/opt/cray/lustre-utils # ./lustre_control.sh filesystem.fs_defs umount_clients
boot:/etc/opt/cray/lustre-utils # ./lustre_control.sh filesystem.fs_defs stop
```

10. Test the `/etc/init.d/lustre` script. Start the Lustre servers and mount the Lustre clients.

    ```
    boot:/etc/opt/cray/lustre-utils # /etc/init.d/lustre start
    ```

11. Verify that the Lustre clients have the Lustre file systems mounted.

    ```
    boot:/etc/opt/cray/lustre-utils # ssh login mount | grep lustre
    12@gni:/lus0 on /mnt/filesystem type lustre (rw,flock)
    ```

12. (Optional) Set permissions for the Lustre file system. Presently, the permissions on the top level directory of this Lustre file system are set to `0755` with an owner and group of `root`. Run the following command to change the permissions and allow non-root users to create files.

```
boot:/etc/opt/cray/lustre-utils # ssh login chmod 1777 /mnt/filesystem
```

13. Exit from the boot node.

    ```
    boot:/etc/opt/cray/lustre-utils # exit
    ```

14. Update the boot automation scripts on the SMW. Any site boot automation scripts need to be changed to reflect the new method of starting Lustre servers and mounting Lustre file systems on the clients. Type these commands, adding the line as shown.

    Note: You must start Lustre on the service nodes before you boot the compute nodes.

```
smw:~# vi /opt/cray/etc/auto.xthostname
lappend actions { crms_exec_on_bootnode "root" "/etc/init.d/lustre start"}
```

6.3.3 Mounting Lustre Clients

Service node and compute node clients reference Lustre as a local file system. Service nodes mount Lustre by using the Lustre startup scripts before the compute nodes boot. The Lustre file systems are mounted on compute nodes automatically during startup if they are included in the corresponding `/etc/fstab` file. Follow Procedure 44 on page 128 to mount Lustre services on compute nodes.

Procedure 44. Creating Lustre clients for compute nodes

> **Note:** To make these changes for a system partition, rather than for the entire system, replace `/opt/xt-images/templates` with `/opt/xt-images/templates-p`*N*, where *N* is the partition number.

1. Edit the `/etc/fstab` file in the default CNL boot image template. Add an entry for each file system you have configured. For example, if `FSNAME=`*lus0*, `MOUNT_POINT=`*/mnt/filesystem*, `NETTYPE=`*gni* and `MDSHOST=nid00012`:

   ```
   smw:~# vi /opt/xt-images/templates/default/etc/fstab
   12@gni:/lus0 /mnt/filesystem  lustre  rw,flock  0 0
   ```

2. For each Lustre file system you have configured, create the file system mount point in the default boot image template. For example, if `MOUNT_POINT=`*/mnt/filesystem*:

   ```
   smw:~# mkdir -p /opt/xt-images/templates/default/mnt/filesystem
   ```

3. Update the boot image to include these changes.

   Note: You can defer this step and update the boot image **once** before you finish booting the system.

6.4 Configuring Striping on Lustre File Systems

Striping is the process of distributing data from a single file across more than one device. You specify values to create a default striping pattern for each file system when you create the *filesystem*`.fs_defs` configuration file. Cray recommends the following configuration options for striping on Lustre file systems.

- Striping files across two to four OSTs. Setting the stripe count value to 2 gives good performance for many types of jobs. For larger file systems, a larger stripe width may improve performance.
- Choosing the default stripe size of 1 MB (1048576 bytes). You can increase stripe size by powers of two, but there is rarely a need to configure a stripe size greater than 2 MB.

  Note: Do not choose a smaller stripe size, even for files with writes that are smaller than the stripe size. The system caches smaller writes.

In the configuration example described in Procedure 43 on page 125, you configure striping by setting `STRIPE_SIZE` and `STRIPE_COUNT`. For more detailed information on configuring striping, see *Managing Lustre for the Cray Linux Environment (CLE)* or the *Lustre Operations Manual* from Oracle. For additional information about Lustre configuration parameters, see the `lustre.fs_defs`(5) man page.

6.5 Configuring Lustre Failover

Lustre object storage server (OSS) and metadata server (MDS) failover is a service that switches activity from the primary server to a standby server when the primary server fails or the service is temporarily shut down for maintenance. After cabling and configuring the file system, you have the option of creating a failover on a Lustre node to a backup node. This capability is a standard Lustre feature. The CLE release provides a mechanism for configuring Lustre to invoke a failover automatically. With this feature configured, a failing Lustre node alerts a Lustre proxy service (`xt-lustre-proxy`), which in turn, invokes the failover commands without administrator intervention. In addition, you can configure an imperative recovery option for Lustre failover that potentially allows file systems to recover faster.

Implementing failover requires specific cabling and configuration of the storage devices and the Lustre file system. For additional information, see *Managing Lustre for the Cray Linux Environment (CLE)*.

6.6 Configuring Additional Lustre Features

A more complete description of Lustre file systems, instructions to perform ongoing maintenance, and instructions to configure various optional features are provided in *Managing Lustre for the Cray Linux Environment (CLE)*. The following topics are discussed:

- Storage considerations
- Configuring striping for Lustre file systems
- Setting secondary group permissions with `group_upcall`
- Automatic start up and mounting of Lustre
- Lustre system administration
- Lustre failover and failback
- Troubleshooting

Additional information is available by accessing Lustre man pages and Lustre documentation from Oracle listed in Lustre File System Documentation on page 119.

Part II: Update and Upgrade Installations

Preparing to Update or Upgrade CLE Software [7]

Refer to this chapter and the next (Chapter 8, Updating or Upgrading Your CLE Software on page 137) to update or upgrade your Cray Linux Environment (CLE) software in the following scenarios:

Update — A software update installation involves applying an update package for a release that is already running on your system. For example, you can update your system from the CLE 4.0 base release to CLE 4.0.UP03.

Upgrade — A software upgrade installation involves moving to the next release. For example, if your system is currently running CLE 3.1 on a Cray XE system, you can upgrade to release CLE 4.0.

For CLE 4.0, the procedures to install an update package and to upgrade from CLE 3.1 are the same, except that an upgrade requires mounting the `Cray-CLEbase11sp1` DVD to perform the SLES11 to SLES11 SP1 base OS upgrade.

Note: In the following chapters, some examples are left-aligned to better fit the page. Left alignment has no special significance.

7.1 Before You Start the Update or Upgrade Process

Perform the following tasks before you install the CLE release package.

- Read the *README* file provided with the release for any installation-related requirements and corrections to this installation guide.

  Additional installation information may also be included in the following documents: *CLE 4.0 Release Errata*, *Limitations for CLE 4.0*, *Cray Linux Environment (CLE) Software Release Overview*, and *Cray Linux Environment (CLE) Software Release Overview Supplement*.

- Verify that your System Management Workstation (SMW) is running Cray SMW Release 6.0 or later. You must install the SMW 6.0 release or later on your SMW before installing the CLE 4.0 release. If a specific SMW update package is required for your installation, that information is documented in the *README* file provided with the CLE 4.0 release. Type the following command to determine the HSS/SMW version:

```
crayadm@smw:~> cat /opt/cray/hss/default/etc/smw-release
6.0.UP03
```

7.2 Backing Up Your Current Software

Before you install the release package, back up the contents of the system set being updated or upgraded. Use the `xthotbackup` command to back up one system set to a second system set. For more information about using system sets, see About System Set Configuration in `/etc/sysset.conf` on page 48 and the `sysset.conf`(5) man page.

By default, `xthotbackup` copies only the boot node root and shared root file systems. Specify the `-a` option to copy all file systems in the system set (except for swap and Lustre) or specify the `-f` option to select a customized set of file system functions. The `-b` option makes the back-up or destination system set bootable by changing the appropriate boot node and service node entries in `/etc/fstab`. For more information, see the `xthotbackup`(8) man page.

Procedure 45. Backing up current software

Use the `xthotbackup` command to copy the disk partitions in one system set to a back-up system set.

Warning: Do not use the `xthotbackup` command when either the source or destination system set is booted. Running `xthotbackup` with a booted system set or partition could cause data corruption.

1. If the Cray system is booted, use your site-specific procedures to shut down the system. For example, to shutdown using an automation file:

   ```
   crayadm@smw:~> xtbootsys -s last -a auto.xtshutdown
   ```

   For more information about using automation files see the `xtbootsys`(8) man page.

   Although not the preferred method, alternatively execute these commands as `root` from the boot node to shutdown your system.

   ```
   boot:~ # xtshutdown -y
   boot:~ # shutdown -h now;exit
   ```

2. Log on to the SMW as `root`.

   ```
   crayadm@smw:~> su - root
   ```

3. Run the `xthotbackup` command to copy from the source system set to the back-up or destination system set. For example, if *BLUE* is the label for the source system set and *GREEN* is the label for the back-up system set, execute the following command as `root`:

   ```
   smw:~ # xthotbackup -a -b BLUE GREEN
   ```

   Note: The `-a` option specifies all file system functions in the system set (except swap and Lustre). To specify a site-specific set of functions, use the `-f` option.

   `xthotbackup` does not copy the swap partition for the boot node, however, if the `-b` option is specified, `mkswap` is invoked on the swap partition for the boot node in the destination system set to prepare a swap partition.

   For more information, see the `xthotbackup`(8) man page.

You are now ready to begin installing the software release package.

Updating or Upgrading Your CLE Software [8]

You must be running either the CLE 3.1 or the CLE 4.0 release in order to update or upgrade the CLE software on your Cray system by using the procedures in this chapter.

8.1 Before You Begin

All upgrades and configuration changes are installed from the SMW to the `bootroot`, `sharedroot`, and (if applicable) the persistent `/var` file systems before booting the upgraded file systems. These file systems are mounted and modified during the procedure to install the release package.

An update or upgrade release package can be installed to an alternative root location if a system is configured to have more than one system set. A significant portion of the upgrade work can be done without using dedicated time if your Cray system is booted from a different system set. The `/etc/sysset.conf` file describes which devices and disk partitions on the boot RAID are used for which system sets. For more information, see About System Set Configuration in `/etc/sysset.conf` on page 48 and the `sysset.conf`(5) man page.

If you are updating or upgrading a system set that is not running, you do not need to shut down your Cray system before you install the release package.

Warning: If you are updating or upgrading a system set that **is** running, you **must** shut down your Cray system before installing the release package. For more information about system sets and system startup and shutdown procedures, see *Managing System Software for Cray XE and Cray XK Systems* (S–2393).

Warning: If the persistent `/var` file system is shared between multiple system sets, you must verify that it is **not** mounted on the Cray system before you install the release package.

8.2 Installing CLE Release Software on the SMW

Two DVDs are required to install the CLE 4.0 release on a Cray system. The first is labeled `Cray CLE 4.0.UP`*nn* `Software` and contains software specific to Cray systems. Optionally, you may have an ISO image called `xe-sles11sp1-`*4.0.46e04*`.iso`, where *n.n.nn* indicates the CLE release build level, and *avv* indicates the installer version.

To upgrade to the CLE 4.0 release from CLE 3.1 requires a second DVD labeled `Cray-CLEbase11sp1-`*yyyymmdd* and contains the CLE 4.0 base operating system, which is based on SLES 11 SP1.

Procedure 46. Copying the software to the SMW

1. Log on to the SMW as `root`.

   ```
   crayadm@smw:~> su - root
   ```

2. Mount the release media by using one of the following commands, depending on your media type.

   If installing the release package from disk, place the `Cray CLE 4.0.UP`*nn* `Software` DVD in the CD/DVD drive and mount it.

   ```
   smw:~# mount /dev/cdrom /media/cdrom
   ```

   Or

   To mount the release media using the ISO image, execute the following command, where `xe-sles11sp1-`*4.0.46e04*`.iso` is the path name to the ISO image file.

   ```
   smw:~# mount -o loop,ro xe-sles11sp1-4.0.46e04.iso /media/cdrom
   ```

3. Copy all files to a directory on the SMW in `/home/crayadm/install.`*xtrel*, where `xtrel` is a site-determined name specific to the release being installed. For example:

   ```
   smw:~# mkdir /home/crayadm/install.4.0.46
   smw:~# cp -pr /media/cdrom/* /home/crayadm/install.4.0.46
   ```

4. Unmount the `Cray CLE 4.0.UP`*nn* `Software` media.

   ```
   smw:~# umount /media/cdrom
   ```

5. If upgrading from CLE 3.1 to CLE 4.0, you must mount the SLES 11 SP1 base media. Insert the `Cray-CLEbase11sp1` DVD into the SMW DVD drive and mount it.

   ```
   smw:~# mount /dev/cdrom /media/cdrom
   ```

   Or

   To mount the base operating system media using the ISO image, execute the following command, where `Cray-CLEbase11sp1-`*yyyymmdd*`.iso` is the path name to the ISO image file.

```
smw:~# mount -o loop,ro Cray-CLEbase11sp1-yyyymmdd.iso /media/cdrom
```

Procedure 47. Running CRAYCLEinstall.sh

1. As root, execute the installation script to update or upgrade the Cray CLE software on the SMW.

   ```
   smw:~# /home/crayadm/install.4.0.46/CRAYCLEinstall.sh \
   -m /home/crayadm/install.4.0.46 -u -v -w
   ```

2. At the prompt 'Do you wish to continue?', type **y** and press Enter.

   The output of the installation script is displayed to the console.

   > **Note:** If this script fails, you can restart it with the same options. However, rerunning this script may generate numerous error messages as it attempts to install already-installed RPMs. You may safely ignore these messages.

8.3 Preparing the Configuration for Software Installation

You may need to update the CLEinstall.conf configuration file. The CLEinstall.conf file that was created during the first installation of this system can be used during an installation to the alternative root location. For a description of the contents of this file, see Chapter 4, About Installation Configuration Files on page 25 or the CLEinstall.conf(5) man page.

Based on the settings you choose in the CLEinstall.conf file, the CLEinstall program then updates other configuration files. A template CLEinstall.conf is provided on the distribution media. Your site-specific copy is located in the installation directory from the previous installation; for example /home/crayadm/install.*3.1.60*/CLEinstall.conf.

> **Warning:** Any configuration data which is in CLEinstall.conf that was manually changed on a system after the last software update must be kept up to date before running CLEinstall for an upgrade or an update. Doing so will avoid spending much time tracking down problems that could have been avoided.

> **Note:** If problems with the hosts file are detected after the update or upgrade, you may need to use the copies of /etc/hosts that CLEinstall saves on bootroot, and /opt/xt-images/templates/default/etc with hosts.preinstall.$$ and hosts.postinstall.$$.

Procedure 48. Preparing the CLEinstall.conf configuration file

1. If you have an existing CLEinstall.conf file, use the diff command to compare it to the template in /home/crayadm/install.*xtrel*. For example:

```
smw:~# diff /home/crayadm/install.4.0.46/CLEinstall.conf \
 /home/crayadm/install.3.1.60/CLEinstall.conf
21c21
< xthostname=mycray
---
> xthostname=crayhostname
24c24
< node_class_login_hostname=mycray
---
> node_class_login_hostname=crayhostname
smw:~ #
```

   > **Note:** The CLEinstall program generates INFO messages suggesting that you remove deprecated parameters from your local CLEinstall.conf file.

2. Edit the CLEinstall.conf file in the temporary directory /home/crayadm/install.*xtrel* and make necessary changes to enable any new features you are configuring for the first time with this system software upgrade.

   > **Note:** The CLEinstall program checks that the /etc/opt/cray/sdb/node_classes file and the node_class[*] parameters in CLEinstall.conf agree. If you made changes to /etc/opt/cray/sdb/node_classes since your last CLE software installation or upgrade, make the same changes to CLEinstall.conf.

```
smw:~# cp -p /home/crayadm/install.4.0.46/CLEinstall.conf \
/home/crayadm/install.4.0.46/CLEinstall.conf.save
smw:~# chmod 644 /home/crayadm/install.4.0.46/CLEinstall.conf
smw:~# vi /home/crayadm/install.4.0.46/CLEinstall.conf
```

   For a complete description of the contents of this file, see Chapter 4, About Installation Configuration Files on page 25.

   > **Tip:** Use the rtr --system-map command to translate between NIDs and physical ID names.

8.4 Running the CLEinstall Installation Program

The CLEinstall installation program upgrades the CLE software for your configuration by using information in the CLEinstall.conf and sysset.conf configuration files.

> **Important:** CLEinstall modifies Cray system entries in /etc/hosts each time you update or upgrade your CLE software. For additional information, see Maintaining Node Class Settings and Hostname Aliases on page 27.

> **Important:** During a CLE update or upgrade, `CLEinstall` disables the execution bits of all scripts in the `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly`, and `/etc/cron.monthly` directories of the `bootroot` and default view of the shared root with a `chmod uog-x` command. If there are site-specific `cron` scripts in these directories, you will need to re-enable the execute permission on them after doing a CLE update or upgrade. Any scripts in these directories which have been node-specialized or class-specialized via `xtopview` will not be changed by the CLE update or upgrade. Only the `bootroot` and the default view of the shared root will be modified.

The following `CLEinstall` options are required or recommended for this type of installation:

`--upgrade` Specify that this is an update or upgrade rather than a full system installation.

`--label=`*system_set_label*

Specify the system set that you are using to install the release.

`--XTrelease=`*release_number*

Specify the target CLE release and build level, for example 4.0.46.

`--CLEmedia=`*directory*

Specify the directory on the SMW where you copied the CLE software media. For example, `/home/crayadm/install.`*release_number*.

`--configfile=`*CLEinstall_configuration_file*

Specify the path to the `CLEinstall.conf` file that you edited in Procedure 48 on page 140.

For a full description of the `CLEinstall` command options and arguments, see Running the `CLEinstall` Installation Program on page 57 or the `CLEinstall`(8) man page.

Procedure 49. Running `CLEinstall`

1. Invoke the `CLEinstall` program on the SMW. `CLEinstall` is located in the directory you created in Procedure 46 on page 138.

   ```
   smw:~# /home/crayadm/install.4.0.46/CLEinstall --upgrade \
   --label=system_set_label --XTrelease=4.0.46 \
   --configfile=/home/crayadm/install.4.0.46/CLEinstall.conf \
   --CLEmedia=/home/crayadm/install.4.0.46
   ```

   If upgrading from CLE 3.1 to CLE 4.0, you must mount the SLES 11 SP1 base media prior to running `CLEinstall`. If you do not mount the SLES 11 SP1 base media prior to running `CLEinstall`, you will get the following error:

```
09:18:59 FATAL: The Basemedia directory at /media/cdrom does not appear to have the CLE \
Base operating system software required to upgrade the base operating system from \
ULC11SP0 to ULC11SP1.
09:18:59 CLEinstall has completed with fatal errors.
```

2. Examine the initial messages and note the process ID (`pid`) of the `CLEinstall` process. Log files are created in `/var/adm/cray/logs` and named by using this `pid`. For example:

```
09:20:21 Installation output will be captured in /var/adm/cray/logs/\
CLEinstall.stdout.27670
09:20:21 Installation errors (stderr) will be captured in /var/adm/cray/logs/\
CLEinstall.stderr.27670
09:20:21 Installation debugging messages will be captured in /var/adm/cray/logs/\
CLEinstall.debug.27670
```

3. `CLEinstall` validates `sysset.conf` and `CLEinstall.conf` configuration settings and then confirms the expected status of your boot node and file systems.

   Confirm that the installation is proceeding as expected, respond to warnings and prompts, and resolve any issues. For example:

   - If you are installing to a system set that is not running, and you did not shut down your Cray system, respond to the following warning and prompt:

     ```
     WARNING: Your bootnode is booted.  Please confirm that the
     system set you are intending to update is not booted.
     Do you wish to proceed?[n]:
     ```

Warning: If the boot node has a file system mounted and `CLEinstall` on the SMW creates a new file system on that disk partition, the running system will be corrupted.

- If you have configured file systems that are shared between two system sets, respond to the following warning and prompt to confirm creation of new file systems:

```
09:21:24 INFO: The PERSISTENT_VAR disk function for the LABEL system set is marked shared.
09:21:24 INFO: The /dev/sdr1 disk partition will be mounted on the SMW for PERSISTENT_VAR
disk function.  Confirm that it is not mounted on any nodes in a running XE
system before continuing.
Do you wish to proceed?[n]:y
```

- If the `node_class[`*idx*`]` parameters do not match the existing `/etc/opt/cray/sdb/node_classes` file, you are asked to confirm that your hardware configuration has changed. If your hardware has not changed, abort `CLEinstall` and correct the node class configuration in `CLEinstall.conf` and/or the `node_classes` file. Respond to the following warning and prompt:

```
09:21:41 INFO: There are 5 WARNINGs about discrepancies between CLEinstall.conf
and /etc/opt/cray/sdb/node_classes
09:21:41 INFO: If you ARE doing a migration from XT to XE, then you may proceed and CLEi
nstall will adjust the /etc/opt/cray/sdb/node_classes file to match the setting
s in CLEinstall.conf and may remove some node-specialized files from the shared
root specialized /etc.
09:21:41 INFO: If you ARE NOT doing a migration from XT to XE, then stop CLEinstall now
to correct the problem.
Do you wish to proceed?[n]:
```

`CLEinstall` may resolve some issues after you indicate that you want to proceed; for example, disk devices are already mounted, boot image file or links already exist, HSS daemons are stopped on the SMW.

Caution: Some problems can be resolved only through manual intervention via another terminal window or by rebooting the SMW; for example, a process is using a mounted disk partition, preventing `CLEinstall` from unmounting the partition.

4. After you have resolved all issues, complete these steps.

   a. Monitor the debug output. Create another terminal window and invoke the `tail` command by using the path and *pid* displayed in step 2 (after `CLEinstall` was invoked).

   ```
   smw:~# tail -f /var/adm/cray/logs/CLEinstall.debug.pid
   ```

   b. In the `CLEinstall` console window, locate the following prompt and type **`y`**.

```
*** Preparing to UPGRADE software on system set label system_set_label.
Do you wish to proceed? [n]
```

5. The `CLEinstall` program now installs the release software.

   Note: This command runs for 30 minutes or more for updates and 90 minutes for an upgrade, depending on your system configuration.

   Monitor the output to ensure that your installation is proceeding without error.

   Note: Several error messages from the `tar` command are displayed as the persistent `/var` is updated for each service node. You may safely ignore these messages.

6. Confirm that the `CLEinstall` program has completed successfully.

   On completion, the `CLEinstall` program generates a list of suggested commands to be run as the next steps in the update or upgrade process. These commands are customized, based on the variables in the `CLEinstall.conf` and `sysset.conf` files, and include runtime variables such as PID numbers in filenames.

Complete the installation and configuration of your Cray system by using both the commands that the `CLEinstall` program provides and the information in the remaining sections of this chapter.

As you complete these procedures, you can cut and paste the suggested commands from the output window or from the window created in step 4, which tailed the debug file. The log files created in `/var/adm/cray/logs` for `CLEinstall.stdout.`*pid* and `CLEinstall.debug.`*pid* also contain the suggested commands.

8.5 Creating Boot Images

The Cray CNL compute nodes and Cray service nodes use RAM disks for booting. Service nodes and CNL compute nodes use the same `initramfs` format and workspace environment. This space is created in `/opt/xt-images/`*machine-xtrelease-partition/nodetype*, where *machine* is the Cray hostname, *xtrelease* is the build level for the CLE release, *partition* describes either the full machine or a system partition, and *nodetype* is either `compute` or `service`.

⚠ **Caution:** Existing files in `/opt/xt-images/templates/default` are copied into the new bootimage work space. In most cases, you can use the older version of the files with your updated or upgraded system. However, some file content may have changed with the new release; you must verify that site-specific modifications are compatible. For example, you can use existing copies of `/etc/hosts`, `/etc/passwd` and `/etc/modprobe.conf`, but if you changed `/init` for the template, the site-modified version that is copied and used for CLE 4.0 may cause a boot failure.

Follow the procedures in this section to prepare the work space in `/opt/xt-images`. For more information about configuring boot images for service and compute nodes, see the `xtclone`(8) and `xtpackage`(8) man pages.

Procedure 50. Preparing compute and service node boot images

The `shell_bootimage_`*LABEL*`.sh` script prepares boot images for the system set specified by *LABEL*. For example, if your system set has the label *BLUE* in `/etc/sysset.conf`, invoke `shell_bootimage_`*BLUE*`.sh` to prepare a boot image. This script uses `xtclone` and `xtpackage` to prepare the work space in `/opt/xt-images`.

For more information about configuring boot images for service and compute nodes, see the `xtclone`(8) and `xtpackage`(8) man pages.

1. Log on to the SMW as `root`.

   ```
   crayadm@smw:~> su - root
   ```

2. Run the `shell_bootimage_`*LABEL*`.sh` script, where *LABEL* is the system set label specified in `/etc/sysset.conf` for this boot image.

   Specify the `-c` option to automatically create and set the boot image for the next boot. For example, if the system set label is *BLUE*:

   ```
   smw:~# /var/opt/cray/install/shell_bootimage_BLUE.sh -c
   ```

   For information about additional options accepted by this script, use the `-h` option to display a help message.

Procedure 51. Enabling boot-node failover

Optional: Boot-node failover is an optional CLE feature.

If you have configured boot-node failover for the first time, follow these steps. If you did not configure boot-node failover, skip this procedure.

To enable boot-node failover, you must set `bootnode_failover` parameters in the `CLEinstall.conf` file before you run the `CLEinstall` program. For more information, see Configuring Boot-node Failover on page 35.

In this example, the primary boot node is *c0-0c0s0n1* (`node_boot_primary=1`) and the backup or alternate boot node is *c0-0c1s1n1* (`node_boot_alternate=61`).

Tip: Use the `rtr --system-map` command to translate between NIDs and physical ID names.

1. As `crayadm` on the SMW, halt the primary and alternate boot nodes.

   Warning: Verify that your system is shut down before you invoke the `xtcli halt` command.

   ```
   crayadm@smw:~> xtcli halt c0-0c0s0n1,c0-0c1s1n1
   ```

2. Specify the primary and backup boot nodes in the boot configuration.

   If the partition variable in `CLEinstall.conf` is `s0`, type the following command to select the boot node for the entire system.

   ```
   crayadm@smw:~> xtcli boot_cfg update -b c0-0c0s0n1,c0-0c1s1n1
   ```

   Or

   If the partition variable in `CLEinstall.conf` is a partition value such as `p0`, `p1`, and so on, type the following command to select the boot node for the designated partition.

   ```
   crayadm@smw:~> xtcli part_cfg update pN -b c0-0c0s0n1,c0-0c1s1n1
   ```

3. To use boot-node failover, you must enable the STONITH capability on the blade or module of the primary boot node. Use the `xtdaemonconfig` command to determine the current STONITH setting.

   ```
   crayadm@smw:~> xtdaemonconfig c0-0c0s0n1 | grep stonith
   c0-0c0s0: stonith=false
   crayadm@smw:~>
   ```

   > **Note:** If you have a partitioned system, invoke `xtdaemonconfig` with the `--partition p`*n* option.

   > **Caution:** STONITH is a per blade setting, not a per node setting. You must ensure that your primary boot node is not assigned to a blade that hosts services with conflicting STONITH requirements, such as Lustre.

4. To enable STONITH on your primary boot node blade, type the following command:

   ```
   crayadm@smw:~> xtdaemonconfig c0-0c0s0 stonith=true
   c0-0c0s0: stonith=true
   crayadm@smw:~> xtcli halt c0-0c0s0n1, c0-0c1s1n1
   crayadm@smw:~> xtcli boot_cfg update -b c0-0c0s0n1, c0-0c1s1n1
   crayadm@smw:~>
   ```

   > **Note:** If you have a partitioned system, invoke `xtdaemonconfig` with the `--partition p`*n* option.

Procedure 52. Enabling SDB node failover

> **Optional:** SDB node failover is an optional CLE feature.

If you have configured SDB node failover for the first time, follow these steps. If you did not configure SDB node failover, skip this procedure.

> **Note:** In addition to this procedure, refer to Procedure 42 on page 116 **after** you have completed the remaining configuration steps and have booted and tested your system.

To enable SDB node failover, you must set `sdbnode_failover` parameters in the `CLEinstall.conf` file before you run the `CLEinstall` program. For more information, see Configuring SDB Node Failover on page 37.

In this example, the primary SDB node is *c0-0c0s2n1* (`node_sdb_primary=5`) and the backup or alternate SDB node is *c0-0c1s3n1* (`node_sdb_alternate=57`).

Tip: Use the `rtr --system-map` command to translate between NIDs and physical ID names.

1. Invoke `xtdaemonconfig` to determine the current STONITH setting on the blade or module of the primary SDB node. For example:

   ```
   crayadm@smw:~> xtdaemonconfig c0-0c0s2n1 | grep stonith
   c0-0c0s2: stonith=false
   crayadm@smw:~>
   ```

   Note: If you have a partitioned system, invoke `xtdaemonconfig` with the `--partition p`*n* option.

   Caution: STONITH is a per blade setting, not a per node setting. You must ensure that your primary SDB node is not assigned to a blade that hosts services with conflicting STONITH requirements, such as Lustre.

2. Enable STONITH on your primary SDB node. For example:

   ```
   crayadm@smw:~> xtdaemonconfig c0-0c0s2n1 stonith=true
   c0-0c0s2: stonith=true
   The expected response was received.
   crayadm@smw:~>
   ```

   Note: If you have a partitioned system, invoke `xtdaemonconfig` with the `--partition p`*n* option.

3. Specify the primary and backup SDB nodes in the boot configuration.

   For example, if the partition variable in `CLEinstall.conf` is `s0`, type the following command to select the primary and backup SDB nodes.

   ```
   crayadm@smw:~> xtcli halt c0-0c0s2n1,c0-0c1s3n1
   crayadm@smw:~> xtcli boot_cfg update -d c0-0c0s2n1,c0-0c1s3n1
   ```

   Or

   If the partition variable in `CLEinstall.conf` is a partition value such as `p0`, `p1`, and so on, type the following command:

   ```
   crayadm@smw:~> xtcli part_cfg update pN -d c0-0c0s2n1,c0-0c1s3n1
   ```

Procedure 53. Running post-CLEinstall commands

1. Unmount and eject the release software DVD from the SMW DVD drive if it is still loaded.

   ```
   smw:~# umount /media/cdrom
   smw:~# eject
   ```

2. Run the `shell_post_install.sh` script on the SMW to unmount the boot root and shared root file systems and perform other cleanup as needed.

```
smw:~# /var/opt/cray/install/shell_post_install.sh /bootroot0 /sharedroot0
```

Warning: Exercise care when you mount and unmount file systems. If you mount a file system on the SMW and boot node simultaneously, you may corrupt the file system.

3. Confirm that the `shell_post_install.sh` script successfully unmounted the boot root and shared root file systems.

   If a file system does not unmount successfully, the script displays information about open files and associated processes (by using the `lsof` and `fuser` commands). Attempt to terminate processes with open files and if necessary, reboot the SMW to resolve the problem.

8.6 Updating the SDB Database Schema

In general, MySQL database schema changes to the Service Database (SDB) are restricted to major CLE releases. However, in the event that it is necessary to update the SDB schema to resolve a critical problem, a schema change may be included in an update package. The *README* file included with the release package will document this change. In addition, the `CLEinstall` program detects that the SDB schema needs to be updated and generates a list of instructions similar to the procedure in Appendix D, Upgrading the SDB Schema on page 165.

If the release package you are installing requires an update to the SDB schema, follow Appendix D, Upgrading the SDB Schema on page 165 before continuing.

Note: If you are upgrading from CLE 3.1 to CLE 4.0, you will be required to update the SDB database schema.

8.7 Configuring Optional Services

If you enabled an optional service you were not previously using in Procedure 48 on page 140, you may need to perform additional configuration steps. Follow the procedures in the appropriate optional section in Chapter 5, Installing CLE on a New System on page 53 or in *Managing System Software for Cray XE and Cray XK Systems*.

If you configured an optional CLE feature or service during a **previous** installation or upgrade, no additional steps are required.

8.8 Booting and Testing the System

Important: If you configured optional services for the first time during this update or upgrade and deferred updating the boot image, update the boot image now by following Procedure 50 on page 145.

Your system is now upgraded.

Procedure 54. Rebooting the Cray System

1. Use your site-specific procedures to shut down the system. For example, to shutdown using an automation file type the following:

   ```
   crayadm@smw:~> xtbootsys -s last -a auto.xtshutdown
   ```

   For more information about using automation files see the `xtbootsys`(8) man page.

   Although not the preferred method, alternatively execute these commands as `root` from the boot node to shutdown your system.

   ```
   boot:~ # xtshutdown -y
   boot:~ # shutdown -h now;exit
   ```

2. Edit the boot automation file and make site-specific changes as needed.

   ```
   crayadm@smw:~> vi /opt/cray/etc/auto.xthostname
   ```

3. Use the `xtbootsys` command to boot the Cray system.

⚠ **Caution:** You must shut down your Cray system before you invoke the `xtbootsys` command. If you are installing to an alternate system set, you must shut down the currently running system before you boot the new boot image.

Type this command to boot the entire system.

```
crayadm@smw:~> xtbootsys -a auto.xthostname
```

Or

Type this command to boot a partition.

```
crayadm@smw:~> xtbootsys --partition pN -a auto.xthostname
```

Procedure 55. Testing the system for basic functionality

1. If the system was shut down by using `xtshutdown`, remove the `/etc/nologin` file from all service nodes to permit a non-root account to log on.

   ```
   smw:~# ssh root@boot
   boot:~ # xtunspec -r /rr/current -d /etc/nologin
   ```

2. Log on to the login node as `crayadm`.

   ```
   boot:~ # ssh crayadm@login
   ```

3. Use system-status commands, such as `xtnodestat`, `xtprocadmin`, and `apstat`.

   The `xtnodestat` command displays the current allocation and status of the compute nodes, organized by physical cabinet. The last line of the output shows the number of available compute nodes.

```
crayadm@login:~> xtnodestat
Current Allocation Status at Fri May 21 07:11:48 2010

     C0-0     C1-0     C2-0     C3-0
  n3 ;;;;;;;; ;;;;;;;; ;;;;;;;; ;;;;;;;;
  n2 ;;;;;;;; ;;;;;;;; ;;;;;;;; ;;;;;;;;
  n1 ;;;;;;;; ;;;;;;;; ;;;;;;;; ;;;;;;;;
c2n0 ;;;;;;;; ;;;;;;;; ;;;;;;;; ;;;;;;;;
  n3 ;;;;;;;; ;;;;;;;; ;S;S;S;S ;;;;;;;;
  n2 ;;;;;;;; ;;;;;;;; ;S;S;S;S ;;;;;;;;
  n1 ;;;;;;;; ;;;;;;;; ;S;S;S;S ;;;;;;;;
c1n0 ;;;;;;;; ;;;;;;;; ;S;S;S;S ;;;;;;;;
  n3 S;S;S;S; ;;;;;;;; ;;;;;;;; ;;;;;;;;
  n2 S;S;S;S; ;;;;;;;; ;;;;;;;; ;;;;;;;;
  n1 S;S;S;S; ;;;;;;;; ;;;;;;;; ;;;;;;;;
c0n0 S;S;S;S; ;;;;;;;; ;;;;;;;; ;;;;;;;;
    s01234567 01234567 01234567 01234567

Legend:
   nonexistent node                 S  service node
;  free interactive compute node    -  free batch compute node
A  allocated, but idle compute node ?  suspect compute node
X  down compute node                Y  down or admindown service node
Z  admindown compute node

Available compute nodes:        352 interactive,          0 batch
```

The `xtprocadmin` command displays the current values of processor flags and node attributes.

```
crayadm@login:~> xtprocadmin
   NID    (HEX)    NODENAME     TYPE    STATUS        MODE
     0      0x0  c0-0c0s0n0  service        up       other
     1      0x1  c0-0c0s0n1  service        up interactive
     2      0x2  c0-0c0s1n0  compute        up interactive
     3      0x3  c0-0c0s1n1  compute        up interactive
     4      0x4  c0-0c0s2n0  service        up interactive
     5      0x5  c0-0c0s2n1  service        up interactive
     6      0x6  c0-0c0s3n0  compute        up interactive
     7      0x7  c0-0c0s3n1  compute        up interactive
     8      0x8  c0-0c0s4n0  service        up interactive
     9      0x9  c0-0c0s4n1  service        up       other
    10      0xa  c0-0c0s5n0  compute        up interactive
    11      0xb  c0-0c0s5n1  compute        up interactive
    12      0xc  c0-0c0s6n0  service        up interactive
    13      0xd  c0-0c0s6n1  service        up interactive
    14      0xe  c0-0c0s7n0  compute        up interactive
    15      0xf  c0-0c0s7n1  compute        up interactive
    16     0x10  c0-0c0s7n2  compute        up interactive
    17     0x11  c0-0c0s7n3  compute        up interactive
    18     0x12  c0-0c0s6n2  service        up interactive
    19     0x13  c0-0c0s6n3  service        up interactive
    20     0x14  c0-0c0s5n2  compute        up interactive
    21     0x15  c0-0c0s5n3  compute        up interactive
    22     0x16  c0-0c0s4n2  service        up       other
    23     0x17  c0-0c0s4n3  service        up       other
    24     0x18  c0-0c0s3n2  compute        up interactive
    25     0x19  c0-0c0s3n3  compute        up interactive
    26     0x1a  c0-0c0s2n2  service        up interactive
    27     0x1b  c0-0c0s2n3  service        up interactive
    28     0x1c  c0-0c0s1n2  compute        up interactive
    29     0x1d  c0-0c0s1n3  compute        up interactive
    30     0x1e  c0-0c0s0n2  service        up interactive
    31     0x1f  c0-0c0s0n3  service        up interactive
    32     0x20  c0-0c1s0n0  compute        up interactive
    33     0x21  c0-0c1s0n1  compute        up interactive
    34     0x22  c0-0c1s1n0  compute        up interactive
    35     0x23  c0-0c1s1n1  compute        up interactive
    36     0x24  c0-0c1s2n0  compute        up interactive
    37     0x25  c0-0c1s2n1  compute        up interactive
    38     0x26  c0-0c1s3n0  compute        up interactive
    39     0x27  c0-0c1s3n1  compute        up interactive
    40     0x28  c0-0c1s4n0  compute        up interactive
...
```

The `apstat` command displays the current status of all applications running on the system.

```
crayadm@login:~> apstat -v
Compute node summary
    arch config     up    use   held  avail   down
      XT    352    352      0      0    362      0

No pending applications are present

No placed applications are present
```

4. Run a simple job on the compute nodes.

   At the conclusion of the installation process, the `CLEinstall` program provides suggestions for runtime commands and indicates how many compute nodes are available for use with the `aprun -n` option.

   > **Note:** For `aprun` to work cleanly, the current working directory on the login node should also exist on the compute node. Change your current working directory to either `/tmp` or to a directory on a mounted Lustre file system.

   For example, type the following.

```
crayadm@login:~> cd /tmp
crayadm@login:~> aprun -b -n 16 -N 1 /bin/cat /proc/sys/kernel/hostname
```

   This command returns the hostname of each of the 16 computes nodes used to execute the program.

```
nid00002
nid00003
nid00006
nid00007
nid00010
nid00011
nid00014
nid00016
nid00015
nid00017
nid00020
nid00021
nid00025
nid00024
nid00028
nid00029
Application 108 resources: utime ~0s, stime ~0s
```

5. Test file system functionality. For example, if you have a Lustre file system named */mylusmnt/filesystem*, type the following.

```
crayadm@login:~> cd /mylusmnt/filesystem
crayadm@login:/mylustremnt/filesystem> echo lustretest > testfile
crayadm@login:/mylustremnt/filesystem> aprun -b -n 5 -N 1 /bin/cat ./testfile
lustretest
lustretest
lustretest
lustretest
lustretest
Application 109 resources: utime ~0s, stime ~0s
```

6. Test the optional features that you have configured on your system.

   a. To test RSIP functionality, log on to an RSIP client node (compute node) and ping the IP address of the SMW or other host external to the Cray system. For example, if `c0-0c0s7n2` is an RSIP client, type the following commands.

```
crayadm@login:~> exit
boot:~ # ssh root@c0-0c0s7n2
root@c0-0c0s7n2's password:
Welcome to the initramfs
# ping 172.30.14.55
172.30.14.55 is alive!
# exit
Connection to c0-0c0s7n2 closed.
boot:~ # exit
```

Note: RSIP clients on the compute nodes make connections to the RSIP server(s) during system boot. Initiation of these connections is staggered at a rate of 10 clients per second, until they are all connected; during that time, connectivity over RSIP tunnels is unreliable. Avoid using RSIP services for several minutes following a system boot.

   b. To check the status of DVS, type the following command on the DVS server node.

```
crayadm@login:~> ssh root@nid00019 /etc/init.d/dvs status
DVS service: ..running
```

To test DVS functionality, invoke the mount command on any compute node.

```
crayadm@login:~> ssh root@c0-0c0s7n2 mount | grep dvs
/dvs-shared on /dvs type dvs (rw,blksize=16384,nodename=c0-0c0s4n3,nocache,nodatasync,\
retry,userenv,clusterfs,maxnodes=1,nnodes=1)
```

Create a test file on the DVS mounted file system. For example, type the following.

```
crayadm@login:~> cd /dvs
crayadm@login:/dvs> echo dvstest > testfile
crayadm@login:/dvs> aprun -b -n 5 -N 1 /bin/cat ./testfile
dvstest
dvstest
dvstest
dvstest
dvstest
Application 121 resources: utime ~0s, stime ~0s
```

7. Following a successful installation, the file `/etc/opt/cray/release/clerelease` is populated with the installed release level. For example,

   ```
   crayadm@login:~> cat /etc/opt/cray/release/clerelease
   4.0.UP03
   ```

If the preceding simple tests ran successfully, the system is operational. Cray recommends that you use the `xthotbackup` utility to create a backup of your newly updated or upgraded system. For more information, see the `xthotbackup`(8) man page.

Installing Additional Software [A]

A.1 Installing the Cray Application Developer's Environment

The Cray Application Developer's Environment (CADE) is available from Cray Inc. as a separate software package.

CADE consists of the basic libraries and components needed to develop and compile code on Cray systems, including the GNU Fortran, C, and C++ compilers. This package does **not** include the Cray Compiling Environment (CCE) or compilers from the Portland Group (PGI), Intel, or PathScale. All compilers other than the GNU compilers are sold, installed, and licensed separately.

For installation and upgrade instructions, see the *Cray Application Developer's Environment Installation Guide* (S–2465).

A.2 Installing Cray Performance Analysis Tools

CrayPat and Cray Apprentice2 are available from Cray Inc. as part of a separate software package, Cray Performance Analysis Tools. For installation and upgrade instructions, see *Cray Performance Analysis Tools Release Overview and Installation Guide* (S–2474).

A.3 Installing a Batch System

Batch system software products for Cray systems are available by contacting the appropriate vendor. For more information about these products see the following websites.

PBS Professional:	Altair Engineering, Inc.	http://www.altair.com
Moab and TORQUE:	Adaptive Computing	http://www.adaptivecomputing.com
Platform LSF:	Platform Computing Corporation	http://www.platform.com

For the most up-to-date information regarding batch system software compatibility with CLE releases, access the **3rd Party Batch SW** link on the CrayPort website at http://crayport.cray.com.

Note: PBS Professional uses a license manager. You must have a network connection between the license server and the SDB node in order to use the license manager for PBS Professional on a Cray system. For information, see *Managing System Software for Cray XE and Cray XK Systems*.

A.4 Installing Optional Compilers

The following compilers are available for Cray systems. They are sold, installed, and licensed separately.

Cray Compiling Environment (CCE):	Cray Inc.	*Cray Compiling Environment Release Overview and Installation Guide* (S–5212)
Chapel Compiler:	Cray Inc.	http://chapel.cray.com
PGI Compiler:	The Portland Group, Inc.	http://www.pgroup.com
PathScale Compiler Suite:	PathScale Inc.	http://www.pathscale.com
Intel Composer XE for Linux:	Intel Corporation	http://software.intel.com

Installing RPMs [B]

A variety of software packages are distributed as standard Linux RPM Package Manager (RPM) packages. RPM packages are self-contained installation files that must be executed with the `rpm` command in order to create all required directories and install all component files in the correct locations.

B.1 Generic RPM Usage

To install RPMs on a Cray system, you must use `xtopview` on the boot node to access and modify the shared root. The `rpm` command is not able to modify the RPM database from a login node or other service node; the root directory is read-only from these nodes.

Any changes to the shared root apply to all service nodes. If the RPM you are installing modifies files in `/etc`, you must invoke `xtopview` to perform any class or node specialization that may be required. `xtopview` specialization applies only to `/etc` in the shared root.

For some Cray distributed RPMs, you can set the `CRAY_INSTALL_DEFAULT` environment variable to configure the new version as the default. Set this variable before you install the RPM. For more information, see the associated installation guide.

For more information on installing RPMs, see the `xtopview`(8) man page and the installation documentation for the specific software package you are installing.

Example 9. Installing an RPM on the SMW

As `root`, use the following command:

```
smw:~# rpm -ivh /directorypath/filename.rpm
```

Example 10. Installing an RPM on the boot node root

As `root`, use the following command:

```
boot:~ # rpm -ivh /directorypath/filename.rpm
```

Example 11. Installing an RPM on the shared root

As `root`, use the following commands:

> **Note:** If the SDB node has not been started, you must include the `-x /etc/opt/cray/sdb/node_classes` option when you invoke the `xtopview` command.

```
boot:~ # cp -a /tmp/filename.rpm /rr/current/software
boot:~ # xtopview
default/:/ # rpm -ivh /software/filename.rpm
```

Updating the Time Zone [C]

When you install the Cray Linux Environment (CLE) operating system, the Cray system time is set at US/Central Standard Time (CST), which is six hours behind Greenwich Mean Time (GMT). You can change this time.

> **Note:** When a Cray system is initially installed, the time zone set on the SMW is copied to the boot root, shared root and CNL boot images.

To change the time zone on the SMW, L0 controller, L1 controller, boot root, shared root or for a CNL image, follow the appropriate procedure below.

Procedure 56. Changing the time zone for the SMW and the L1 and L0 controllers

> **Warning:** Perform this procedure while the Cray system is shut down; do not flash L0 and L1 controllers while the Cray system is booted.

You must be logged on as `root`. In this example, the time zone is changed from `"America/Chicago"` to `"America/New_York"`.

1. Ensure the L0 and L1 controllers are responding.

   ```
   smw:~ # xtalive -a l0sysd s0
   ```

2. Check the current time zone setting for the SMW and controllers.

   ```
   smw:~ # date
   Wed Aug 25 21:30:06 CDT 2010

   smw:~ # xtrsh -l root -s /bin/date s0
   c0-0c0s2 : Wed Aug 25 21:30:51 CDT 2010
   c0-0c0s5 : Wed Aug 25 21:30:51 CDT 2010
   c0-0c0s7 : Wed Aug 25 21:30:51 CDT 2010
   c0-0c1s1 : Wed Aug 25 21:30:51 CDT 2010
   .
   .
   .
   c0-0 : Wed Aug 25 21:30:52 CDT 2010
   ```

3. Verify that the `zone.tab` file in the `/usr/share/zoneinfo` directory contains the time zone you want to set.

   ```
   smw:~ # grep America/New_York /usr/share/zoneinfo/zone.tab
   US      +404251-0740023 America/New_York      Eastern Time
   ```

4. Create the time conversion information files.

   ```
   smw:~ # date
   Wed Aug 25 21:32:52 CDT 2010
   smw:~ # /usr/sbin/zic -l America/New_York
   smw:~ # date
   Wed Aug 25 22:33:05 EDT 2010
   ```

5. Modify the `clock` file in the `/etc/sysconfig` directory to set the `DEFAULT_TIMEZONE` and the `TIMEZONE` variables to the new time zone.

   ```
   smw:/etc/sysconfig # grep TIMEZONE /etc/sysconfig/clock
   TIMEZONE="America/Chicago"
   DEFAULT_TIMEZONE="US/Eastern"
   smw:~ # vi /etc/sysconfig/clock
   make changes
   smw:~ # grep TIMEZONE /etc/sysconfig/clock
   TIMEZONE="America/New_York"
   DEFAULT_TIMEZONE="US/Eastern"
   ```

6. Copy the `/etc/localtime` directory to `/opt/tfptboot` and restart `rsms`.

   ```
   smw:~ # cp /etc/localtime /opt/tftpboot
   smw:~ # /etc/init.d/rsms restart
   ```

7. If this is the first time the time zone has been modified, complete this step. If the time zone has been changed already, skip this step and perform step 8.

   a. Exit from the `root` login.

      ```
      smw:~ # exit
      ```

   b. Erase the flash memory of the L1s and flash the updated time zone.

      ```
      crayadm@smw:~> fm -w -t l1
      crayadm@smw:~> xtflash -t l1
      ```

   c. Erase the flash memory of the L0s and flash the updated time zone.

      ```
      crayadm@smw:~> fm -w -t l0
      crayadm@smw:~> xtflash -t l0
      ```

   d. Check the current time zone setting for the SMW and controllers.

      ```
      crayadm@smw:~> date
      Wed Aug 25 23:07:07 EDT 2010
      crayadm@smw:~> xtrsh -l root -s /bin/date s0
      c0-0c1s1 : Wed Aug 25 23:07:16 EDT 2010
      c0-0c0s7 : Wed Aug 25 23:07:16 EDT 2010
      c0-0c1s3 : Wed Aug 25 23:07:16 EDT 2010
      .
      .
      .
      c0-0 : Wed Aug 25 23:07:17 EDT 2010
      ```

8. If the time zone has been changed already, complete this step. If this is the first time the time zone has been modified, perform step 7.

a. To update the L1's time zone:

```
smw:~ # xtrsh -l root -m ^c[0-9]+-[0-9]+$ -s 'atftp -g -r localtime \
-l $(readlink /etc/localtime) router && cp /etc/localtime /var/tftp'
```

b. To update the L0's time zone:

```
smw:~ # xtrsh -l root -m s -s 'atftp -g -r localtime -l $(readlink /etc/localtime) router'
```

9. Bounce the system.

```
crayadm@smw:~> xtbounce s0
```

Procedure 57. Changing the time zone on the boot root and shared root

Perform the following steps to change the time zone. You must be logged on as `root`. In this example, the time zone is changed from `"America/Chicago"` to `"Europe/London"`.

1. Confirm the time zone setting on the SMW.

```
smw:~ # cd /etc/sysconfig
smw:~ # grep TIMEZONE clock
TIMEZONE="Europe/London"
DEFAULT_TIMEZONE="Europe/London"
```

2. Log on to the boot node.

```
smw:~ # ssh root@boot
boot:~ #
```

3. Verify that the `zone.tab` file in the `/user/share/zoneinfo` directory contains the time zone you want to set.

```
boot:~ # cd /usr/share/zoneinfo
boot:~ # grep Europe/London zone.tab
GB +512830-0001845 Europe/London Great Britain
```

4. Create the time conversion information files.

```
boot:~ # date
Fri Mar 10 05:19:38 CST 2007
boot:~ # /usr/sbin/zic -l Europe/London
boot:~ # date
Fri Mar 10 11:21:31 GMT 2007
```

5. Modify the `clock` file in the `/etc/sysconfig` directory to set the `DEFAULT_TIMEZONE` and the `TIMEZONE` variables to the new time zone.

```
boot:~ # cd /etc/sysconfig
boot:~ # grep TIMEZONE clock
TIMEZONE="America/Chicago"
DEFAULT_TIMEZONE="America/Chicago"
boot:~ # vi clock
make changes
boot:~ # grep TIMEZONE clock
TIMEZONE="Europe/London"
DEFAULT_TIMEZONE="Europe/London"
```

6. Switch to the default view by using `xtopview`.

   Note: If the SDB node has not been started, you must include the `-x /etc/opt/cray/sdb/node_classes` option when you invoke the `xtopview` command.

   ```
   boot:~ # xtopview
   default/:/ #
   ```

7. Verify that the `zone.tab` file in the `/user/share/zoneinfo` directory contains the time zone you want to set.

   ```
   default/:/ # cd /usr/share/zoneinfo
   default/:/ # grep Europe/London zone.tab
   GB +512830-0001845 Europe/London Great Britain
   ```

8. Create the time conversion information files.

   ```
   default/:/ # date
   Fri Mar 10 05:22:38 CST 2007
   default/:/ # /usr/sbin/zic -l Europe/London
   default/:/ # date
   Fri Mar 10 11:24:31 GMT 2007
   ```

9. Modify the `clock` file in the `/etc/sysconfig` directory to set the `DEFAULT_TIMEZONE` and the `TIMEZONE` variables to the new time zone.

   ```
   default/:/ # cd /etc/sysconfig
   default/:/ # grep TIMEZONE clock
   TIMEZONE="America/Chicago"
   DEFAULT_TIMEZONE="America/Chicago"
   default/:/ # vi clock
   make changes
   default/:/ # grep TIMEZONE clock
   TIMEZONE="Europe/London"
   DEFAULT_TIMEZONE="Europe/London"
   ```

10. Exit `xtopview`.

    ```
    default/:/ # exit
    boot:~ #
    ```

Procedure 58. Changing the time zone for compute nodes

1. Confirm the time zone setting on the SMW.

   ```
   smw:~ # cd /etc/sysconfig
   smw:~ # grep TIMEZONE clock
   TIMEZONE="America/Chicago"
   DEFAULT_TIMEZONE="America/Chicago"
   ```

2. Copy the new `/etc/localtime` file from the SMW to the bootimage template directory.

   ```
   smw:~# cp -p /etc/localtime
   /opt/xt-images/templates/default/etc/localtime
   ```

3. Update the boot image to include these changes; follow the steps in Procedure 8 on page 64.

   The time zone is not changed until you boot the compute nodes with the updated boot image.

Upgrading the SDB Schema [D]

Procedure 59. Upgrading the database utilities with an update package

Follow this procedure to update the SDB database schema with a CLE update package.

After running the `CLEinstall` program and before booting and testing the upgraded system, perform these steps. Your Cray system should be shut down.

1. As `crayadm` on the SMW, invoke the `xtbootsys` command to boot the boot and SDB nodes.

   ```
   crayadm@smw:~> xtbootsys -a auto.bootnode+sdb
   ```

   Or

   Include the `--partition p`*N* (where *N* is the partition number) to boot a partition.

   ```
   crayadm@smw:~> xtbootsys --partition pN -a auto.bootnode+sdb
   ```

   You are prompted for the root password.

   ```
   Enter your mainframe's root password (or just hit return)
   ```

2. From the boot node, `ssh` to the SDB.

   ```
   crayadm@smw:~> ssh root@boot
   boot:~ # ssh root@sdb
   ```

3. Stop the SDB.

   ```
   sdb:~ # /etc/init.d/sdb stop
   ```

4. Start the MySQL server.

   ```
   sdb:~ # /etc/init.d/mysql start
   ```

5. Run the upgrade script. When prompted, enter the MySQL root password.

   ```
   sdb:~ # /software/mysql/shell_mysql_upgrade.sh
   ```

6. Stop the MySQL server.

   ```
   sdb:~ # /etc/init.d/mysql stop
   Shutting down MySQL..                      done
   ```

7. Start the SDB.

   ```
   sdb:~ # /etc/init.d/sdb start
   starting sdb
   XT release: using release 4.0.n
   Starting MySQL.                                           done
   waiting for mysql to accept connections
   Initializing SDB Tables
   Initializing processor table
   Connected
   Initializing attributes tables
   Connected
   Initializing segment tables
   Connected
   Initializing service_processor table
   Connected
   Initializing Lustre recovery table
   Initializing Lustre failover table
   Initializing accounting tables
   sdb:~ #
   ```

8. Use your site-specific procedures to shut down the boot and SDB nodes. For example, to shut down using an automation file:

   ```
   crayadm@smw:~> xtbootsys -s last -a auto.xtshutdown
   ```

   For more information about using automation files, see the `xtbootsys`(8) man page.

   Although not the preferred method, alternatively, you can execute these commands as `root`.

   Shut down the SDB and boot nodes:

   ```
   sdb:~ # shutdown -h now; exit
   ```

   After waiting until the SDB node has finished its shutdown, shut down the boot node:

   ```
   boot:~ # shutdown -h now; exit
   ```

Complete the remaining procedures to install your update package and boot the system.

Configuring Primary and Extended File Partitions [E]

Use the `fdisk` command to configure three types of file partitions: primary, extended, and logical. The partition table, which stores the size and location of partitions for each device, is limited to four primary partitions. When more partitions are required, you must create an extended partition. This form of primary partition can contain multiple logical partitions.

There are six parameters for `fdisk` that you often use:

`p`	Print (view) the partition table
`n`	Create a new partition
`d`	Delete an existing partition
`t`	Change the partition type
`q`	Quit without saving changes
`w`	Write the new partition table and exit

E.1 Creating a Primary Partition

This example uses the `fdisk` command to set up a device, `/dev/sdc`, that is partitioned into two primary partitions, `/dev/sdc1` and `/dev/sdc2`. Use this procedure to set up the primary partitions for the devices that are described in Table 6.

Example 12. Configuring a primary partition with the `fdisk` command

As `root`, use the `fdisk` command:

```
smw:~# fdisk /dev/sdc
```

An informational message is displayed, then the `fdisk` command prompt is displayed. Type **`p`** to print the current hard drive geometry and configuration information (if any).

```
Note: sector size is 4096 (not 512)

The number of cylinders for this disk is set to 5000.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sdc: 41.9 GB, 41943040000 bytes
64 heads, 32 sectors/track, 5000 cylinders
Units = cylinders of 2048 * 4096 = 8388608 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1               1        3577    29302656   83  Linux
/dev/sdc2            3578        4770     9773056   83  Linux
```

Assuming the device is not configured, type **`n`** to create a new partition.

```
Command (m for help): n
```

You are prompted to specify whether it is a primary (p) or extended (e) partition.

```
Command action
   e   extended
   p   primary partition
```

Type **`p`** to create a primary partition. You are prompted to specify the partition number. Type the partition number as defined in Table 6.

```
p
Partition number (1-4): 1
```

You are prompted to specify the first cylinder in this partition. Press `Enter` to accept the default.

```
First cylinder (1-14593, default 1): (Press Enter)
```

You are prompted to specify either the last cylinder or the size of the partition in gigabytes. Type the size as defined in Table 6.

```
Last cylinder or +size or +sizeM or +sizeK
  (1-14593, default 14593): +30G
```

Repeat this process for the next partition in this device:

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (3578-14593, default 3578): <CR>
Using default value 3578
Last cylinder or +size or +sizeM or +sizeK
   (3578-14593, default 14593): +10G
```

Use the **p** command to verify the partitioning.

```
Command (m for help): p

Note: sector size is 4096 (not 512)

Disk /dev/sdc: 41.9 GB, 41943040000 bytes
64 heads, 32 sectors/track, 5000 cylinders
Units = cylinders of 2048 * 4096 = 8388608 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1               1        3577    29302656   83  Linux
/dev/sdc2            3578        4770     9773056   83  Linux

Command (m for help):
```

When you are satisfied with the partitioning, type **w** to write and exit.

```
Command (m for help): w
```

Use the preceding example as a guide to continue creating the primary partitions `/dev/sdd` through `/dev/sdk`, as needed, employing the values in Table 6.

E.2 Creating an Extended Partition and Logical Partitions

Primary partitions are numbered from 1 to 4. An extended partition is a primary partition that is subdivided into one or more logical partitions. Logical partition numbering starts at 5, regardless of the number of primary partitions.

This example uses the `fdisk` command to set up a device with extended and logical partitions; for example, `/dev/sdf`. Use this procedure to set up the extended and logical partitions for the devices that are described in Table 6.

Example 13. Configuring extended and logical partitions with the `fdisk` command

Use the `fdisk` command:

```
smw:~ # fdisk /dev/sde
```

An informational message is displayed, then the `fdisk` command prompt is displayed. Type **p** to print the current hard drive geometry and configuration information (if any).

```
The number of cylinders for this disk is set to 4461.
There is nothing wrong with that, but this is larger than 1024,
 and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)
Warning: invalid flag 0x0000 of partition table 4
 will be corrected by w(rite)
```

Assuming the device is not configured, type **n** to create a new partition.

```
Command (m for help): n
```

You are prompted to specify whether it is a primary (p) or extended (e) partition.

```
Command action
   e extended
   p primary partition (1-4)
```

Type **e** to create an extended partition. You are prompted to specify the partition number. Type the partition number as defined in Table 6.

```
e
Partition number (1-4): 1
```

You are prompted to specify the first cylinder in this partition. Press `Enter` to accept the default.

```
First cylinder (1-4461, default 1): (Press Enter)

Using default value 1
```

You are prompted to specify either the last cylinder in this partition or the size of the partition. Press `Enter` to accept the default.

```
Last cylinder or +size or +sizeM or +sizeK
   (1-4461, default 4461): <CR>

Using default value 4461
```

Type **p** to verify partitioning.

```
Command (m for help): p

Disk /dev/sdf: 293.6 GB, 293601280000 bytes
255 heads, 63 sectors/track, 4461 cylinder
Units = cylinders of 16065 * 4096 = 65802240 bytes

Device Boot Start End Blocks Id System
/dev/sdf1 1 4461 286663608 5 Extended
```

Type **n** to create the next new partition in this device.

```
Command (m for help): n
```

Type **`l`** to create a logical partition.

```
Command action
l logical (5 or over)
p primary partition (1-4)
l
```

You are prompted to specify the first cylinder in this partition. Press `Enter` to accept the default.

```
First cylinder (1-4461, default 1): 1
```

You are prompted to specify either the last cylinder in this partition or the size of the partition in gigabytes. Type the size as defined in Table 6.

```
Last cylinder or +size or +sizeM or +sizeK
  (5-4461, default 4461): +30G
```

Type **`p`** to verify partitioning.

```
Command (m for help): p

Disk /dev/sdf: 293.6 GB, 293601280000 bytes
255 heads, 63 sectors/track, 4461 cylinders
Units = cylinders of 16065 * 4096 = 65802240 bytes

Device Boot Start End Blocks Id System
/dev/sdf1 1 4461 286663608 5 Extended
/dev/sdf5 1 461 29623860 83 Linux
```

Repeat the process for the next partition. Type **`n`** to create the next new partition in this device.

```
Command (m for help): n
```

Specify **`l`** for logical partition to create a logical partition.

```
Command action
l logical (5 or over)
p primary partition (1-4)
l
```

You are prompted to specify the first cylinder in this partition. Press `Enter` to accept the default.

```
First cylinder (462-4461, default 462): (Press Enter)
Using default value 462
```

You are prompted to specify either the last cylinder in this partition or the size of the partition in gigabytes. Type the size as defined in Table 6.

```
Last cylinder or +size or +sizeM or +sizeK
  (462-4461, default 4461): +180G
```

Type **p** to verify the partitioning.

```
Command (m for help): p
Disk /dev/sdf: 293.6 GB, 293601280000 bytes
255 heads, 63 sectors/track, 4461 cylinders
Units = cylinders of 16065 * 4096 = 65802240 bytes

Device   Boot Start End     Blocks Id System
/dev/sdf1 1   4461 286663608  5    Extended
/dev/sdf5 1   461  29623860   83   Linux
/dev/sdf6 462 3197 175815108   83  Linux
```

Use the preceding example as a guide to continue creating the extended and logical partitions for `/dev/sdf` and `/dev/sdi`, as needed, employing the values in Table 6.

Configuring SuSEfirewall2 for a Login or Network Node [F]

Warning: The default /etc/sysconfig/SuSEfirewall2 file contains the configuration setting FW_DEV_EXT="any". When FW_DEV_EXT is set to "any", all traffic on all interfaces on the node will be filtered and the boot node will lose contact with the node over HSN. The FW_DEV_EXT parameter **must** be set to the external Ethernet interface; for example, FW_DEV_EXT="eth0".

Execute the following commands for any network or login node that will be running the SuSEfirewall filter. This example assumes the login or network node is nid 8.

Procedure 60. Configuring SuSEfirewall2 for a login or network node

1. Specialize the /etc/sysconfig/SuSEfirewall2 file for this node.

```
boot:~ # xtopview -n 8
node/8:/ # xtspec -n 8 /etc/sysconfig/SuSEfirewall2
```

2. Edit the configuration file to make the desired changes. Change the FW_DEV_EXT, FW_SERVICES_EXT_TCP, and FW_SERVICES_EXT_UDP variables so they are specific to your site.

```
node/8:/ # vi /etc/sysconfig/SuSEfirewall2
```

Change the following lines in the file.

a. Change variable FW_DEV_EXT from FW_DEV_EXT="any" to FW_DEV_EXT="eth*X*" where *X* is the Ethernet interface number; for example, eth0.

b. Change FW_SERVICES_EXT_TCP and FW_SERVICES_EXT_UDP from

```
FW_SERVICES_EXT_TCP=""
FW_SERVICES_EXT_UDP=""
```

to

```
FW_SERVICES_EXT_TCP="ssh"
FW_SERVICES_EXT_UDP="ssh"
```

FW_SERVICES_EXT_TCP="ssh" and FW_SERVICES_EXT_UDP="ssh" allow external ssh connections. If your site requires other services via the external interface, include them here. For additional information, see the /etc/sysconfig/SuSEfirewall2 file.

3. Execute the following commands to start the firewall at boot time:

```
node/8:/ # chkconfig SuSEfirewall2_init on
node/8:/ # chkconfig SuSEfirewall2_setup on
```

4. Exit the xtopview session:

```
node/8:/ # exit
```

5. Start the firewall on the node with the modified configuration (in this example, nid00008):

```
boot:~ # ssh nid00008
nid00008:~ # /etc/init.d/SuSEfirewall2_init start
nid00008:~ # /etc/init.d/SuSEfirewall2_setup start
```

Creating Partitions on a Cray XE System [G]

The xtcli part_cfg command updates partition configurations to define a *logical machine* within a Cray XE system. Partition IDs are predefined as p0 to p31. p0 (the default) is reserved as the complete system. See the xtcli_part(8) man page for more information.

> **Note:** Contact your Cray service representative before creating partitions to ensure that the members/components of each partition will be a routable configuration.

Partition requirements include:

- Each partition must contain the normal set of service nodes: boot, sdb, syslog, ufs, login, and so on. A service node is a member of exactly one partition at a time as well as being part of p0, the whole system.
- Each partition should have an individual CLEinstall.conf defining that partition's specific configuration.
- The IP addresses should be set to unique values for each partition.

By convention, s0 or p0, the entire system, uses these settings:

```
partition=s0
xthostname=mycray
node_class_login_hostname=mycray
bootimage_bootnodeip=10.131.255.254
bootnode_failover_IPaddr=10.131.255.254
persistent_var_IPaddr=10.131.255.254
sdbnode_failover_IPaddr=10.131.255.253
node_sdb_hostname=sdb
node_ufs_hostname=ufs
node_syslog_hostname=syslog
node_boot_hostname=boot
```

For partition p1, the same IP address could be used, but it is wise to set the hostnames to include p1. When logged into nodes in the p1 partition, the boot, sdb, ufs, and syslog hostname aliases will refer to boot-p1, sdb-p1, ufs-p1, and syslog-p1:

```
partition=p1
xthostname=mycray-p1
node_class_login_hostname=mycray-p1
bootimage_bootnodeip=10.131.255.254
bootnode_failover_IPaddr=10.131.255.254
persistent_var_IPaddr=10.131.255.254
sdbnode_failover_IPaddr=10.131.255.253
node_sdb_hostname=sdb-p1
node_ufs_hostname=ufs-p1
node_syslog_hostname=syslog-p1
node_boot_hostname=boot-p1
```

For partition p2, a different set of IP addresses and hostnames should be used. When logged into nodes in the p2 partition, the boot, sdb, ufs, and syslog hostname aliases will refer to boot-p2, sdb-p2, ufs-p2, and syslog-p2:

```
partition=p2
xthostname=mycray-p2
node_class_login_hostname=mycray-p2
bootimage_bootnodeip=10.131.255.252
bootnode_failover_IPaddr=10.131.255.252
persistent_var_IPaddr=10.131.255.252
sdbnode_failover_IPaddr=10.131.255.251
node_sdb_hostname=sdb-p2
node_ufs_hostname=ufs-p2
node_syslog_hostname=syslog-p2
node_boot_hostname=boot-p2
```

On a partitioned system, the System Management Workstation (SMW) has aliases for boot-p1, boot-p2, etc. in /etc/hosts. The boot hostname is an alias to boot-p1, so it is best to develop the habit of using the boot-p*N* hostname when connecting from the SMW to the boot node of partition p*N*.

Procedure 61. Connecting from the SMW to the boot node of partition

1. Modify the /etc/sysset.conf to reflect the correct hostnames for your nodes in each partition. The system set that is to be used with partition p1 must be modified to use boot-p1, sdb-p1, etc. in the host column.

2. Check the current partition information for your system.

   Note: The `bootimage` is specified as a file instead of a raw disk device to provide clarity to the examples, but you could use different raw disk devices instead of the files. The example `bootimage` locations have the format `hostname-partition-label-version` in the `/bootimagedir` directory where label is the system set label and version is the CLE version.

   ```
   smw:~ # xtcli part_cfg show
   Network topology: class 0
   === part_cfg ===
   ------------------
   [partition]: p0: enable (noflags|)
   [members]: c0-0, c1-1
   [boot]: c0-0c0s0n1:ready
   [sdb]: c0-0c0s2n1:ready
   [cpio_path]: /bootimagedir/mycray-p0-BLUE-4.0.15
   ================
   ```

   Note: It is best to change the partition configuration only when the system is not booted. The boot node and SDB node shown in the preceding listing are in the `ready` state, indicating that they are booted. If they are booted, shut them down before continuing.

3. Add partition `p1` and partition `p2` by specifying the partition components (comma separated), boot node, SDB node, and boot image. In this example, partition `p1` uses the `GREEN` system set and partition `p2` uses the `RED` system set.

```
smw:~ # xtcli part_cfg add p1 -m c0-0 -b c0-0c0s0n1 -d c0-0c0s2n1 \
-i /bootimagedir/mycray-p1-GREEN-4.0.15
Network topology: class 0
=== part_cfg ===
------------------
[partition]: p1: disabled (noflags|)
[members]: c0-0
[boot]: c0-0c0s0n1:halt
[sdb]: c0-0c0s2n1:halt
[cpio_path]: /bootimagedir/mycray-p1-GREEN-4.0.15
================

smw:~ # xtcli part_cfg add p2 -m c0-1 -b c0-1c0s0n1 -d c0-1c0s2n1 \
-i /bootimagedir/mycray-p2-RED-4.0.15
Network topology: class 0
=== part_cfg ===
------------------
[partition]: p2: disabled (noflags|)
[members]: c0-0c1
[boot]: c0-1c0s0n1:halt
[sdb]: c0-1c0s2n1:halt
[cpio_path]: /bootimagedir/mycray-p2-RED-4.0.15
================
```

4. Check the partition configuration; it should show `p0`, `p1`, and `p2`. Only partition `p0` is enabled.

   ```
   smw:~ # xtcli part_cfg show
   Network topology: class 0
   === part_cfg ===
   ------------------
   [partition]: p0: enable (noflags|)
   [members]: c0-0
   [boot]: c0-0c0s0n0:halt
   [sdb]: c0-0c0s0n3:halt
   [cpio_path]: /bootimagedir/mycray-p0-BLUE-4.0.15
   ------------------
   [partition]: p1: disabled (noflags|)
   [members]: c0-0
   [boot]: c0-0c0s0n1:halt
   [sdb]: c0-0c0s2n1:halt
   [cpio_path]: /bootimagedir/mycray-p1-GREEN-4.0.15
   ------------------
   [partition]: p2: disabled (noflags|)
   [members]: c0-1
   [boot]: c0-1c0s0n1:halt
   [sdb]: c0-1c0s2n1:halt
   [cpio_path]: /bootimagedir/mycray-p2-RED-4.0.15
   ================
   ```

5. Before a partition can be used, it must be activated. This changes the state from disabled to enabled. Deactivating the partition changes the state from enabled to disabled. `P0`, however, cannot be active when other partitions are active.

   a. Deactivate the `p0` partition:

      ```
      smw:~ # xtcli part_cfg deactivate p0
      Network topology: class 0
      === part_cfg ===
      ------------------
      [partition]: p0: disabled (noflags|)
      [members]: c0-0,c0-1
      [boot]: c0-0c0s0n1:halt
      [sdb]: c0-0c0s2n1:halt
      [cpio_path]: /bootimagedir/mycray-p0-BLUE-4.0.15
      ================
      ```

b. Activate the p1 and p2 partitions:

```
smw:~ # xtcli part_cfg activate p1
Network topology: class 0
=== part_cfg ===
------------------
[partition]: p1: enable (noflags|)
[members]: c0-0
[boot]: c0-0c0s0n1:halt
[sdb]: c0-0c0s2n1:halt
[cpio_path]: /bootimagedir/mycray-p1-GREEN-4.0.15

smw:~ # xtcli part_cfg activate p2
Network topology: class 0
=== part_cfg ===
------------------
[partition]: p2: enable (noflags|)
[members]: c0-1
[boot]: c0-1c0s0n1:halt
[sdb]: c0-1c0s2n1:halt
[cpio_path]: /bootimagedir/mycray-p2-RED-4.0.15
```

6. Check the partition configuration; it should now show p0 disabled and both p1 and p2 enabled:

```
smw:~ # xtcli part_cfg show
Network topology: class 0
=== part_cfg ===
------------------
[partition]: p0: disabled (noflags|)
[members]: c0-0
[boot]: c0-0c0s0n0:halt
[sdb]: c0-0c0s0n3:halt
[cpio_path]: /bootimagedir/mycray-p0-BLUE-4.0.15
------------------
[partition]: p1: enable (noflags|)
[members]: c0-0
[boot]: c0-0c0s0n1:halt
[sdb]: c0-0c0s2n1:halt
[cpio_path]: /bootimagedir/mycray-p1-GREEN-4.0.15
------------------
[partition]: p2: enable (noflags|)
[members]: c0-1
[boot]: c0-1c0s0n1:halt
[sdb]: c0-1c0s2n1:halt
[cpio_path]: /bootimagedir/mycray-p2-RED-4.0.15
================
```

7. After the partitions have been configured, run CLEinstall to install CLE 4.0 on the system sets chosen for these partitions.