CLE Installation and Configuration Guide

S-2444-5204

Contents

About CLE Initial Installations, Updates, and Upgrades

This guide contains procedures for installation and configuration of the Cray Linux Environment (CLE) 5.2.UP04 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 5.2.UP04 release includes Cray's customized version of the SUSE Linux Enterprise Server (SLES™) 11 SP3 operating system.

Throughout this document, any reference to *Cray systems* refers to Cray XC30 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 5.2.UP04 on a system that is already running an earlier version of CLE 5.2 is considered an update installation.

Upgrade

An upgrade installation involves moving to the next release. For example, installing CLE 5.2.UP04 on a system that is running CLE 5.1 is considered an upgrade.

This guide describes procedures for the following types of installations.

- Initial or new software installations. Follow *Prepare to Install a New System* on page 11.
- Update and Upgrade installations. Follow *Prepare to Update or Upgrade CLE Software* on page 106 to update or upgrade an existing system.

The procedures in this document require that you have already installed the appropriate System Management Workstation (SMW) software release on the SMW. See *Before the CLE Software Installation* on page 11.

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

Related Publications

The following documents contain additional information that can 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 the Cray Linux Environment* (S-2393)
- *Managing Lustre for the Cray Linux Environment (CLE)* (S-0010)
- *Introduction to Cray Data Virtualization Service* (S-0005)
- *Using and Configuring System Environment Data Collections (SEDC)* (S-2491)
- *Using Compute Unit Affinity on Cray Systems* (S-0030)
- *Using Balanced Injection in Cray Systems* (S-0040)
- *Cray Programming Environments Installation Guide* (S-2372)
- *Installing, Configuring, and Managing SMW Failover on the Cray XC System* (S-0044)

Distribution Media

The CLE 5.2 release distribution media includes three DVDs required to install the CLE 5.2 release on a Cray XC30 system. The first is labeled Cray CLE 5.2.UP*nn* Software and contains software specific to Cray systems. The second is labeled Cray-CLEbase11-*yyyymmdd* and contains the CLE 5.2 base operating system which is based on SLES 11 SP3. The third is labeled CentOS-6.5-x86_64-bin-DVD1.iso and contains the CentOS™ 6.5 base operating system for CLE direct-attached Lustre® (DAL) nodes.

Prepare to Install a New System

Follow these procedures to perform an initial software installation of the Cray Linux Environment (CLE) 5.2.UP04 software release for a new Cray XC30 system.

Before the CLE Software Installation

Perform the following tasks before you install the CLE 5.2.UP04 software release.

- Review release package documentation. Read the *CLE 5.2.UP04 Release Errata*, *Limitations for CLE 5.2.UP04* 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* (S-2425) and *Cray Linux Environment (CLE) Software Release Overview Supplement* (S-2497).
- Confirm the SMW software release level. You must install the SMW 7.2.UP04 release or later on your SMW before installing the CLE 5.2.UP04 release. If a specific SMW update package is required for your installation, that information is documented in the *README* file provided with the CLE 5.2.UP04 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
 7.2.UP04
```

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 *Change the Default System Passwords*.

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* (S-2480).

Access to MySQL™ databases requires a user name and password. The MySQL accounts and privileges are shown in *MySQL Database Accounts and Privileges* on page 12.

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.

For steps to change MySQL account passwords, see *Change Default MySQL Passwords on the SDB* on page 66.

Configure the Boot RAID

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

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, DataDirect™ Networks (DDN™) and NetApp Corporation. You may also have a QLogic™ SANbox™ Fibre Channel switch from QLogic Corporation or a serial-attached SCSI (SAS) switch from NetApp.

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

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* (S-2480).

After *Configure the Boot RAID LUNs or Volume Groups* on page 14, follow the procedures for *Boot LUN Partitions* on page 16.

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 SP3, and Cray CLE software packages onto the boot RAID disks.

> **NOTE:** For a Cray XC30 system configured for SMW high availability (HA) with the SMW failover feature, the boot RAID is also used for the shared log, MySQL database, and `/home` file system.

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.

Configure the Boot RAID LUNs or Volume Groups

Follow the procedures in *Installing Cray System Management Workstation (SMW) Software* (S-2480) to configure your boot RAID. You must configure the boot RAID with at least six LUNs; this number ensures that you have enough space to support the various system management file systems and a backup. The recommended configuration listed in *Recommended Boot RAID LUN Values* on page 14 describes nine LUNs, spanning three system sets: a backup set for the previous release labeled `BLUE`, a production set for the current release labeled `GREEN`, and a set for testing the deployment of a pending upgrade labeled `RED`. 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* (S-2480) to `telnet` to the RAID controller and use the `lun add` and `lun delete` commands to configure LUNs following the recommendations in *Recommended Boot RAID LUN Values* on page 14.

If you have NetApp, Inc. Engenio devices, follow the boot RAID configuration procedures in *Installing Cray System Management Workstation (SMW) Software* (S-2480) to use the SANtricity Storage Manager software to create the boot RAID volume group and configure the volumes following the recommendations in *Recommended Boot RAID LUN Values*.

The example below uses a combined SDB, UFS and `syslog` node; if you are using a dedicated node for either of these functions, they should each have their storage on a separate LUN. A single LUN should not serve multiple physical nodes.

WARNING: Some third-party batch systems require additional space (possibly upwards of 50GB) in the `PERSISTENT_VAR` partition. You should review the requirements of the batch system you intend to deploy in order to determine the appropriate size for this partition and the `shroot*` LUN(s). The size given for the `UFS` partition, and therefore the `sdb*` LUN(s), is based on the assumption that it will be used for the `crayadm` home directory and `/ufs/alps_shared`, and that general users home directories will be on another file system. This is the recommended configuration for best performance. However, if the `UFS` partition will be used by users it should be 40 GB at an absolute minimum and should likely be put on its own separate LUN for better performance.

Table 3. Recommended Boot RAID LUN Values

LUN	Label	Size (1-50 Cabinets)	Size (50+ Cabinets)	Segment Size
0	bootroot0	40GB	70GB	256KB
1	shroot0	280GB	370GB	256KB
2	sdb0	60GB	80GB	256KB
3	bootroot1	40GB	70GB	256KB
4	shroot1	280GB	370GB	256KB
5	sdb1	60GB	80GB	256KB
6	bootroot2	40GB	70GB	256KB

LUN	Label	Size (1-50 Cabinets)	Size (50+ Cabinets)	Segment Size
7	shroot2	280GB	370GB	256KB
8	sdb2	60GB	80GB	256KB

NOTE: For a Cray XC30 system configured for SMW HA with the SMW failover feature, the boot RAID must also include sufficient space for the shared log, MySQL database, and `/home` file system.

Zone 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.

Zone 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* (S-2480). Use the `zoning` command to edit each port number and map the LUNs; follow the recommendations in *Recommended DDN Zoning*.

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. *Partition the LUNs* on page 17 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.

Zone the QLogic FC Switch

If you have a QLogic Fibre Channel Switch, follow the procedures described in *Installing Cray System Management Workstation (SMW) Software* (S-2480) 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 *Recommended QLogic Zoning* on page 16. These recommendations presuppose that the disk device has four host ports connected to ports 0-3 for the QLogic SANbox switch. 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. *Partition the LUNs* on page 17 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.

Boot LUN Partitions

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

Example of Boot LUN Partitions contains an example of a partition layout using the three system sets mentioned above. The SMW Device names in column 5 are consistent with rack-mount SMW hardware with four internal disk drives.

WARNING: Please note that `/dev/sd`*x* names are not persistent; the names in the example may be different from your system and may change between reboots, so while it is acceptable to use them and the output of `lsscsi` while partitioning, ensure that you are targeting the appropriate device. After initial partitioning, you should always address the storage via its persistent `/dev/disk/by-id/` name. For more information, see *About Persistent Boot RAID Device Names* on page 38.

Table 6. Example of Boot LUN Partitions

LUN	System Set	Part Num	Part Type	SMW Device	Size (1-50 Cabinets)	Size (50+ Cabinets)	Type	Description
0	BLUE	1	Primary	sde1	30GB	60GB	Linux	Boot node root file system

LUN	System Set	Part Num	Part Type	SMW Device	Size (1-50 Cabinets)	Size (50+ Cabinets)	Type	Description
0	BLUE	2	Primary	sde2	10GB	10GB	Swap	Boot node swap
1	BLUE	1	Primary	sdf1	210GB	250GB	Linux	Shared root
1	BLUE	2	Primary	sdf2	10GB	10GB	Linux	Boot image 1
1	BLUE	3	Primary	sdf3	10GB	10GB	Linux	Boot image 2
1	BLUE	4	Primary	sdf4	50GB	100GB	Linux	Persistent `/var`
2	BLUE	1	Primary	sdg1	20GB	20GB	Linux	Service database (sdb)
2	BLUE	2	Primary	sdg2	20GB	20GB	Linux	UFS
2	BLUE	3	Primary	sdg3	20GB	40GB	Linux	syslog
3	GREEN	1	Primary	sdh1	30GB	60GB	Linux	Boot node root file system
3	GREEN	2	Primary	sdh2	10GB	10GB	Swap	Boot node swap
4	GREEN	1	Primary	sdi1	210GB	250GB	Linux	Shared root
4	GREEN	2	Primary	sdi2	10GB	10GB	Linux	Boot image 1
4	GREEN	3	Primary	sdi3	10GB	10GB	Linux	Boot image 2
4	GREEN	4	Primary	sdi4	50GB	100GB	Linux	Persistent `/var`
5	GREEN	1	Primary	sdj1	20GB	20GB	Linux	Service database (sdb)
5	GREEN	2	Primary	sdj2	20GB	20GB	Linux	UFS
5	GREEN	3	Primary	sdj3	20GB	40GB	Linux	syslog
6	RED	1	Primary	sdk1	30GB	60GB	Linux	Boot node root file system
6	RED	2	Primary	sdk2	10GB	10GB	Swap	Boot node swap
7	RED	1	Primary	sdl1	210GB	250GB	Linux	Shared root
7	RED	2	Primary	sdl2	10GB	10GB	Swap	Boot image 1
7	RED	3	Primary	sdl3	10GB	10GB	Linux	Boot image 2
7	RED	4	Primary	sdl4	50GB	100GB	Linux	Persistent `/var`
8	RED	1	Primary	sdm1	20GB	20GB	Linux	Service database (sdb)
8	RED	2	Primary	sdm2	20GB	20GB	Linux	UFS
8	RED	3	Primary	sdm3	20GB	40GB	Linux	syslog

Partition 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. The first SMW device is the first non-ATA device listed. On an SMW with four internal SATA drives, the output should resemble the following example. Note that four of the disks are ATA, not DDN or NetApp, Inc. Engenio disks. Depending on when the disks were obtained, they might be `LSI`, `Engenio`, or `NetApp`. The `lsscsi` output may be different on your system.

```
smw:~ # lsscsi
[0:0:0:0]    disk    ATA      FUJITSU MHZ2160B             8A22    /dev/
sda
[0:0:1:0]    disk    ATA      ST91000640NS                 AA02    /dev/sdb
[0:0:2:0]    disk    ATA      FUJITSU MHZ2160B             8A22    /dev/sdc
[0:0:3:0]    disk    ATA      FUJITSU MHZ2160B             8A22    /dev/sdd
[1:0:0:0]    disk    LSI      INF-01-00                    0777    /dev/sde
[1:0:0:1]    disk    LSI      INF-01-00                    0777    /dev/sdf
[1:0:0:2]    disk    LSI      INF-01-00                    0777    /dev/sdg
[1:0:0:3]    disk    LSI      INF-01-00                    0777    /dev/sdh
[1:0:0:4]    disk    LSI      INF-01-00                    0777    /dev/sdi
[1:0:0:5]    disk    LSI      INF-01-00                    0777    /dev/sdj
[1:0:0:6]    disk    LSI      INF-01-00                    0777    /dev/sdk
[1:0:0:7]    disk    LSI      INF-01-00                    0777    /dev/sdl
[1:0:0:8]    disk    LSI      INF-01-00                    0777    /dev/sdm
```

3. Create the partitions shown in *Example of Boot LUN Partitions* on page 16 by using the `fdisk` command. If you are not familiar with `fdisk`, see *Configure Primary and Extended File Partitions* on page 174 and the `fdisk(8)` man page.

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

   In this example, repeat the previous command for `/dev/sdf` through `/dev/sdm`; use the values in *Example of Boot LUN Partitions* on page 16 for each `fdisk` session. Changes to the partition table are not effective until entering **`w`** to write and exit.

4. Invoke the `fdisk` command with the `-l` option to verify that the LUNs (volumes) are configured according to *Example of Boot LUN Partitions* on page 16. LUN sizes may be slightly different; for example, 43G instead of 40G, as listed in the table. The following output represents the example; output is specific to the actual LUN configuration.

```
smw:~ # fdisk -l
Disk /dev/sda: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders, total 312581808 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000081

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1              63    67103504    33551721   82  Linux swap / Solaris
/dev/sda2   *    67103505   312576704   122736600   83  Linux

Disk /dev/sdb: 1000.2 GB, 1000204886016 bytes
36 heads, 63 sectors/track, 861342 cylinders, total 1953525168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000083

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048  1953525167   976761560   83  Linux

Disk /dev/sdc: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders, total 312581808 sectors
Units = sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000080

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1              63    67103504    33551721   82  Linux swap / Solaris
/dev/sdc2   *    67103505   312576704   122736600   83  Linux


Disk /dev/sdd: 160.0 GB, 160041885696 bytes
6 heads, 63 sectors/track, 826936 cylinders, total 312581808 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000082

   Device Boot      Start         End      Blocks   Id  System
/dev/sdd1            2048   312581807   156289880   83  Linux


Disk /dev/sde: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders, total 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

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

Disk /dev/sdf: 300.6 GB, 300647710720 bytes
255 heads, 63 sectors/track, 36551 cylinders, total 587202560 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

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

Disk /dev/sdg: 64.4 GB, 64424509440 bytes
255 heads, 63 sectors/track, 7832 cylinders, total 125829120 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

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

Disk /dev/sdh: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders, total 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

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

Disk /dev/sdi: 300.6 GB, 300647710720 bytes
255 heads, 63 sectors/track, 36551 cylinders, total 587202560 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk identifier: 0x00000000

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

Disk /dev/sdj: 64.4 GB, 64424509440 bytes
255 heads, 63 sectors/track, 7832 cylinders, total 125829120 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

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

Disk /dev/sdk: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders, total 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

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

Disk /dev/sdl: 300.6 GB, 300647710720 bytes
255 heads, 63 sectors/track, 36551 cylinders, total 587202560 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

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

Disk /dev/sdm: 64.4 GB, 64424509440 bytes
255 heads, 63 sectors/track, 7832 cylinders, total 125829120 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

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

About Installation Configuration Files

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.

About `CLEinstall.conf` Parameters That Must Be Defined

A template `CLEinstall.conf` is delivered on the Cray CLE 5.2.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 *Create the Installation Configuration Files* on page 42, 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.

User home directories

Set `home_directory_ufs=no` and enter values for `home_directory_server_hostname`, `home_directory_server_IPaddr`, and `home_directory_server_path` that point to your site-specific user file system NFS™ server; Cray does not recommend using the boot RAID (`/ufs`) for user files on production systems.

> **IMPORTANT:** This section is referred to as "UFS (home directory) for login nodes" in older versions of `CLEinstall.conf`.

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 *Maintain Node Class Settings and Hostname Aliases* on page 22 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 GPUs installed. Setting this parameter to `yes` will install the required RPMs and code for GPU systems. If your machine has GPU blades without GPUs, or it does not have GPU blades installed, set this parameter to `no`.

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

Maintain 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`.

`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`.

Set 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.

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
```

About CLEinstall.conf Parameters with Standard Settings

The standard or default values for settings in the following categories are appropriate in many 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
- SDB node network settings
- Persistent var settings
- syslog settings
- Partition setting
- SDB database settings
- Writeable /tmp for CNL setting
- Writeable /var/tmp for CNL setting
- Node Health Check (NHC) on boot
- NHC communication over Secure Sockets Layer (SSL)

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

Change the Default High-speed Network (HSN) Settings

By default, the HSN IP address is 10.128.0.0. You can modify these parameters to configure another valid address; for example 10.33.0.0. In most cases, the default value is acceptable. Modify the followoing HSN settings as needed:

```
HSN_byte1=
HSN_byte2=
```

The HSN_byte1= and HSN_byte2= parameters specify the HSN IP address. The default values are HSN_byte1=10, HSN_byte2=128. Cray recommends that the values for HSN_byte1 and HSN_byte2 do not overlap subnets listed as default IP addresses in *Installing Cray System Management Workstation (SMW) Software* (S-2480).

When HSN_byte1 and HSN_byte2 are changed 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: Because of site-specific local modifications, additional files might require updating when the HSN IP address changes. For example, such files as `/etc/hosts.allow`, `/etc/hosts.deny`, `/etc/exports`, and `/etc/security/access.conf` might require updating.

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.

Change 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.

nfsserver

Specifies whether or not the `nfsserver` service should be enabled on all service nodes. If this parameter is set to `yes`, all service nodes will run the `nfsserver` service. If this parameter is set to `no`, only the boot, sdb, and ufs nodes will run the `nfsserver` service. The default value is `no`.

Change 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`. 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`. 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)` man page. The default value is `no`.

Change Turbo Boost Limit

Because processors have a high degree of variability in the amount of turbo boost each processor can supply, limiting the amount of turbo boost can reduce performance variability and reduce power consumption. The limit applies only when a high number of cores are active. On an N-core processor, the limit is in effect when the active core count is N, N-1, N-2, or N-3. On a 12-core processor, the limit is in effect when 12, 11, 10, or 9 cores are active.

NOTE: Turbo boost is not supported on SandyBridge processors.

Set the following CLEinstall.conf parameter to adjust the turbo boost limit.

```
turbo_boost_limit=999
```

The valid values are `100`, `200` and `999`. The default setting is `999`. When `turbo_boost_limit=100`, 100 MHz is the limit. A value of `200` limits turbo boost to 200 MHz. A value of `999` implies no turbo boost limit is applied.

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.

Enable NHC Communication Over Secure Sockets Layer (SSL)

The `NHC_SSL` parameter is set to `yes` by default in `CLEinstall.conf`. This parameter generates the appropriate keys for Node Health Checker (NHC) to use SSL for communication with the compute nodes. Cray recommends sites configure NHC to use SSL.

The key (`rsa_key`), certificate (`rsa_cert`), and the certificate signing request (`servercsr`) are created in the `/root/.nodehealth` directory on the shared root and copied to the `/opt/xt-images/templates/default-`*`pX`*`/root/.nodehealth` directory for the compute node images. If any one of these three required files are missing from the shared root, they are all generated again and copied back out. If the `NHC_SSL` setting in `CLEinstall.conf` is set to `no`, the files are removed from both locations.

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

The `CLEinstall.conf` file contains settings for optional functionality and subsystems. To configure and enable a particular functionality, its settings require modifications. Settings for unused features and subsystems can be ignored.

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

Lustre File System Support and Tuning

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 *Install and Configure Direct-Attached Lustre* on page 93, as you complete the installation or upgrade process.

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 setting this parameter to the total number of compute nodes and login nodes configured on the 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.

direct_attached_lustre=no

When this setting is `yes`, then direct-attached Lustre (DAL) is enabled. `CLEinstall` uses IMPS to prepare the image to be added to the unified bootimage so that nodes chosen to be internal Lustre service nodes can be booted with the CentOS operating system with Cray additions for Lustre servers. When this setting is `yes`, the `--Centosmedia` option is required when running `CLEinstall`. `CLEinstall` provides command hints for the rest of the DAL configuration steps.

Configure Boot Node Failover

Boot-node Failover is an optional CLE feature that sets up a backup boot node to automatically take over when the primary boot node fails.

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

In addition, 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. These tasks are done after creating boot images later in the new install, update, or upgrade processes.

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`. In most cases, the default value is acceptable. You must modify the 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`. 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`. In most cases, the default value is acceptable.

For additional information, including manual boot node failover configuration steps, see *Managing System Software for the Cray Linux Environment* (S-2393).

Configure SDB Node Failover

Service database (SDB) Node failover is an optional CLE feature that enables automatic failover to a backup 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.

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`.

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 the Cray Linux Environment* (S-2393).

In addition, 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. These tasks are done after creating boot images later in the new install, update, or upgrade processes.

After booting and testing your system, follow *Configure Boot Automation for SDB Node Failover* on page 90 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`. 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`. 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`. In most cases, the default value is acceptable.

Include DVS in the Compute Node Boot Image

The following `CLEinstall.conf` parameter configures `CLEinstall` to include the Data Virtualization Service (DVS) RPM in the compute node boot image. Cray DVS is an optional CLE feature. In addition to setting this parameter, refer to *Configure Cray DVS* on page 78, 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, edit the `/var/opt/cray/install/shell_bootimage_LABEL.sh` script and specify `CNL_DVS=y` before updating the CNL boot image.

For additional information about DVS, see *Introduction to Cray Data Virtualization Service* (S-0005).

Configure DSL and CNRTE

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, configure the shared root as a DVS-projected file system. Dynamic shared objects and libraries (DSL) and the compute node root runtime environment (CNRTE) are optional.

To configure DSL and the compute node root runtime environment for your Cray system, follow these steps.

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. Modify DSL-specific parameters according to the system configuration by editing the `CLEinstall.conf` file (*Create the Installation Configuration Files* on page 42).
3. Configuring compute nodes as compute node root servers requires additional configuration. Cray recommends configuring the nodes as repurposed compute nodes. Complete *Repurpose Compute Nodes as Service Nodes* on page 43 before running `CLEinstall`.

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 the Cray Linux Environment* (S-2393). For additional information about DVS, see *Introduction to Cray Data Virtualization Service* (S-0005).

Set the following parameters in the `CLEinstall.conf` file to cause the `CLEinstall` program to automatically configure the system for the compute node root runtime environment.

```
DSL=yes
```

Set this parameter to `yes` to enable dynamic shared objects and libraries and the CNRTE. The default is `no`. Setting this option to `yes` enables 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 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.

Configure Realm-Specific IP Addressing (RSIP)

Realm-Specific IP Addressing allows CLE compute and service nodes to share IP addresses configured on the external Gigabit and 10 Gigabit Ethernet interfaces of network nodes. RSIP is an optional CLE feature. 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.

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 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 the Cray Linux Environment* (S-2393). 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.

Configure Service Node MAMU

Service Node Multiple Application Multiple User (MAMU) support provides the ability to set aside a small number of re-purposed compute nodes for serial workload. Serial workload nodes can be configured in advance, while the node is booted as a service node. See *Managing System Software for the Cray Linux Environment* for more details on this feature.

CAUTION: If you use the CLE installer, the nodes you specify as service MAMU nodes must be compute nodes when you start their configuration and installation.

The following `CLEinstall.conf` parameters configure service node MAMU.

```
SERVICE_MAMU=no
```

Set this parameter to `yes` to use a set of nodes to serve as a separate workload management pool of execution (MOM) nodes. These execution nodes do not run ALPS or manage Cray Workload, but are available for core level-scheduling directly through the workload manager.

```
SERVICE_MAMU_classes=postproc
```

Uncomment this line to set up a default class, `postproc`, for serial workload nodes. To specify more than one class, use a space to separate the class names. For example, `SERVICE_MAMU_classes=postproc serial`.

```
node_class[6]=postproc 15 16 17 18
```

Uncomment this line, which appears in the node class settings section of CLEinstall.conf.

Define service node MAMU NIDs

Use this class entry to set up nodes with NIDs `15`, `20`, `33`, and `37`.

```
node_class[n]=postproc 15 20 33 37
```

The value of *n* is the next unused node class defined in `CLEinstall.conf`. Configure a node class for each class listed in the `SERVICE_MAMU_classes` setting.

Configure DataWarp

Cray DataWarp provides an intermediate layer of high bandwidth, file-based storage to applications running on compute nodes. It is comprised of commercial SSD hardware and software, Linux community software, and Cray system hardware and software. DataWarp storage is located on server nodes connected to the Cray system's high speed network (HSN). I/O operations to this storage completes faster than I/O to the attached parallel file

system (PFS), allowing the application to resume computation more quickly and resulting in improved application performance. DataWarp storage is transparently available to applications via standard POSIX I/O operations and can be configured in multiple ways for different purposes. DataWarp capacity and bandwidth are dynamically allocated to jobs on request and can be scaled up by adding DataWarp server nodes to the system.

Use the following `CLEinstall.conf` parameters to enable and configure DataWarp.

```
datawarp=no
```

Set this parameter to `yes` to enable DataWarp. Turning on DataWarp creates a command hint displayed by the `CLEinstall` command during fresh installs to edit the boot automation file. The command hint is also available in the `/var/adm/cray/logs/CLEinstall.P#.YYYYMMDDhhmmss.$LABEL.command_hints.log` file on the SMW. The command hint is similar to this message:

```
9.5) Add the DataWarp Manager nodes with SSDs to the boot automation file. These
commands should be added after the SDB node yet before other service nodes
have started.

  lappend actions { crms_sleep 5 }
  lappend actions  [list crms_boot_loadfile SNL0 service node_list linux numa=fake=2]
```

Where *node_list* is a comma-separated list of DataWarp manager node cnames.

```
datawarp_manager_nodes=
```

A comma-separated list of nodes (cnames) on which the DataWarp manager will be run. DataWarp managers manage the storage capacity of the SSDs, and the specified nodes must have SSDs on them.

```
datawarp_api_gateways=
```

The list of DataWarp API gateways nodes—required to be either internal login or MOM node(s)—must contain one or more comma-separated triplets, in this format:

```
<hostname>:<node>:<port>
```

DataWarp API Gateway nodes act as an interface from users to the rest of the DataWarp service. *hostname* can be cname, nid*#####*, or a *hostname* that other nodes use to reach this node. *node* is the node number, an integer. *port* is the port that the DataWarp daemon listens on. Port 81 is standard.

```
datawarp_admin_uid_list=
```

A comma-separated list of user IDs that are administrators for DataWarp. Both `root` and `crayadm` are already on this list.

Configure Compute Node Swap

```
CNL_swap=no
```

Set `CNL_swap=yes` to enable the configuration of compute nodes as service nodes. Enabling the swap parameter causes `CLEinstall` to install two RPMs, one on the compute node image and the other on the shared root for the service nodes.

Configure Graphics Processing Units

The following `CLEinstall.conf` parameters configure graphics processing units (GPUs).

```
GPU=no
```

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

For systems with GPUs, you must configure and enable the alternative compute node root run time environment for dynamic shared objects and libraries (DSL). GPU blades require access to shared libraries to support GPUs. The parameters are `DSL=yes`, `DSL_nodes`, `DSL_mountpoint`, and `DSL_attrcache_timeout`. For more information, see *Configure DSL and CNRTE* on page 31.

Configure Intel Xeon Phi Coprocessors

The following `CLEinstall.conf` parameters configure Intel Xeon Phi™ coprocessors.

```
KNC=no
```

Specifies whether Xeon Phi coprocessors are present in the system. Setting `KNC=yes` enables support for Xeon Phi compute blades. The default value is `no`.

```
KNC_BASE=50000
```

Defines an offset that is used to prepare hostnames and IP addresses for the Xeon Phi nodes. If the CNL node hosting a Xeon Phi coprocessor is `nid00032`, then the coprocessor has a hostname of `nid50032` and a hostname alias of `acc50032`.

Also notice that the cname for the Intel MIC has `a0` added to the `c0-0c0s8n0` hostname. Below are the two entries in `/etc/hosts` for a compute node and the Xeon Phi coprocessor node hosted by that compute node.

```
10.128.0.33      nid00032       c0-0c0s8n0
10.128.196.249   nid50032       c0-0c0s8n0a0    acc00032
```

Configure `ntpclient` for Clock Synchronization

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. The `ntpclient` is an optional CLE feature.

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.

Configure the Parallel Command (pcmd) Tool for Unprivileged Users

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

```
NHC_pcmd_suid=no
```

The `pcmd` is installed as a root-only tool by default. To allow non-root users to run the tool, `pcmd` can be installed as a `setuid root` program. Do this at installation time by specifying `NHC_pcmd_suid=yes` in the `CLEinstall.conf` file.

Configure High Speed Network Metrics

NOTE: Configuring network metrics monitoring is an optional feature for Cray XC30, Cray XE, and Cray XK systems. Also, using the Lightweight Distributed Metric Service (LDMS) within OVIS to collect network metrics is optional. Administrators can configure network metric collection without using OVIS as a client application. However, this feature is provided with LDMS metric aggregation in mind and in that use case LDMS is responsible for aggregating and collecting compute node metrics to the SMW. For more information on installing and configuring LDMS and OVIS on Cray systems, see *https://ovis.ca.sandia.gov/mediawiki/index.php/CRAY-LDMS*.

High speed network (HSN) metrics monitoring is a CLE feature that provides on-node metric collection and aggregation for system nodes. Cray provides kernel modules and utilities for metric collection. Per-NIC HSN metrics collected include: injection and ejection bandwidths, kernel output and input bandwidths. For each ASIC link, dimension metrics collected include: reception data, packet counts (XE/XK only), time stalled, and lane status. The values provided are not continuous values but by using associated timestamps, rates can be determined.

Use the following `CLEinstall.conf` parameter to enable network metric collection.

```
CNL_network_metrics=no
```

Set `CNL_network_metrics` to `yes` in `CLEInstall.conf` to enable network metric collection. For more information on network metric collection, see *Managing System Software for the Cray Linux Environment*.

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.

A system administrator can use this file to 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 *Create Configuration Files* on page 41, 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 5.2.UP`*nn* `Software` DVD (also see */etc/sysset.conf Examples* on page 180 for reference). 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 (or LVM device names for LVM devices, which are also persistent). For more information, see *About Persistent Boot RAID Device Names* on page 38. For more information about LVM configuration, see *Configure LVM for System Backups* on page 179.
- 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 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 *Create the Installation Configuration Files* on page 42, you are directed to create and edit your `/etc/sysset.conf` file. Make all site-specific changes at that point in the installation process.

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

Check the boot RAID configuration and QLogic switch zoning (for QLogic Fibre Channel switch or DDN device), SAS switch zoning, 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 the `/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`.

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 `/dev/disk/by-id` persistent device names or LVM device names (if your system is configured to use LVM), which are also persistent.

CAUTION: You must use `/dev/disk/by-id` or an LVM device name 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.

The code that follows is the 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
```

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

When you create a site-specific `/etc/sysset.conf` file (*Create the Installation Configuration Files* on page 42), 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 37, 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
```

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

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
```

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

Install CLE on a New System

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 SP3) 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 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 already running CLE software on your system, see *Prepare to Update or Upgrade CLE Software* on page 106.

Install CLE Software on the SMW

Three DVDs are required to install the CLE 5.2.UP04 release on a Cray system. The first is labeled `Cray CLE 5.2.UP`*nn* `Software` and contains software specific to Cray systems. Optionally, you may have an ISO image called `xc-sles11sp3-`*n.n.nn*`avv.iso`, where *n.n.nn* indicates the CLE release build level, and *avv* indicates the installer version.

The second DVD is labeled `Cray-CLEbase11sp3-`*yyyymmdd* and contains the CLE 5.2 base operating system which is based on SLES 11 SP3. The third DVD is labeled `CentOS-6.5-x86_64-bin-DVD1.iso` and contains the CentOS 6.5 base operating system for CLE direct-attached Lustre (DAL) nodes.

Copy 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 5.2.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 `xc-sles11sp3-`*5.2.55d05*`.iso` is the path name to the ISO image file.

```
smw:~# mount -o loop,ro xc-sles11sp3-5.2.55d05.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.5.2.55
smw:~# cp -pr /media/cdrom/* /home/crayadm/install.5.2.55
```

5. Unmount the `Cray CLE 5.2.UP`*nn* `Software` DVD and eject it.

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

6. Insert the `Cray-CLEbase11sp3` 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-CLEbase11sp3-`*yyyymmdd*`.iso` is the path name to the ISO image file.

```
smw:~# mount -o loop,ro Cray-CLEbase11sp3-yyyymmdd.iso /media/cdrom
```

Install CLE software on the SMW

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

```
smw:~# /home/crayadm/install.5.2.55/CRAYCLEinstall.sh \
-m /home/crayadm/install.5.2.55 -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. If this script fails, restart it with the same options. However, rerunning this script may generate numerous error messages as it attempts to install already-installed RPMs. Do not be concerned about these messages.

> **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.

Create Configuration Files

CAUTION: *About Installation Configuration Files* on page 21 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 36, the CLE 5.2 release software can be installed on a system that has never had the CLE 5.2 release installed on it, or the release can be installed to an alternative root location.

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

When installing direct-attached Lustre, make sure that the `CLEinstall.conf` specifies this setting: `direct_attached_lustre=yes`.

To configure a system for future LVM backups, follow the procedures in *Configure LVM for System Backups* on page 179 before the CLE install. Install CLE on the LVM-configured system set and you'll be able to use LVM snapshots for future system backups.

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 *About Installation Configuration Files* on page 21 or the `sysset.conf(5)` and `CLEinstall.conf(5)` man pages.

Log out and back in again to access man pages that were installed in *Install CLE software on the SMW* on page 41.

Create the Installation Configuration Files

1. Edit the `/home/crayadm/install.`*`xtrel`*`/CLEinstall.conf` configuration file. Carefully follow *About Installation Configuration Files* on page 21 and make modifications for your specific configuration.

   ```
   smw:~# chmod 644 /home/crayadm/install.5.2.55/CLEinstall.conf
   smw:~# vi /home/crayadm/install.5.2.55/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 there is already 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.5.2.55/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 38 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 38.

```
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 36.

Repurpose Compute Nodes as Service Nodes

CLE and SMW software include functionality to optionally change the role of compute nodes and boot the hardware with service node images. Use this functionality to add service nodes for services that do not require external connectivity, such as `DSL_nodes`. When a compute node is configured with a service node role, that node is referred to as a *repurposed compute node*.

Do not repurpose compute nodes that are intended to be service MAMU nodes until after running the `CLEinstall` program. For more information, see *Configure Service Node MAMU* on page 33.

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.

Mark 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: Aries
           Nodeid: Service  Core Arch|  Comp state      [Flags]
------------------------------------------------------------------------
       c0-0c0s7n0: service  MC24   OP|          on      [noflags|]
------------------------------------------------------------------------
crayadm@smw:~>
```

Run the CLEinstall Program

The CLEinstall 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 | --reconfigure

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 specifying 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. For a reconfiguration of CLE features or hardware changes, use the --reconfigure option. This 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.

--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.*timestamp*.

--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, 5.2.55.

> **NOTE:** The --XTrelease option is required with the --upgrade and --reconfigure options, and it 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, *`XT_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 backup SMW that is not connected to the Hardware Supervisory System (HSS) network, or when you configure an unavailable partition on a partitioned system.

The *`XT_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 on the CLE nodes running the Lustre client. For example, `2.4`. If this option is not specified, `CLEinstall` will use the default version of Lustre. The version specified here does not affect the version of Lustre server that runs on direct-attached Lustre nodes.

`--Centosmedia=directory`

Specify the directory where the CentOS™ software media has been mounted. The `--Centosmedia` option is required when installing or upgrading CLE with direct-attached Lustre. For example, the CentOS image mount point could be `/media/Centosbase`.

`--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. Specify this option when restarting `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.

Run `CLEinstall`

1. For direct-attached Lustre implementations, also mount the `CentOS-6.5-x86_64-bin-DVD1.iso` ISO image if it is not already mounted.

```
smw:~# mkdir /media/Centosbase
smw:~# mount -o loop,ro CentOS-6.5-x86_64-bin-DVD1.iso /media/Centosbase
```

Also include the `--Centosmedia=`*directory* option when invoking `CLEinstall`. In this example, the option is **`--Centosmedia=/media/Centosbase`**.

2. Invoke the `CLEinstall` program on the SMW. `CLEinstall` is located in the directory you created in *Copy the Software to the SMW* on page 40.

```
smw:~# /home/crayadm/install.5.2.55/CLEinstall --install --label=system_set_label \
--configfile=/home/crayadm/install.5.2.55/CLEinstall.conf \
--CLEmedia=/home/crayadm/install.5.2.55
```

3. Examine the initial messages directed to standard output. Log files are created in `/var/adm/cray/logs` and named by using a timestamp that indicates when the install script began executing. For example:

```
08:57:48 Installation output will be captured in /var/adm/cray/logs/\
CLEinstall.p3.20140911085748.CLE52-P3.stdout.log
08:57:48 Installation errors (stderr) will be captured in /var/adm/cray/logs/\
CLEinstall.p3.20140911085748.CLE52-P3.stderr.log
08:57:48 Installation debugging messages will be captured in /var/adm/cray/logs/\
CLEinstall.p3.20140911085748.CLE52-P3.debug.log
```

The naming conventions of these logs are:

CLEinstall.p*#*.*YYYYMMDDhhmmss*.*$LABEL*.*logtype*.log

where

p*#* is the partition number specified in the CLEinstall.conf file.

YYYYMMDDhhmmss is the timestamp in year, month, day, hour, minute, and second format.

$LABEL is the system set label (in the example above, `CLE52-P3`).

logtype is `stdout` (standard output), `stderr` (standard error), or `debug`.

Also, log files are created in `/var/adm/cray/logs` each time `CLEinstall` calls `CRAYCLEinstall.sh`. For example:

```
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.01-B.log
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.02-B.log
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.03-B.log
.
.
.
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.17-S.log
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.18-S.log
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.19-S.log
```

The naming conventions of these logs are:

CRAYCLEinstall.sh.p*#*.*YYYYMMDDhhmmss*.*$LABEL*.*sequence#-root.log*

where

p*#* is the partition number specified in the `CLEinstall.conf` file.

YYYYMMDDhhmmss is the timestamp in year, month, day, hour, minute, and second format. This is the same timestamp used for the log files of the `CLEinstall` program instance that called `CRAYCLEinstall.sh`.

$LABEL is the system set label.

sequence# is an increasing count that specifies each invocation of `CRAYCLEinstall.sh` by `CLEinstall`.

root is either `B` (bootroot) or `S` (sharedroot), specifying the root modified by the `CRAYCLEinstall.sh` call.

4. `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 rerun `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.

5. Monitor the debug output. Create another terminal window and invoke the `tail` command by using the path and timestamp displayed when `CLEinstall` was run.

   ```
   smw~:# tail -f /var/adm/cray/logs/CLEinstall.p#.YYYYMMDDhhmmss.$LABEL.debug.log
   ```

6. Locate the following warning and prompt in the `CLEinstall` console window 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]
   ```

7. 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.

8. 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 installation 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 file names. The list of suggested commands is written to the `/var/adm/cray/logs/CLEinstall.P#.`*`YYYYMMDDhhmmss`*`.`*`$LABEL`*`.command_hints.log` file in the installer log directory.

   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 *Install and Configure Direct-Attached Lustre* on page 93.

   As you complete these procedures, you can cut and paste the suggested commands from the output window or from the window created in a previous step that tailed the debug file. The log files created in `/var/adm/cray/logs` for `CLEinstall.P#.`*`YYYYMMDDhhmmss`*`.`*`$LABEL`*`.stdout.log` and `CLEinstall.P#.`*`YYYYMMDDhhmmss`*`.`*`$LABEL`*`.debug.log` also contain the suggested commands.

Create 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-LABEL-partition/nodetype`*, where *`machine`* is the Cray hostname, *`xtrelease`* is the build level for the CLE release, *`LABEL`* is the system set label used from `/etc/sysset.conf`, *`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 the upgraded system. However, some file content may have changed with the new release. Verify that site-specific modifications are compatible. For example, use existing copies of `/etc/hosts`, `/etc/passwd` and `/etc/modprobe.conf`, but if `/init` changed for the template, the site-modified version that is copied and used for CLE 5.2 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.

Modify Boot Image Parameters for Service Nodes

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

If the `bootimage_temp_directory` is `/home/crayadm/boot`, the modified `parameters` file is:

`/home/crayadm/boot/bootimage.default.`*`LABEL`*`.xtrelease.`*`timestamp`*`/SNL0/parameters`

For example, the default `parameters` file is:

```
/home/crayadm/boot/bootimage.default.CLE52.5.2.14.201403060906/SNL0/parameters-snl
```

The contents of the `parameters` file is a single line, but the following example is 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, this file does not need to be changed.

   ```
   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
   ```

Prepare 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`*. When `shell_bootimage_`*`LABEL`*`.sh` is run, it creates a log file in `/var/adm/cray/logs/shell_bootimage_`*`LABEL`*`.sh.`*`PID`*`.log`. 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.

-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. This option is not applicable to Cray XC30 systems. For more information, see the `xtbounce(8)` man page.

Optionally, this script includes `CNL_*` parameters that can be used to modify the CNL boot image configuration 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 the `-c` option was specified, the script invokes these commands automatically and the remaining steps in this procedure should be skipped.

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.

   For Cray XC30 systems, type this command (for partitioned systems, replace `s0` with `p`*N* where *N* is the partition number for which the image is being built):

   ```
   smw:~# xtbootimg \
   -L /opt/xt-images/hostname-xtrelease-LABEL-s0/compute/CNL0.load \
   -L /opt/xt-images/hostname-xtrelease-LABEL-s0/service/SNL0.load \
   -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 being used. 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 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
   ```

Enable Boot Node Failover

NOTE: Boot node failover is an optional CLE feature.

If boot-node failover has been configured for the first time, follow these steps. If boot-node failover has not been configured, skip this procedure.

To enable bootnode failover, you must set `bootnode_failover` parameters in the `CLEinstall.conf` file before you run the `CLEinstall` program. For more information, see *Configure Boot Node Failover* on page 28.

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 the 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, 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
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` 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 the 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
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` option.

Enable SDB Node Failover

NOTE: SDB node failover is an optional CLE feature.

If SDB node failover has been configured for the first time, follow these steps. If SDB node failover has not been configured, skip this procedure.

In addition to this procedure, refer to *Configure Boot Automation for SDB Node Failover* on page 90 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 *Configure SDB Node Failover* on page 29.

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
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` 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.
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` 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
```

Run 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:~# umount /media/Centosbase
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.

Boot and Log on to the Boot Node

At this point, boot and log on to the boot node on the Cray system. The remaining procedures require dedicated Cray system time.

WARNING: Before starting this procedure, verify that the boot root and shared root file systems are no longer mounted on the SMW. Mounting the file systems on the SMW and boot node simultaneously can corrupt the file systems.

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 displays a series of questions. Cray recommends answering yes by typing **`Y`** in response 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 ...
   6) boot all_serv ...
   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 ...
   16) boot all_serv and wait ...
   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`**.

   ```
   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
   Enter a space-separated list of additional boot parameters (or nothing to do
   nothing): <Press the Enter key.>
   ```

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

   Use this window later to boot the SDB, service, and compute nodes. If the window gets closed, 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 using the `ssh` command to log on to the boot node as root, use one of two fixes for the problem.

The first way is to remove the boot node keys using the following commands:

```
smw:~# smw:~# ssh-keygen -R boot
smw:~# su - crayadm
crayadm@smw> ssh-keygen -R boot
crayadm@smw> exit
```

The second way to fix the problem is to 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.

Change Passwords on the Boot and Service Nodes

For security immediately change the `root` and `crayadm` passwords immediately after login on the boot node for the first time.

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
```

   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.

Change 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 the path specified in the following commands, `/opt/xt-images/templates/default`, with `/opt/xt-images/templates/default-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 *Change Passwords on the Boot and Service Nodes* on page 55, 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 *Prepare Compute and Service Node Boot Images* on page 49.

   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 82.

Modify SSH Keys for Compute Nodes

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

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. Generate a `dropbear` compatible RSA key.

```
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
```

4. Generate a `dropbear` compatible DSS key.

```
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/
kjyDE5IaxjJpIBF262N9UsxleTX6F65OjNoL84fcKqlSL6NV5XJ5O00SKgTuVZjpXO913q9SEhkcI0Zy
0vRQ8
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
```

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

   For the RSA key:

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

```
crayadm@smw:~/dropbear_ssh_keys> su root
smw:/home/crayadm/dropbear_ssh_keys # cp -p ssh_host_dss_key.db /opt/xt-images/
templates/default/etc/ssh/ssh_host_dss_key
```

6. Update the boot image to include these changes; follow the steps in *Prepare Compute and Service Node Boot Images* on page 49.

Modify the `/etc/hosts` File

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

Site-specifc networking requirements can make changes to the `/etc/hosts` file necessary. Follow this procedure to edit the hosts file for the boot node, service nodes, and CNL compute nodes.

IMPORTANT: `CLEinstall` modifies Cray system entries in `/etc/hosts` each time you update or upgrade your CLE software. For additional information, see *Maintain Node Class Settings and Hostname Aliases* on page 22.

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 the 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 *Prepare Compute and Service Node Boot Images* on page 49.

      NOTE: You can defer this step and update the boot image once before you *Finish Booting the System* on page 82.

Configure 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.

Configure 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.

   IMPORTANT: If you are configuring an RSIP gateway, you must disable GRO in the ETHTOOL_OPTIONS of the network interface by adding the following line:

   ```
   ETHTOOL_OPTIONS="-K iface gro off"
   ```

   Repeat the previous steps in this procedure 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
```

Configure 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. 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 *Configure cron Services* on page 75)

   `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
   ```

Configure 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 connected service nodes on a Cray system, acting as Lustre servers, to external storage devices. These nodes are commonly referred to as LNET routers. Follow *Configure InfiniBand on Service Nodes* on page 61.
- IB can provide IP connectivity between devices on the fabric. To configure IP over InfiniBand (IPoIB), follow *Configure IP Over InfiniBand (IPoIB) on Cray Systems* on page 63.
- If you are using devices that require the SCSI RDMA Protocol (SRP), follow *Configure and enable SRP on Cray Systems* on page 63.

  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 the Cray Linux Environment* (S-2393).

Configure 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 the 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: The system administrator is prompted to type **c** and enter a brief comment describing the changes made. To complete the comment, type **Ctrl-d** or a period on a line by itself. Do this each time `xtopview` is exited to log a record of revisions into an RCS system.

6. Proper IPoIB operation requires additional configuration. See *Configure IP Over InfiniBand (IPoIB) on Cray Systems* on page 63.

Configure 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: To configure IPoIB for another IB interface connected to this node, repeat step *2* on page 63 and step *3* on page 63 for `/etc/sysconfig/network/ifcfg-ib`*n*.
For LNET traffic, each IB interface should be assigned a unique IP address from the subnet that it will operate on. For TCP/IP traffic, multiple IB interfaces on a node must be assigned unique IP addresses from different subnets.

Configure and enable 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/infinibandand 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. Optional: 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:~ #
```

Complete 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 ...
6) boot all_serv ...
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 ...
16) boot all_serv and wait ...
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:
```

Boot and Configure the SDB Node

Continue in the terminal session for `xtbootsys` that you started in *Boot and Log on to the Boot Node* on page 53.

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

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

   ```
   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
   Enter a space-separated list of additional boot parameters (or nothing to do
   nothing): <Press the Enter key.>
   ```

   NOTE: Until you start the SDB MySQL database in later in the installation process, 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
   ```

   This command generates `ssh` DSA, RSA, and ECDSA 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 on Cray XE and Cray XK systems.

   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.
```

For the ECDSA key:

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

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
```

Change Default MySQL Passwords on the SDB

Access to MySQL databases requires a user name and password. The MySQL accounts and privileges are

MySQL basic	Read access to most tables; most applications use this account
MySQL sys_mgmt	Most privileged; access to all information and commands

For security, Cray recommends changing the default passwords for MySQL database accounts. The valid characters for use in MySQL passwords are:

```
  ! " # $ % & ' ( )
* + , - . / 0 1 2 3
4 5 6 7 8 9 : ; < =
> ? @ A B C D E F G
H I J K L M N O P Q
R S T U V W X Y Z [
\ ] ^ _ ` a b c d e
f g h i j k l m n o
p q r s t u v w x y
z { | } ~
```

1. If a site-specific MySQL password for root has not been set, complete this step.

a. Log on to the SDB.

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

b. Invoke the MySQL monitor.

```
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: 5.5.31-log Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

c. Set passwords. Use the actual name of the system database (SDB) node if it is not named `sdb`. For example, the node could be named `sdb-p3` on a partitioned system.

```
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. Change the password for the `sys_mgmt` account. This requires an update to `.my.cnf` in step *4* on page 68.

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

b. Change the password for the `basic` account. This requires an update to `/etc/opt/cray/sysadm/odbc.ini` in step *5* on page 69.

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)
```

The connection may time out when making changes to the MySQL database, but it is automatically reconnected. If this happens, messages similar to the following are displayed. 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
```

4. Optional: If a site-specific password for `sys_mgmt` was set, update the `.my.cnf` file for `root` with the new password. Additionally, update the `.odbc.ini.root` file with the new password.

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

   ```
   boot# vi /root/.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
   ```

   c. Exit `xtopview` and edit `.odbc.ini.root` for the root on the boot node. Update **each** database section with the new password.

   ```
   default/:/ # exit
   boot# vi /root/.odbc.ini.root
   ```

   ```
   Driver          = MySQL_ODBC
   Description     = Connector/ODBC Driver DSN
   USER            = sys_mgmt
   PASSWORD        = newpassword
   ```

   d. Copy `.odbc.ini.root` to `.odbc.ini`.

   ```
   boot:~ # cp -p /root/.odbc.ini.root /root/.odbc.ini
   ```

   e. Invoke `xtopview` and edit `.odbc.ini.root` for the root in the shared root. Update **each** database section with the new password.

   ```
   boot# xtopview
   default/:/ # vi /root/.odbc.ini.root
   ```

   ```
   Driver          = MySQL_ODBC
   Description     = Connector/ODBC Driver DSN
   USER            = sys_mgmt
   PASSWORD        = newpassword
   ```

   f. Exit `xtopview` and restart the SDB service on all service nodes.

   ```
   default/:/ # exit
   boot:~ # pdsh -a /etc/init.d/sdb restart
   ```

5. Optional: If a site-specific password for `basic` was set, update **each** datasource name in the `/etc/opt/cray/sysadm/odbc.ini` file with the new password. Additionally, update the `/root/.odbc.ini` file with the new password.

   a. Edit `/etc/opt/cray/sysadm/odbc.ini` for the basic user on the boot node. Update **each** database section with the new password.

   ```
   boot# vi /etc/opt/cray/sysadm/odbc.ini
   ```

   ```
   Driver          = MySQL_ODBC
   Description     = Connector/ODBC Driver DSN
   USER            = basic
   PASSWORD        = newpassword
   ```

   b. Invoke `xtopview` and edit `/root/odbc.ini` in the shared root. Update **each** database section with the new password.

   ```
   boot# xtopview
   default/:/# vi /root/odbc.ini
   ```

   ```
   Driver          = MySQL_ODBC
   Description     = Connector/ODBC Driver DSN
   USER            = basic
   PASSWORD        = newpassword
   ```

   c. Exit `xtopview`.

Add 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`. Other optional services can be added using this procedure.

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

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

   Use this command also to show the services already assigned.

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

   NOTE: Do not add `SYSLOG` or `CRON` by using `xtservconfig`. Follow *Configure cron Services* on page 75 and *Configure System Message Logs* on page 81 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.

Configure 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.

Configure Service Node MAMU

Service Node MAMU support provides the ability to set aside a small number of repurposed compute nodes for serial workload. These nodes serve as workload management execution nodes, specifically designated for serial workload, and intra-node MPI. The workload manager manages these as standard Linux nodes and support core level placement.

Repurpose Compute Nodes as Service Nodes

CLE and SMW software include functionality to optionally change the role of compute nodes and boot the hardware with service node images. Use this functionality to add service nodes for services that do not require external connectivity, such as `DSL_nodes`. When a compute node is configured with a service node role, that node is referred to as a *repurposed compute node*.

Do not repurpose compute nodes that are intended to be service MAMU nodes until after running the `CLEinstall` program. For more information, see *Configure Service Node MAMU* on page 33.

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.

Configure UFS with Postprocessing Nodes

Use of UFS for user accounts is discouraged. UFS must be used for system accounts like `crayadm`. The directory created provides a minimal administrative account directory for `crayadm`. Configuring the home directory on `/ufs` puts a large load on the Boot RAID and can lead to instability. In this example, `postproc` is a node class created for MAMU nodes and *`ufs`* is the value of `node_ufs_hostname` in the `CLEinstall.conf` file. Do this task for each defined class that consists of MAMU nodes.

```
boot# xtopview -c postproc
class/postproc:/# xtspec /etc/fstab
```

Add this line:

```
ufs:/ufs/home         /ufs/home     nfs      tcp,rw  0 0
```

```
class/postproc:/# exit
boot# xtopview
default/:/# mkdir -p /ufs/home
default/:/# exit
```

Boot the Remaining Service Nodes

Continue in the terminal session for `xtbootsys` that you started in *Boot and Log on to the Boot Node* on page 53.

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

1. Select option `13` to boot the service nodes.

```
Enter your boot choice: 13
```

2. Type `p0` to boot the remaining service nodes in the entire system, or `pN` (where *N* is the partition number) to boot a partition. Confirm the selection of nodes and send the ec_boot event by pressing the Enter key or typing **`Y`**.

```
Enter a service list (or nothing to do nothing): p0
'xtcli status -a p0' completed with status 0
'xtcli status -t aries_lcb p0' completed with status 0
Do you want to boot service c0-0c0s0n1,<complete node list not shown>,c0-0c1s2n2 ? [Yn] Y
Do you want to send the ec_boot event ('no' means to only load memory) ? [Yn] Y
Enter a space-separated list of additional boot parameters (or nothing to do nothing): <Press the
Enter key.>
```

3. After the specified service nodes are booted, you are prompted to `Enter your boot choice` again. Do not close the `xtbootsys` terminal session. Use this terminal session later to boot the compute nodes.

Populate the known_hosts 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`.

Configure Lustre File Systems

If you plan to configure Lustre file systems, follow the procedures in *Install and Configure Direct-Attached Lustre* on page 93 and then return here to continue the installation.

Create New Login Accounts

NOTE: 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; 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 the Cray Linux Environment* (S-2393).

Configure the Login Failure Logging PAM

NOTE: 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`.

Configure `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.
   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.

   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]
   auth required pam_tally.so [file=alternatepath]
   ```

   This example shows the file modified to report failed login using an alternate location for the tally file.

   ```
   #
   # /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
   ```

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]
```

This example shows the file modified to report failed login using an alternate location for the tally file.

```
#
# /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
```

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]
```

This example shows the file modified to report failed login using an alternate location for the tally file.

```
#
# /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
```

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
```

Configure the Load Balancer

NOTE: 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.

Configure `lbnamed` on the SMW

1. 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.

   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`.

   ```
   smw:/etc/opt/cray-xt-lbnamed/ # vi lbnamed.conf
   ```

   ```
   $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. 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.

   ```
   smw:/etc/opt/cray-xt-lbnamed/ # vi poller.conf
   ```

   ```
   #
   # groups
   # ------------------------
   # login     mycray1-mycray3
   ```

```
mycray1 1 login
mycray2 1 login
mycray3 1 login
```

Install the Load Balancer on an External "White Box" Server

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 5.0.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.

Configure cron Services

NOTE: 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 *Configure cron for the SMW and the Boot Node* on page 75.

On the shared root, configuring `cron` for CLE depends on whether persistent `/var` is set up. If persistent `/var` exists, follow *Configure cron for the Shared Root with Persistent /var* on page 76; otherwise, follow *Configure cron for the Shared Root without Persistent /var* on page 76.

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 they must be manually enabled to be used.

Configure cron for the SMW and the Boot Node

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 SMW:

   ```
   smw:~# chkconfig cron
   cron on
   ```

   On the boot node:

```
boot:~ # chkconfig cron
cron off
```

2. Configure the `cron` daemon to start.
   For this example, enable `cron` on the boot node:

   ```
   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 the system does not have a persistent `/var`, Cray recommends following *Configure cron for the Shared Root without Persistent /var*.

Configure 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 site 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
   ```

Configure 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 must be turned on with the `xtservconfig` command to specify the nodes on which the daemon should run.

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 site 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
```

   f. Enable the `cron` service on a single login node (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.

Configure IP Routes

NOTE: 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.

Configure 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 *Prepare Compute and Service Node Boot Images* on page 49. This step can be deferred by updating the boot image once before you *Finish Booting the System* on page 82.

Configure Cray DVS

NOTE: 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 Cray service nodes acting only as DVS clients, you will need to insert the following line into the specialized `/etc/modprobe.d/dvs` file for that class of service node:

```
options dvsipc dvsipc_config_type=0
```

For more information, see *Introduction to Cray Data Virtualization Service* (S-0005) and the `dvs(5)` man page.

Configure 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.

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. 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. Update the boot image with the DVS configuration by preparing new compute and service node boot images.

   The `CNL_dvs` parameter in the `CLEinstall.conf` file should be enabled before the `CLEinstall` program ran. If DVS was not turned on, edit the `/var/opt/cray/install/shell_bootimage_LABEL.sh` script and set `CNL_DVS=y` before updating the boot image.

NOTE:

It is important to keep `CLEinstall.conf` consistent with changes made to the system configuration in order to avoid unexpected changes during upgrades or updates. Remember to set `CNL_dvs` equal to `yes` in `CLEinstall.conf`.

You can defer updating the boot image and update it once before you *Finish Booting the System* on page 82.

Configure System Message Logs

Use the Lightweight Log Management (LLM) System for message log configuration. This system forwards Syslog and other logs to the SMW without keeping a local copy on the Cray system. This system does not normally require additional configuration on the service nodes beyond setting `LLM=yes` in the `CLEinstall.conf` file prior to installing or upgrading the CLE software.

Configure 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`).

Customize Intel Xeon Phi Coprocessor Nodes

The `/opt/xt-images/templates/default` area is used to customize files for compute node `initramfs`. For CLE systems with Intel Xeon Phi Coprocessor nodes, there are some additional customizations that can be done.

1. To customize any file underneath `/opt/xt-images/templates/default` to be different for Intel Xeon Phi coprocessor nodes than for other compute nodes, create a copy of the file with the `.knc` suffix. During the creation of a bootimage with the `shell_bootimage_LABEL.sh` script, any file with the `.knc` suffix will be removed from the `initramfs` for service and compute nodes, but will be renamed for the `initramfs` used to boot the Intel Xeon Phi coprocessor nodes. The `CLEinstall` program configures `/dsl` and writeable `tmp` for the Xeon Phi node (with any differences in syntax) just like they are for the compute node based on settings in the `CLEinstall.conf` file.

2. If the compute nodes mount a Lustre file system, then a special customization is needed for `/opt/xt-images/templates/default/etc/fstab.knc`. Compute nodes mount a file system of type Lustre, but the Intel Xeon Phi coprocessor nodes need to use DVS to access the Lustre file system.

   For example, if the compute nodes have this entry in `/opt/xt-images/templates/default/etc/fstab` to mount a Lustre files system from `nid 74` in `/lus/dal`:

   ```
   74@gni:/dal /lus/dal lustre rw,flock,user_xattr 0 0
   ```

then the `/opt/xt-images/templates/default/etc/fstab.knc` file should change that entry to this one to mount Lustre from a DVS server node (nid 5), which is projecting all Lustre file systems underneath the `/lus` mount point:

```
/lus /lus dvs path=/lus,nodename=c0-0c0s1n1,blksize=1048576
```

After changing any files in `/opt/xt-images/templates/default` the boot image will need to be rebuilt. For more information, see *Create Boot Images* on page 48.

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 *Configure Boot Automation on the SMW* on page 89.

IMPORTANT: If you deferred updating the boot image in any of the previous procedures, update the boot image now by following the steps in *Prepare Compute and Service Node Boot Images* on page 49.

Boot the 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 *Boot and Log on to the Boot Node* on page 53.

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: <Press the Enter key.>
Enter an alternative CPIO archive name (or nothing): <Press the Enter key.>
Do you want to send the ec_boot event ('no' means to only load memory) ? [Yn]
Enter a space-separated list of additional boot parameters (or nothing to do nothing): <Press the
Enter key.>
```

3. The nodes boot, then this prompt is displayed when password-less ssh is not set up. Type **`n`** to continue.

```
You have not set up password-less access for this account to
execute this command and you also have not provided a password to use.
Answering 'no' to the next question will let you try again.
Do you want to quit this application ? [Yn] n
Enter the default root password:
```

Enter the default root password (**`initial0`**).

4. After all the compute nodes are booted, return to the `xtbootsys` menu. Type **q** to exit the `xtbootsys` program. The output after successful compute node booting is similar to the following:

```
Enter your boot choice: q
'cpio -it -F /tmp/boot/venus-ORANGE-5.2.82.cpio' completed with status 0
'cpio -ivdmu -F /tmp/boot/venus-ORANGE-5.2.82.cpio SNL0/parameters' completed with status 0
Gathering ko files from bootnode:/rr/current/lib/modules/`uname -r`
spawn ssh -o StrictHostKeyChecking=no root@boot-p0 uname -r
Warning: Permanently added 'boot-p0,10.3.1.254'(ECDSA) to the list of known hosts.
3.0.101-0.46.1_1.0502.8871-cray_ari_s
'ssh -o StrictHostKeyChecking=no root@boot-p0 uname -r'
process exited with status '0'
spawn ssh -o StrictHostKeyChecking=no root@boot-p0 file /rr/current/lib/modules/
3.0.101-0.46.1_1.0502.8871-cray_ari_s
/rr/current/lib/modules/3.0.101-0.46.1_1.0502.8871-cray_ari_s:directory
'ssh -o StrictHostKeyChecking=no root@boot-p0 file /rr/current/lib/modules/
3.0.101-0.46.1_1.0502.8871-cray_ari_s'
process exited with status '0'
'rsync -rptgov --copy-unsafe-links --copy-dirlinks --filter "+ */" --filter "+ *.ko"--filter "- *" \
   root@boot-p0:/rr/current/lib/modules/3.0.101-0.46.1_1.0502.8871-cray_ari_s/var/opt/cray/debug/ \
   p0-20150910t094046/shared-root
process exited with status '0'
Gathering ko files from bootnode:/lib/modules/`uname -r`
spawn ssh -o StrictHostKeyChecking=no root@boot-p0 uname -r
3.0.101-0.46.1_1.0502.8871-cray_ari_s
'ssh -o StrictHostKeyChecking=no root@boot-p0 uname -r'
process exited with status '0'
spawn ssh -o StrictHostKeyChecking=no root@boot-p0 file \
   /lib/modules/3.0.101-0.46.1_1.0502.8871-cray_ari_s
/lib/modules/3.0.101-0.46.1_1.0502.8871-cray_ari_s: directory
'ssh -o StrictHostKeyChecking=no root@boot-p0 file \
   /lib/modules/3.0.101-0.46.1_1.0502.8871-cray_ari_s'
process exited with status '0'
'rsync -rptgov --copy-unsafe-links --copy-dirlinks --filter "+ */" --filter "+ *.ko" --filter "- *" \
   root@boot-p0:/lib/modules/3.0.101-0.46.1_1.0502.8871-cray_ari_s/var/opt/cray/debug/ \
   p0-20150910t094046/boot-root'
process exited with status '0'
This session took 3223 seconds (53 minutes, 43 seconds).
######################################################################
Session Boot Summary:  14 nodes completed their boot
######################################################################
INFO: closing exp8
data(config,auto_flood_control) is off
######################################################################
Your boot session identifier is p0-20150910t094046
######################################################################
```

Flash the nvBIOS for Kepler GPUs

A Cray XC30 system with NVIDIA® Tesla® SXM modules requires an update to the NVIDIA BIOS (nvBIOS) for the NVIDIA K20X and K40s graphics processing units (GPUs). The nvBIOS is unique for each SXM-1 Kepler™ SKU, based on the type of heat sink, as shown below.

GPU Type	Board SKU	Production Firmware Image Version
Kepler K20X (13 fin)	P2085 SKU 202	80.10.44.00.02
Kepler K20X (20 fin)	P2085 SKU 212	80.10.44.00.04
Kepler K20X (30 fin)	P2085 SKU 222	80.10.44.00.05
Kepler K40s (13 fin)	P2085 SKU 209	80.80.4B.00.03
Kepler K40s (20 fin)	P2085 SKU 219	80.80.4B.00.04

GPU Type	Board SKU	Production Firmware Image Version
Kepler K40s (30 fin)	P2085 SKU 229	80.80.4B.00.05

The CLE software includes a script that automatically determines the SKU version and flashes the nvBIOS with the appropriate firmware.

> **TIP:** You can use the `cnselect` command to identify the number and location of the Kepler GPUs. This example shows a system with K20X GPUs on four nodes.
>
> ```
> login:~# cnselect -c "subtype.eq.'nVidia_Kepler'"
> 4
> login:~# cnselect -e "subtype.eq.'nVidia_Kepler'"
> 70-73
> ```

1. As root on the login node, set the allocation mode for all compute nodes to interactive.

   ```
   login:~# xtprocadmin -km interactive
   ```

2. Change to the directory containing the NVIDIA scripts.

   ```
   login:~# cd /opt/cray/cray-nvidia/default/bin
   ```

3. To flash the Kepler K20X GPUs, for example, choose one of the following options.

   - To update the entire system:

     ```
     login:~# aprun -n `cnselect -c "subtype.eq.'nVidia_Kepler'"` \
     -N 1 -L `cnselect -e "subtype.eq.'nVidia_Kepler'"` \
     ./nvFlashBySKU -b
     ```

   - To update a single node or set of nodes:

     ```
     login:~# aprun -n PEs -N 1 -L node_list ./nvFlashBySKU -b
     ```

     > **NOTE:** Replace *PEs* with the number of nodes; replace *node_list* with a comma-separated list of NIDs.
     >
     > For example, to flash four GPUs on nodes 70-73:
     >
     > ```
     > login:~# aprun -n 4 -N 1 -L 70,71,72,73 ./nvFlashBySKU -b
     > c0-0c0s1n0: Nid 70: Successful Cray Graphite K20X nvBIOS flash
     > c0-0c0s1n1: Nid 71: Successful Cray Graphite K20X nvBIOS flash
     > c0-0c0s1n2: Nid 72: Successful Cray Graphite K20X nvBIOS flash
     > c0-0c0s1n3: Nid 73: Successful Cray Graphite K20X nvBIOS flash
     > ```

4. If there is a flash failure, `nvFlashBySKU` displays an error message with the failing node ID, as in this example:

   ```
   c0-0c0s1n3: Nid 73: Failed Cray Graphite K20X nvBIOS flash
   ```

   Depending on the type of failure, `nvFlashBySKU` might display additional information, if available. No flashing is done on unsupported SKUs.

   If a GPU fails to flash, the SXM-1 card must be replaced.

5. After flashing is successful, use `xtbootsys` to reboot the nodes from the SMW. For example:

```
crayadm@smw:~> xtbootsys --reboot -L CNL0 -r "rebooting after nvBIOS update" \
c0-0c0s1n0,c0-0c0s1n1,c0-0c0s1n2,c0-0c0s1n3
```

TIP: You can use `xtprocadmin` on the login node to determine each node name from the `cnselect` output, as in this example:

```
login:~# xtprocadmin -n `cnselect -e "subtype.eq.'nVidia_Kepler'"`
   NID    (HEX)    NODENAME     TYPE    STATUS        MODE
    70     0xf8  c0-0c0s1n0  compute        up       batch
    71     0xf9  c0-0c0s1n1  compute        up       batch
    72     0xfa  c0-0c0s1n2  compute        up       batch
    73     0xfb  c0-0c0s1n3  compute        up       batch
```

6. After the reboot is successful, log on to the login node as root and change to the directory containing the NVIDIA scripts.

```
login:~# cd /opt/cray/cray-nvidia/default/bin
```

To verify that all GPUs are reporting the correct nvBIOS version (see the table above), choose one of the following options:

- To display the nvBIOS versions for the entire system:

```
login:~# aprun -n `cnselect -c "subtype.eq.'nVidia_Kepler'"` \
-N 1 -L `cnselect -e "subtype.eq.'nVidia_Kepler'"` \
./xkcheck -n -c -f | grep Version
```

- To display the nvBIOS versions for a single node or set of nodes:

```
login:~# aprun -n PEs -N 1 -L node_list \
./xkcheck -n -c -f | grep Version
```

Replace *PEs* with the number of nodes; replace *node_list* with a comma-separated list of NIDs.

For example:

```
login:~# aprun -n 4 -N 1 -L 70,71,72,73 \
./xkcheck -n -c -f | grep Version
4 nodes report VBIOS Version                  : 80.10.3D.00.05
```

7. Reset the compute nodes to the normal batch or interactive mode using the `xtprocadmin` command.

Test 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. The output for Cray XE and Cray XK systems follows.

```
crayadm@login:~> xtnodestat
Current Allocation Status at Fri May 21 07:11:48 2012

     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 output for Cray XC30 systems follows.

```
crayadm@login:~> xtnodestat
Current Allocation Status at Fri Feb 15 15:01:38 2013

     C0-0             C1-0             C2-0             C3-0
  n3 ---a-------a---- ---a------------ --aa-------a---- ----------------
  n2 ---a-------a---- ---a------------ --aa-------a---- ----------------
  n1 ---a-------a---- ---a------------ --Xa-------a---- ----------------
c2n0 ---a-------a---- ---a------------ --aa-------a---- ----------------
  n3 -----bX--a-a---- ---a-------a-a-- - a--------a---- ---a------------
  n2 -----ba--a-a---- ---a-------a-a-- -Sa--------a---- ---X------------
  n1 -----ba--a-a---- ---a-------a-a-- -Sa--------a---- ---a------------
c1n0 -----ba--a-a---- ---a-------X-a-- - a--------a---- ---a------------
  n3   -a------a-----   ---------a---- - -a------------  --a-------a----
  n2 SS-a------a----- SS---------a---- -S-a------------ S--X-------a----
  n1 SS-X------a----- SS---------a---- -S-a------------ S--a------Aa----
c0n0   -a------a-----   ---------a---- - -X------------  --a------Aa----
    s0123456789abcdef 0123456789abcdef 0123456789abcdef 0123456789abcdef


Legend:
   nonexistent node                      S  service node
;  free interactive compute node         -  free batch compute node
A  allocated (idle) compute or ccm node  ?  suspect compute node
W  waiting or non-running job            X  down compute node
Y  down or admindown service node        Z  admindown compute node

Available compute nodes:          0 interactive,        650 batch
```

The `xtprocadmin` command displays the current values of processor flags and node attributes. The output for Cray XE and Cray XK systems follows.

```
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 output for Cray XC30 systems follows.

```
crayadm@login:~> xtprocadmin
   NID    (HEX)    NODENAME     TYPE    STATUS        MODE
     1      0x1  c0-0c0s0n1  service        up       batch
     2      0x2  c0-0c0s0n2  service        up       batch
     5      0x5  c0-0c0s1n1  service        up       batch
     6      0x6  c0-0c0s1n2  service        up       batch
     8      0x8  c0-0c0s2n0  compute        up       batch
     9      0x9  c0-0c0s2n1  compute        up       batch
    10      0xa  c0-0c0s2n2  compute        up       batch
```

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    733    733    107     89    626      0

Total pending applications: 4
Pending Pid     User     w:d:N NID     Age Command  Why
      17278  crayadm 1848:1:24   5   0h53m  ./app1 Busy
      17340  crayadm 1848:1:24   5   0h53m  ./app1 Busy
      17469  crayadm 1848:1:24   5   0h52m  ./app1 Busy
      26155  crayadm 1848:1:24   5   0h12m  ./app2 Busy

Total placed applications: 2
   Apid ResId   User  PEs Nodes    Age State        Command
1631095   135 alan-1   64     4  0h31m   run            mcp
1631145   140  flynn  128     8  0h05m   run  TRON-JA307020
```

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.

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
5.2.UP04
```

If the preceding simple tests ran successfully, the system is operational. Cray recommends using the `xthotbackup` utility to create a backup of a newly updated or upgraded system. For more information, see the `xthotbackup(8)` man page.

Configure Boot Automation on the SMW

A sample boot automation file, `/opt/cray/hss/default/etc/auto.generic.cnl`, is provided as a basis for further customizing the boot process. A system shutdown is required to test the customized boot automation files.

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

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
'xtcli shutdown $data(idlist)' completed with status 0

connecting to boot node (boot-p0)

spawn ssh -o StrictHostKeyChecking=no -x root@boot-p0
Warning: Permanently added 'boot-p0' (ECDSA) to the list of known hosts.
Password:

You have not set up password-less ssh from this
account on the SMW to 'root' on the boot node
and you also have not provided a password to use.
** Answering 'no' to the next question will let you try again. **
Do you want to quit this application ? [Yn] n
Enter your root password :
```

Enter the default root password (**initial0**).

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/hss/default/etc/auto.generic.cnl \
/opt/cray/hss/default/etc/auto.xthostname
```

3. Edit the boot automation file.

```
crayadm@smw:~> vi /opt/cray/hss/default/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, see *Configure a Boot Automation File for DAL* on page 104.

   c. Make additional site-specific changes as needed and save the file.

4. Use the `xtbootsys` command to boot the Cray system.

   CAUTION: Shut down the Cray system before invoking the `xtbootsys` command. If installing to an alternate system set, shut down the currently running system before booting 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. Reboot the 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.

Configure Boot Automation for SDB Node Failover

If SDB node failover is configured on the system and there are commands in the boot automation script that apply to the SDB, follow these steps to ensure the appropriate boot automation commands are invoked if an SDB node fails.

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.

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, 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
   ```

Configure Boot Automation for DataWarp

When Cray DataWarp is enabled and configured on the system, edit the boot automation file to add a required kernel boot parameter for the DataWarp manager nodes for increased performance.

1. Edit the boot automation file.

   ```
   crayadm@smw:~> vi /opt/cray/hss/default/etc/auto.xthostname
   ```

2. Add the DataWarp manager nodes with SSDs to the boot automation file. Add these commands after the service database (SDB) node, yet before other service nodes are started.

   ```
   lappend actions { crms_sleep 5 }
   lappend actions  [list crms_boot_loadfile SNL0 service node_list linux numa=fake=2]
   ```

   Where *node_list* is a comma-separated list of DataWarp manager node cnames.

   Further guidance for booting these nodes

   After the system is booted using a boot automation file containing the DataWarp edits, using the `xtbootsys --reboot` command reboots the DataWarp manager nodes and maintains the `numa=fake=2` kernel parameter. If the DataWarp manager nodes are rebooted manually using `xtcli boot` directly, this parameter is not preserved. When a manual boot is needed, be sure to pass the `numa=fake=2` parameter on the `xtcli` command line. For example:

   ```
   crayadm@smw> xtcli boot SNL0 c0-0c0s3n2 -- numa=fake=2
   ```

Post Installation System Management

The Cray system running CLE software should now have an operational . For information about additional software you may need on your system, including programming environment and batch software, see *Install Additional Software* on page 161. *Install RPMs* on page 165 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 the Cray Linux Environment* (S-2393). 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 the Cray Linux Environment (S-2393) 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 wish to use the following optional features, additional configuration is required. See *Managing System Software for the Cray Linux Environment* (S-2393) for more information.
- Dynamic Shared Objects and Cluster Compatibility Mode (CCM)
- Comprehensive System Accounting (CSA)

Install and Configure Direct-Attached Lustre

NOTE: The direct-attached Lustre file system is optional; your storage RAID may be external to the mainframe.

This chapter contains the information and procedures required to initially install and configure a direct-attached Lustre (DAL) file system on a new Cray system. Installation and configuration of DAL differs from that of external Lustre. Because the Lustre community no longer supports Lustre on SLES-based servers, Cray service nodes that support DAL must now use a CentOS™ operating system running on ramdisk, as opposed to the shared root file system.

WARNING: The procedures in this chapter are for initial DAL installation only. Data will be lost if these procedures are followed on an existing DAL file system. If DAL is running on the system, see *Update CLE Software* on page 127 or *Upgrade CLE Software* on page 109.

Installation and configuration of DAL is facilitated by Cray's Image Management and Provisioning System (IMPS), a new set of features that changes how software is installed, managed, provisioned, booted and configured. DAL is currently the only Cray product installed using IMPS; however, in future releases, IMPS will support the installation of other Cray products.

For an introduction to IMPS concepts and the commands used in this procedure, see *IMPS Guide for DAL Installation* (S-0049).

Before Starting the DAL Installation

IMPORTANT: Prior to starting the instructions in this chapter, install the Cray Linux Environment (CLE) base operating system and Cray CLE software packages on your system as described in *Install CLE on a New System* on page 40. Be certain that the following were true for the CLE installation:

- The parameter direct_attached_lustre was set to yes in `CLEinstall.conf`
- The --Centosmedia=*`directory`* option was included when `CLEinstall` was executed
- The -d option was included when the `shell_bootimage` script was executed

Build the DAL Image Root

An *image root* is a directory of the contents a bootable image will eventually contain and is built using an image recipe.

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

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

2. Build the image root.

```
smw:~ # impscli build image_recipe dal_cle_5.2up04_centos_6.5_x86-64_net
```

   Where *net* is `ari` for Cray XC30 systems or `gem` for Cray XE/XK systems. This image root will later be provisioned to generate an image for deployment.

Create the Config Set for DAL Nodes

A *config set* contains site specific settings used by services throughout the Cray system. The IMPS Configurator creates a config set by interactively guiding the administrator through the process of providing needed configuration values. As the Configurator prompts for information about the system, a description and guidance, including a reasonable default value, are provided for each query.

The prompts generated by the IMPS Configurator vary from site to site due to system differences and, therefore, are not displayed in this guide. Key prompts include requests for information regarding node identifiers for DAL and InfiniBand, name service details, time zone information, and LNET routing information.

1. Gather system configuration details. It is helpful to gather the following information before launching the Configurator.
   - Node Identifiers (NIDs) and HSN IP addresses of the DAL nodes
   - NIDs of InfiniBand nodes (if applicable)
   - Is TCP/IP over InfiniBand supported on this system?
   - Is this system configured for multipath?
   - Do service nodes utilize a name service such as LDAP? What are the server addresses?
   - LNET configuration details
   - LNET routing network information (e.g., tcp0, gni, gni32)
   - Will the Lustre Monitoring Tool (LMT) be utilized? What will be the LMT administrator's password for this tool?
   - Is `ssh` allowed?
   - Time zone

2. Launch the Configurator to create the config set. Config sets for DAL nodes reside on the SMW and are named for the partition to be booted, e.g., `p0`, `p1`, `p2`.

   CAUTION: In previous releases, `s`*N* was a valid partition name. This is no longer true, only `p`*N* is a valid partition name.

```
smw:~ # impscli create config_set pN with images \
dal_cle_5.2up04_centos_6.5_x86-64_net
```

   Where *net* is `ari` for Cray XC30 systems or `gem` for Cray XE/XK systems.

Create a Valid `multipath.conf` File

If the DAL nodes utilize multipath, a valid `multipath.conf` file must exist. If a `multipath.conf` file is present, multipath is started when the DAL nodes are booted. To facilitate creating a valid `multipath.conf` file, a template file is available on the SMW in `/etc/opt/cray/share/`*`pN`*`/dist.d/multipath.conf.cray` after the configuration step.

Although no modifications to the `multipath.conf` file are required, most sites will want to add aliases for the OST devices. Configuring Lustre is easier when using aliases. If aliases are not defined, the DAL nodes must be booted with the `multipath.conf` file in place in order for multipath to generate new OST device names. There is an example of defining an alias at the bottom of the template file, similar to the following:

```
# Example of using alias names instead of WWID.s
# Then use /dev/mapper/<alias> in the lustre xxx.fs_defs
# multipaths {#multipaths {
# multipath {
#   wwid   360001ff08052b000000000308aa20000
#  alias  ccsfs-ost000
# }
#}
```

1. Optional: Edit the template file to have the correct values for the system.

   ```
   smw:~ # vi /etc/opt/cray/share/pN/dist.d/multipath.conf.cray
   ```

2. Optional: Verify the proper zoning of the system.

3. Optional: Copy the template file to `/etc/opt/cray/share/`*`pN`*`/files/class/ib-oss/etc/multipath.conf`. When this file is in place, multipath will be activated at boot time for all nodes in the `ib-oss` class.

   ```
   smw:~ # cp /etc/opt/cray/share/pN/dist.d/multipath.conf.cray \
   /etc/opt/cray/share/pN/files/class/ib-oss/etc/multipath.conf
   ```

4. Optional: Use the multipath devices in `/dev/mapper` when editing the Lustre *`fs_name`*`.fs_defs` file a later DAL configuration task.

Provision the DAL Image

Provisioning transforms the image contents to the proper format for deployment.

```
smw:~ # impscli provisiondal image dal_cle_5.2up04_centos_6.5_x86-64_net to \
/opt/xt-images/machine-xtrelease-LABEL-partition
```

Where *`net`* is `ari` for Cray XC30 systems or `gem` for Cray XE/XK systems. The image is stored in the directory `/opt/xt-images/`*`machine-xtrelease-LABEL-partition`*, where *`LABEL`* is specified in the output of `CLEinstall`.

Informational messages are displayed. Warning messages, related to creating a CentOS image on an SLES system, are also displayed and can safely be ignored. Finally, a message similar to the following is displayed after the provisioning completes successfully.

```
INFO - Provisioning of DAL image 'dal_cle_5.2up04_centos_6.5_x86-64_net' successful.
```

Create a Boot Image That Includes the DAL Image

A CentOS boot package must be prepared for the system with the specified *LABEL* and placed in the */bootimagedir* directory.

1. Execute the `shell_bootimage` script indicated in the output of `CLEinstall`.

   ```
   smw:~ # /var/opt/cray/install/shell_bootimage_LABEL.sh -d -c -b \
   /bootimagedir/bootimage.cpio
   ```

2. Copy the boot package on the SMW to the same directory on the boot node:

   ```
   smw:~ # scp -p /bootimagedir/bootimage.cpio root@boot:/bootrootdir/bootimagedir/
   bootimage.cpio
   ```

Boot DAL Service Nodes with CentOS Boot Image

IMPORTANT: The system must be up to complete this process; boot the system if necessary.

Currently, all service nodes are running SLES; those intended for DAL must be halted and then rebooted with CentOS.

Wait for a sufficient time for the service nodes to reboot before proceeding.

```
crayadm@smw:~> xtbootsys --partition pN --reboot -L \
dal_cle_5.2up04_centos_6.5_x86-64_net cnames_of_dal_nodes
```

Where *cnames_of_dal_nodes* is a comma-separated list and *net* is `ari` for Cray XC30 systems or `gem` for Cray XE/XK systems.

Set Up ssh Keys

Edit the `/root/.ssh/known_hosts` file to remove the old `ssh` keys. To reset the keys, run the `ssh.sh` script to set up the new CentOS nodes with proper authentication.

```
crayadm@smw:~> ssh root@boot-pN
boot:~ # /var/opt/cray/install/shell_ssh.sh
boot:~ # exit
```

Lustre Post Boot Configuration

For each file system created, perform the following steps after the system has been booted with the DAL nodes for the first time only.

Configure Lustre

Configuring DAL nodes is slightly different than on legacy internal Lustre server configurations because the control files are located in the config set on the SMW instead of on the boot node.

On the SMW, create and edit a file system definition file, *fs_name*.fs_defs, entering the appropriate node and device information.

```
crayadm@smw:~> su -
smw:~ # cd /opt/cray-xt-lustre-utils/default/etc
smw:~ # cp example.fs_defs fs_name.fs_defs
smw:~ # vi fs_name.fs_defs
```

Cray recommends updating only the following portion of the fs_defs file.

```
# Lustre server hosts to LNET NIDs mapping.
# Multiple lines are additive.
# Use multiple lines with the same nodes if you have several nids for the same
# nodes.
# Use pdsh hostlist expressions.
# i.e. prefix[a,k-l,...] where a,k,l, are integers with k < l, etc
# Each line should have a one-to-one mapping between the nodes and nids.
nid_map: nodes=nid000[27-29,31] nids=[27-29,31]@gni

# Device configuration. Components can be spread across multiple lines.
## node     specifies the primary device host
## dev      specifies the device path
## fo_node  specifies the backup device host
## fo_dev   specifies the backup device path. Only needed if different from
##          the primary device path
## jdev     specifies the external journal device (OST configuration only).
## index    Force a particular OST or MDT index. If this component is specified
##          for one OST or MDT it should be specified for all of them. By
##          default the index is zero based and is assigned based on the order
##          in which devices are defined in this file. i.e. the first 'ost:'
##          has index '0', the second has index '1', the first 'mdt:' has index
##          '0', the second has index '1', etc.
##
##          Note: A combined MGT/MDT target is not supported with multiple MDTs.
##          Note: A separate MGT and MDT can be co-located on a single server.

## MGT
## Management Target
mgt: node=nid00027
     dev=/dev/disk/by-id/scsi-360001ff020021101061ad79111170000

## MDT
## MetaData Target(s)
mdt: node=nid00027
     dev=/dev/disk/by-id/scsi-360001ff020021101061ad79111170100
     index=0
mdt: node=nid00029
```

```
     dev=/dev/disk/by-id/scsi-360001ff020021101061ad79111170200
     index=1

## OST
## Object Storage Target(s)
ost: node=nid00028
     dev=/dev/disk/by-id/scsi-360001ff020021101061ad79111170300
     index=0
ost: node=nid00031
     dev=/dev/disk/by-id/scsi-360001ff020021101061ad7a811170400
     index=1
```

CAUTION: Be sure to save a copy of the updated `fs_defs` file in another location because the `/opt/cray-xt-lustre-utils/default` link changes with CLE software updates.

Install File System Definition Files

The full path to the *fs_name*`.fs_defs` file must be provided for the `install` command.

```
smw:~ # lustre_control install -I /etc/opt/cray/share/pN/lustre fs_name.fs_defs
Performing 'install' from smw at Thu Oct 31 16:16:32 CDT 2013

Parsing file system definitions file: fs_name.fs_defs
Parsed file system definitions file: fs_name.fs_defs
The 'fs_name' file system definitions were successfully installed!
```

Format the Lustre File System

On the boot node, format the file system.

```
smw:~ # ssh root@boot
boot:~ # lustre_control reformat -f fs_name
Performing 'reformat' from boot-pN at Fri Feb 7 13:28:08 CST 2014

About to reformat all targets for the filesystem(s):
fs_name

Continue? (y|n|q)
y
Formatting filesystem fs_name
Operation successful
To complete the reformat operation start Lustre services in the order
MGS->MDT(s)->OST(s) by executing the following command:
lustre_control start -p -f fs_name
```

Start the Lustre File System

From the boot node, start the Lustre file system.

```
boot:~ # lustre_control start -p -f fs_name
Performing 'start' from boot-pN at Fri Nov 8 13:31:47 CST 2013
Starting filesystem(s):
```

```
fs_name
All targets mounted successfully
```

Add Lustre Mount Point for Service Nodes

For the client mount of Lustre to operate correctly, the Lustre entry must also be added to `/etc/fstab` in class login. The `noauto` option should be specified.

```
boot:~ # xtopview -c login
default/:/ # vi /etc/fstab
MGS-nid number@gni:/fs_name /lus/fs_name lustre rw,flock,user_xattr,noauto 0 0
```

Create File System Mount Point for Service Nodes

Also in the `xtopview` session, create the *`/lus/fs_name`* mount point.

```
default/:/ # mkdir -p /lus/fs_name
default/:/ # exit
```

Mount File System on the Login Node

Mount each file system on the login node.

```
boot:~ # ssh login
login:~ # mount /lus/fs_name
```

CAUTION: Do not continue until the mount command is successful.

Verify Write Access to File System

Verify write access to *`/lus/fs_name`* on the login node by checking the timestamp of `test.txt`.

```
login:~ # touch /lus/fs_name/test.txt
login:~ # ls -la /lus/fs_name
total 12
drwxr-xr-x 3 root root 4096 Nov 8 13:49 .
drwxr-xr-x 3 root root 4096 Nov 8 13:43 ..
drwxr-xr-x 3 root root 4096 Nov 8 13:31 .lustre
-rw-r--r-- 1 root root 0 Nov 8 13:49 test.txt
login:~ # exit
boot:~ # exit
```

Configure File System for Compute Nodes

The file system mount point must be created and added to `/etc/fstab`. A new boot image is built to include the file system.

1. Create the file system mount point.

```
smw:~ #  mkdir -p /opt/xt-images/templates/default-pN//lus/fs_name
```

2. Add mount point to `/etc/fstab`

```
smw:~ # vi /opt/xt-images/templates/default-pN/etc/fstab
MGS-nid number@gni:/fs_name /lus/fs_name lustre rw,flock,user_xattr 0 0
```

3. Create a boot image that includes the file system.

```
smw:~ # /var/opt/cray/install/shell_bootimage_LABEL.sh -d -c -b /bootimagedir/
bootimage.cpio
```

   Where *LABEL* is specified in the output of `CLEinstall`.

4. Log out as `root`.

```
smw:~ # exit
```

Reboot Compute Nodes

The compute nodes must be rebooted to recognize the mounted file system.

```
crayadm@smw:~> xtbootsys -reboot -L CNL0 cnames_of_compute_nodes
```

Where *cnames_of_compute_nodes* is a comma-separated list.

Configure File System for Boot and Shutdown

Edit `xt.lustre.config` to enable the `/etc/init.d/lustre` `startup` or `stop` script to start or stop the Lustre file system at boot or shutdown time, respectively. Add *fs_name* to the `FILESYSTEMS=` line in `xt.lustre.config`. For example:

```
crayadm@smw:~> ssh root@boot
boot:~ # vi /etc/opt/cray/lustre-utils/xt.lustre.config
FILESYSTEMS="lus0"
```

If you have more than one Lustre file system, include all configured file system names, separated by a space. For example:

```
FILESYSTEMS="lus0 lus1"
```

Configure the MySQL Database for LMT

If the Lustre Monitoring Tool (LMT) will be used, a MySQL server instance can be set up on the MGS node for storing real time and historical LMT data. This procedure must be executed on the MGS.

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

```
boot:~ # ssh nidMGS
```

Where *nidMGS* is the NID of the MGS.

2. Start the MySQL server daemon (if not already running).
```
root@mgs:~ # /sbin/service mysqld start
```

3. To improve the security of the MySQL server instance, run the `mysql_secure_installation` script, which allows for the creation of a password for the MySQL `root` user, configures remote access for the MySQL `root` user, removes anonymous users and a test database, and reloads privileges.

```
root@mgs:~ # mysql_secure_installation
```

The following are the prompts generated by the script and Cray's recommended responses.

```
Enter current password for root (enter for none): <Enter>

Set root password? [Y/n] Y
New password: Enter a secure password
Re-enter new password: Enter the secure password again

Remove anonymous users? [Y/n] Y

Disallow root login remotely? [Y/n] Y

Remove test database and access to it? [Y/n] Y

Reload privilege tables now? [Y/n] Y
```

4. Ensure `root`-only access to the LMT user configuration file `/usr/share/lmt/mkusers.sql`.

```
root@mgs:~ #  chmod 600 /usr/share/lmt/mkusers.sql
```

5. Edit the LMT user configuration file `/usr/share/lmt/mkusers.sql`.

```
root@mgs:~ # vi /usr/share/lmt/mkusers.sql
```

This file contains MySQL statements that create users named `lwatchclient` and `lwatchadmin`. It gives them privileges only on databases that start with `filesystem_`. Cray recommends making the following changes to `mkusers.sql`:

- Grant privileges only on filesystem_*fsname*.*, where *fsname* is the file system name. This will grant permissions only on the database for the file system being monitored.
- Ensure that both `lwatchadmin` and `lwatchclient` are password protected.

```
CREATE USER 'lwatchclient'@'localhost' IDENTIFIED BY 'passwd1';
GRANT SELECT ON filesystem_fsname.* TO 'lwatchclient'@'localhost';

CREATE USER 'lwatchadmin'@'localhost' IDENTIFIED BY 'passwd2';
GRANT SELECT,INSERT,DELETE ON filesystem_fsname.* TO 'lwatchadmin'@'localhost';
GRANT CREATE,DROP          ON filesystem_fsname.* TO 'lwatchadmin'@'localhost';

FLUSH PRIVILEGES;
```

6. Save the changes and execute the following command:

```
root@mgs:~ # mysql -u root -p < /usr/share/lmt/mkusers.sql
```

This prompts for the MySQL `root` user password, which was set when `mysql_secure_installation` was executed.

7. Ensure `root`-only access to the LMT configuration file `/etc/lmt/lmt.conf`.

```
root@mgs:~ #  chmod 600 /etc/lmt/lmt.conf
```

8. Edit the LMT configuration file to know about the password for `lwatchclient`.

```
root@mgs:~ # vi /etc/lmt/lmt.conf
```

Change:

```
lmt_db_ropasswd = nil
```

To:

```
lmt_db_ropasswd = "passwd1"
```

Where *passwd1* is the password for user `lwatchclient`.

9. Create the file `/etc/lmt/rwpasswd` and place the password for `lwatchadmin` in it. The configuration file reads the password for `lwatchadmin` from `/etc/lmt/rwpasswd`.

```
root@mgs:~ # touch /etc/lmt/rwpasswd
root@mgs:~ # chmod 600 /etc/lmt/rwpasswd
root@mgs:~ # vi /etc/lmt/rwpasswd
```

passwd2

10. Create the database for the file system to be monitored.

```
root@mgs:~ # lmtinit -a fsname
```

Where *fsname* is the name of the DAL file system. LMT data will be inserted into the LMT MySQL database the next time the Cerebro service is restarted on the MGS.

11. Restart Cerebro.

```
root@mgs:~ # service cerebrod restart
```

12. Verify that LMT is adding data to the MySQL database.

a. Initiate the LMT shell.

```
root@mgs:~ # lmtsh -f fsname
```

b. List tables.

```
fsname> t
```

c. List tables again after several seconds to verify that `Row Count` is increasing.

Configure the LMT GUI

LMT includes the `lmt-gui` package, which contains a GUI called `lwatch` and a command-line tool (`lstat`) to view live data. The configuration file `.lmtrc` must be set up prior to using either tool. This procedure must be executed on the MGS.

1. Edit the sample configuration file `/usr/share/doc/packages/lmt-gui/sample.lmtrc` to reflect the site specific LMT configuration, where *db_name* is set to the name of the MySQL database used by LMT, that is, filesystem_*fsname*.

```
# LMT Configuration File - place in $HOME/.lmtrc

filesys.1.name=<insert_fsname_here>
filesys.1.mountname=<insert_/path/to/mountpoint_here>
filesys.1.dbhost=<insert_db_host_ip_here>
filesys.1.dbport=<insert_db_port_here>
filesys.1.dbuser=<insert_db_client_username_here>
# Leave dbauth blank if the given client has no password
filesys.1.dbauth=<insert_db_client_password_here>
filesys.1.dbname=<insert_db_name_here>
```

2. Save the updated `.lmtrc` as `~/.lmtrc`.

   Both `lwatch` and `lstat` are now usable.

   To run the GUI from a remote node, the MySQL database must be configured to allow remote access for the read-only user, `lwatchclient`.

Configure the MySQL Database for Remote Access

To run the LMT GUI on a separate node from the LMT server, the MySQL server instance running on the LMT server must be configured to allow remote access for the LMT read-only user, `lwatchclient`. The following MySQL statements can be added to the LMT user configuration file prior to executing the statements in that file or they can be executed directly. In these examples, *passwd1* is the password for user `lwatchclient` and *fsname* is the name of the file system being monitored.

This example allows the user `lwatchclient` to connect from any hostname.

```
CREATE USER 'lwatchclient'@'%' IDENTIFIED BY 'passwd1';
GRANT SELECT ON filesystem_fsname.* TO 'lwatchclient'@'%';
```

This example allows the user `lwatchclient` to connect only from a specific IP address, *IP_addr*.

```
CREATE USER 'lwatchclient'@'IP_addr' IDENTIFIED BY 'passwd1';
GRANT SELECT ON filesystem_fsname.* TO 'lwatchclient'@'IP_addr';
```

Configure a Boot Automation File for DAL

Follow the steps in this section to configure and test the boot automation file for DAL. For more information about boot automation, see the `xtbootsys(8)` man page.

Create Script on Boot Node

NOTE: This step is only necessary for systems with InfiniBand storage connected to the DAL nodes.

Create and save the following script on the boot node in `/root/bin/local.dal-opensm`. This script is called by your boot automation file to ensure that DAL nodes running IB are discovering their LUNs upon boot and must be executable.

```
#!/bin/sh
#
# Local widget to work around opensm startup at boot time on
# dal nodes with Infiniband.
#
pdsh -w cnames_of_dal_nodes "service opensm restart"
```

Where *cnames_of_dal_nodes* is a comma-separated list.

```
boot:~ # vi /root/bin/local.dal-opensm
boot:~ # chmod 755 /root/bin/local.dal-opensm
```

Shutdown the System

From the SMW, use the site-specific procedures to shut down the system. For example, to shut down using an automation file, enter 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
```

Edit the Boot Automation File for DAL

1. Edit the boot automation file.

   ```
   crayadm@smw:~> vi /opt/cray/hss/default/etc/auto.xthostname
   ```

2. Add the following line to boot the DAL nodes after booting the boot and SDB nodes:

   ```
   lappend actions [list crms_boot_loadfile dal_cle_5.2up04_centos_6.5_x86-64_net \
   service cnames_of_your_DAL_nodes linux]
   ```

Where *cnames_of_your_DAL_nodes* is a comma-separated list.

3. If there are DAL nodes that have InfiniBand attached storage, add the following line to restart `opensm` on those nodes after booting the service nodes:

```
lappend actions { crms_exec_on_bootnode "root" "/root/bin/local.dal-opensm" }
```

4. If Network Address Translation (NAT) IP forwarding is configured for MDS nodes to route to external LDAP servers, add the following lines prior to starting Lustre:

```
lappend actions { crms_exec_via_bootnode "NAT_router_node" "root" \
"path_to_start-nat_script" }
lappend actions { crms_exec_via_bootnode "MDS_node" "root" "/sbin/route \
add default gw NAT_router_node_IP"
```

   For additional information on NAT IP forwarding, see *Managing System Software for the Cray Linux Environment* (S-2393).

5. Add the following line to start Lustre on the DAL nodes after booting the service nodes:

```
lappend actions { crms_exec_on_bootnode "root" \
"/opt/cray/lustre-utils/default/bin/lustre_control start -f fs_name" }
```

6. Add the following line to mount the DAL clients after starting Lustre on the DAL nodes:

```
lappend actions { crms_exec_on_bootnode "root" \
"/opt/cray/lustre-utils/default/bin/lustre_control mount_clients -f fs_name -w
login"}
```

Boot Using the Autoboot File

From this point forward, boot your system using the autoboot file. This ensures that the entire system boots normally, including DAL service nodes, and verifies that the Lustre file system is installed and working correctly.

1. On the SMW, set the boot image:

```
crayadm@smw:~> xtcli part_cfg update pN -i /bootimagedir/image_name.cpio
```

2. Boot the system:

```
crayadm@smw:~> xtbootsys --partition pN -a auto.xthostname
```

Verify Shutdown/Reboot Procedures (Optional)

Reboot your system and confirm that shutdown and boot procedures operate as expected.

Prepare to Update or Upgrade CLE Software

Refer to this chapter and the next two chapters 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 5.2 base release to CLE 5.2.UP04.

Upgrade

A software upgrade installation involves moving to the next release. For example, if your system is currently running CLE 5.1 for XC30, you can upgrade to release CLE 5.2.UP04.

Before Starting 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 5.2.UP04 Release Errata*, *Limitations for CLE 5.2.UP04*, *Cray Linux Environment (CLE) Software Release Overview* (S-2425), and *Cray Linux Environment (CLE) Software Release Overview Supplement* (S-2497).
- Verify that your System Management Workstation (SMW) is running Cray SMW Release 7.2.UP04 or later. You must install the SMW 7.2.UP04 release or later on your SMW before installing the CLE 5.2.UP04 release. If a specific SMW update package is required for your installation, that information is documented in the *README* file provided with the CLE 5.2.UP04 release. Type the following command to determine the HSS/SMW version:

```
crayadm@smw:~> cat /opt/cray/hss/default/etc/smw-release
7.2.UP04
```

Back Up the 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 36 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 backup or destination system set bootable by changing the appropriate boot node and service node entries in `/etc/fstab`. Doing a live backup (`xthotbackup -L`) can reduce the amount of time a CLE system is unavailable to the user community for the CLE backup and software upgrade process. For more information, see the `xthotbackup(8)` man page.

Back Up Current Software

Use the `xthotbackup` command to copy the disk partitions in one system set to a backup system set.

WARNING: If the source system set is booted, you should use the `xthotbackup -L` option. If not using the `xthotbackup -L` option, neither the source system set nor the destination system set should be used by a booted CLE system. 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 backup or destination system set. For example, if *BLUE* is the label for the source system set and *GREEN* is the label for the backup 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.

Back Up Current Software Using `xthotbackup -L`

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

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

2. Run the `xthotbackup` command to copy a booted system from the source system set to the backup or destination system set. For example, if `BLUE` is the label for the source system set and `GREEN` is the label for the backup system set, execute the following command:

```
smw:~ # xthotbackup -L -a -b BLUE GREEN
```

> **NOTE:** The `-L` option will connect to the boot node (or other nodes that mount the file systems in the source system set) to backup the file systems in the destination system set. 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.

The `xthotbackup` command 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.

You are now ready to begin installing the software release package.

Upgrade CLE Software

A Cray system must be running CLE 4.0 or higher in order to upgrade the CLE software by using the procedures in this chapter.

Before You Begin

All upgrades, updates, 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. For example, if your system is running on the `BLUE` system set, and the `GREEN` system set is a backup of `BLUE`, then you can perform a large amount of the CLE upgrade process on the `GREEN` system set while the system is booted, thus reducing the amount of system downtime during upgrades/updates. These instructions will inform you when dedicated time is required.
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 36 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 the Cray Linux Environment* (S-2393).

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.

Install CLE Release Software on the SMW

Three DVDs are provided to install the CLE 5.2 release on a Cray system. The first is labeled `Cray CLE 5.2.UP`*`nn`* `Software` and contains software specific to Cray systems. Optionally, you may have an ISO image called `xc-sles11sp3-`*`5.2.55d05`*`.iso`, where *`5.2.55`* indicates the CLE release build level, and *`d05`* indicates the installer version.

To upgrade to the CLE 5.2 release from CLE or CLE 4.2 requires a second DVD labeled `Cray-CLEbase11sp3-`*`yyyymmdd`* and contains the CLE 5.2 base operating system, which is based on SLES 11 SP3. The third DVD is

labeled `CentOS-6.5-x86_64-bin-DVD1.iso` and contains the CentOS 6.5 base operating system for CLE direct-attached Lustre (DAL) nodes.

Copy 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 5.2.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 `xc-sles11sp3-`*`5.2.55d05`*`.iso` is the path name to the ISO image file.

   ```
   smw:~# mount -o loop,ro xc-sles11sp3-5.2.55d05.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.5.2.55
   smw:~# cp -pr /media/cdrom/* /home/crayadm/install.5.2.55
   ```

4. Unmount the `Cray CLE 5.2.UP`*`nn`* `Software` media.

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

5. For upgrading from CLE 5.1 or CLE 4.2 to CLE 5.2, you must mount the SLES 11 SP3 base media. Insert the `Cray-CLEbase11sp3` 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-CLEbase11SP3-`*`yyyymmdd`*`.iso` is the path name to the ISO image file.

   ```
   smw:~# mount -o loop,ro Cray-CLEbase11SP3-yyyymmdd.iso /media/cdrom
   ```

6. For direct-attached Lustre implementations, also mount the `CentOS-6.5-x86_64-bin-DVD1.iso` image.

   ```
   smw:~# mkdir /media/Centosbase
   smw:~# mount -o loop,ro CentOS-6.5-x86_64-bin-DVD1.iso /media/Centosbase
   ```

Install CLE on the SMW

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

   ```
   smw:~# /home/crayadm/install.5.2.56/CRAYCLEinstall.sh \
   -m /home/crayadm/install.5.2.56 -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. 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.

Prepare 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 *About Installation Configuration Files* on page 21 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.5.2.14/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 prevent spending much time tracking down problems that could have been avoided.

During update and upgrade installations, the `/opt/cray/hss/default/etc/auto.xtshutdown` automated shut-down file is overwritten by the newer shut-down file that corresponds to the update/upgrade release. The old shut-down file will be saved as `/opt/cray/hss/default/etc/auto.xtshutdown.rpmsave`. If your site has made local changes to the autofile, you will need to review the changes and reapply them to the new file following the update or upgrade.

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.$$` and `hosts.postinstall.$$`.

Prepare 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.5.2.56/CLEinstall.conf \
    /home/crayadm/install.5.2.14/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.5.2.56/CLEinstall.conf \
/home/crayadm/install.5.2.56/CLEinstall.conf.save
smw:~# chmod 644 /home/crayadm/install.5.2.56/CLEinstall.conf
smw:~# vi /home/crayadm/install.5.2.56/CLEinstall.conf
```

   For a complete description of the contents of this file, see *About Installation Configuration Files* on page 21.

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

Run 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 *Maintain Node Class Settings and Hostname Aliases* on page 22.

If the update or upgrade you are applying modifies configuration information in the `alps.conf` file, your existing `alps.conf` parameters will be automatically merged into the new file and your original file will be saved (in the same directory) as `alps.conf.unmerged`. If you experience problems with ALPS immediately following an update or upgrade, you can replace `alps.conf` with `alps.conf.unmerged` and execute `/etc/init.d/alps restart` on the boot and SDB nodes to restore your original configuration.

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 ugo-x` command. If there are site-specific `cron` scripts in these directories, you will need to re-enable the execute permission on them after performing 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 that you are upgrading to, for example 5.2.55.

`--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 *Preparing the `CLEinstall.conf` configuration file*.

`--Basemedia=directory`

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

`--Centosmedia=directory`

Specify the directory where the CentOS software media has been mounted. The `--Centosmedia` option is required when installing or upgrading CLE with direct-attached Lustre (DAL). For example, the CentOS image mount point could be `/media/Centosbase`.

For a full description of the `CLEinstall` command options and arguments, see *Run the CLEinstall Program* on page 44 or the `CLEinstall(8)` man page.

Run `CLEinstall`

1. Invoke the `CLEinstall` program on the SMW. `CLEinstall` is located in the directory you created in *Copy the Software to the SMW* on page 128.

   ```
   smw:~# /home/crayadm/install.5.2.55/CLEinstall --upgrade \
   --label=system_set_label --XTrelease=5.2.55 \
   --configfile=/home/crayadm/install.5.2.55/CLEinstall.conf \
   --CLEmedia=/home/crayadm/install.5.2.55 \
   --Basemedia=/media/cdrom
   ```

   When DAL is enabled, the `CLEinstall` program requires the --Centosmedia option.

   ```
   smw:~# /home/crayadm/install.5.2.55/CLEinstall --upgrade \
   --label=system_set_label --XTrelease=5.2.55 \
   --configfile=/home/crayadm/install.5.2.55/CLEinstall.conf \
   --CLEmedia=/home/crayadm/install.5.2.55 \
   --Basemedia=/media/cdrom --Centosmedia=/media/Centosbase
   ```

2. Examine the initial messages directed to standard output. Log files are created in `/var/adm/cray/logs` and named by using a timestamp that indicates when the install script began executing. For example:

   ```
   08:57:48 Installation output will be captured in /var/adm/cray/logs/\
   CLEinstall.p3.20140911085748.CLE52-P3.stdout.log
   08:57:48 Installation errors (stderr) will be captured in /var/adm/cray/logs/\
   CLEinstall.p3.20140911085748.CLE52-P3.stderr.log
   08:57:48 Installation debugging messages will be captured in /var/adm/cray/logs/\
   CLEinstall.p3.20140911085748.CLE52-P3.debug.log
   ```

   The naming conventions of these logs are:

CLEinstall.p*#*.*YYYYMMDDhhmmss*.*$LABEL*.*logtype*.log

where

p*#* is the partition number specified in the CLEinstall.conf file.

YYYYMMDDhhmmss is the timestamp in year, month, day, hour, minute, and second format.

$LABEL is the system set label (in the example above, `CLE52-P3`).

logtype is `stdout` (standard output), `stderr` (standard error), or `debug`.

Also, log files are created in `/var/adm/cray/logs` each time `CLEinstall` calls `CRAYCLEinstall.sh`. For example:

```
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.01-B.log
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.02-B.log
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.03-B.log
.
.
.
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.17-S.log
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.18-S.log
CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.19-S.log
```

The naming conventions of these logs are:

CRAYCLEinstall.sh.p*#*.*YYYYMMDDhhmmss*.*$LABEL*.*sequence#*-*root*.*log*

where

p*#* is the partition number specified in the `CLEinstall.conf` file.

YYYYMMDDhhmmss is the timestamp in year, month, day, hour, minute, and second format. This is the same timestamp used for the log files of the `CLEinstall` program instance that called `CRAYCLEinstall.sh`.

$LABEL is the system set label.

sequence# is an increasing count that specifies each invocation of `CRAYCLEinstall.sh` by `CLEinstall`.

root is either `B` (bootroot) or `S` (sharedroot), specifying the root modified by the `CRAYCLEinstall.sh` call.

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 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, 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 adding service nodes, then you may proceed and CLEinstall
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 adding service nodes, 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. Monitor the debug output. Create another terminal window and invoke the `tail` command by using the path and timestamp displayed when `CLEinstall` was run.

```
smw~:# tail -f /var/adm/cray/logs/CLEinstall.p#.YYYYMMDDhhmmss.$LABEL.debug.log
```

5. Locate the following warning and prompt in the `CLEinstall` console window and type **`y`**.

```
*** Preparing to UPGRADE software on system set label system_set_label. Do you
wish to proceed? [n]
```

   The `CLEinstall` program now installs the release software. This command runs for 30 minutes or more for updates and 90 minutes for an upgrade, depending on your system configuration.

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

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

   On completion, the `CLEinstall` program generates a list of command hints 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 file names. The list of command hints is written to the `CLEinstall.command_hints.`*timestamp* file in the installer log directory.

   Complete the upgrade/update 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 a previous step that tailed the debug file. The log files created in `/var/adm/cray/logs` for `CLEinstall.P#.`*YYYYMMDDhhmmss*`.`*$LABEL*`.stdout.log` and `CLEinstall.P#.`*YYYYMMDDhhmmss*`.`*$LABEL*`.debug.log` also contain the suggested commands.

A CLE upgrade requires corrective boots of the system, which are contained in the command hints. The command hints contain commands for the following tasks:

1. Boot the boot node and SDB node (all upgrades).
2. Generate the new ECDSA SSH host key because of the upgrade to SLES 11 SP3 (all upgrades).
3. Update the SDB schema for upgrades from older CLE versions, which is described in *Upgrade the SDB Database Utilities with a CLE Update Package* on page 172 (some upgrades).
4. Shut down the boot and SDB nodes, and boot the full system (all upgrades).
5. Run `shell_ssh.sh` once the service nodes are booted up.

When upgrading DAL, the command hints contain instructions for building the DAL image, creating the IMPS config set, provisioning the DAL image, and further Lustre configuration tasks. The details for these DAL tasks are described in *Upgrade DAL on XE Systems* on page 146.

Create 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-LABEL-partition/nodetype`*, where *`machine`* is the Cray hostname, *`xtrelease`* is the build level for the CLE release, *`LABEL`* is the system set label used from `/etc/sysset.conf`, *`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 the upgraded system. However, some file content may have changed with the new release. Verify that site-specific modifications are compatible. For example, use existing copies of `/etc/hosts`, `/etc/passwd` and `/etc/modprobe.conf`, but if `/init` changed for the template, the site-modified version that is copied and used for CLE 5.2 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.

Prepare 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`.

NOTE: When upgrading a system using direct-attached Lustre (DAL), use the `-d` option. This option specifies that the CentOS DAL be included. For more information, see *Create a Boot Image That Includes the DAL Image* on page 96.

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_.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.

Enable Boot Node Failover

NOTE: Boot node failover is an optional CLE feature.

If boot-node failover has been configured for the first time, follow these steps. If boot-node failover has not been configured, skip this procedure.

To enable bootnode failover, you must set `bootnode_failover` parameters in the `CLEinstall.conf` file before you run the `CLEinstall` program. For more information, see *Configure Boot Node Failover* on page 28.

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 the 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, 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
   ```

   NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` 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 the 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
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` option.

Enable SDB Node Failover

NOTE: SDB node failover is an optional CLE feature.

If SDB node failover has been configured for the first time, follow these steps. If SDB node failover has not been configured, skip this procedure.

In addition to this procedure, refer to *Configure Boot Automation for SDB Node Failover* on page 90 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 *Configure SDB Node Failover* on page 29.

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
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` 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.
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` 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
```

Run 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:~# umount /media/Centosbase
   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.

Update the SDB Database Schema

Upgrading to CLE 5.2 may require that the Service Database (SDB) database schema be upgraded. The command hints indicate whether this is necessary. Refer to *Upgrade the SDB Database Utilities with a CLE Update Package* on page 172 before continuing.

Configure Optional Services

If you enabled an optional service you were not previously using in *Prepare the CLEinstall.conf Configuration File* on page 111, you may need to perform additional configuration steps. Follow the procedures in the appropriate optional section in *Install CLE on a New System* on page 40 or in *Managing System Software for the Cray Linux Environment* (S-2393).

If you configured an optional CLE feature or service during a previous installation or upgrade, no additional steps are required.

Boot and Test the System

IMPORTANT: If you configured optional services for the first time during this upgrade and deferred updating the boot image, update the boot image now by following *Prepare Compute and Service Node Boot Images* on page 134.

Your system is now upgraded. Boot the system using either `xtbootsys` interactive mode or a boot automation file.

Boot the System with Interactive `xtbootsys`

1. Boot the boot node, followed by the SDB node, and then all remaining service nodes, but do not boot the CNL compute nodes.

   ```
   crayadm@smw> xtbootsys
   ```

2. Update the SSH known host keys for `root@boot` by running this script on the boot node after all of the service nodes complete their boot.

   ```
   smw# ssh root@boot
   boot# /var/opt/cray/install/shell_ssh.sh
   ```

3. Boot the CNL compute nodes one of these ways.

   a. Use menu option 17 in `xtbootsys`.

   b. Execute this command.

   ```
   crayadm@smw> xtcli -s boot CNL0 -o compute s0
   ```

Boot the System with a Boot Automation File

1. Merge the old boot automation file with `/opt/cray/hss/default/etc/auto.generic.cnl` to create a new boot automation file, `/opt/cray/hss/default/etc/auto.mycray`.

2. Boot the CLE system using a boot automation file.

   ```
   crayadm@smw> xtbootsys -a auto.mycray
   ```

3. Update the SSH known host keys for `root@boot` by running this script on the boot node after all of the service nodes complete their boot.

   ```
   smw# ssh root@boot
   boot# /var/opt/cray/install/shell_ssh.sh
   ```

Configure MAMU Nodes

1. On the boot node, run this script, which ensures the keys are correct on the MAMU nodes (the `postproc` node class in this and previous examples).

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

2. Modify the `sshd_config` and `/etc/fstab` files for the new `postproc` class.

```
boot# xtopview -c postproc -m "setting up postproc nodes"
class/postproc:/# xtspec /etc/fstab
```

Add this line:

```
ufs:/ufs/home         /ufs/home     nfs      tcp,rw  0 0
```

```
class/postproc:/ # xtspec /etc/ssh/sshd_config
class/postproc:/ # vi /etc/ssh/sshd_config
```

Strip any `MatchUser` blocks from the bottom of the `sshd_config` file. Save and close the file.

3. Run these commands to restrict logins on the `postproc` nodes to only the `crayadm` administrative account and `root`, which is necessary to provide out of memory protection.

```
class/postproc:/ # xtspec /etc/ssh/sshd_config
class/postproc:/ # echo "AllowUsers root crayadm" >> /etc/ssh/sshd_config
class/postproc:/ # exit
```

Flash the nvBIOS for Kepler GPUs

A Cray XC30 system with NVIDIA® Tesla® SXM modules requires an update to the NVIDIA BIOS (nvBIOS) for the NVIDIA K20X and K40s graphics processing units (GPUs). The nvBIOS is unique for each SXM-1 Kepler™ SKU, based on the type of heat sink, as shown below.

GPU Type	Board SKU	Production Firmware Image Version
Kepler K20X (13 fin)	P2085 SKU 202	80.10.44.00.02
Kepler K20X (20 fin)	P2085 SKU 212	80.10.44.00.04
Kepler K20X (30 fin)	P2085 SKU 222	80.10.44.00.05
Kepler K40s (13 fin)	P2085 SKU 209	80.80.4B.00.03
Kepler K40s (20 fin)	P2085 SKU 219	80.80.4B.00.04
Kepler K40s (30 fin)	P2085 SKU 229	80.80.4B.00.05

The CLE software includes a script that automatically determines the SKU version and flashes the nvBIOS with the appropriate firmware.

> **TIP:** You can use the `cnselect` command to identify the number and location of the Kepler GPUs. This example shows a system with K20X GPUs on four nodes.
>
> ```
> login:~# cnselect -c "subtype.eq.'nVidia_Kepler'"
> 4
> login:~# cnselect -e "subtype.eq.'nVidia_Kepler'"
> 70-73
> ```

1. As root on the login node, set the allocation mode for all compute nodes to interactive.

```
login:~# xtprocadmin -km interactive
```

2. Change to the directory containing the NVIDIA scripts.

```
login:~# cd /opt/cray/cray-nvidia/default/bin
```

3. To flash the Kepler K20X GPUs, for example, choose one of the following options.

   - To update the entire system:

```
login:~# aprun -n `cnselect -c "subtype.eq.'nVidia_Kepler'"` \
-N 1 -L `cnselect -e "subtype.eq.'nVidia_Kepler'"` \
./nvFlashBySKU -b
```

   - To update a single node or set of nodes:

```
login:~# aprun -n PEs -N 1 -L node_list ./nvFlashBySKU -b
```

     NOTE: Replace *PEs* with the number of nodes; replace *node_list* with a comma-separated list of NIDs.

     For example, to flash four GPUs on nodes 70-73:

```
login:~# aprun -n 4 -N 1 -L 70,71,72,73 ./nvFlashBySKU -b
c0-0c0s1n0: Nid 70: Successful Cray Graphite K20X nvBIOS flash
c0-0c0s1n1: Nid 71: Successful Cray Graphite K20X nvBIOS flash
c0-0c0s1n2: Nid 72: Successful Cray Graphite K20X nvBIOS flash
c0-0c0s1n3: Nid 73: Successful Cray Graphite K20X nvBIOS flash
```

4. If there is a flash failure, `nvFlashBySKU` displays an error message with the failing node ID, as in this example:

```
c0-0c0s1n3: Nid 73: Failed Cray Graphite K20X nvBIOS flash
```

   Depending on the type of failure, `nvFlashBySKU` might display additional information, if available. No flashing is done on unsupported SKUs.

   If a GPU fails to flash, the SXM-1 card must be replaced.

5. After flashing is successful, use `xtbootsys` to reboot the nodes from the SMW. For example:

```
crayadm@smw:~> xtbootsys --reboot -L CNL0 -r "rebooting after nvBIOS update" \
c0-0c0s1n0,c0-0c0s1n1,c0-0c0s1n2,c0-0c0s1n3
```

   TIP: You can use `xtprocadmin` on the login node to determine each node name from the `cnselect` output, as in this example:

```
login:~# xtprocadmin -n `cnselect -e "subtype.eq.'nVidia_Kepler'"`
   NID    (HEX)    NODENAME     TYPE    STATUS        MODE
    70     0xf8  c0-0c0s1n0  compute        up       batch
    71     0xf9  c0-0c0s1n1  compute        up       batch
    72     0xfa  c0-0c0s1n2  compute        up       batch
    73     0xfb  c0-0c0s1n3  compute        up       batch
```

6. After the reboot is successful, log on to the login node as root and change to the directory containing the NVIDIA scripts.

```
login:~# cd /opt/cray/cray-nvidia/default/bin
```

   To verify that all GPUs are reporting the correct nvBIOS version (see the table above), choose one of the following options:

   - To display the nvBIOS versions for the entire system:

```
login:~# aprun -n `cnselect -c "subtype.eq.'nVidia_Kepler'"` \
-N 1 -L `cnselect -e "subtype.eq.'nVidia_Kepler'"` \
./xkcheck -n -c -f | grep Version
```

- To display the nvBIOS versions for a single node or set of nodes:

  ```
  login:~# aprun -n PEs -N 1 -L node_list \
  ./xkcheck -n -c -f | grep Version
  ```

  Replace *PEs* with the number of nodes; replace *node_list* with a comma-separated list of NIDs.

  For example:

  ```
  login:~# aprun -n 4 -N 1 -L 70,71,72,73 \
  ./xkcheck -n -c -f | grep Version
  4 nodes report VBIOS Version                  : 80.10.3D.00.05
  ```

7. Reset the compute nodes to the normal batch or interactive mode using the `xtprocadmin` command.

Test 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. The output for Cray XE and Cray XK systems follows.

   ```
   crayadm@login:~> xtnodestat
   Current Allocation Status at Fri May 21 07:11:48 2012

        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 output for Cray XC30 systems follows.

```
crayadm@login:~> xtnodestat
Current Allocation Status at Fri Feb 15 15:01:38 2013

     C0-0             C1-0             C2-0             C3-0
  n3 ---a-------a---- ---a------------ --aa-------a---- ----------------
  n2 ---a-------a---- ---a------------ --aa-------a---- ----------------
  n1 ---a-------a---- ---a------------ --Xa-------a---- ----------------
c2n0 ---a-------a---- ---a------------ --aa-------a---- ----------------
  n3 -----bX--a-a---- ---a-------a-a-- - a--------a---- ---a------------
  n2 -----ba--a-a---- ---a-------a-a-- -Sa--------a---- ---X------------
  n1 -----ba--a-a---- ---a-------a-a-- -Sa--------a---- ---a------------
c1n0 -----ba--a-a---- ---a-------X-a-- - a--------a---- ---a------------
  n3   -a------a-----   ---------a---- - -a------------  --a-------a----
  n2 SS-a------a----- SS---------a---- -S-a------------ S--X-------a----
  n1 SS-X------a----- SS---------a---- -S-a------------ S--a------Aa----
c0n0   -a------a-----   ---------a---- - -X------------  --a------Aa----
    s0123456789abcdef 0123456789abcdef 0123456789abcdef 0123456789abcdef


Legend:
   nonexistent node                       S  service node
;  free interactive compute node          -  free batch compute node
A  allocated (idle) compute or ccm node   ?  suspect compute node
W  waiting or non-running job             X  down compute node
Y  down or admindown service node         Z  admindown compute node

Available compute nodes:          0 interactive,        650 batch
```

The `xtprocadmin` command displays the current values of processor flags and node attributes. The output for Cray XE and Cray XK systems follows.

```
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 output for Cray XC30 systems follows.

```
crayadm@login:~> xtprocadmin
   NID    (HEX)    NODENAME     TYPE    STATUS        MODE
     1      0x1  c0-0c0s0n1  service        up       batch
     2      0x2  c0-0c0s0n2  service        up       batch
     5      0x5  c0-0c0s1n1  service        up       batch
     6      0x6  c0-0c0s1n2  service        up       batch
     8      0x8  c0-0c0s2n0  compute        up       batch
```

```
     9       0x9  c0-0c0s2n1  compute         up        batch
    10       0xa  c0-0c0s2n2  compute         up        batch
```

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    733    733    107     89    626      0

Total pending applications: 4
Pending Pid     User     w:d:N NID     Age Command  Why
      17278  crayadm 1848:1:24   5   0h53m  ./app1 Busy
      17340  crayadm 1848:1:24   5   0h53m  ./app1 Busy
      17469  crayadm 1848:1:24   5   0h52m  ./app1 Busy
      26155  crayadm 1848:1:24   5   0h12m  ./app2 Busy

Total placed applications: 2
   Apid ResId   User  PEs Nodes    Age State        Command
1631095   135 alan-1   64     4  0h31m   run            mcp
1631145   140  flynn  128     8  0h05m   run  TRON-JA307020
```

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.

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
5.2.UP04
```

If the preceding simple tests ran successfully, the system is operational. Cray recommends using the `xthotbackup` utility to create a backup of a newly updated or upgraded system. For more information, see the `xthotbackup(8)` man page.

Update CLE Software

To use the procedures in this chapter, a Cray system must be running CLE 5.2.UP00 to update the CLE software to a newer version of CLE 5.2.

Before You Begin

All upgrades, updates, 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. For example, if your system is running on the `BLUE` system set, and the `GREEN` system set is a backup of `BLUE`, then you can perform a large amount of the CLE upgrade process on the `GREEN` system set while the system is booted, thus reducing the amount of system downtime during upgrades/updates. These instructions will inform you when dedicated time is required.
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 36 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 the Cray Linux Environment* (S-2393).

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.

Install CLE Release Software on the SMW

Two DVDs are provided to install the CLE 5.2 release on a Cray system. The first is labeled `Cray CLE 5.2.UP`*`nn`* `Software` and contains software specific to Cray systems. Optionally, you may have an ISO image called `xc-sles11sp3-`*`5.2.55d05`*`.iso`, where *`5.2.55`* indicates the CLE release build level, and *`d05`* indicates the installer version.

The second DVD is labeled `CentOS-6.5-x86_64-bin-DVD1.iso` and contains the CentOS 6.5 base operating system for CLE direct-attached Lustre (DAL) nodes.

Copy the Software to the SMW

1. Optional: Log on to the SMW as root.

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

2. Optional: 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 5.2.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 xc-sles11sp3-*5.2.56b12*.iso is the path name to the ISO image file.

   ```
   smw:~# mount -o loop,ro xc-sles11sp3-5.2.56b12.iso /media/cdrom
   ```

3. Optional: 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.5.2.56
   smw:~# cp -pr /media/cdrom/* /home/crayadm/install.5.2.56
   ```

4. Optional: Unmount the Cray CLE 5.2.UP*nn* Software media.

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

5. Optional: For direct-attached Lustre implementations, also mount the `CentOS-6.5-x86_64-bin-DVD1.iso` image.

   ```
   smw:~# mkdir /media/Centosbase
   smw:~# mount -o loop,ro CentOS-6.5-x86_64-bin-DVD1.iso /media/Centosbase
   ```

Install CLE on the SMW

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

   ```
   smw:~# /home/crayadm/install.5.2.56/CRAYCLEinstall.sh \
   -m /home/crayadm/install.5.2.56 -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. 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.

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 *About Installation Configuration Files* on page 21 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.`*`5.2.14`*`/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 prevent spending much time tracking down problems that could have been avoided.

During update and upgrade installations, the `/opt/cray/hss/default/etc/auto.xtshutdown` automated shut-down file is overwritten by the newer shut-down file that corresponds to the update/upgrade release. The old shut-down file will be saved as `/opt/cray/hss/default/etc/auto.xtshutdown.rpmsave`. If your site has made local changes to the autofile, you will need to review the changes and reapply them to the new file following the update.

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.$$` and `hosts.postinstall.$$`.

Prepare 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.5.2.56/CLEinstall.conf \
 /home/crayadm/install.5.2.14/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.5.2.56/CLEinstall.conf \
/home/crayadm/install.5.2.56/CLEinstall.conf.save
smw:~# chmod 644 /home/crayadm/install.5.2.56/CLEinstall.conf
smw:~# vi /home/crayadm/install.5.2.56/CLEinstall.conf
```

For a complete description of the contents of this file, see *About Installation Configuration Files* on page 21.

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

Run 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 *Maintain Node Class Settings and Hostname Aliases* on page 22.

If the update or upgrade you are applying modifies configuration information in the `alps.conf` file, your existing `alps.conf` parameters will be automatically merged into the new file and your original file will be saved (in the same directory) as `alps.conf.unmerged`. If you experience problems with ALPS immediately following an update or upgrade, you can replace `alps.conf` with `alps.conf.unmerged` and execute `/etc/init.d/alps restart` on the boot and SDB nodes to restore your original configuration.

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 ugo-x` command. If there are site-specific `cron` scripts in these directories, you will need to re-enable the execute permission on them after performing 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 that you are upgrading to, for example 5.2.55.

`--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 *Preparing the `CLEinstall.conf` configuration file*.

`--Centosmedia=directory`

Specify the directory where the CentOS software media has been mounted. The `--Centosmedia` option is required when installing or upgrading CLE with direct-attached Lustre (DAL). For example, the CentOS image mount point could be `/media/Centosbase`.

For a full description of the `CLEinstall` command options and arguments, see *Run the CLEinstall Program* on page 44 or the `CLEinstall(8)` man page.

Run CLEinstall

1. Invoke the `CLEinstall` program on the SMW. `CLEinstall` is located in the directory you created in *Copy the Software to the SMW* on page 128.

   ```
   smw:~# /home/crayadm/install.5.2.56/CLEinstall --upgrade \
   --label=system_set_label --XTrelease=5.2.56 \
   --configfile=/home/crayadm/install.5.2.56/CLEinstall.conf \
   --CLEmedia=/home/crayadm/install.5.2.56 \
   ```

   Also include the `--Centosmedia=`*directory* option when invoking `CLEinstall`. In this example, the option is **`--Centosmedia=/media/Centosbase`**.

2. Examine the initial messages directed to standard output. Log files are created in `/var/adm/cray/logs` and named by using a timestamp that indicates when the install script began executing. For example:

   ```
   08:57:48 Installation output will be captured in /var/adm/cray/logs/\
   CLEinstall.p3.20140911085748.CLE52-P3.stdout.log
   08:57:48 Installation errors (stderr) will be captured in /var/adm/cray/logs/\
   CLEinstall.p3.20140911085748.CLE52-P3.stderr.log
   08:57:48 Installation debugging messages will be captured in /var/adm/cray/logs/\
   CLEinstall.p3.20140911085748.CLE52-P3.debug.log
   ```

   The naming conventions of these logs are:

   CLEinstall.p*#*.*YYYYMMDDhhmmss*.*$LABEL*.*logtype*.log

   where

   p*#* is the partition number specified in the CLEinstall.conf file.

   YYYYMMDDhhmmss is the timestamp in year, month, day, hour, minute, and second format.

   $LABEL is the system set label (in the example above, `CLE52-P3`).

   logtype is `stdout` (standard output), `stderr` (standard error), or `debug`.

   Also, log files are created in `/var/adm/cray/logs` each time `CLEinstall` calls `CRAYCLEinstall.sh`. For example:

   ```
   CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.01-B.log
   CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.02-B.log
   CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.03-B.log
   .
   .
   .
   CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.17-S.log
   CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.18-S.log
   CRAYCLEinstall.sh.p3.20140911085748.CLE52-P3.19-S.log
   ```

   The naming conventions of these logs are:

CRAYCLEinstall.sh.p*#.YYYYMMDDhhmmss.$LABEL.sequence#-root.log*

where

p*#* is the partition number specified in the `CLEinstall.conf` file.

YYYYMMDDhhmmss is the timestamp in year, month, day, hour, minute, and second format. This is the same timestamp used for the log files of the `CLEinstall` program instance that called `CRAYCLEinstall.sh`.

$LABEL is the system set label.

sequence# is an increasing count that specifies each invocation of `CRAYCLEinstall.sh` by `CLEinstall`.

root is either `B` (bootroot) or `S` (sharedroot), specifying the root modified by the `CRAYCLEinstall.sh` call.

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 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, 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 adding service nodes, then you may proceed and CLEinstall
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 adding service nodes, 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. Monitor the debug output. Create another terminal window and invoke the `tail` command by using the path and timestamp displayed when `CLEinstall` was run.

   ```
   smw~:# tail -f /var/adm/cray/logs/CLEinstall.p#.YYYYMMDDhhmmss.$LABEL.debug.log
   ```

5. Locate the following warning and prompt in the `CLEinstall` console window and type **`y`**.

   ```
   *** Preparing to UPGRADE software on system set label system_set_label. Do you
   wish to proceed? [n]
   ```

   The `CLEinstall` program now installs the release software. This command runs for 30 minutes or more for updates and 90 minutes for an upgrade, depending on your system configuration.

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

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

   On completion, the `CLEinstall` program generates a list of command hints 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 file names. The list of command hints is written to the `CLEinstall.command_hints.`*`timestamp`* file in the installer log directory.

   Complete the upgrade/update 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 a previous step that tailed the debug file. The log files created in `/var/adm/cray/logs` for `CLEinstall.P#.`*`YYYYMMDDhhmmss`*`.`*`$LABEL`*`.stdout.log` and `CLEinstall.P#.`*`YYYYMMDDhhmmss`*`.`*`$LABEL`*`.debug.log` also contain the suggested commands.

Create 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-LABEL-partition/nodetype`*, where *`machine`* is the Cray hostname, *`xtrelease`* is the build level for the CLE release, *`LABEL`* is the system set label used from `/etc/sysset.conf`, *`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 the upgraded system. However, some file content may have changed with the new release. Verify that site-specific modifications are compatible. For example, use existing copies of `/etc/hosts`, `/etc/passwd` and `/etc/modprobe.conf`, but if `/init` changed for the template, the site-modified version that is copied and used for CLE 5.2 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.

Prepare 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`.

> **NOTE:** When upgrading a system using direct-attached Lustre (DAL), use the `-d` option. This option specifies that the CentOS DAL be included.

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_.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.

Enable Boot Node Failover

> **NOTE:** Boot node failover is an optional CLE feature.

If boot-node failover has been configured for the first time, follow these steps. If boot-node failover has not been configured, skip this procedure.

To enable bootnode failover, you must set `bootnode_failover` parameters in the `CLEinstall.conf` file before you run the `CLEinstall` program. For more information, see *Configure Boot Node Failover* on page 28.

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 the 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, 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
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` 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 the 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
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` option.

Enable SDB Node Failover

NOTE: SDB node failover is an optional CLE feature.

If SDB node failover has been configured for the first time, follow these steps. If SDB node failover has not been configured, skip this procedure.

In addition to this procedure, refer to *Configure Boot Automation for SDB Node Failover* on page 90 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 *Configure SDB Node Failover* on page 29.

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
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` 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.
```

NOTE: If the system is partitioned, invoke `xtdaemonconfig` with the `--partition pn` 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
```

Update Direct-Attached Lustre

NOTE: The direct-attached Lustre (DAL) file system is optional; your storage RAID may be external to the mainframe.

Installation and configuration of DAL differs from that of external Lustre. Cray service nodes that support DAL use a CentOS operating system running on ramdisk as opposed to the shared root file system.

1. Build the DAL image root.

```
smw:~ # impscli build image_recipe dal_cle_5.2up04_centos_6.5_x86-64_ari
```

2. Update the config set. The IMPS Configurator updates the config set by interactively guiding the administrator through the process of providing needed configuration values. As the Configurator prompts for information about the system, a description and guidance, including a reasonable default value, are provided for each query.

```
smw:~ # impscli update config_set pN with images \
dal_cle_5.2up04_centos_6.5_x86-64_ari
```

Where p*N* is a valid partition name.

3. Provision the DAL image.

```
smw:~ # impscli provisiondal image dal_cle_5.2up04_centos_6.5_x86-64_ari to \
/opt/xt-images/machine-xtrelease-LABEL-partition
```

Where *LABEL* is the same as in *Prepare Compute and Service Node Boot Images* on page 134.

Informational messages are displayed. Warning messages, related to creating a CentOS image on an SLES system, are also displayed and can safely be ignored. Finally, a message similar to the following is displayed after the provisioning completes successfully.

```
INFO - Provisioning of DAL image 'dal_cle_5.2up04_centos_6.5_x86-64_ari'
successful.
```

4. Recreate the boot image to include DAL.

```
smw:~ # /var/opt/cray/install/shell_bootimage_LABEL.sh -d -c -b /bootimagedir/
bootimage.cpio
```

Where *LABEL* is the same as in *Prepare Compute and Service Node Boot Images* on page 134.

5. Copy the boot package on the SMW to the same directory on the boot node.

```
smw:~ # cp -p /bootimagedir/bootimage.cpio /bootrootdir/bootimagedir/bootimage.cpio
```

6. Remember to update the version level of the boot image for the DAL service nodes when editing the boot automation file `/opt/cray/hss/default/etc/auto.`*xthostname* during *Boot and Test the System* on page 138.

```
lappend actions [list crms_boot_loadfile \
dal_cle_5.2up04_centos_6.5_x86-64_ari service \
cnames_of_your_DAL_nodes linux]
```

Where *cnames_of_your_DAL_nodes* is a comma-separated list.

7. Unmount the CentOS media.

```
smw:~ # umount /media/Centosbase
```

Run Post-`CLEinstall` Commands

1. 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.

2. 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.

Configure Optional Services

If you enabled an optional service you were not previously using in *Prepare the CLEinstall.conf Configuration File* on page 111, you may need to perform additional configuration steps. Follow the procedures in the appropriate optional section in *Install CLE on a New System* on page 40 or in *Managing System Software for the Cray Linux Environment* (S-2393).

If you configured an optional CLE feature or service during a previous installation or upgrade, no additional steps are required.

Configure MAMU Nodes

1. On the boot node, run this script, which ensures the keys are correct on the MAMU nodes (the `postproc` node class in this and previous examples).

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

2. Modify the `sshd_config` and `/etc/fstab` files for the new `postproc` class.

   ```
   boot# xtopview -c postproc -m "setting up postproc nodes"
   class/postproc:/# xtspec /etc/fstab
   ```

   Add this line:

   ```
   ufs:/ufs/home         /ufs/home     nfs      tcp,rw  0 0
   ```

   ```
   class/postproc:/ # xtspec /etc/ssh/sshd_config
   class/postproc:/ # vi /etc/ssh/sshd_config
   ```

   Strip any `MatchUser` blocks from the bottom of the `sshd_config` file. Save and close the file.

3. Run these commands to restrict logins on the `postproc` nodes to only the `crayadm` administrative account and `root`, which is necessary to provide out of memory protection.

   ```
   class/postproc:/ # xtspec /etc/ssh/sshd_config
   class/postproc:/ # echo "AllowUsers root crayadm" >> /etc/ssh/sshd_config
   class/postproc:/ # exit
   ```

Boot and Test 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 *Prepare Compute and Service Node Boot Images* on page 134.

Your system is now upgraded.

Reboot the Cray System

1. Use 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/hss/default/etc/auto.xthostname
   ```

3. Use the `xtbootsys` command to boot the Cray system.

   CAUTION: Shut down your Cray system before invoking the `xtbootsys` command. If installing to an alternate system set, shut down the currently running system before booting 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
   ```

Flash the nvBIOS for Kepler GPUs

A Cray XC30 system with NVIDIA® Tesla® SXM modules requires an update to the NVIDIA BIOS (nvBIOS) for the NVIDIA K20X and K40s graphics processing units (GPUs). The nvBIOS is unique for each SXM-1 Kepler™ SKU, based on the type of heat sink, as shown below.

GPU Type	Board SKU	Production Firmware Image Version
Kepler K20X (13 fin)	P2085 SKU 202	80.10.44.00.02
Kepler K20X (20 fin)	P2085 SKU 212	80.10.44.00.04
Kepler K20X (30 fin)	P2085 SKU 222	80.10.44.00.05
Kepler K40s (13 fin)	P2085 SKU 209	80.80.4B.00.03
Kepler K40s (20 fin)	P2085 SKU 219	80.80.4B.00.04
Kepler K40s (30 fin)	P2085 SKU 229	80.80.4B.00.05

The CLE software includes a script that automatically determines the SKU version and flashes the nvBIOS with the appropriate firmware.

> **TIP:** You can use the `cnselect` command to identify the number and location of the Kepler GPUs. This example shows a system with K20X GPUs on four nodes.
>
> ```
> login:~# cnselect -c "subtype.eq.'nVidia_Kepler'"
> 4
> login:~# cnselect -e "subtype.eq.'nVidia_Kepler'"
> 70-73
> ```

1. As root on the login node, set the allocation mode for all compute nodes to interactive.

   ```
   login:~# xtprocadmin -km interactive
   ```

2. Change to the directory containing the NVIDIA scripts.

   ```
   login:~# cd /opt/cray/cray-nvidia/default/bin
   ```

3. To flash the Kepler K20X GPUs, for example, choose one of the following options.

   - To update the entire system:

     ```
     login:~# aprun -n `cnselect -c "subtype.eq.'nVidia_Kepler'"` \
     -N 1 -L `cnselect -e "subtype.eq.'nVidia_Kepler'"` \
     ./nvFlashBySKU -b
     ```

   - To update a single node or set of nodes:

     ```
     login:~# aprun -n PEs -N 1 -L node_list ./nvFlashBySKU -b
     ```

     > **NOTE:** Replace *PEs* with the number of nodes; replace *node_list* with a comma-separated list of NIDs.
     >
     > For example, to flash four GPUs on nodes 70-73:
     >
     > ```
     > login:~# aprun -n 4 -N 1 -L 70,71,72,73 ./nvFlashBySKU -b
     > c0-0c0s1n0: Nid 70: Successful Cray Graphite K20X nvBIOS flash
     > c0-0c0s1n1: Nid 71: Successful Cray Graphite K20X nvBIOS flash
     > c0-0c0s1n2: Nid 72: Successful Cray Graphite K20X nvBIOS flash
     > c0-0c0s1n3: Nid 73: Successful Cray Graphite K20X nvBIOS flash
     > ```

4. If there is a flash failure, `nvFlashBySKU` displays an error message with the failing node ID, as in this example:

   ```
   c0-0c0s1n3: Nid 73: Failed Cray Graphite K20X nvBIOS flash
   ```

   Depending on the type of failure, `nvFlashBySKU` might display additional information, if available. No flashing is done on unsupported SKUs.

   If a GPU fails to flash, the SXM-1 card must be replaced.

5. After flashing is successful, use `xtbootsys` to reboot the nodes from the SMW. For example:

   ```
   crayadm@smw:~> xtbootsys --reboot -L CNL0 -r "rebooting after nvBIOS update" \
   c0-0c0s1n0,c0-0c0s1n1,c0-0c0s1n2,c0-0c0s1n3
   ```

   > **TIP:** You can use `xtprocadmin` on the login node to determine each node name from the `cnselect` output, as in this example:

```
login:~# xtprocadmin -n `cnselect -e "subtype.eq.'nVidia_Kepler'"`
   NID    (HEX)    NODENAME     TYPE    STATUS        MODE
    70     0xf8  c0-0c0s1n0  compute        up       batch
    71     0xf9  c0-0c0s1n1  compute        up       batch
    72     0xfa  c0-0c0s1n2  compute        up       batch
    73     0xfb  c0-0c0s1n3  compute        up       batch
```

6. After the reboot is successful, log on to the login node as root and change to the directory containing the NVIDIA scripts.

   ```
   login:~# cd /opt/cray/cray-nvidia/default/bin
   ```

   To verify that all GPUs are reporting the correct nvBIOS version (see the table above), choose one of the following options:

   - To display the nvBIOS versions for the entire system:

     ```
     login:~# aprun -n `cnselect -c "subtype.eq.'nVidia_Kepler'"` \
     -N 1 -L `cnselect -e "subtype.eq.'nVidia_Kepler'"` \
     ./xkcheck -n -c -f | grep Version
     ```

   - To display the nvBIOS versions for a single node or set of nodes:

     ```
     login:~# aprun -n PEs -N 1 -L node_list \
     ./xkcheck -n -c -f | grep Version
     ```

     Replace *PEs* with the number of nodes; replace *node_list* with a comma-separated list of NIDs.

     For example:

     ```
     login:~# aprun -n 4 -N 1 -L 70,71,72,73 \
     ./xkcheck -n -c -f | grep Version
     4 nodes report VBIOS Version                  : 80.10.3D.00.05
     ```

7. Reset the compute nodes to the normal batch or interactive mode using the `xtprocadmin` command.

Test 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. The output for Cray XE and Cray XK systems follows.

   ```
   crayadm@login:~> xtnodestat
   Current Allocation Status at Fri May 21 07:11:48 2012
   ```

```
     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 output for Cray XC30 systems follows.

```
crayadm@login:~> xtnodestat
Current Allocation Status at Fri Feb 15 15:01:38 2013

     C0-0             C1-0             C2-0             C3-0
  n3 ---a-------a---- ---a------------ --aa-------a---- ----------------
  n2 ---a-------a---- ---a------------ --aa-------a---- ----------------
  n1 ---a-------a---- ---a------------ --Xa-------a---- ----------------
c2n0 ---a-------a---- ---a------------ --aa-------a---- ----------------
  n3 -----bX--a-a---- ---a-------a-a-- - a--------a---- ---a------------
  n2 -----ba--a-a---- ---a-------a-a-- -Sa--------a---- ---X------------
  n1 -----ba--a-a---- ---a-------a-a-- -Sa--------a---- ---a------------
c1n0 -----ba--a-a---- ---a-------X-a-- - a--------a---- ---a------------
  n3   -a------a-----   ---------a---- - -a------------  --a-------a----
  n2 SS-a------a----- SS---------a---- -S-a------------ S--X-------a----
  n1 SS-X------a----- SS---------a---- -S-a------------ S--a------Aa----
c0n0   -a------a-----   ---------a---- - -X------------  --a------Aa----
    s0123456789abcdef 0123456789abcdef 0123456789abcdef 0123456789abcdef


Legend:
   nonexistent node                       S  service node
;  free interactive compute node          -  free batch compute node
A  allocated (idle) compute or ccm node   ?  suspect compute node
W  waiting or non-running job             X  down compute node
Y  down or admindown service node         Z  admindown compute node

Available compute nodes:          0 interactive,        650 batch
```

The `xtprocadmin` command displays the current values of processor flags and node attributes. The output for Cray XE and Cray XK systems follows.

```
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 output for Cray XC30 systems follows.

```
crayadm@login:~> xtprocadmin
   NID    (HEX)    NODENAME     TYPE    STATUS        MODE
     1       0x1  c0-0c0s0n1  service         up       batch
     2       0x2  c0-0c0s0n2  service         up       batch
     5       0x5  c0-0c0s1n1  service         up       batch
     6       0x6  c0-0c0s1n2  service         up       batch
     8       0x8  c0-0c0s2n0  compute         up       batch
     9       0x9  c0-0c0s2n1  compute         up       batch
    10       0xa  c0-0c0s2n2  compute         up       batch
```

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    733    733    107     89    626      0

Total pending applications: 4
Pending Pid     User     w:d:N NID     Age Command  Why
      17278  crayadm 1848:1:24   5   0h53m  ./app1 Busy
      17340  crayadm 1848:1:24   5   0h53m  ./app1 Busy
      17469  crayadm 1848:1:24   5   0h52m  ./app1 Busy
      26155  crayadm 1848:1:24   5   0h12m  ./app2 Busy

Total placed applications: 2
   Apid ResId   User  PEs Nodes    Age State        Command
1631095   135 alan-1   64     4  0h31m   run            mcp
1631145   140  flynn  128     8  0h05m   run  TRON-JA307020
```

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.

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
   5.2.UP04
   ```

   If the preceding simple tests ran successfully, the system is operational. Cray recommends using the `xthotbackup` utility to create a backup of a newly updated or upgraded system. For more information, see the `xthotbackup(8)` man page.

Upgrade DAL on XE Systems

This section contains the information and procedures to upgrade a direct-attached Lustre (DAL) file system 1.8.x on a Cray XE system running CLE 4.2 to CLE 5.2 with Lustre 2.5. Because the Lustre community no longer supports Lustre on SLES-based servers, Cray service nodes that support DAL must now use a CentOS™ operating system running on ramdisk, as opposed to the shared root file system.

Installation and configuration of DAL is facilitated by Cray's Image Management and Provisioning System (IMPS), a new set of features that changes how software is installed, managed, provisioned, booted and configured. DAL is currently the only Cray product installed using IMPS; however, in future releases, IMPS will support the installation of other Cray products.

For an introduction to IMPS concepts and the commands used in this procedure, see *IMPS Guide for DAL Installation* (S-0049).

Lustre 2.5

Lustre 2.5 includes significant advances in Lustre design that were introduced in Lustre 2.4 plus additional new features and improvements. To accommodate these advancements, the on-disk file system format is slightly different than the 1.8.x format. These differences are limited to Lustre's internal file system metadata and fall into two categories: those related to file identifiers (FIDs) and those related to quota support. The format and storage of user data are not affected by these differences.

Lustre 2.5 provides tools to add the new metadata structures to an existing 1.8.x file system. Some of these tools run automatically when the 2.5 servers are started; some require the administrator to perform explicit upgrade operations at the time 2.5 is installed.

For further information regarding Lustre 2.x, see the Lustre Operations Manual available at *https://wiki.hpdd.intel.com/display/PUB/Documentation*.

File Identifiers

A file identifier (FID) is a unique identifier for a Lustre file or object. It is independent of the back end file system, for example, ldiskfs. In Lustre 1.8.x, inodes are used to uniquely identify the objects belonging to a file. In Lustre 2.5, FIDs replace inodes for this purpose. The drawback of using inodes is that a file's inode can change over the life of the file. For example, when a file is restored from a file level backup, it will be assigned a new inode/generation number. Afterwards, Lustre 1.8.x can no longer locate its objects. FIDs, on the other hand, never change once assigned. Following restoration from a file level backup, the FIDs are still correct and Lustre 2.5 can locate its objects. In addition to supporting the device level backup of 1.8.x, Lustre 2.5 also supports file level backup and restore. Lustre 2.5 still uses inodes internally to interact with the `ldiskfs` back end file system.

To facilitate this interaction, Lustre 2.5 maintains a mapping of FIDs to inodes called the Object Index (OI). When upgrading from 1.8.x to 2.5 the OI is created through a process called OI Scrub. The OI Scrub occurs automatically when a 1.8.x file system without an OI is mounted by 2.5 servers. An OI Scrub may also be

triggered if Lustre discovers a missing or bad FID to inode mapping during normal operation. Finally, an OI Scrub is started manually by running `lfsck`.

`lfsck` checks and repairs errors in the OI. The OI maintains the FID to inode mapping, and the inode contains the inode to FID mapping. The FID of the object identified by the inode is stored in the extended attributes area of the inode known as *linkEA*. The inode also contains the FID of the file's parent directory. This feature is known as *FID-in-dirent*. The linkEA and FID-in-dirent information enables Lustre to efficiently generate full path names from the inode. These names are used in POSIX-style path name permission checks to produce better error messages and to support changelog applications like `lustre_rsync`.

Storing the parent FID in the inode also improves the performance of `readdir` and other directory operations. To populate the inodes with the appropriate FIDs, the Lustre administrator must set the `dirdata` attribute on each metadata target (MDT) and then run `lfsck` with the `-t namespace` option. The `lfsck` process runs in the background; the rate at which it updates inodes is a tunable parameter under the control of the system administrator, as described in the Lustre Operations Manual, available at *https://wiki.hpdd.intel.com/display/PUB/Documentation*.

> **NOTE:** After `dirdata` has been enabled in LFSCK: Add FIDs to inode Attributes, fallback is not possible, meaning, the Lustre file system cannot be downgraded to work with 1.8.x servers.

Quota Support

Lustre 2.5 includes improvements that address several limitations of the 1.8.x quota design:

- Quota limits can be changed while slaves are offline
- Object Storage Targets (OSTs) can be added and deleted without corrupting space usage statistics
- Master recovery can be completed without all targets being online
- A full `quotacheck` is no longer required following `e2fsck`
- Quota enforcement is enabled/disabled by file system rather than per-target
- Infrastructure is restructured for future growth, better performance, and improved functionality

To support these improvements, both the on-disk format of quota information and the user interface to quota functionality have changed.

Quota specification has three components: space usage accounting, quota limit definition, and enabling enforcement.

Space Usage Accounting

In previous versions of Lustre, the `lfs quotacheck` had to be run to generate the database of space used on each target by each user and group. In Lustre 2.5, the `quotacheck` command is deprecated. Instead, newly formatted 2.5 file systems have space usage accounting enabled by default and the statistics are automatically kept up-to-date. When upgrading from 1.8.x to 2.5, the usage statistics must be initialized for existing files before quota limits can be enforced. Initializing usage statistics is a one-time operation, which is done on Cray systems with the `lustre_control` command as described in Enable Quotas.

Quota Limit Definition

Although the definition of user and group quota limits does not change with 2.5, the storage format of these limits does change. With Lustre 1.8.x, the quota limits were stored in a file specific to the back end file system. In 2.5, this information is stored in a Lustre-defined index along with other Lustre metadata. The quota limits are converted automatically to the new format and storage location when the MDT is upgraded to 2.5.

Enabling Quota Enforcement

In Lustre 2.5, quota enforcement is independent of space usage accounting. The accounting information is always maintained, even when enforcement is disabled. Enforcement is enabled/disabled for an entire file system through the `lctl` command. The `lfs quota on | off` command and per-target `quota_type` parameter are no longer used in Lustre 2.5.

To enable enforcement of inode quotas for *`fs_name`* (must be done on MGS):

```
mgs:~ # lctl conf_param fs_name.quotamdt=value
```

Where *`value`* determines for whom quotas are enforced; set to: `u` for users, `g` for groups, `ug` for users and groups, or `none`.

To enable enforcement of block quotas for *`fs_name`* (must be done on MGS):

```
mgs:~ # lctl conf_param fs_name.quota.ost=value
```

Where *`value`* determines for whom quotas are enforced; set to: `u` for users, `g` for groups, `ug` for users and groups, or `none`.

For further information on `lctl` and enforcing quotas, see the Lustre Operations Manual at *https://wiki.hpdd.intel.com/display/PUB/Documentation*.

Performance Expectations

For optimum long term performance and functionality, all the disk format changes described above are recommended; however, each of the upgrade processes has a performance cost.

Object Index (OI) Creation and Repair

OI Scrub is designed to have minimal impact on system performance. It runs in the background while the file system remains online and clients continue to access files. The system administrator can control the overhead of the scrubbing process by tuning the maximum number of objects examined per second. If no limit is set, OI Scrub will run as fast as possible. On an unloaded system, with no limit set, experiments have shown OI Scrub to process in excess of 100,000 objects per second. For further information on OI Scrub, see the Lustre Operations Manual, available at *https://wiki.hpdd.intel.com/display/PUB/Documentation*.

OI Scrub status is monitored through the `/proc` file: `/fs/lustre/osd-ldiskfs/`*`mdt_device`*`/oi_scrub`

Adding FIDs to inodes

Updating all inode extended attributes to include the related FID and parent FID information is a one-time operation, although updates to individual inodes may occur when `lfsck` repairs file system corruption. The process is similar to an explicitly invoked OI Scrub and has similar performance characteristics.

Progress of the inode update is monitored through the `/proc` file: `fs/lustre/mdd/`*`mdt_device`*`/fsck_namespace`

Space Usage Statistics

When upgrading a 1.8.x formatted file system, a database of the space usage statistics must be created. The usage data is gathered and stored when the quota flag is set on each target. The speed of the data collection in 2.5 is similar to the speed of an `lfs quotacheck` in earlier Lustre versions.

After the initial creation, Lustre updates the space usage statistics automatically as files change. Updating the statistics imposes some overhead and has been reported to affect metadata performance by as much as 5%. Cray internal testing has shown that the accounting overhead has no measurable effect on performance.

Before Starting the DAL Upgrade

Prior to starting the DAL upgrade process, complete the upgrade procedure for Cray Linux Environment (CLE) base operating system and Cray CLE software packages on your system as described in *Upgrade CLE Software* on page 109. Be certain that the following were true for the CLE upgrade:

- The parameter `direct_attached_lustre` was set to yes in `CLEinstall.conf`
- The `--Centosmedia=directory` option was included when `CLEinstall` was executed
- The `-d` option was included when the `shell_bootimage` script was executed

Build the DAL Image Root

An *image root* is a directory of the contents a bootable image will eventually contain and is built using an image recipe.

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

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

2. Build the image root.

   ```
   smw:~ # impscli build image_recipe dal_cle_5.2up04_centos_6.5_x86-64_gem
   ```

   This image root is provisioned later to generate an image for deployment.

Create the Config Set for DAL Nodes

A *config set* contains site specific settings used by services throughout the Cray system. The IMPS Configurator creates a config set by interactively guiding the administrator through the process of providing needed configuration values. As the Configurator prompts for information about the system, a description and guidance, including a reasonable default value, are provided for each query.

The prompts generated by the IMPS Configurator vary from site to site due to system differences and, therefore, are not displayed in this guide. Key prompts include requests for information regarding node identifiers for DAL and InfiniBand, name service details, time zone information, and LNET routing information.

1. Gather system configuration details. It is helpful to gather the following information before launching the Configurator.
   - Node Identifiers (NIDs) and HSN IP addresses of the DAL nodes

- NIDs of InfiniBand nodes (if applicable)
- Is TCP/IP over InfiniBand supported on this system?
- Is this system configured for multipath?
- Do service nodes utilize a name service such as LDAP? What are the server addresses?
- LNET configuration details
- LNET routing network information (e.g., tcp0, gni, gni32)
- Will the Lustre Monitoring Tool (LMT) be utilized? What will be the LMT administrator's password for this tool?
- Is `ssh` allowed?
- Time zone

2. Launch the Configurator to create the config set. Config sets for DAL nodes reside on the SMW and are named for the partition to be booted, e.g., `p0`, `p1`, `p2`.

   CAUTION: In previous releases, `s`*N* was a valid partition name. This is no longer true, only `p`*N* is a valid partition name.

   ```
   smw:~ # impscli create config_set pN with images \
   dal_cle_5.2up04_centos_6.5_x86-64_net
   ```

   Where *net* is `ari` for Cray XC30 systems or `gem` for Cray XE/XK systems.

Create a Valid `multipath.conf` File

If the DAL nodes utilize multipath, a valid `multipath.conf` file must exist. If a `multipath.conf` file is present, multipath is started when the DAL nodes are booted. To facilitate creating a valid `multipath.conf` file, a template file is available on the SMW in `/etc/opt/cray/share/`*pN*`/dist.d/multipath.conf.cray` after the configuration step.

Although no modifications to the `multipath.conf` file are required, most sites will want to add aliases for the OST devices. Configuring Lustre is easier when using aliases. If aliases are not defined, the DAL nodes must be booted with the `multipath.conf` file in place in order for multipath to generate new OST device names. There is an example of defining an alias at the bottom of the template file, similar to the following:

```
# Example of using alias names instead of WWID.s
# Then use /dev/mapper/<alias> in the lustre xxx.fs_defs
# multipaths {#multipaths {
# multipath {
#   wwid   360001ff08052b000000000308aa20000
#  alias  ccsfs-ost000
# }
#}
```

1. Optional: Edit the template file to have the correct values for the system.

   ```
   smw:~ # vi /etc/opt/cray/share/pN/dist.d/multipath.conf.cray
   ```

2. Optional: Verify the proper zoning of the system.

3. Optional: Copy the template file to `/etc/opt/cray/share/pN/files/class/ib-oss/etc/multipath.conf`. When this file is in place, multipath will be activated at boot time for all nodes in the `ib-oss` class.

```
smw:~ # cp /etc/opt/cray/share/pN/dist.d/multipath.conf.cray \
/etc/opt/cray/share/pN/files/class/ib-oss/etc/multipath.conf
```

4. Optional: Use the multipath devices in `/dev/mapper` when editing the Lustre *fs_name*`.fs_defs` file a later DAL configuration task.

Provision the DAL Image

Provisioning transforms the image contents to the proper format for deployment.

```
smw:~ # impscli provisiondal image dal_cle_5.2up04_centos_6.5_x86-64_net to \
/opt/xt-images/machine-xtrelease-LABEL-partition
```

Where *net* is `ari` for Cray XC30 systems or `gem` for Cray XE/XK systems. The image is stored in the directory `/opt/xt-images/`*machine-xtrelease-LABEL-partition*, where *LABEL* is specified in the output of `CLEinstall`.

Informational messages are displayed. Warning messages, related to creating a CentOS image on an SLES system, are also displayed and can safely be ignored. Finally, a message similar to the following is displayed after the provisioning completes successfully.

```
INFO - Provisioning of DAL image 'dal_cle_5.2up04_centos_6.5_x86-64_net' successful.
```

Create a Boot Image That Includes the DAL Image

A CentOS boot package must be prepared for the system with the specified *LABEL* and placed in the */bootimagedir* directory.

1. Execute the `shell_bootimage` script indicated in the output of `CLEinstall`.

```
smw:~ # /var/opt/cray/install/shell_bootimage_LABEL.sh -d -c -b \
/bootimagedir/bootimage.cpio
```

2. Copy the boot package on the SMW to the same directory on the boot node:

```
smw:~ # scp -p /bootimagedir/bootimage.cpio root@boot:/bootrootdir/bootimagedir/
bootimage.cpio
```

Unmount File Systems and Release Media

Run the post-CLEinstall script to unmount any mounted file systems.

```
smw:~ # /var/opt/cray/install/shell_post_install.sh /bootroot.LABEL /sharedroot.LABEL
smw:~ # umount /media/cdr /media/Centosbase
```

Unmount and eject the software release DVD if it is still loaded.

```
smw:~ # umount /media/cdr /media/Centosbase
smw:~ # exit
```

Boot DAL Service Nodes with CentOS Boot Image

IMPORTANT: The system must be up to complete this process; boot the system if necessary.

Currently, all service nodes are running SLES; those intended for DAL must be halted and then rebooted with CentOS.

Wait for a sufficient time for the service nodes to reboot before proceeding.

```
crayadm@smw:~> xtbootsys --partition pN --reboot -L \
dal_cle_5.2up04_centos_6.5_x86-64_net cnames_of_dal_nodes
```

Where *cnames_of_dal_nodes* is a comma-separated list and *net* is `ari` for Cray XC30 systems or `gem` for Cray XE/XK systems.

Set Up `ssh` Keys

Edit the `/root/.ssh/known_hosts` file to remove the old `ssh` keys. To reset the keys, run the `ssh.sh` script to set up the new CentOS nodes with proper authentication.

```
crayadm@smw:~> ssh root@boot-pN
boot:~ # /var/opt/cray/install/shell_ssh.sh
boot:~ # exit
```

Lustre Post Boot Configuration

For each file system created, perform the following steps after the system has been booted with the DAL nodes for the first time only.

Configure Lustre

Configuring DAL nodes is slightly different than on legacy internal Lustre server configurations because the control files are located in the config set on the SMW instead of the boot root file system. Therefore, the `fs_defs` file must be copied from `/etc/opt/cray/lustre-utils` on the boot node to `/opt/cray-xt-lustre-utils/default/etc` on the SMW.

Copy the fs_defs file from the boot root file system to the SMW.

```
crayadm@smw:~> su -
smw:~ # cd /opt/cray-xt-lustre-utils/default/etc
smw:~ # scp -p root@boot-pN:/etc/opt/cray/lustre-utils/fs_name.fs_defs .
```

IMPORTANT: Save a copy of the `fs_defs` file in another location because the `/opt/cray-xt-lustre-utils/default` link changes with CLE software updates.

Install File System Definition Files

The `fs_defs` file must be installed and the configuration log must be regenerated. The full path to the *fs_name*`.fs_defs` file must be provided.

1. Install the `fs_defs` file.

   ```
   smw:~ # lustre_control install -I /etc/opt/cray/share/pN/lustre fs_name.fs_defs
   Performing 'install' from smw at Thu Oct 31 16:16:32 CDT 2013

   Parsing file system definitions file: fs_name.fs_defs
   Parsed file system definitions file: fs_name.fs_defs
   The 'fs_name' file system definitions were successfully installed!
   ```

2. Log onto the boot node.

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

3. Regenerate the file system's Lustre configuration log.

   ```
   boot:~ # lustre_control write_conf -f fs_name
   ```

   At this point, the file system is still compatible with 1.8.x servers. The upgrade of the file system occurs later in this process.

Start the Lustre File System

From the boot node, start the Lustre file system.

```
boot:~ # lustre_control start -p -f fs_name
Performing 'start' from boot-pN at Fri Nov 8 13:31:47 CST 2013
Starting filesystem(s):
fs_name
All targets mounted successfully
```

Verify Mount Points for Service Nodes

1. Verify that the Lustre entry exists in `/etc/fstab` in class login. If necessary, add missing entries.

   ```
   boot:~ # xtopview -c login
   default/:/ # cat /etc/fstab
   MGS-nid number@gni:/fs_name /lus/fs_name lustre rw,flock,user_xattr,noauto 0 0
   ```

2. Verify that the file system mount point */lus/fs_name* exists.

```
default/:/ # ls -l /lus/fs_name
```

3. If necessary, create the */lus/fs_name* mount point.

```
default/:/ # mkdir -p /lus/fs_name
default/:/ # exit
```

Mount File System on the Login Node

Mount each file system on the login node.

```
boot:~ # ssh login
login:~ # mount /lus/fs_name
```

CAUTION: Do not continue until the mount command is successful.

Verify Write Access to File System

Verify write access to */lus/fs_name* on the login node by checking the timestamp of `test.txt`.

```
login:~ # touch /lus/fs_name/test.txt
login:~ # ls -la /lus/fs_name
total 12
drwxr-xr-x 3 root root 4096 Nov 8 13:49 .
drwxr-xr-x 3 root root 4096 Nov 8 13:43 ..
drwxr-xr-x 3 root root 4096 Nov 8 13:31 .lustre
-rw-r--r-- 1 root root 0 Nov 8 13:49 test.txt
login:~ # exit
boot:~ # exit
```

Verify Mount Points for Compute Nodes

1. Verify that the Lustre entry exists in the `/etc/fstab` file. If necessary, add missing entries.

```
smw:~ # cat /opt/xt-images/templates/default-pN/etc/fstab
MGS-nid number@gni:/fs_name /lus/fs_name lustre rw,flock,user_xattr 0 0
```

2. Verify that the file system mount point */lus/fs_name* exists.

```
smw:~ # mkdir -p /opt/xt-images/templates/default-pN/lus/fs_name
```

3. If changes were made to this compute node `fstab` file or mount point creation, rebuild the boot image to include these changes.

```
smw:~ # /var/opt/cray/install/shell_bootimage_LABEL.sh -d -c -b /bootimagedir/
bootimage.cpio
```

   Where *LABEL* is specified in the output of `CLEinstall`.

4. Log out as `root`.

```
smw:~ # exit
```

Reboot Compute Nodes

The compute nodes must be rebooted to recognize the mounted file system.

```
crayadm@smw:~> xtbootsys -reboot -L CNL0 cnames_of_compute_nodes
```

Where *cnames_of_compute_nodes* is a comma-separated list.

Configure File System for Boot and Shutdown

Edit `xt.lustre.config` to enable the `/etc/init.d/lustre` `startup` or `stop` script to start or stop the Lustre file system at boot or shutdown time, respectively. Add *fs_name* to the `FILESYSTEMS=` line in `xt.lustre.config`. For example:

```
crayadm@smw:~> ssh root@boot
boot:~ # vi /etc/opt/cray/lustre-utils/xt.lustre.config
FILESYSTEMS="lus0"
```

If you have more than one Lustre file system, include all configured file system names, separated by a space. For example:

```
FILESYSTEMS="lus0 lus1"
```

Upgrade to Lustre 2.5

Upgrading to Lustre 2.5 can happen at this point in the upgrade process or at a later time; however, postponing the upgrade too long may cause problems later. Additionally, existing quotas will not work until the upgrade is completed. Skip to *Configure a Boot Automation File for DAL* on page 104 if the upgrade to Lustre 2.5 is postponed.

LFSCK: Add FIDs to inode Attributes

To populate the inodes with the appropriate FIDs, the `dirdata` attribute is set on each MDT. To determine whether the `dirdata` attribute is set, dump the superblock of the MDTs using the `dump2fs` command:

```
crayadm@smw:~> ssh root@boot
boot:~ # ssh MDS_node
mds:~> dumpe2fs -h mdt_device | grep 'Filesystem features'
```

For example, before the `dirdata` attribute is set, the output resembles:

```
Filesystem features: has_journal ext_attr resize_inode dir_index filetype flex_bg
sparse_super large_file huge_file uninit_bg dir_nlink
```

And after the `dirdata` attribute is set, the output resembles:

```
Filesystem features: has_journal ext_attr resize_inode dir_index filetype flex_bg
dirdata sparse_super large_file huge_file uninit_bg dir_nlink
```

The inode attributes are updated using `lfsck`. While executing, `lfsck` performs periodic checkpoints so it can resume operation from where it left off if it is stopped or interrupted. See the Lustre Manual, available at *https://wiki.hpdd.intel.com/display/PUB/Documentation* for more information about `lfsck`.

Progress of the inode update is monitored by watching the `lfsck_namespace` file as follows:

```
boot:~ # ssh MDS_node
mds:~ # lctl get_param mdd/fs_name-MDT0000/lfsck_namespace
```

Before executing `lfsck -t namespace`, the output looks like:

```
mdd.fs_name-MDT0000.lfsck_namespace
name: lfsck_namespace
magic: 0xa0629d03
version: 2
status: init
flags:
param:
time_since_last_completed: N/A
time_since_latest_start: N/A
time_since_last_checkpoint: N/A
latest_start_position: N/A, N/A, N/A
last_checkpoint_position: N/A, N/A, N/A
first_failure_position: N/A, N/A, N/A
checked_phase1: 0
checked_phase2: 0
updated_phase1: 0
updated_phase2: 0
failed_phase1: 0
failed_phase2: 0
dirs: 0
M-linked: 0
nlinks_repaired: 0
lost_found: 0
success_count: 0
run_time_phase1: 0 seconds
run_time_phase2: 0 seconds
average_speed_phase1: 0 items/sec
average_speed_phase2: 0 objs/sec
real-time_speed_phase1: N/A
real-time_speed_phase2: N/A
current_position: N/A
```

These values are updated while `lfsck` is running. Once it has completed, the `success_count` value will increase by 1.

1. Log onto boot node.

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

2. Shutdown Lustre on both the compute and service nodes, then stop the file system.

   ```
   boot:~ # lustre_control umount_clients -c -f fs_name
   boot:~ # lustre_control umount_clients -f fs_name
   boot:~ # lustre_control stop -f fs_name
   ```

3. Set `dirdata` attribute on the MDT.

   WARNING: After `dirdata` has been enabled, the Lustre file system cannot be downgraded to work with 1.8.x servers.

   ```
   boot:~ # ssh MDS_node
   mds:~ # tune2fs -O dirdata mdt_device
   mds:~ # exit
   ```

4. Start Lustre servers.

   ```
   boot:~ # lustre_control start -f fs_name
   ```

5. Updating the inode attributes.

   ```
   boot:~ # ssh MDS_node
   mds:~ # lctl lfsck_start -M fs_name-MDT0000 -t namespace
   mds:~ # exit
   ```

Enable Quotas

Existing quota limit definitions are converted automatically during this upgrade process; nothing explicit needs to be done to retain existing quota limits.

1. Stop Lustre servers:

   ```
   boot:~ # lustre_control stop -f fs_name
   ```

2. Enable accounting and create the space usage database.
   The `lustre_control set_quota_flag` operation executes `tunefs.lustre --quota` on each Lustre target in the specified file system. The `tunefs.lustre` command sets the QUOTA feature flag in the superblock of the target and runs `e2fsck` to build the per-UID/GID disk usage database.

   ```
   boot:~ # lustre_control set_quota_flag -f fs_name
   ```

3. Optional: Enable enforcement.

   If quotas were previously configured and enforcement is desired going forward, execute `lctl conf_param` on the MGS. If this command is run for a file system without quotas configured, an error message is displayed.

   If continuing from the `lustre_control set_quota_flag` operation, start the Lustre servers.

   ```
   boot:~ # lustre_control start -f fs_name
   ```

   To enable enforcement of inode quotas for *fs_name* (must be done on MGS):

   ```
   mgs:~ # lctl conf_param fs_name.quota.mdt=value
   ```

   Where *value* determines for whom quotas are enforced; set to: `u` for users, `g` for groups, `ug` for users and groups, or `none`.

   For further information on `lctl` and enforcing quotas, see the Lustre Manual at *https://wiki.hpdd.intel.com/display/PUB/Documentation*.

OI Scrub

An OI Scrub to create the FID to inode mappings is done automatically as part of the upgrade process. The following commands to start and stop an OI Scrub are not needed to complete the upgrade, but are included here for completeness. These commands are run on the MDS.

- Start OI Scrub

```
mds:~ # lctl lfsck_start -M fs_name-MDT0000
```

- Stop OI Scrub

```
mds:~ # lctl lfsck_stop -M fs_name-MDT0000
```

- Tune OI Scrub speed

```
mds:~ # lctl lfsck_start -M fs_name-MDT0000 -s <max objects/second>
```

- Monitor OI Scrub status

```
mds:~ # lctl get_param osd-ldiskfs/fs_name-MDT0000/oi_scrub
```

Configure a Boot Automation File for DAL

Follow the steps in this section to configure and test the boot automation file for DAL. For more information about boot automation, see the `xtbootsys(8)` man page.

Create Script on Boot Node

NOTE: This step is only necessary for systems with InfiniBand storage connected to the DAL nodes.

Create and save the following script on the boot node in `/root/bin/local.dal-opensm`. This script is called by your boot automation file to ensure that DAL nodes running IB are discovering their LUNs upon boot and must be executable.

```
#!/bin/sh
#
# Local widget to work around opensm startup at boot time on
# dal nodes with Infiniband.
#
pdsh -w cnames_of_dal_nodes "service opensm restart"
```

Where *cnames_of_dal_nodes* is a comma-separated list.

```
boot:~ # vi /root/bin/local.dal-opensm
boot:~ # chmod 755 /root/bin/local.dal-opensm
```

Shutdown the System

From the SMW, use the site-specific procedures to shut down the system. For example, to shut down using an automation file, enter 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
```

Edit the Boot Automation File for DAL

1. Edit the boot automation file.

   ```
   crayadm@smw:~> vi /opt/cray/hss/default/etc/auto.xthostname
   ```

2. Add the following line to boot the DAL nodes after booting the boot and SDB nodes:

   ```
   lappend actions [list crms_boot_loadfile dal_cle_5.2up04_centos_6.5_x86-64_net \
   service cnames_of_your_DAL_nodes linux]
   ```

   Where *cnames_of_your_DAL_nodes* is a comma-separated list.

3. If there are DAL nodes that have InfiniBand attached storage, add the following line to restart `opensm` on those nodes after booting the service nodes:

   ```
   lappend actions { crms_exec_on_bootnode "root" "/root/bin/local.dal-opensm" }
   ```

4. If Network Address Translation (NAT) IP forwarding is configured for MDS nodes to route to external LDAP servers, add the following lines prior to starting Lustre:

   ```
   lappend actions { crms_exec_via_bootnode "NAT_router_node" "root" \
   "path_to_start-nat_script" }
   lappend actions { crms_exec_via_bootnode "MDS_node" "root" "/sbin/route \
   add default gw NAT_router_node_IP"
   ```

   For additional information on NAT IP forwarding, see *Managing System Software for the Cray Linux Environment* (S-2393).

5. Add the following line to start Lustre on the DAL nodes after booting the service nodes:

   ```
   lappend actions { crms_exec_on_bootnode "root" \
   "/opt/cray/lustre-utils/default/bin/lustre_control start -f fs_name" }
   ```

6. Add the following line to mount the DAL clients after starting Lustre on the DAL nodes:

   ```
   lappend actions { crms_exec_on_bootnode "root" \
   "/opt/cray/lustre-utils/default/bin/lustre_control mount_clients -f fs_name -w
   login"}
   ```

Boot Using the Autoboot File

From this point forward, boot your system using the autoboot file. This ensures that the entire system boots normally, including DAL service nodes, and verifies that the Lustre file system is installed and working correctly.

1. On the SMW, set the boot image:

```
crayadm@smw:~> xtcli part_cfg update pN -i /bootimagedir/image_name.cpio
```

2. Boot the system:

```
crayadm@smw:~> xtbootsys --partition pN -a auto.xthostname
```

Verify Shutdown/Reboot Procedures (Optional)

Reboot your system and confirm that shutdown and boot procedures operate as expected.

Install Additional Software

This section details additional software installation that may be needed on a Cray system.

Install the Cray Programming Environments

The Cray Application Developer's Environment (CADE) for Cray XE and Cray XK systems and the Cray Developers Toolkit (CDT) for Cray XC30 systems are available from Cray Inc. as separate software packages. The two packages are comparable but not interchangeable.

Both packages consist of the basic libraries and components needed to develop and compile code on Cray systems, including the GNU Fortran, C, and C++ compilers. The CADE 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.

The CDT package does include CCE, as well as the GNU compilers, but requires a valid license key before CCE can be installed. All other compilers are sold, installed, and licensed separately.

For installation and upgrade instructions, see the *Cray Programming Environments Installation Guide* (S-2372).

Install 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 Measurement and Analysis Tools Installation Guide* (S-2474).

Install 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*.

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 the Cray Linux Environment* (S-2393).

Install 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*

Modify Configuration Values for DAL Service Nodes

The process to modify configuration values for DAL CentOS service nodes is different than the process for SLES service nodes because DAL service nodes are supported by Cray's Image Management and Provisioning System (IMPS). With IMPS, a *config set* contains site specific settings used by services throughout the Cray system. The IMPS Configurator creates a config set by interactively guiding the administrator through the process of providing needed configuration data as in *Create the Config Set for DAL Nodes* on page 94. Modifying configuration values simply involves updating the config set as described in the procedure below.

Among the possible configuration values to reconfigure are members of classes, enable or disable features like multipath, and changing the time zone. For the most specific information, please refer to the information provided during the configuration process.

1. Create a clone of the config set as a backup in the event that the update produces unwanted results.

   ```
   smw:~ # impscli clone config_set pN to backup_config_set
   INFO - Successfully cloned config_set pN to backup_config_set.
   ```

2. Launch the Configurator to update the config set.

   ```
   smw:~ # impscli update config_set pN
   ```

   The Configurator prompts for information as it does when creating a config set, including a description, guidance, and a reasonable default value.

3. Reboot service nodes to pick up new configuration information.

   ```
   smw:~ # exit
   crayadm@smw:~> xtbootsys --partition pN --reboot \
   -L dal_cle_5.2up00_centos_6.5_x86-64_net cnames_of_dal_nodes
   ```

   Where *net* is ari for Cray XC30 systems or gem for Cray XE/XK systems and *cnames_of_dal_nodes* is a comma-separated list.

   Modify LNET Routing Network Information for DAL

   ```
   smw:~ # impscli clone config_set p0 to p0-backup
   INFO - Successfully cloned config_set p0 to p0-backup
   smw:~ # impscli update config_set p0
   ...
   Description:
   Allows you to configure the network(s) that LNET will use for routing
   traffic. Each
   network is a comma seperated string indicating the network name, e.g.,
   'o2ib(ib0)', 'gni'.

   Short Description: A comma seperated list of networks to route traffic
   ```

```
through.

Specify 'gni' for use with a Cray.

Enter string value (press return for default value of 'gni'): gni2538

...
smw:~ # exit
crayadm@smw:~> xtbootsys --partition pN --reboot \
-L dal_cle_5.2up00_centos_6.5_x86-64_net cnames_of_dal_nodes
```

Install RPMs

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.

Generic RPM Usage

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.

To install RPMs on a Cray system, 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 being installed modifies files in `/etc`, it is necessary to 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, the `CRAY_INSTALL_DEFAULT` environment variable can be set to configure the new version as the default. Set this variable before installing 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.

Install an RPM on the SMW

As `root`, use the following command:

```
smw:~# rpm -ivh /directorypath/filename.rpm
```

Install an RPM on the Boot Node Root

As `root`, use the following command:

```
boot:~ # rpm -ivh /directorypath/filename.rpm
```

Install an RPM on the Shared Root

As `root`, use the following commands:

NOTE: If the SDB node has not been started, include the `-x /etc/opt/cray/sdb/node_classes` option when invoking the `xtopview` command.

```
boot:~ # cp -a /tmp/filename.rpm /rr/current/software
boot:~ # xtopview
default/:/ # rpm -ivh /software/filename.rpm
```

Update the Time Zone

When the Cray Linux Environment (CLE) operating system is installed, the Cray system time is set at US/Central Standard Time (CST), which is six hours behind Greenwich Mean Time (GMT). An administrator can change this time.

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 the compute node image, follow the appropriate procedure below.

Change the time zone for the SMW and the blade and cabinet controllers on XE systems

CAUTION: Perform this procedure while the Cray system is shut down; do not flash blade and cabinet 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 blade and cabinet controllers are responding. For example:

```
smw:~ # xtalive -a l0sysd s0
```

2. Optional: Check the current time zone setting for the SMW and controllers.

```
smw:~ # date
Wed Aug 01 21:30:06 CDT 2012

smw:~ # xtrsh -l root -s /bin/date s0
c0-0c0s2 : Wed Aug 01 21:30:51 CDT 2012
c0-0c0s5 : Wed Aug 01 21:30:51 CDT 2012
c0-0c0s7 : Wed Aug 01 21:30:51 CDT 2012
c0-0c1s1 : Wed Aug 01 21:30:51 CDT 2012
.
.
.
c0-0 : Wed Aug 01 21:30:52 CDT 2012
```

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 01 21:32:52 CDT 2012
smw:~ # /usr/sbin/zic -l America/New_York
smw:~ # date
Wed Aug 01 22:33:05 EDT 2012
```

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` file to `/opt/tftpboot`, and then restart the log system and `rsms`.

```
smw:~ # cp /etc/localtime /opt/tftpboot
smw:~ # /etc/init.d/cray-syslog restart
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 proceed to the next step.

   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 01 23:07:07 EDT 2012
   crayadm@smw:~> xtrsh -l root -s /bin/date s0
   c0-0c1s1 : Wed Aug 01 23:07:16 EDT 2012
   c0-0c0s7 : Wed Aug 01 23:07:16 EDT 2012
   c0-0c1s3 : Wed Aug 01 23:07:16 EDT 2012
   .
   .
   .
   c0-0 : Wed Aug 01 23:07:17 EDT 2012
   ```

   Skip the next step and proceed with bouncing the system, below.

8. If the time zone has been changed already, complete this step.

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
```

> **NOTE:** An incompatibility exists between the current version of `/etc/localtime` and earlier versions that may be on the system. This incompatibility causes the `date` command to report an incorrect time on the compute nodes. To resolve this incompatibility, after updating the SMW software you will also need to update the time zone on the compute nodes as described in the procedure *Changing the time zone for compute nodes* in *Installing and Configuring Cray Linux Environment (CLE) Software*.

Change the Time Zone on the Boot Root and Shared Root

Prerequisites

User `root` privileges are required for this procedure.

Perform the following steps to change the time zone. In this example, the time zone is changed from `"America/Chicago"` to `"America/New_York"`.

1. Confirm the time zone setting on the SMW.

```
smw:~ # cd /etc/sysconfig
smw:/etc/sysconfig # grep TIMEZONE clock
TIMEZONE="America/New_York"
DEFAULT_TIMEZONE="US/Eastern"
```

2. Log on to the boot node.

```
smw:/etc/sysconfig # ssh root@boot
```

3. Verify that the `zone.tab` file in the `/user/share/zoneinfo` directory contains the appropriate time zone.

```
boot:~ # cd /usr/share/zoneinfo
boot:/usr/share/zoneinfo # grep America/New_York zone.tab
US      +404251-0740023 America/New_York        Eastern Time
```

4. Create the time conversion information files.

```
boot:/usr/share/zoneinfo # date
Mon Jul 30 22:50:52 CDT 2012
boot:/usr/share/zoneinfo # /usr/sbin/zic -l America\New_York
boot:/usr/share/zoneinfo # date
Mon Jul 30 23:59:38 EDT 2012
```

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:/usr/share/zoneinfo # cd /etc/sysconfig
boot:~ # grep TIMEZONE clock
TIMEZONE="America/Chicago"
DEFAULT_TIMEZONE="US/Eastern"
boot:~ # vi clock
make changes
boot:~ # grep TIMEZONE clock
TIMEZONE="America/New_York"
DEFAULT_TIMEZONE="US/Eastern"
```

```
boot:/usr/share/zoneinfo # cd /etc/sysconfig
boot:~ # grep TIMEZONE clock
TIMEZONE="America/Chicago"
DEFAULT_TIMEZONE="US/Eastern"
boot:~ # vi clock
```

Make site-specific changes.

```
boot:~ # grep TIMEZONE clock
TIMEZONE="America/New_York"
DEFAULT_TIMEZONE="US/Eastern"
```

6. Switch to the default view by using `xtopview`.

> **IMPORTANT:** If the SDB node has not been started, the `-x /etc/opt/cray/sdb/node_classes` option must be included when invoking the `xtopview` command.

```
boot:~ # xtopview
```

7. Verify that the `zone.tab` file in the `/user/share/zoneinfo` directory contains the appropriate time zone.

```
default/:/ # grep America/New_York /usr/share/zoneinfo zone.tab
US      +404251-0740023 America/New_York        Eastern Time
```

8. Create the time conversion information files.

```
default/:/ # date
Mon Jul 30 23:10:52 CDT 2012
default/:/ # /usr/sbin/zic -l America/New_York
default/:/ # date
Tue Jul 31 00:11:38 EDT 2012
```

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/:/etc/sysconfig # grep TIMEZONE clock
TIMEZONE="America/Chicago"
DEFAULT_TIMEZONE="US/Eastern"
default/:/etc/sysconfig # vi clock
make changes
default/:/etc/sysconfig # grep TIMEZONE clock
TIMEZONE="America/New_York"
DEFAULT_TIMEZONE="US/Eastern"
```

10. Exit `xtopview`.

```
default/:/etc/sysconfig # exit
```

Change the Time Zone for Compute Nodes

1. Exit from the boot node and confirm the time zone setting on the SMW.

```
boot:/usr/share/zoneinfo # exit
smw:/etc/sysconfig # grep TIMEZONE clock
TIMEZONE="America/New_York"
DEFAULT_TIMEZONE="US/Eastern"
```

2. Copy the new `/etc/localtime` file from the SMW to the bootimage template directory.

```
smw:/etc/sysconfig # cp -p /etc/localtime \
/opt/xt-images/templates/default/etc/localtime
```

3. Copy the new `/usr/share/zoneinfo` file from the SMW to the bootimage template directory. The directory to contain the time zone file must be created in the bootimage template area.

   NOTE: This procedure enables a single time zone for the compute nodes. If users will be setting the `TIMEZONE` variable to time zones which are not the system default, you may wish to either copy a few of the common time zones used by the user community or the entire `/usr/share/zoneinfo` directory to the `/opt/xt-images/templates/default/` area.

```
smw:/etc/sysconfig # mkdir -p \
/opt/xt-images/templates/default/usr/share/zoneinfo/America
smw:~# cp -p /usr/share/zoneinfo/America/New_York \
/opt/xt-images/templates/default/usr/share/zoneinfo/America/New_York
```

4. Update the boot image to include these changes; follow the steps in *Prepare Compute and Service Node Boot Images* on page 49.

   The time zone is not changed until you boot the compute nodes with the updated boot image.

Upgrade the SDB Database Utilities with a CLE Update Package

After running the `CLEinstall` program and before booting and testing the upgraded system, perform these steps. The 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 pN` (where *N* is the partition number) to boot a partition.

   ```
   crayadm@smw:~> xtbootsys --partition pN
   ```

   You are prompted for the root password.

   ```
   Enter your mainframe's root password (or just hit return)
   ```

2. If you are using site-specific passwords for MySQL accounts (which is recommended), you should verify that these passwords are correct in all of the required locations. These required locations changed in CLE 4.1, so if you are upgrading from a CLE 4.0 version, you must complete *Change Default MySQL Passwords on the SDB* on page 66 for your system to function properly. Verify that your site-specific password exists in all of the required locations in *Change Default MySQL Passwords on the SDB* on page 66 and return here to complete the database update.

3. From the boot node, `ssh` to the SDB.

   ```
   crayadm@smw:~> ssh root@boot
   boot:~ # ssh root@sdb
   ```

4. Stop the SDB.

   ```
   sdb:~ # /etc/init.d/sdb stop
   ```

5. Start the MySQL server.

   ```
   sdb:~ # /etc/init.d/mysql start
   ```

6. Run the upgrade script. When prompted, enter the MySQL root password.

   ```
   sdb:~ # /software/mysql/shell_mysql_upgrade.sh
   ```

7. Stop the MySQL server.

```
sdb:~ # /etc/init.d/mysql stop
Shutting down MySQL..                          done
```

8. Start the SDB.

```
sdb:~ # /etc/init.d/sdb start
starting sdb
XT release: using release 5.2.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
```

9. 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.

Configure Primary and Extended File Partitions

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

Create 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 *Example of Boot LUN Partitions* on page 16.

Configure 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 *Example of Boot LUN Partitions* on page 16.

```
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 *Example of Boot LUN Partitions* on page 16.

```
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 *Example of Boot LUN Partitions* on page 16.

Create 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 *Example of Boot LUN Partitions* on page 16.

Configure 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 *Example of Boot LUN Partitions* on page 16.

```
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 *Example of Boot LUN Partitions* on page 16.

```
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 *Example of Boot LUN Partitions* on page 16.

```
 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 *Example of Boot LUN Partitions* on page 16.

Configure LVM for System Backups

This section covers preparing a system set for Logical Volume Manager (LVM) operation and activating or deactivating volume groups on the System Management Workstation (SMW).

Prepare a System Set for LVM Snapshot Backups

1. Identify a set of partitions that you will set up for LVM, such as a backup system set for your site.

   IMPORTANT: Do not use your current production system set for this procedure. The `pvcreate` command will destroy all of the data on the disk.

2. Create LVM physical volumes on the disk partitions.

   ```
   smw:~ # pvcreate /dev/disk/by-id/scsi-3600a0b800050da0e0000135e4c2a0ce5-part1
   Writing physical volume data to disk "/dev/disk/by-id/scsi-3600a0b800050da0e0000135e4c2a0ce5-part1"
   Physical volume "/dev/disk/by-id/scsi-3600a0b800050da0e0000135e4c2a0ce5-part1" successfully created
   ```

3. Create LVM volume groups on the physical volumes.

   ```
   smw:~ # vgcreate BLUE-VG1 /dev/disk/by-id/scsi-3600a0b800050da0e0000135e4c2a0ce5-part1
   Volume group "BLUE-VG1" successfully created
   ```

4. Create LVM logical volumes on the volume groups.

   ```
   smw:~ # lvcreate -L 30G -n BLUE-BOOTNODE_ROOT BLUE-VG1
   Logical volume "BLUE-BOOTNODE_ROOT" created
   ```

5. Edit the `/etc/sysset.conf` file to include the paths to the logical volumes instead of the traditional `/dev/disk/by-id/...` convention. For example, below is an original non-LVM `/etc/sysset.conf` entry:

   ```
   BOOTNODE_ROOT   /dev/disk/by-id/scsi-3600a0b800050da0e0000135e4c2a0ce5-part1    boot \
   /dev/disk/by-id/scsi-3600a0b800050da0e0000135e4c2a0ce5-part1    /       no
   ```

   For this LVM example, the `SMWdevice` and `hostdevice` columns use the logical volume paths:

   ```
   BOOTNODE_ROOT    /dev/BLUE-VG1/BLUE-BOOTNODE_ROOT       boot   \
   /dev/BLUE-VG1/BLUE-BOOTNODE_ROOT       /        no
   ```

   For reference, */etc/sysset.conf Examples* on page 180 shows complete `/etc/sysset.conf` files before and after LVM backup conversion.

6. Repeat step *2* through step *5* on page 179 for each system set function (e.g., `SHAREDROOT`, `SDB`, `SYSLOG`) you are converting to support LVM.

When you have finished these steps, you are ready to begin an initial install on this system set, or you can use `xthotbackup` to copy your production system set to this label and begin the CLE update or upgrade process here. The `xthotbackup` program will automatically activate and deactivate source and target volume groups on the SMW for its own operation. If `xthotbackup -L` is used to do a live backup, it will automatically activate and deactivate the target volume groups on the booted CLE system.

IMPORTANT: When the installation process is completed on an LVM system set, you must deactivate the volume groups on the SMW. See *Deactivate Volume Groups on the SMW Following an Installation or Upgrade* on page 180 for more information.

LVM Volume Group Activate/Deactivation on SMW

LVM volume groups must be activated on the SMW prior to performing a CLE installation, upgrade, or update for a system set which uses LVM logical volumes. After a CLE installation, upgrade, or update of a system set, the LVM volume groups must be deactivated on the SMW so that they can be used by the CLE service nodes, which will mount file systems on those logical volumes.

Deactivate Volume Groups on the SMW Following an Installation or Upgrade

After all maintenance on the SMW has been completed for a system set, the volume groups should be deactivated on the SMW so that CLE service nodes can activate and use these volumes. Perform the following steps to deactivate the volume groups on the SMW.

1. Optional: View the `/etc/sysset.conf` file and identify which entries in the system set are configured for LVM.

   These entries have LVM volume group names in the `SMWdevice` paths (such as `/dev/`*`BLUE-VG1/`*) instead of the usual `/dev/disk/by-id/scsi...` path name convention.

2. Optional: Deactivate the LVM volume groups you identified in the previous step with the `vgchange` command.

```
smw:~ # vgchange -an BLUE-VG1 BLUE-VG2 BLUE-VG3 BLUE-VG4 BLUE-VG5 BLUE-VG6
```

Activate volume groups on the SMW for a CLE system set

Before performing maintenance on the SMW for a system set, the volume groups should not be in use on any booted CLE service nodes and will need to be activated on the SMW for access. Perform the following steps to activate the volume groups on the SMW.

1. View your `/etc/sysset.conf` file and identify which entries in your system set are configured for LVM.

   These entries will have LVM volume group names in the `SMWdevice` paths (such as `/dev/BLUE-VG1/`) instead of the usual `/dev/disk/by-id/scsi...` path name convention.

2. Activate the LVM volume groups you identified in the previous step with the `vgchange` command.

```
smw:~ # vgchange -ay BLUE-VG1 BLUE-VG2 BLUE-VG3 BLUE-VG4 BLUE-VG5 BLUE-VG6
```

/etc/sysset.conf Examples

Sample /etc/sysset.conf non-LVM system set

```
LABEL:BLUE
DESCRIPTION:BLUE CLE 4.1.40
# function      SMWdevice       host    hostdevice  mountpoint
shared
#
BOOTNODE_ROOT   /dev/disk/by-id/
scsi-3600a0b800050da0e0000135e4c2a0ce5-part1    boot \
/dev/disk/by-id/scsi-3600a0b800050da0e0000135e4c2a0ce5-
part1    /       no
BOOTNODE_SWAP   /dev/disk/by-id/
scsi-3600a0b800050da0e0000135e4c2a0ce5-part2    boot \
/dev/disk/by-id/scsi-3600a0b800050da0e0000135e4c2a0ce5-part2
swap    no
SHAREDROOT      /dev/disk/by-id/
scsi-3600a0b800050da0e000013624c2a0d62-part5    boot \
/dev/disk/by-id/scsi-3600a0b800050da0e000013624c2a0d62-part5    /
rr     no
BOOT_IMAGE0     /dev/disk/by-id/
scsi-3600a0b800050da0e000013624c2a0d62-part7    boot \
/dev/disk/by-id/scsi-3600a0b800050da0e000013624c2a0d62-part7    /
raw0   no
BOOT_IMAGE1     /dev/disk/by-id/
scsi-3600a0b800050da0e000013624c2a0d62-part8    boot \
/dev/disk/by-id/scsi-3600a0b800050da0e000013624c2a0d62-part8    /
raw1   no
SDB             /dev/disk/by-id/
scsi-3600a0b800050d78200001f944c2a0e75-part1    sdb \
/dev/disk/by-id/scsi-3600a0b800050d78200001f944c2a0e75-part1    /var/
lib/mysql  no
SYSLOG          /dev/disk/by-id/
scsi-3600a0b800050da0e000013704c2a0e04-part1    syslog \
/dev/disk/by-id/scsi-3600a0b800050da0e000013704c2a0e04-part1    /
syslog no
UFS             /dev/disk/by-id/
scsi-3600a0b800050d78200001f944c2a0e75-part2    ufs \
/dev/disk/by-id/scsi-3600a0b800050d78200001f944c2a0e75-part2    /
ufs    no
PERSISTENT_VAR  /dev/disk/by-id/
scsi-3600a0b800050da0e000013624c2a0d62-part6    boot \
/dev/disk/by-id/scsi-3600a0b800050da0e000013624c2a0d62-part6    /
snv    no
```

Sample /etc/sysset.conf LVM system set

```
LABEL:BLUE
DESCRIPTION:BLUE CLE 4.1.40
# function      SMWdevice       host    hostdevice  mountpoint
shared
#
BOOTNODE_ROOT   /dev/BLUE-VG1/BLUE-BOOTNODE_ROOT      boot \
/dev/BLUE-VG1/BLUE-BOOTNODE_ROOT      /       no
BOOTNODE_SWAP   /dev/disk/by-id/
scsi-3600a0b800050da0e0000135e4c2a0ce5-part2    boot \
```

```
/dev/disk/by-id/scsi-3600a0b800050da0e0000135e4c2a0ce5-part2
swap    no
SHAREDROOT      /dev/BLUE-VG2/BLUE-SHAREDROOT boot \
/dev/BLUE-VG2/BLUE-SHAREDROOT /rr     no
BOOT_IMAGE0     /dev/disk/by-id/
scsi-3600a0b800050da0e000013624c2a0d62-part7    boot \
/dev/disk/by-id/scsi-3600a0b800050da0e000013624c2a0d62-part7        /
raw0   no
BOOT_IMAGE1     /dev/disk/by-id/
scsi-3600a0b800050da0e000013624c2a0d62-part8    boot \
/dev/disk/by-id/scsi-3600a0b800050da0e000013624c2a0d62-part8        /
raw1   no
SDB             /dev/BLUE-VG3/BLUE-SDB        sdb \
/dev/BLUE-VG3/BLUE-SDB        /var/lib/mysql  no
SYSLOG          /dev/BLUE-VG4/BLUE-SYSLOG     syslog \
/dev/BLUE-VG4/BLUE-SYSLOG     /syslog no
UFS             /dev/BLUE-VG5/BLUE-UFS        ufs \
/dev/BLUE-VG5/BLUE-UFS        /ufs    no
PERSISTENT_VAR  /dev/BLUE-VG6/BLUE-PERSISTENT_VAR     boot \
/dev/BLUE-VG6/BLUE-PERSISTENT_VAR     /snv    no
```

Set Permissions for /sbin/lvdisplay

Prerequisites

Log on to the SMW and know the `root` user password.

The `xtfsconflict(8)` command requires access to the `/sbin/lvdisplay` command to determine logical volume conflicts in systems utilizing multipath for SMW or SDB failover, for example. Editing the `/etc/sudoers` file permits `xtfsconflict` this access. `xtfsconflict` and `/sbin/lvdisplay` are a "read-only" commands that only report system information. These commands do not alter anything in the system.

1. Edit the /etc/sudoers file:

   ```
   smw:~# sudo /usr/sbin/visudo
   ```

2. Provide the `root` password. A vi editor session starts for the `/etc/sudoers` file.

3. Add the following lines to `/etc/sudoers`:

   ```
   crayadm ALL= NOPASSWD : /sbin/lvdisplay
   ```

4. Exit the editor by entering **`:wq`**.

Configure SuSEfirewall2for a Login or Network Node

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`.

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
```

Create Partitions on a Cray System

The `xtcli part_cfg` command updates partition configurations to define a *logical machine* within a Cray 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*.

Connect 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.

   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
   ================
   ```

   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 5.2.UP04 on the system sets chosen for these partitions.