



**XC™ Series CLE 5.2 to CLE 6.0 Software
Migration using a Physical SMW (CLE
6.0.UP03) S-2580 Rev A**

Contents

1 About XC™ Series CLE 5.2 to CLE 6.0 Software Migration using a Physical SMW (CLE 6.0.UP03) S-2580 Rev A.....	7
2 Introduction to Software Migration from CLE 5.2 to CLE 6.0 using a Physical SMW.....	10
3 Migration Training.....	14
4 Migration Planning.....	17
4.1 Information to Collect Before Installation.....	19
5 Preparation of Migration SMW and Boot RAID.....	22
5.1 Prepare for an SMW/CLE Fresh Install.....	22
5.1.1 Network Connections.....	22
5.1.2 SMW Internal Disk Requirements.....	23
5.1.3 Configuration Values.....	24
5.1.4 Passwords.....	25
5.2 Install the Base Operating System on the Migration SMW.....	25
5.2.1 Prepare to Install the Base Linux Distribution.....	26
5.2.2 Install the SLES 12 Base Linux Distribution on the Migration SMW.....	46
5.2.3 Configure Boot RAID Devices.....	51
5.2.4 Make a Snapshot Manually.....	57
5.3 Install the SMW and CLE Software on the Migration SMW.....	58
5.3.1 Start a Typescript File.....	59
5.3.2 Prepare to Bootstrap the SMW Installation.....	60
5.3.3 Determine the Persistent Device Name for a LUN.....	62
5.3.4 RAID Disk Space Requirements.....	63
5.3.5 Bootstrap the SMW Installation.....	65
5.3.6 Provision SMW Storage.....	71
5.3.7 Run the Installer for an Initial Installation.....	72
5.3.8 Set Default Snapshot and Boot the SMW.....	73
5.4 Configure Other Features and Services.....	74
5.4.1 Set or Change the HSS Data Store (MariaDB) Root Password.....	75
5.4.2 Start a Typescript File.....	76
5.4.3 Make a Post-install Snapshot using snaputil.....	77
5.4.4 Update install.cle.conf for Software Updates.....	77
5.4.5 Configure Power Management.....	78
5.4.6 Reduce Impact of Btrfs Periodic Maintenance on SMW Performance	82
5.4.7 Configure the Simple Event Correlator (SEC).....	82
5.4.8 Prevent Unintentional Re-creation of Mail Configuration Files.....	83

5.4.9 Install the Dell Systems Management Tools and Documentation DVD.....	83
6 Preparation of Configuration Data and Software Images on the Migration SMW.....	85
6.1 Start a Typescript File.....	85
6.2 Extract Configuration Data from the CLE 5.2 / SMW 7.2 System.....	86
6.3 Read Man Pages for New Commands.....	98
6.4 Transfer Configuration Data to Configuration Worksheets.....	98
6.4.1 Prepare Global Worksheets for Migration.....	100
6.4.2 Prepare CLE Worksheets for Migration.....	137
6.5 Load and Validate Configuration Data on the Migration SMW.....	301
6.5.1 Disable Pre- and Post-configuration Scripts.....	301
6.5.2 Update Global Config Set from Worksheets.....	302
6.5.3 Create New CLE Config Set from Worksheets.....	302
6.5.4 Update CLE Config Set	303
6.5.5 Update /etc/motd for Nodes.....	304
6.5.6 Copy Files for External Lustre Fine-grained Routing.....	305
6.5.7 Configure Files for Cray Simple Sync Service.....	305
6.5.8 Validate Config Sets.....	312
6.5.9 Ensure Time Zone Setting Accessible by Cabinet and Blade Controllers.....	313
6.5.10 Continue Initial DataWarp Configuration.....	314
6.6 Update Non-config-set Configuration Files on the Migration SMW.....	319
6.6.1 Create Hardware Test Configuration with xtdiscover.....	319
6.6.2 Assign Service Nodes Manually and Disable Components.....	320
6.6.3 Check for Site Modifications in SMW xtrim.conf	321
6.6.4 Check for Site Modifications to SEDC Files.....	322
6.6.5 Check for Site Modifications to SMW Firewall and IP Tables.....	322
6.7 Choose Image Recipes to Build.....	323
6.7.1 Where to Place the Root File System—tmpfs versus Netroot.....	324
6.7.2 Create a NIMS Map.....	325
6.7.3 About Image Groups and How to Customize Them.....	326
6.7.4 About the Admin Image.....	328
6.8 Build Image Roots and Boot Images from Recipes.....	328
6.8.1 Bootstrap NIMS with imgbuilder.....	329
6.8.2 Install SMW/CLE Patches on the Migration SMW.....	330
6.8.3 Prepare Cray Image Groups and Custom Recipes.....	332
6.8.4 Assign Nodes to New NIMS Groups.....	335
6.8.5 Build Images and Map Them to NIMS Groups.....	339
6.9 Assign Kernel Parameters to Nodes.....	341
6.9.1 Set the Turbo Boost Limit.....	341

6.10 Check NIMS Information	342
6.11 Identify and Port Site-local Scripts.....	343
6.12 Install Cray Programming Environment (PE) Software.....	344
6.12.1 Install Additional Cray Programming Environment (PE) Software Releases to Image Root .	347
7 Preservation of Other Data Prior to Final Shutdown.....	350
8 Shutdown and Switch using a Physical Migration SMW.....	353
8.1 Shut Down the CLE System.....	353
8.2 Put the SMW HA Cluster into Maintenance Mode during a Migration.....	354
8.3 Switch Cabling to Migration SMW and Boot RAID.....	354
8.4 Discover XC System Hardware.....	356
8.4.1 Start a Typescript File.....	356
8.4.2 Bootstrap Hardware Discovery.....	357
8.4.3 Update Firmware.....	359
8.4.4 Discover Hardware and HSN Routing, Prepare STONITH	360
8.4.5 (Optional) Configure Partitions.....	362
8.4.6 Repurpose Compute Nodes.....	363
8.4.7 Finish Configuring the SMW for the CLE System Hardware.....	363
8.4.8 Enable System Environmental Data Collections (SEDC).....	364
8.5 Complete CLE Configuration.....	365
8.5.1 Update and Validate Global Config Set after Migration Switch.....	365
8.5.2 Update and Validate CLE Config Sets after Migration Switch.....	366
8.5.3 Check CLE Hostnames in /etc/hosts File.....	366
8.5.4 Display and Capture all Config Set Information.....	367
8.5.5 Make a Post-config Snapshot using snaputil.....	368
8.5.6 Make a Post-config Backup of Current Global and CLE Config Sets.....	369
8.5.7 Check NIMS Information	369
8.6 Boot the CLE System during a Migration.....	370
8.6.1 Boot the Boot and SDB Nodes.....	371
8.6.2 Restore ALPS Files to /alps_shared.....	371
8.6.3 Push Diag Image to Boot Node.....	372
8.6.4 Push Netroot Images to Boot Node.....	373
8.6.5 Push PE Image Root to Boot Node.....	374
8.6.6 Boot the Rest of the System using a Boot Automation File.....	376
8.6.7 Run Tests after Boot is Complete.....	378
8.6.8 Prepare Site and Software Revision Information Reporting using xtgetrev and xtshowrev.....	379
8.6.9 Test xtdumpsys and cdump.....	380
8.6.10 Make a Post-boot Snapshot using snaputil.....	382
8.6.11 Make a Post-boot Backup of Current Global and CLE Config Sets.....	383

8.7 Complete Post-boot Configuration of Config Services.....	384
8.7.1 Apply Site Firewall and IP Tables Configuration via Config Set and Ansible Play	386
8.7.2 Update and Validate CLE Config Set for Post-boot Changes.....	387
8.7.3 Configure Direct-attached Lustre (DAL).....	388
8.7.4 LMT Configuration for DAL.....	395
8.8 Install and Configure Additional Software.....	400
8.8.1 Complete DataWarp Configuration.....	401
8.8.2 Install and Configure a Workload Manager (WLM).....	403
8.9 Back Up the Newly Installed and Configured SMW/CLE Software.....	404
8.10 Back Up Site Data.....	404
8.11 Restore Operational Data during a Migration.....	406
8.12 Roll Back Changes during a Migration.....	407
9 Supplemental Information.....	410
9.1 About Cray Scalable Services.....	410
9.2 Cray XC System Configuration.....	412
9.3 About Config Sets.....	414
9.4 About Config Set Caching.....	415
9.5 About Variable Names in the Configurator and Configuration Worksheets.....	416
9.6 About Node Groups.....	416
9.7 About Simple Sync.....	419
9.8 About Boot Automation Files.....	423
9.9 About Snapshots and Config Set Backups during a Migration.....	424
9.10 Install Third-Party Software with a Custom Image Recipe.....	424
9.11 Prefixes for Binary and Decimal Multiples.....	430
10 Checklists for Migration using a Physical Migration SMW.....	431
10.1 Master Checklist for Migration using a Physical Migration SMW.....	431
10.2 Installation Checklist 1: Install the Base Operating System on the SMW.....	433
10.3 Installation Checklist 2: Install the SMW and CLE Software.....	434
10.4 Migration Checklist 1.1-P: Configure Other Features and Services.....	434
10.5 Migration Checklist 2.1: Extract Configuration Data.....	435
10.6 Migration Checklist 2.2: Transfer Global Configuration Data.....	435
10.7 Migration Checklist 2.3: Transfer CLE Configuration Data.....	436
10.8 Migration Checklist 2.4: Load and Validate Configuration Data on the Migration SMW.....	438
10.9 Migration Checklist 2.5-P: Update Non-config-set Configuration Files on the Migration SMW.....	439
10.10 Migration Checklist 2.6: Build Image Roots and Boot Images from Recipes.....	439
10.11 Migration Checklist 3.2-P: Discover XC System Hardware.....	440
10.12 Migration Checklist 3.3-P: Complete CLE Configuration.....	440
10.13 Migration Checklist 3.4: Boot the CLE System.....	441

10.14 Migration Checklist 3.5: Complete Post-boot Configuration of Config Services.....442

10.15 Migration Checklist 3.6-P: Install and Configure Additional Software.....443

1 About XC™ Series CLE 5.2 to CLE 6.0 Software Migration using a Physical SMW (CLE 6.0.UP03) S-2580 Rev A

XC™ Series CLE 5.2 to CLE 6.0 Software Migration using a Physical SMW (CLE 6.0.UP03) S-2580 Rev A, published 04 May 2017, supersedes XC™ Series CLE 5.2 to CLE 6.0 Software Migration using a Physical SMW, which was published 17 March 2017.

Scope and Audience

XC™ Series CLE 5.2 to CLE 6.0 Software Migration using a Physical SMW (S-2580) provides an overview and details of the tasks necessary to migrate from CLE 5.2.UP04 / SMW 7.2.UP04 software to CLE 6.0.UP03 / SMW 8.0.UP03 software on Cray XC™ series hardware.

The migration from CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0 requires assistance from Cray. This publication is intended to be used under the guidance of Cray field support staff.

CLE 6.0.UP03 / SMW 8.0.UP03 Release

XC™ Series CLE 5.2 to CLE 6.0 Software Migration using a Physical SMW (CLE 6.0.UP03) S-2580 Rev A supports Cray software release CLE 6.0.UP03 / SMW 8.0.UP03 for Cray XC™ Series systems, released on 16 February 2017.

New in Revision A

- The [Set the Turbo Boost Limit](#) on page 341 procedure was updated to reflect currently released processors.

Command Prompt Conventions

- Host name and account in command prompts**
- The host name in a command prompt indicates where the command must be run. The account that must run the command is also indicated in the prompt.
- The `root` or super-user account always has the `#` character at the end of the prompt.
 - Any non-`root` account is indicated with `account@hostname>`. A user account that is neither `root` nor `crayadm` is referred to as `user`.

<code>smw#</code>	Run the command on the SMW as <code>root</code> .
<code>cmc#</code>	Run the command on the CMC as <code>root</code> .
<code>sdb#</code>	Run the command on the SDB node as <code>root</code> .

<code>crayadm@boot></code>	Run the command on the boot node as the <code>crayadm</code> user.
<code>user@login></code>	Run the command on any login node as any non- <code>root</code> user.
<code>hostname#</code>	Run the command on the specified system as <code>root</code> .
<code>user@hostname></code>	Run the command on the specified system as any non- <code>root</code> user.
<code>smw1#</code> <code>smw2#</code>	For a system configured with the SMW failover feature there are two SMWs—one in an active role and the other in a passive role. The SMW that is active at the start of a procedure is <code>smw1</code> . The SMW that is passive is <code>smw2</code> .
<code>smwactive#</code> <code>smwpassive#</code>	In some scenarios, the active SMW is <code>smw1</code> at the start of a procedure—then the procedure requires a failover to the other SMW. In this case, the documentation will continue to refer to the formerly active SMW as <code>smw1</code> , even though <code>smw2</code> is now the active SMW. If further clarification is needed in a procedure, the active SMW will be called <code>smwactive</code> and the passive SMW will be called <code>smwpassive</code> .

Command prompt inside chroot

If the `chroot` command is used, the prompt changes to indicate that it is inside a `chroot` environment on the system.

```
smw# chroot /path/to/chroot
chroot-smw#
```

Directory path in command prompt

Example prompts do not include the directory path, because long paths can reduce the clarity of examples. Most of the time, the command can be executed from any directory. When it matters which directory the command is invoked within, the `cd` command is used to change into the directory, and the directory is referenced with a period (`.`) to indicate the current directory.

For example, here are actual prompts as they appear on the system:

```
smw:~ # cd /etc
smw:/etc# cd /var/tmp
smw:/var/tmp# ls ./file
smw:/var/tmp# su - crayadm
crayadm@smw:~> cd /usr/bin
crayadm@smw:/usr/bin> ./command
```

And here are the same prompts as they appear in this publication:

```
smw# cd /etc
smw# cd /var/tmp
smw# ls ./file
smw# su - crayadm
crayadm@smw> cd /usr/bin
crayadm@smw> ./command
```

Typographic Conventions

Monospace	Indicates program code, reserved words, library functions, command-line prompts, screen output, file/path names, and other software constructs.
Monospaced Bold	Indicates commands that must be entered on a command line or in response to an interactive prompt.
<i>Oblique or Italics</i>	Indicates user-supplied values in commands or syntax definitions.
Proportional Bold	Indicates a graphical user interface window or element and key strokes (e.g., Enter , Alt-Ctrl-F).
\ (backslash)	At the end of a command line, indicates the Linux® shell line continuation character (lines joined by a backslash are parsed as a single line). Do not type anything after the backslash or the continuation feature will not work correctly.

Feedback

Your feedback is important to us. Visit the Cray Publications Portal at <http://pubs.cray.com> and make comments online using the **Contact Us** button in the upper-right corner, or email comments to pubs@cray.com.

Trademarks

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

2 Introduction to Software Migration from CLE 5.2 to CLE 6.0 using a Physical SMW

The Cray CLE 6.0 / SMW 8.0 releases use SUSE® Linux Enterprise Server (SLES) 12 as the base operating system on the SMW. The most recent prior release, CLE 5.2.UP04 / SMW 7.2.UP04, was based on SLES 11. Because SLES 12 represents a major change in architecture and features, the transition from SLES 11 to SLES 12 requires a software *migration* rather than a software *upgrade*.

To help with this, Cray has created a one-time CLE/SMW migration process and esLogin migration process for use by customer staff and on-site Cray field support staff to migrate from CLE 5.2.UP04 / SMW 7.2.UP04 (SLES 11) to CLE 6.0.UP03 / SMW 8.0.UP03 (SLES 12).

Migration Goal and the "Migration SMW"

The goal of the migration process is to minimize system downtime (unavailability to run user jobs) while preserving necessary configuration and operation data. To achieve this goal, the CLE/SMW migration process begins with planning and extensive preparation. Thorough preparation, which is the key to a successful migration, requires the use of a *migration SMW*. The migration SMW can be either a virtual SMW hosted on an on-site Cray laptop or a spare physical SMW and boot RAID that are not connected to the currently running XC system.

This publication provides details of the process that uses a spare physical migration SMW to migrate CLE/SMW configuration, image, and operational data from the current system to a CLE 6.0.UP03 / SMW 8.0.UP03 system.

- For the process that uses a virtual SMW to migrate the necessary data, see *XC™ Series CLE 5.2 to CLE 6.0 Software Migration using a Virtual SMW (CLE 6.0.UP03) S-2575*.
- For an overview of both CLE/SMW migration processes, see *XC™ Series CLE 5.2 to CLE 6.0 Software Migration Overview (CLE 6.0.UP03) S-2574*.
- For the process to migrate external login nodes, see *XC™ Series esLogin to eLogin Migration Guide S-2584 Rev A*.

This publication also provides migration checklists to use for tracking progress through this process: [Checklists for Migration using a Physical Migration SMW](#) on page 431.

Migration Scope

- **Software Releases.** Migration of CLE and SMW software is supported from CLE 5.2.UP04 and SMW 7.2.UP04 to CLE 6.0.UP03 and SMW 8.0.UP03 only. To migrate from an earlier release, update to CLE 5.2.UP04 and SMW 7.2.UP04 first, then use this migration process. To migrate to a later CLE 6.0 / SMW 8.0 release, use this migration process, then update from CLE 6.0.UP03 and SMW 8.0.UP03.
- **Hardware.** This migration process applies to XC series systems only. CLE 6.0 / SMW 8.0 releases do not support XE or XK series hardware. Also, the CLE 6.0.UP03 / SMW 8.0.UP03 release does not support Intel® Xeon Phi™ "Knight's Corner" (KNC) nodes. To reduce risk, this software migration process assumes that no hardware upgrades will be attempted during the migration.

- **CIMS/esLogin.** Migration is supported from a CIMS (Cray Integrated Management Services) running Bright Computing software to manage esLogin-based CDL (Cray Development and Login) nodes to a CMC (Cray Management Controller) running CSMS (Cray System Management Software) and OpenStack software to manage eLogin-based CDL nodes. The process to extract information from a CDL node managed by a CIMS running Bright Computing software is provided in *XC™ Series esLogin to eLogin Migration Guide S-2584 Rev A*.
- **SMW HA.** Migration of an SMW HA system is supported from SLEHA11.SP3.UP02 to SLEHA12.SP0.UP03 in two stages: migrate the first SMW using this migration process, and then proceed with the standard SMW HA fresh install process for installing and configuring SMW HA software on the first SMW, setting up the second SMW, and configuring the SMW HA cluster, as described in *XC™ Series SMW HA Installation Guide (S-0044)*. The detailed migration guides contain some "SMW HA only" steps and notes that apply to migration of the first SMW.
- **Power Management.** This migration process does not migrate power management data. See *XC™ Series Power Management Administration Guide (CLE 6.0.UP03) S-0043* for information pertaining to backing up the database. The erfs data must be regenerated.

Migration Process Phases

1. **Training.** This first phase consists of learning about the new Cray Management System (CMS) through Cray training classes, Cray technical publications, and publicly available Ansible documentation (Ansible is leveraged heavily by the new management system).
2. **Planning.** This second phase plans the hardware and software configurations needed for a successful migration. Sites are encouraged to contact their district service manager for help with this phase.
3. **Preparation of Physical Migration SMW and Boot RAID.** This phase performs a fresh install of the CLE 6.0.UP03 / SMW 8.0.UP03 software release on the spare physical SMW that will be used for migration.
4. **Preparation of Configuration Data and Images.** This phase extracts configuration data from the currently running system and transfers that data to configuration worksheets in the new release. It also builds the appropriate software images and assigns them to nodes. Most of the procedures in this phase are done on the migration SMW.
5. **Preservation of Other Data.** This phase captures accounting, operational, and user data from the currently running CLE system just prior to shutting it down.
6. **Shutdown and Switch.** This final phase disconnects the original SMW and boot RAID from the XC hardware and connects the migration SMW and new boot RAID to the XC hardware. It also completes any configuration requiring connection to XC hardware. Note that rolling back to the CLE 5.2 / SMW 7.2 system at this point in the migration process is possible only with a physical migration SMW.

Migration Caveats

The following list of features and components have one or more associated caveats to be aware of.

Intel® Xeon Phi™ "Knight's Corner" (KNC) nodes	The CLE 6.0.UP03 / SMW 8.0.UP03 release does not support the KNC processor. If this system contains KNC nodes, do not attempt a hardware upgrade while migrating. See the "Migration Planning" section for more information.
SMW HA	When doing a migration where the end result is SMW HA, the two SMWs that will run SMW 8.0 / CLE 6.0 with SLEHA12.SP0.UP03 must be matched hardware:

same model (both Dell PowerEdge™ R815 Rack Servers or both Dell PowerEdge™ R630 Rack Servers)
same memory and processor speed
same number of disk drives in each SMW
same capacity disk drives in each SMW
same capacity disk in the matching drive bays of the two SMWs
(if using R630 SMWs) RAID controller has the same configuration to present the disks to Linux: disks in slot 0 through slot 3 as a virtual disk with RAID5 and the disk on slot 4 not in RAID configuration

Lustre

- This migration process does not include a tested procedure for preserving a DAL file system during migration, though it is expected to be possible. That will be the responsibility of each customer site. Sites that use part of the boot RAID for DAL will need to use care when reformatting the boot RAID so that the LUNs used for DAL are preserved.
- External Lustre file systems should not require reformatting.
- Direct-attached Lustre (DAL) file systems should not require reformatting.
- DAL LMT (Lustre Monitoring Tool) database will not be migrated.

DataWarp

- If this site has not reformatted/over-provisioned Intel P3608 SSD cards as directed in FN6121a *Datawarp - Performance Issues*, then these Intel P3608 SSD cards must be reformatted. This will be done during the "Shutdown and Switch" phase of the migration process.
- DataWarp Fusion IO SSDs that are ioMemory3 (for example, SX300) are supported in the CLE 6.0.UP03 / SMW 8.0.UP03 release, but no other models from Fusion IO are supported. The SLES 12 version of SanDisk/Fusion driver (VSL4.2.5) requires firmware version 8.9.5. Sites may need to update (flash) the driver firmware to 8.9.5. However, once updated, the firmware cannot be reverted to the previous version. **DO NOT UPDATE FIRMWARE NOW.** That will be done during the "Shutdown and Switch" phase of the migration process.
 - ⚠ **CAUTION:** Once updated, the firmware revision cannot be reverted to the previous version, so the SSDs will NOT be usable in a CLE 5.2 / SMW 7.2 system.
- DataWarp SanDisk Fusion ioScale2 SSD PCIe boards are no longer supported with CLE 6.0 / SMW 8.0.

Workload managers

Workload manager (WLM) logs are not preserved. If log preservation is desired, speak with the WLM vendor.

Accounting data

Accounting data is not preserved; however, this migration process includes a step for running final accounting reports for the CLE 5.2 / SMW 7.2 system just prior to system shutdown.

FC/SAS/Ethernet cards

Firmware updates for the FC cards, SAS cards, and Ethernet cards that are used in the SMW and CLE nodes should be current. Cray has not tested whether old firmware works after SLES 12 has been installed.

Network interfaces

This migration process does not include a tested procedure for configuring bonded or VLAN network interfaces for a fresh install of CLE 6.0.UP03 / SMW 8.0.UP03.

Warning about Potential Loss of Important Data



WARNING: This migration process includes a fresh install, and when a fresh install is performed on a system, disks are wiped clean. There is a risk of losing important data.

Sites planning to migrate from CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0 should be aware of the consequences of a migration process that includes a fresh install. This migration process will wipe SMW internal disks and the boot RAID. To prevent loss of important data, the "Preparation of Configuration Data and Images" and "Preservation of Other Data" phases of this process provide procedures and guidance for selecting and archiving data prior to shutdown of the CLE 5.2 / SMW 7.2 system.

Sites that use part of the boot RAID for DAL can reformat the boot RAID in such a way that the LUNs used for DAL are preserved.

Here are some ways that SLES 12 and the CLE 6.0.UP03 / SMW 8.0.UP03 release handle SMW and boot RAID storage differently, which is why a fresh install is necessary in this migration process.

- New LUNS created on boot RAID to hold CLE 6.0 / SMW 8.0 file systems.
- Dell R815 SMW uses software RAID1 on two drives for the operating system.
- Dell R630 SMW uses hardware RAID5 on four drives for the operating system.
- Additional SMW disk used for the Power Management (Postgres) database.
- SLES 12 installed on SMW into new disk partitions (`/`, `swap`, `/boot`).
- SMW, boot, and SDB nodes use LVM volume groups on the boot RAID.

3 Migration Training

With the CLE 6.0 / SMW 8.0 releases, Cray has changed the way software is installed, configured, and managed on XC Series systems. Because the CLE 6.0.UP03 / SMW 8.0.UP03 release, which is based on SLES 12, is so different from the CLE 5.2.UP04 / SMW 7.2.UP04 release, which is based on SLES 11, sites experienced with CLE 5.2.UP04 / SMW 7.2.UP04 and older Cray software releases will need to learn about the new Cray Management System (CMS) for a successful migration to the new release.

The new management system uses a common installation process for SMW and CLE, leverages standard Linux and open source tools, and centralizes configuration, keeping configuration data separate from software images until that data is applied to nodes at boot time or whenever `cray-ansible` is run. The core elements of this new management system are:

- IMPS** The Image Management and Provisioning System (IMPS) is responsible for creating and distributing repository content and for prescriptive image creation.
- CMF** The Configuration Management Framework (CMF) comprises the configuration data (stored in config sets on the SMW), tools to manage and distribute that data (e.g., the configurator and the IMPS Distribution System (IDS)), and software to apply the configuration data to the running image (`cray-ansible` and Ansible plays).
- NIMS** The Node Image Mapping Service (NIMS) is responsible for keeping track of which images get booted on which nodes, what additional kernel parameters to pass to nodes at boot time, and which load file to use within a boot image.

To help customer sites learn about the new system software, Cray recommends a combination of resources: publicly available Ansible documentation (books and websites), Cray training, Cray technical publications, and the collection of topics in [Supplemental Information](#) on page 410 at the end of this publication.

Ansible Documentation

Working with the new Cray management software requires a basic understanding of Ansible and Python. Configuration data is applied in large part through Ansible plays, and sites may wish to write their own Ansible plays to supplement that functionality. Some familiarity with the Python programming language will be helpful because Ansible is based on and extendable by Python, and many new Cray tools are written in Python. Get acquainted with this material first, if possible.

Cray recommends reading at least one of these Ansible books (or an equivalent):

- *Ansible: Up & Running: Automating Configuration Management and Deployment the Easy Way*, by Lorin Hochstein
- *Ansible for DevOps: Server and configuration management for humans*, by Jeff Geerling
- *Ansible Playbook Essentials*, by Gourav Shah
- *Mastering Ansible*, by Jesse Keating

Visit these websites for more information about Ansible:

- <https://www.ansible.com/configuration-management>

- <http://docs.ansible.com/>

Cray Training

Cray offers a four-day training course on Cray System Administration, recommended for both site staff and Cray on-site support staff. Even those who have already taken this course with CLE 5.x and SMW 7.x content need to take it again to learn about the new CLE 6.x and SMW 8.x content.

Cray Technical Publications

All Cray technical publications are available at <http://pubs.cray.com>.

To prepare for this migration, Cray strongly recommends reading the following technical publications:

- *What's New for CLE 6.0 and SMW 8.0 (CLE 6.0.UP03) S-2573*
- *XC™ Series CLE 5.2 to CLE 6.0 Software Migration using a Virtual SMW (CLE 6.0.UP03) S-2575* or *XC™ Series CLE 5.2 to CLE 6.0 Software Migration using a Physical SMW (CLE 6.0.UP03) S-2580*
- *XC™ Series Configurator User Guide (CLE 6.0.UP03) S-2560*
- *XC™ Series Cray Ansible Writing Guide (CLE 6.0.UP03) S-2582*
- *XC™ Series esLogin to eLogin Migration Guide S-2584 Rev A*
- *XC™ Series eLogin Installation Guide (CLE 6.0.UP03) S-2566 Rev A*
- *XC™ Series eLogin Administration Guide (CLE 6.0.UP03) S-2570 Rev A*
- *XC™ Series Boot Troubleshooting Guide (CLE 6.0.UP03) S-2565*
- *XC™ Series System Administration Guide (CLE 6.0.UP03) S-2393*
- *XC™ Series Software Installation and Configuration Guide (CLE 6.0.UP03) S-2559* (Note that although the detailed migration guides share most of the content of this publication, there may be other content here that would be useful to reference.)
- *XC™ Series Power Management Administration Guide (CLE 6.0.UP03) S-0043*

If these optional features are or will be used at this site, read these publications also:

- DataWarp:
 - *XC™ Series DataWarp™ Installation and Administration Guide (CLE 6.0.UP03) S-2564*, which supersedes *DataWarp Installation Guide S-2547*
 - *XC™ Series DataWarp™ User Guide (CLE 6.0.UP03) S-2558*

- DVS:

Although the Cray Data Virtualization Service (DVS) is an integral feature of the CLE 6.0.UP03 release, its use by sites to project an external file system or provide access to DataWarp is optional. For those purposes, familiarity with the DVS guide is necessary.

- *XC™ Series DVS Administration Guide (CLE 6.0.UP03) S-0005*
- *XC™ Series GPFS Software Installation Guide (CLE 6.0.UP03) S-2569*
- Lustre: *XC™ Series Lustre® Administration Guide (CLE 6.0.UP03) S-2648*
- Shifter:
 - *XC™ Series Shifter Configuration Guide (CLE 6.0.UP03) S-2572*

- *XC™ Series Shifter User Guide (CLE 6.0.UP03) S-2571*
- *Slurm: XC™ Series Slurm Installation Guide (CLE 6.0.UP03) S-2538*
- *SMW HA (high availability):*
 - *XC™ Series SMW HA Installation Guide (SLEHA12.SP0.UP03) S-0044*
 - *XC™ Series SMW HA Administration Guide (SLEHA12.SP0.UP03) S-2551*
- *SEDC: XC™ Series System Environment Data Collections (SEDC) Guide (CLE 6.0.UP03) S-2491*

4 Migration Planning

For more information and guidance about any of the following planning activities, contact the district service manager (DSM) for this site.

Plan Hardware

This migration process assumes that sites will not upgrade hardware (for example, upgrading Intel® Xeon Phi™ processors, from KNC to KNL) during the migration because of the risk involved in changing software and hardware at the same time. However, some network connection and certain hardware additions may be necessary, depending on the following considerations:

- Is the SDB node connected to the admin network in the CLE 5.2 / SMW 7.2 system?

Every XC system should have an Ethernet switch with a network for the SMW, boot, and SDB nodes (an "admin" network). Sites that did not connect the SDB node to this admin network in their CLE 5.2 / SMW 7.2 system must do so for migration to a CLE 6.0 / SMW 8.0 system. CLE 6.0 / SMW 8.0 requires that SDB connection. If the system to be migrated does not have an Ethernet card for the SDB node, this site must obtain one.

- How many nodes are available for use as tier2 nodes?

New nodes may need to be added to the system if an insufficient number of nodes are available (see "Plan Tier2 Nodes" below).

- Which type of migration SMW will be used: virtual or physical?

This guide assumes that this site has decided to use a physical migration SMW, so a spare SMW and boot RAID must be obtained.

- Is the CLE 5.2 / SMW 7.2 system to be migrated an SMW HA system?

If this system is SMW HA, an additional spare SMW must be obtained, and it must match the spare SMW obtained for the physical migration SMW:

- same model (both Dell PowerEdge™ R815 Rack Servers or both Dell PowerEdge™ R630 Rack Servers)
- same memory and processor speed
- same number of disk drives in each SMW
- same capacity disk drives in each SMW
- same capacity disk in the matching drive bays of the two SMWs
- (if using R630 SMWs) RAID controller has the same configuration to present the disks to Linux: disks in slot 0 through slot 3 as a virtual disk with RAID5 and the disk on slot 4 not in RAID configuration

- Does the system being migrated have Intel® Xeon Phi™ "Knight's Corner" (KNC) nodes?

The CLE 6.0.UP03 / SMW 8.0.UP03 release does not support Intel® Xeon Phi™ "Knight's Corner" (KNC) nodes. If this system has KNC nodes, Cray suggests one of the following options:

Table 1. What to do with KNC nodes/blades

Situation	Recommended option
Remove KNC blades and not replace them.	Option 1: <ol style="list-style-type: none"> 1. Remove the KNC blades and confirm that HSS routing works correctly. 2. Perform the migration.
Replace KNC blades with a new blade type supported by both CLE 5.2 and CLE 6.0.	Option 2: <ol style="list-style-type: none"> 1. Replace the KNC blades with the new blades and confirm that HSS routing works correctly. 2. Perform the migration.
Replace KNC blades with a new blade type supported by CLE 6.0 but NOT supported by CLE 5.2, such as Intel® Xeon Phi™ "Knight's Landing" (KNL).	Option 3: <ol style="list-style-type: none"> 1. Leave the KNC blades in the system (Aries enabled) but disable the KNC nodes. 2. Perform the migration. 3. Remove the KNC blades and confirm that HSS routing works correctly with the blades gone. 4. Add the new blades and confirm that HSS routing works correctly with the new blades.

Plan Tier2 Nodes

Tier2 nodes are an important part of Cray Scalable Services, which enables configuration data and software on the SMW and boot node to be made available to the rest of the system and log data from the system to be aggregated on the SMW. Cray Scalable Services depends on having a hierarchy of nodes: the Server of Authority (SMW), tier1 nodes (boot and SDB nodes), tier2 nodes (designated service and repurposed compute nodes), and tier3 nodes (everything else). For more information, see [About Cray Scalable Services](#) on page 410.

Part of hardware planning is ensuring the correct ratio of tier3 nodes (clients) to tier2 nodes (servers). On a CLE 5.x system, the DSL nodes, which NFS-mounted the sharedroot and then DVS-projected it to compute nodes, are similar to but not the same as tier2 nodes. There may be enough DSL nodes from CLE 5.x to be reused as tier2 nodes, but more tier2 nodes may be required. See the tier2 node FAQ in [Information to Collect Before Installation](#) on page 19 for specific rules on how many tier2 nodes are required to support the number of tier3 nodes in this system and which type of nodes can be used as tier2 nodes. The other information in that topic is intended for a fresh install of CLE 6.0 / SMW 8.0; it is included as part of migration planning because migration using a spare SMW and boot RAID includes a fresh install of CLE 6.0 / SMW 8.0, and that comes next in the process.

Plan Workload Manager (WLM)

Before beginning the migration, check with the WLM vendor to find out what version of their product supports SLES 12 or both SLES 11 SP3 and SLES 12. Also find out if they have any documentation that describes what data can be backed up and restored. For example, Altair provides PBS Pro *Big Book* (available on the PBS Professional documentation site: <http://www.pbsworks.com/PBSProductGT.aspx?n=PBS-Professional&c=Overview-and-Capabilities&d=PBS-Professional,-Documentation>), which lists data that can be backed up and restored (see section 7.7 "Migration Upgrade Under Linux").

Prepare the Physical Migration SMW

To prepare the physical migration SMW by installing the CLE 6.0.UP03 / SMW 8.0.UP03 release on the spare SMW, do the following:

1. Collect the information specified in [Information to Collect Before Installation](#) on page 19.
2. Follow the installation procedures in [Preparation of Migration SMW and Boot RAID](#) on page 22.

This SMW and boot RAID will be used as the migration SMW for the rest of the migration process.

When that basic software installation has been completed, the migration process continues on to configure as much as possible without XC hardware connected, including installation of the PE software to an image root.

4.1 Information to Collect Before Installation

SMW Information

This information will be needed to update the global config set during configuration.

NOTE: (SMW HA only) Sites migrating an SMW HA system should collect this information from both SMWs of the SMW HA pair.

- Network base IP address for SMW eth0
- Netmask for SMW eth0
- Gateway IP address for SMW eth0
- List of IP addresses to use as DNS server
- List of domains to use in the DNS search path for hosts attached to SMW eth0 network
- List of NTP servers
- Host name of the SMW: both the short name and the fully qualified domain name (FQDN)
- IP address of SMW eth0

Hardware Information

When `xtdiscover` is used to discover XC system hardware, it will prompt for this information.

- Maximum cabinet size in the X dimension
- Maximum cabinet size in the Y dimension
- Network topology class (0 or 2 for Cray XC Series liquid-cooled systems, 0 for Cray XC Series air-cooled systems: XC30-AC, XC40-AC)
- Primary boot node (and alternate boot node if enabling boot node failover)
- Primary SDB node (and alternate SDB node if enabling SDB node failover)

Service Node Roles

The XC system being installed and configured must have service nodes designated to function in some or all of the following roles. A node may have more than one role (e.g., boot and tier1). The system at this site may not require all of these roles.

- boot
- SDB
- login
- tier1 (boot node and SDB node)
- tier2 (see Tier2 Node FAQ)
- LNet router to external Lustre server
- realm-specific IP (RSIP) nodes
- DataWarp-managed nodes with SSD hardware
- DataWarp API gateway nodes
- nodes providing a role for a workload manager (WLM)
- DVS servers to an external file system
- Direct-attached Lustre (DAL) MGS, MDS, or OSS nodes
- compute node repurposed to be a service node

Tier2 Node FAQ

- Q. How many tier2 nodes are needed?** **A.** At least one server must be provided, and for resiliency, two nodes placed on different blades. The recommended ratio of tier2 nodes (servers) to tier3 nodes (clients) is 1 to 400.
- Q. Will adding more tier2 nodes help performance?** **A.** Adding more tier2 nodes does not always yield additional performance and is subject to diminishing returns.
- Q. What kind of node can be used as a tier2 node?** **A.** Use these:
- OPTIMAL: dedicated repurposed compute nodes (RCN)
 - dedicated service nodes
 - nodes with uniform light to moderate load
 - nodes with relatively homogeneous single core speeds to reduce resource contention disparity during periods of partial availability
- AVOID these (will result in sub-optimal performance):
- nodes with slower individual CPU cores, such as Intel® Xeon Phi™ "Knights Landing" (KNL) processors
 - direct-attached Lustre (DAL) servers
 - RSIP (realm-specific IP) servers
 - login nodes
- Q. Can a tier2 node have more than one role?** **A.** Small test and development systems (TDS) may use tier2 nodes that have additional roles, but generally, it is better for tier2 nodes to be dedicated.
- Q. Where should tier2 nodes be placed?** **A.** Distribute nodes throughout the system (on different blades) for resiliency in the event of hardware failure.

Service Node Network Information

For each service node with a network interface, either Ethernet or InfiniBand, collect this information.

For each network defined:

- unique identifier for the network (management, login, lnet)
- description or notes about the network (e.g. "Network to external Lustre")
- network base IP address
- netmask
- gateway IP address

For each network interface added to a host

- unique identifier for each interface (primary_ethernet, eth0, eth1, eth2, eth3, ib0, ib1, etc.) on this host
- device name for the interface (eth0, ib1, etc.)
- description or notes about the nterface (e.g., "Ethernet connecting boot node to SMW")
- any host name aliases by which this node should be known
- name of the network to which this interface belongs (see list of networks defined above)
- IPv4 network address for the interface

5 Preparation of Migration SMW and Boot RAID

For this phase of the migration process, use the following procedures in the order listed to install the CLE 6.0.UP03 / SMW 8.0.UP03 release on the spare SMW (and boot RAID) that will serve as the migration SMW. These procedures are performed while the migration SMW is not connected to XC system hardware.

1. [Prepare for an SMW/CLE Fresh Install](#) on page 22
2. [Install the Base Operating System on the Migration SMW](#) on page 25
3. [Install the SMW and CLE Software](#)
4. [Configure Other Features and Services](#) on page 74

Use the checklists in [Checklists for Migration using a Physical Migration SMW](#) on page 431 to track progress while performing these procedures.

5.1 Prepare for an SMW/CLE Fresh Install

In preparation for a fresh install, do the following:

- Verify that everything needed has been saved, because this fresh install will wipe all SMW disks clean.
- Before trying to perform the migration, ensure either (1) access to the physical keyboard, mouse, and monitor of the migration SMW or (2) connection over iDRAC to the migration SMW if it is not physically present.
- Read the *SMW Release Errata* and the *SMW README* provided with the SMW release package for any additional installation-related requirements, corrections to this guide, and other relevant information about the release package.
- Read the *CLE Release Errata* and the *CLE README* provided with the CLE release package for any additional installation-related requirements, corrections to this guide, and other relevant information about the release package.
- Read the Field Notices (FN) related to kernel security fixes to identify any changes to this release package. Apply any needed changes before installing the new software.
- Verify that the network connections are in place (see [Network Connections](#) on page 22).
- Find out how much SMW internal disk space is needed (see [SMW Internal Disk Requirements](#) on page 23).
- Know which configuration values are site-specific and which are defaults (see [Configuration Values](#) on page 24).
- Be familiar with the default passwords used during the installation process (see [Passwords](#) on page 25).

5.1.1 Network Connections

The following network connections are required.

- A stand-alone SMW with a single quad-ethernet card has these private network connections:
 - eth0 - To the customer/management network
 - eth1 - To the hardware supervisory system (HSS) network
 - eth2 - Used for SMW HA (failover) heartbeat 1 network
 - eth3 - To the boot and SDB nodes (the admin network)
- An SMW configured for SMW failover (SMW HA) has a second quad-ethernet card with these connections:
 - eth4 - Used for SMW HA heartbeat 2 network
 - eth5 - Used for SMW HA distributed replicated block device (DRBD)
 - eth6 - Reserved for future use
 - eth7 - Reserved for future use

Things to note about network connections:

- Every XC system should have an Ethernet switch with a network for the SMW, boot, and SDB nodes (an "admin" network). Sites that did not connect the SDB node to this admin network in their CLE 5.2 / SMW 7.2 system must do so for migration to a CLE 6.0 / SMW 8.0 system.
- Ethernet port assignments are valid only after the SMW software installation completes.
- The SMW must have a Fibre Channel or serial attached SCSI (SAS) connection to the boot RAID.
- A boot node must have a Fibre Channel or SAS connection to the boot RAID. If boot node failover is enabled or there are multiple logical CLE partitions, then each boot node should have such a connection to the boot RAID.
- A service database (SDB) node must have a Fibre Channel or SAS connection to the boot RAID. If SDB node failover is enabled or there are multiple logical CLE partitions, then each SDB node should have such a connection to the boot RAID.

IMPORTANT: The SMW must be disconnected from the boot RAID before the initial installation of the SLES software.

IMPORTANT: Ensure that the Fibre Channel optic cable connectors or SAS cable connectors have protective covers when disconnected from the SMW, boot node, SDB node, or boot RAID.

5.1.2 SMW Internal Disk Requirements

Internal SMW disks are used for the boot disk (with optional RAID1 mirroring between two boot drives) and the power management disk (PMDISK).

The PMDISK requires a minimum of 500 GB. This may be a fresh disk or a repurposed disk on an existing SMW. The PMDISK will be allocated to `/var/lib/pgsql` in an ext4 file system.

The boot disk (or pair of boot disks in RAID1 configuration) requires a minimum of 160 GB, but may be larger. If a RAID1 mirror is enabled, the drives in the RAID1 configuration must be the same size. The boot disk has 4 GB allocated to `/boot` in an ext3 file system, 32 GB for swap, and the rest of the disk for the root (`/`) file system in a btrfs file system.

Table 2. SMW Internal Disk Requirements

Mount Point	FS Type	Disk	Size	Description
<code>/boot</code>	ext3	boot	4 GB	Boot information

Mount Point	FS Type	Disk	Size	Description
swap	swap	boot	32 GB	SMW swap
/	btrfs	boot	120+ GB	root file system of SMW with btrfs subvolumes
/var/lib/pgsql	ext4	power management	1000+ GB	Power Management disk

5.1.3 Configuration Values

The following IP addresses are set by default and are not site dependent.

Table 3. Default IP Addresses

IP Address	Description
10.1.0.1	Primary boot RAID controller
10.1.0.2	Secondary boot RAID controller
10.1.0.15	Storage RAID controller
10.1.1.1	SMW eth1 - HSS network
10.2.1.1	(SMW HA only) SMW eth2 - SMW HA heartbeat 1
10.3.1.1	SMW eth3 - admin network
10.3.1.253	SDB node
10.3.1.254	boot node
10.4.1.1	(SMW HA only) SMW eth4 - SMW HA heartbeat 2
10.5.1.1	(SMW HA only) SMW eth5 - SMW HA DRBD
127.0.0.1	localhost (loopback)

The following configuration values are site dependent. Record the actual values for the installation site in the third column.

Table 4. Site-dependent Configuration Values

Description	Example Value	Actual Value
SMW hostname	smw	
Domain	cray.com	
Aliases	cray-smw smw1	
Customer network IP address	192.168.78.68	
Customer network netmask	255.255.255.0	
Default gateway	192.168.78.1	

Description	Example Value	Actual Value
Domain names to search	us.cray.com mw.cray.com	
Nameserver IP address	10.0.73.30 10.0.17.16	
iDRAC hostname	cray-drac	
iDRAC IP address	192.168.78.69	
iDRAC Subnet Mask	255.255.255.0	
iDRAC Default GW	192.168.78.1	
Timezone	US/Central	
NTP servers	ntpghost1 ntpghost2	
X dimension	1-64	
Y dimension	1-32	
Topology Class	0, 2 (see note below)	

NOTE: Regardless of the number of cabinets in the system, Cray XC Series air-cooled systems must be set to topology class 0. Cray XC Series liquid-cooled systems can be topology class 0 or 2.

5.1.4 Passwords

The following default account names and passwords are used throughout the initial software installation process. Cray recommends changing these default passwords during the installation and configuration process at appropriate times before the SMW or network CLE nodes are connected to networks that are external to the XC system.

Table 5. Default System Passwords

Account Name	Password
root	initial0
crayadm	crayadm
mysql	None; a password must be created
root (iDRAC)	initial0

5.2 Install the Base Operating System on the Migration SMW

The base operating system must be installed on the SMW before the Cray SMW and CLE software release packages can be installed. Cray provides two rack-mount SMW models: the Dell PowerEdge™ R815 Rack Server and the Dell PowerEdge™ R630 Rack Server. Earlier desktop SMW hardware is not supported. The figure below shows an easy way to distinguish between the two rack-mount models when viewing them from the front.

Figure 1. Distinguishing Features of Dell R815 and R630 Servers



Dell R815: 2U high and 6 drive bays



Dell R630: 1U high and 8 drive bays

Continue the installation process with [Prepare to Install the Base Linux Distribution](#) on page 26.

5.2.1 Prepare to Install the Base Linux Distribution

About this task

A full initial installation begins with installing the base operating system. This procedure provides initial steps that are common to installing the base OS on both Dell R815 and R630 SMW models.

Procedure

1. Disconnect the SMW connection to the boot RAID.

Disconnect the data cables and place protective covers on the fibre optic cable connectors (if present).

2. Connect the SMW keyboard, monitor, and mouse.

Connect a keyboard, monitor, and mouse to the USB and monitor connectors on the SMW, if not already connected.

NOTE: Once the iDRAC has been configured, the keyboard, monitor, and mouse can be connected to the iDRAC for remote console activities instead of being directly connected to the SMW console.

As the next step in preparing to install the base OS, do one of the following, depending on the SMW model.

- For a Dell R630 SMW, first configure the SMW RAID, then configure the BIOS and iDRAC:
 1. [Configure the Dell R630 SMW RAID Virtual Disks](#) on page 32
 2. [R630 SMW: Change the BIOS and iDRAC Settings](#) on page 36
- For a Dell R815 SMW, just configure the BIOS and iDRAC:
 1. [R815 SMW: Change the BIOS and iDRAC Settings](#) on page 26

After the BIOS and iDRAC settings have been configured, all SMW internal disks that are not to receive the base operating system should be physically ejected from SMW internal disk drive bays.

5.2.1.1 R815 SMW: Change the BIOS and iDRAC Settings

Prerequisites

This procedure assumes that the SMW is disconnected from the boot RAID and connected to a keyboard, monitor, and mouse.

About this task

This procedure changes the system setup for a Dell R815 SMW: the network connections, remote power control, and the remote console. Depending on the server model and version of BIOS configuration utility, there could be minor differences in the steps to configure the system. For more information, refer to the documentation for the Dell server used at this site. Because Cray ships systems with most of the installation and configuration completed, some of the steps may have been done already.

For a Dell R630 SMW, see [R630 SMW: Change the BIOS and iDRAC Settings](#) on page 36.

Procedure

1. Remove SMW non-boot internal drives.

Eject all the internal disk drives from the SMW except for the primary boot disk in slot 0 and the secondary boot disk in slot 1.

2. Power up the SMW. When the BIOS power-on self-test (POST) process begins, **quickly press the F2 key** after the following messages appear in the upper-right of the screen.

```
F2 = System Setup
F10 = System Services
F11 = BIOS Boot Manager
F12 = PXE Boot
```

When the **F2** keypress is recognized, the **F2 = System Setup** line changes to **Entering System Setup**.

After the POST process completes and all disk and network controllers have been initialized, the BIOS **System Setup** menu appears.

3. Change system time.

The system time should be in UTC, not in the local timezone.

- a. Select **System Time** in the **System Setup** menu.

The hours will be highlighted in blue.

- b. Set the correct time.

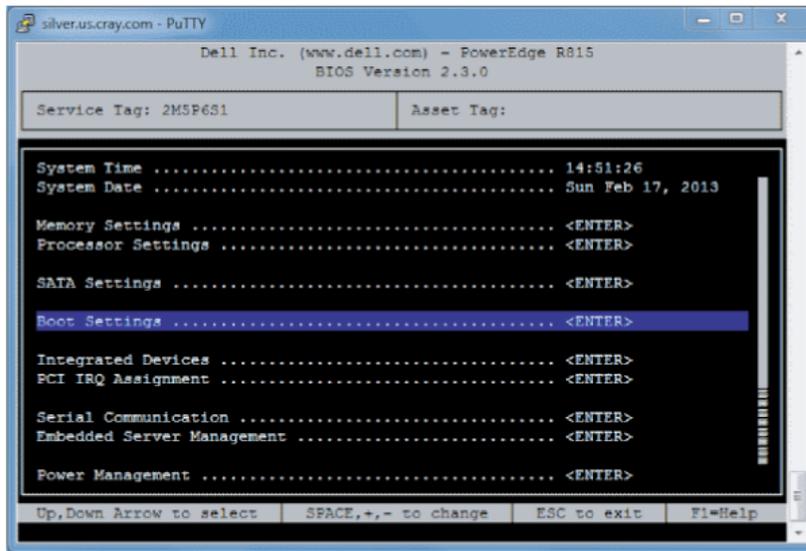
1. Press the space key to change hours.
2. Use the right-arrow key to select minutes, then change minutes with the space key.
3. Use the right-arrow key to select seconds, then change seconds with the space key.

- c. Press **Esc** when the correct time is set.

4. Change boot settings.

- a. Select **Boot Settings** in the **System Setup** menu, then press **Enter**.

Figure 2. Dell R815 SMW Boot Settings Menu



A pop-up menu with the following list appears:

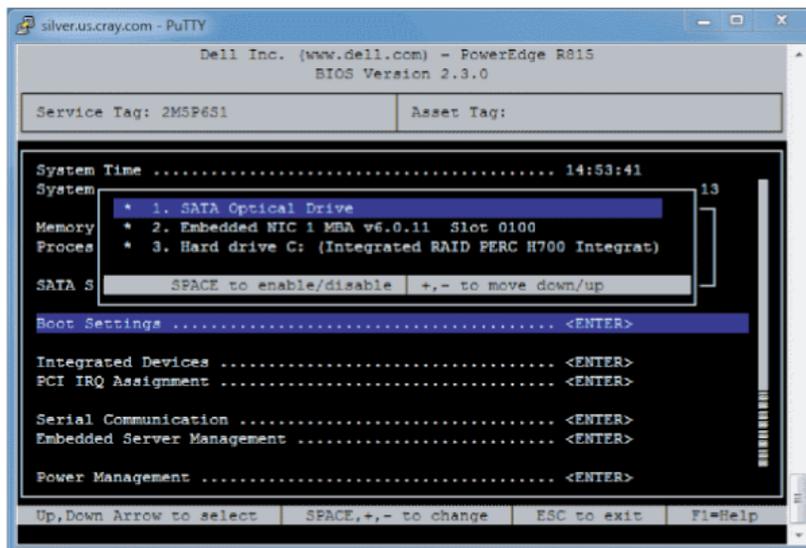
```

Boot Mode ..... BIOS
Boot Sequence ..... <ENTER>
USB Flash Drive Emulation Type..... <ENTER>
Boot Sequence Retry ..... <Disabled>

```

- b. Select **Boot Sequence**, then press **Enter**.

Figure 3. Dell R815 SMW Boot Sequence Settings



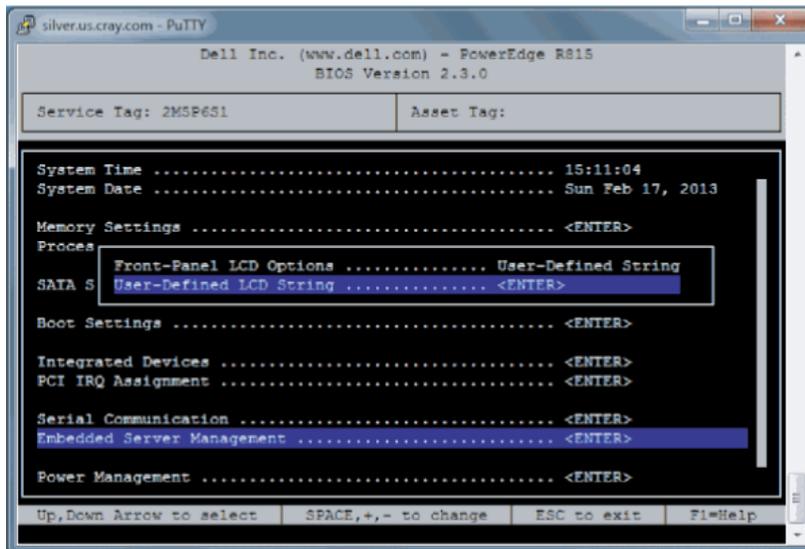
- c. Change the order of items in the **Boot Sequence** list so that the optical (DVD) drive appears first, then the hard drive. If **Embedded NIC** appears in the list, it should end up below the optical drive and hard drive in the list.

- d. Disable embedded NIC.

Select Embedded NIC and press **Enter**, then use the space key to disable it.

- e. Press **Esc** to exit the **Boot Sequence** menu.
 - f. Press **Esc** again to exit the **Boot Settings** menu.
5. Change serial communication.
 - a. Select **Serial Communication** in the **System Setup** menu, then press **Enter**.
 - b. Confirm these settings in the **Serial Communication** menu.
 - **Serial Communication** is set to **On with Console Redirection via COM2**
 - **Serial Port Address** is set to **Serial Device1=COM2, Serial Device2=COM1**
 - **External Serial Connector** is set to **Serial Device2**
 - **Failsafe Baud Rate** is set to **115200**
 - c. Press **Esc** to exit the **Serial Communication** menu.
 6. Select **Embedded Server Management** in the **System Setup** menu, then press **Enter**.

Figure 4. Dell R815 SMW Embedded Server Management Settings



- a. Set **Front-Panel LCD Options** to **User-Defined LCD String** in the **Embedded Server Management** menu. Use the space key to cycle through the choices, then use the down-arrow key.
 - b. Set **User-Defined LCD String** to the login hostname (e.g., cray-drac), then press **Enter**.
 - c. Press **Esc** to exit the **Embedded Server Management** menu.
7. Insert base operating system DVD into SMW.
 Insert the base operating system DVD labeled Cray-SMWbase12-201511021655 into the DVD drive. (The DVD drive on the front of the SMW may be hidden by a removable decorative bezel.)
 8. Save BIOS changes and exit.
 - a. Press **Esc** to exit the BIOS **System Setup** menu.
 A menu with a list of exit options appears.

Save changes and exit

Discard changes and exit
Return to Setup

- b. Ensure that **Save changes and exit** is selected, then press **Enter**.

The SMW resets automatically.

9. Enter BIOS boot manager.

- a. When the BIOS POST process begins again, **quickly press the F11 key** within 5 seconds of when the following messages appear in the upper-right of the screen.

```

          F2 = System Setup
          F10 = System Services
          F11 = BIOS Boot Manager
          F12 = PXE Boot

```

When the **F11** keypress is recognized, the **F11 = BIOS Boot Manager** line changes to **Entering BIOS Boot Manager**.

10. Change the integrated Dell Remote Access Controller (iDRAC) settings.

Watch the screen carefully as text scrolls until the **iDRAC6 Configuration Utility 1.57** line is visible. When the line **Press <Ctrl-E> for Remote Access Setup within 5 sec...** displays, press **Ctrl-E** within 5 seconds.

```

0 5 0 ATA WDC WD5000BPVT-0 1A01 465 GB
LSI Corporation MPT2 boot ROM successfully installed!
iDRAC6 Configuration Utility 1.57
Copyright 2010 Dell Inc. All Rights Reserved
iDRAC6 Firmware Revision version: 1.54.15
Primary Backplane Firmware Revision 1.07
-----
IPv6 Settings
-----
IPv6 Stack : Disabled
Address 1 : ::
Default Gateway : ::
-----
IPv4 Settings
-----
IPv4 Stack : Enabled
IP Address : 172. 31. 73.142
Subnet mask : 255.255.255. 0
Default Gateway : 172. 31. 73. 1
Press <Ctrl-E> for Remote Access Setup within 5 sec...

```

The **iDRAC6 Configuration Utility** menu appears.

11. Set **iDRAC LAN** to **ON**.

12. Configure the iDRAC LAN.

Select **LAN Parameters**, then press **Enter**.

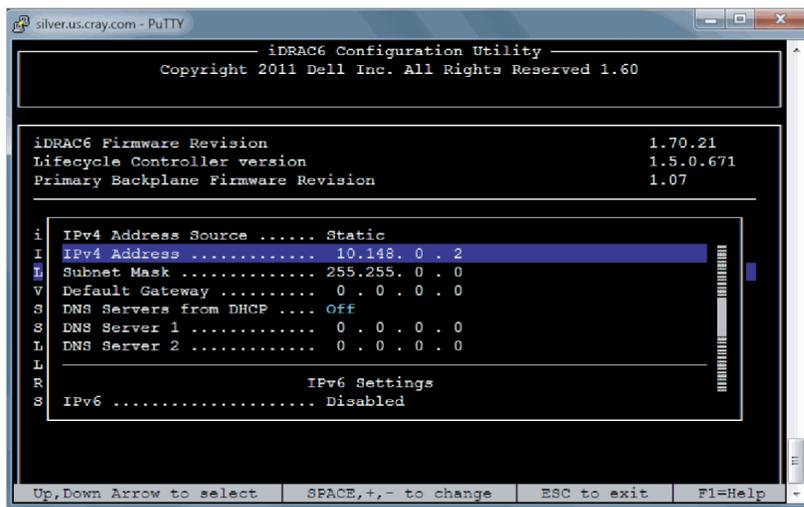
- a. Configure iDRAC6 name.

Use the arrow key to scroll down and select **iDRAC6 Name**, then press **Enter**. Enter a value for **Current DNS iDRAC6 Name** (e.g., smw-drac), then press **Enter**.

Trouble? If unable to set the iDRAC6 name, try this:

1. Temporarily set **Register iDRAC6 Name** to "On."
 2. Set **iDRAC6 Name**.
 3. Return to **Register iDRAC6 Name** and set it to "Off."
- b. Configure domain name.
- Use the arrow key to scroll down and select **Domain Name**, then press **Enter**. Enter a value for **Current Domain Name** (e.g., us.cray.com), then press **Enter**.
- c. Configure hostname string.
- Use the arrow key to scroll down and select **Host Name String**, then press **Enter**. Enter a value for **Current Host Name String** (e.g., smw-drac), then press **Enter**.
- d. Configure IPv4 settings.
- Use the arrow key to scroll down into the **IPv4 Settings** group and confirm that the **IPv4 Address Source** is set to **static**. Then enter values for the following:
- IPv4 Address** (the SMW DRAC IP address)
 - Subnet Mask** (the SMW iDRAC subnet mask)
 - Default Gateway** (the SMW iDRAC default gateway)
 - DNS Server 1** (the first site DNS server)
 - DNS Server 2** (the second site DNS server)

Figure 5. Dell R815 SMW DRAC IPv4 Parameter Settings



- e. Configure IPv6 settings.
- Use the arrow key to scroll down into the **IPv6 Settings** group and ensure that **IPv6** is disabled.
- f. Change the IPMI settings.
- Change the IPMI settings to enable the Serial Over LAN (SOL) console.
1. Use the arrow key to scroll down to the **IPMI SETTINGS** list.
 2. (Stand-alone SMW only) Ensure that **Enable IPMI over LAN** is NOT selected.
 3. (SMW HA only) Ensure that **Enable IPMI over LAN** is selected. This setting is used for both SMWs in an SMW HA pair.

4. Ensure that **Channel Privilege Level Limit** is set to **Administrator**.

g. Press **Esc** to exit **LAN Parameters** and return to the **iDRAC6 Configuration Utility** menu.

13. Configure iDRAC virtual media.

a. Select **Domain Name**, then press **Enter**.

b. Select **Virtual Media Configuration**, then press **Enter**.

c. Select the **Virtual Media** line and press the space key until it indicates **Detached**.

d. Press **Esc** to exit the **Virtual Media Configuration** menu.

14. Set the password for the iDRAC LAN root account.

Using the arrow keys, select **LAN User Configuration**, then press **Enter**. The following configuration is for both SSH and web browser access to the iDRAC.

a. Select **Account User Name** and enter the account name "root."

b. Select **Enter Password** and enter the intended password.

c. Select **Confirm Password** and enter the intended password again.

d. Press **Esc** to return to the **iDRAC6 Configuration Utility** menu.

15. Exit the iDRAC configuration utility.

a. Press **Esc** to exit the **iDRAC6 Configuration Utility** menu.

b. Select **Save Changes and Exit**.

The **BIOS Boot Manager** menu appears.

16. Choose to boot from SATA Optical Drive.

Using the arrow keys, select the **SATA Optical Drive** entry, then press **Enter**.

5.2.1.2 Configure the Dell R630 SMW RAID Virtual Disks

Prerequisites

This procedure assumes that the SMW is disconnected from the boot RAID and connected to a keyboard, monitor, and mouse.

About this task

Before installing and configuring SMW software, the base operating system needs to be installed on the SMW. And before the base operating system can be installed, the internal disk drives of the SMW must be configured as RAID virtual disks, as described in this procedure, and the default system setup for the R630 SMW node must be configured, as described in [R630 SMW: Change the BIOS and iDRAC Settings](#) on page 36.

A Dell R630 SMW has five physical disks. The SMW node must be reconfigured so that the internal Dell PERC RAID controller treats four of these disks as RAID 5 with a hot spare and the fifth disk as non-RAID. This procedure describes how to do that. Because Cray ships systems with most of the installation and configuration completed, some of the steps may be needed only if changes are made to the configuration.

This procedure includes detailed steps for the DELL R630 server using the PERC H330 Mini BIOS Configuration Utility 4.03-0010. Depending on the server model and version of RAID configuration utility, there could be minor differences in the steps to configure the system. For more information, refer to the documentation for the DELL PERC controller or server RAID controller software.

Procedure

1. Connect a keyboard, monitor, and mouse to the front panel USB and monitor connectors on the SMW, if not already connected.
2. Ensure all SMW internal disk drives are inserted into the SMW drive slots.
3. Power up the SMW. As the SMW node reboots, watch for the Power Edge Expandable RAID Controller section and be ready to press **Ctrl-R** when prompted.

Cray recommends using the RAID configuration utility (via **Ctrl-R**) to configure the RAID virtual disks instead of the **System Setup Device Settings** menu.

TIP: In the RAID configuration utility:

- Use the up-arrow or down-arrow key to highlight an item in a list.
- Press the **Enter** key to select an item.
- Press the **F2** key to display a menu of options for an item.
- Use the right-arrow, left-arrow, or **Tab** key to switch between the **Yes** and **No** buttons in a confirmation window.

4. Clear existing/default disk configuration, if necessary.

If any disk groups are currently defined:

- a. Select **Disk Group 0**, then press **F2**.
- b. Select **Delete Disk Group**, then press **Enter**.
- c. Select **Yes** in the pop-up confirmation window to confirm the changes.

5. Switch disk controller from HBA-Mode to RAID-Mode.

Some SMW hardware might be configured for HBA-Mode. If it is, then change it to RAID-Mode using the following substeps. If it is not, then skip these substeps.

- a. Switch disk controller from HBA-Mode to RAID-Mode.
 1. Press **Ctrl-N** (multiple times) to move to the **Ctrl Mgmt** tab.
 2. Press **Tab** (multiple times) to get to **Personality Mode**.
 3. Press **Enter** to see choice between **RAID-Mode** and **HBA-Mode**.
 4. Use the up-arrow or down-arrow key to select **RAID-Mode**, then press **Enter**.
 5. Press **Tab** (multiple times) to get to **Apply**, then press **Enter**. This message appears: "The operation has been performed successfully. Reboot the system for the change to take effect."
 6. Press **Enter**.
- b. Exit RAID configuration utility.
 1. Press **Esc** to exit the RAID configuration utility.

2. Select **OK** to confirm, then press **Enter**.

c. Reboot the SMW.

Press **Ctrl-Alt-Delete** at the prompt to reboot. The server will restart the boot process. Be prepared to press **Ctrl-R** when prompted.

d. Enter RAID configuration utility.

As the SMW node reboots, enter the RAID controller configuration utility by pressing **Ctrl-R** when prompted. This will return to the point prior to switching from HBA-Mode to RAID-Mode.

6. Configure most internal disks to appear as `/dev/sda` in a RAID-5 virtual disk.

This step configures most internal disks to appear as `/dev/sda` in a RAID-5 virtual disk. The R630 has four identical 500-GB drives and one 1-TB drive. The 1-TB drive will be excluded from this RAID-5 configuration. Instead, that drive will be used to hold the postgresql database with Power Management data.

a. Select **No Configuration Present**, then press the **F2** key.

b. Select **Convert to RAID capable**, then press **Enter**. The **Convert Non-RAID Disks to RAID capable** screen appears.

c. Create virtual disk `/sda`

7. Convert non-RAID disks to RAID-capable.

a. Press **Enter** to check the box for a physical disk, which selects it for this RAID-5 disk group. This action also advances the selection to the next disk. In this manner, select all four of the identical 500-GB drives but exclude the 1-TB drive (leave it unselected).

b. Press **Tab** to move to **OK**, then press **Enter**.

8. Verify the virtual disk changes.

To verify the virtual disk changes, compare settings with those shown in the figure.

9. Create virtual disk `sda`.

a. Use up-arrow to return to the **No Configuration Present!** item.

b. Press **F2** to see a pop-up menu.

c. Press **Enter** to choose **Create New VD**.

The **Convert Non - RAID Disks to RAID capable** screen appears. The only disk left on this screen should be the large (1-TB) disk which was excluded earlier. It should not be added to the RAID capable set of disks, so continue to exclude it.

d. Press **Tab** to move from the list of disks to **Cancel**, then press **Enter**.

This cancels the conversion of non-RAID disks to RAID capable. The **Create New VD** screen appears.

10. Create new virtual disk (VD).

a. Press **Enter** to switch from **RAID-0** to other options.

b. Use down-arrow to select **RAID-5**, then press **Enter**.

c. Press **Tab** to move to the **Physical Disks** area.

- d. Press **Enter** to select each disk except one.
One disk should not be selected so that it can become the hot spare (configured in the next step).
- e. Press **Tab** to move to **VD Name**.
- f. Select name sda.
- g. Press **Tab** to move to **Advanced**, then press **Enter**.
The **Create Virtual Disk-Advanced** screen appears.

11. Configure one disk as the hot spare.

- a. Press **Tab** multiple times to move to **Initialize**, then press **Enter** to select it.
A pop-up window with the following text appears: "Initialization will destroy data on the virtual disk. Are you sure you want to continue?"
- b. Press **Tab** or arrow keys to move to **OK**, then press **Enter** to confirm initialization.
- c. Press **Tab** to move to **Configure HotSpare**, then press **Enter** to select it.
- d. Press **Tab** or arrow keys to move to **OK** on the **Create Virtual Disk-Advanced** screen, then press **Enter**.
- e. Press **Tab** or arrow keys to move to **OK** on the **Create New VD** screen, then press **Enter**.
A pop-up window with the following text appears: "Virtual disk is successfully created and initialized."
- f. Press **Tab** or arrow keys to move to **OK**, then press **Enter**.
A pop-up window with the following text appears: "Dedicated Hotspare for Disk Group 0."
- g. Select the disk to be the hot spare, then press **Enter**.
- h. Press **Tab** or arrow keys to move to **OK**, then press **Enter**.
A pop-up window with the following text appears: "Initialization complete on VD 0."
- i. Press **Tab** or arrow keys to move to **OK**, then press **Enter**.
The RAID will initialize in the background.

12. Exit RAID configuration utility.

Exit the RAID configuration utility, reboot, and then begin installing the base operating system.

- a. Press the **Esc** key to exit the RAID configuration utility.
- b. Select **OK**, then press **Enter** to confirm.

13. Reboot the system.

A message appears that prompts to reboot.

ATTENTION: Only the disk drives configured to be the RAID-5 virtual disk sda should be inserted into the SMW internal drive bays when installing SLES 12.

- a. Eject the 1-TB disk (which was not added to the RAID-5 virtual disk sda) from the SMW.
This will be re-inserted when SLES 12 installation is complete.
- b. Press **Ctrl-Alt-Delete**.

The server will restart the boot process and will not interrupt RAID initialization. During the system reboot, be prepared to press **F2** when prompted, to change the system setup.

RAID configuration is now complete. The next step in preparing to install the base operating system is to configure the system setup for the R630 SMW node, as described in [R630 SMW: Change the BIOS and iDRAC Settings](#) on page 36.

5.2.1.3 R630 SMW: Change the BIOS and iDRAC Settings

Prerequisites

This procedure assumes that the internal disk drives of the SMW have just been configured as RAID virtual disks and the system is rebooting. If the system is not rebooting, press **Ctrl-Alt-Delete** to reboot.

About this task

This procedure describes how to change the system setup for the SMW: the network connections, remote power control, and the remote console. This procedure includes detailed steps for the Dell R630 server. Depending on the server model and version of BIOS configuration utility, there could be minor differences in the steps to configure the system. For more information, refer to the documentation for the Dell server used at this site. Because Cray ships systems with most of the installation and configuration completed, some of the steps may have been done already.

For a Dell R815 server, see [R815 SMW: Change the BIOS and iDRAC Settings](#) on page 26.

Procedure

Watch as the system reboots and the BIOS power-on self-test (POST) process begins. Be prepared to press **F2**, when prompted, to change the system setup.

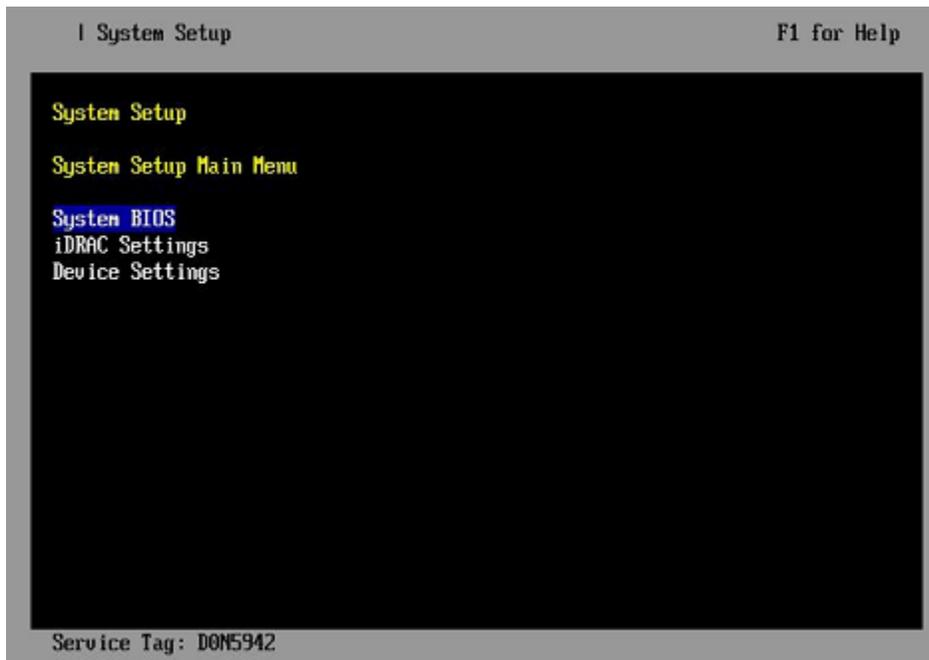
1. Press the **F2** key immediately after the following messages appear in the upper-left of the screen:

```
F2 = System Setup
F10 = System Services
F11 = BIOS Boot Manager
F12 = PXE Boot
```

When the **F2** keypress is recognized, the **F2 = System Setup** line changes color from white-on-black to white-on-blue.

After the POST process completes and all disk and network controllers have been initialized, the Dell **System Setup** screen appears. The following submenus are available on the **System Setup Main Menu** and will be used in subsequent steps: **System BIOS**, **iDRAC Settings**, and **Device Settings**.

Figure 6. Dell R630 System Setup Main Menu



TIP: In system setup screens,

- Use the **Tab** key to move to different areas on the screen.
- Use the up-arrow and down-arrow keys to highlight or select an item in a list, then press the **Enter** key to enter or apply the item.
- Press the **Esc** key to exit a submenu and return to the previous screen.

2. Change the BIOS settings.

- a. Select **System BIOS** on the **System Setup Main Menu**, then press **Enter**.

The **System BIOS Settings** screen appears.

Figure 7. Dell R630 System BIOS Settings Screen



b. Change Boot Settings.

1. Select **Boot Settings** on the **System BIOS Settings** screen, then press **Enter**. The **Boot Settings** screen appears.

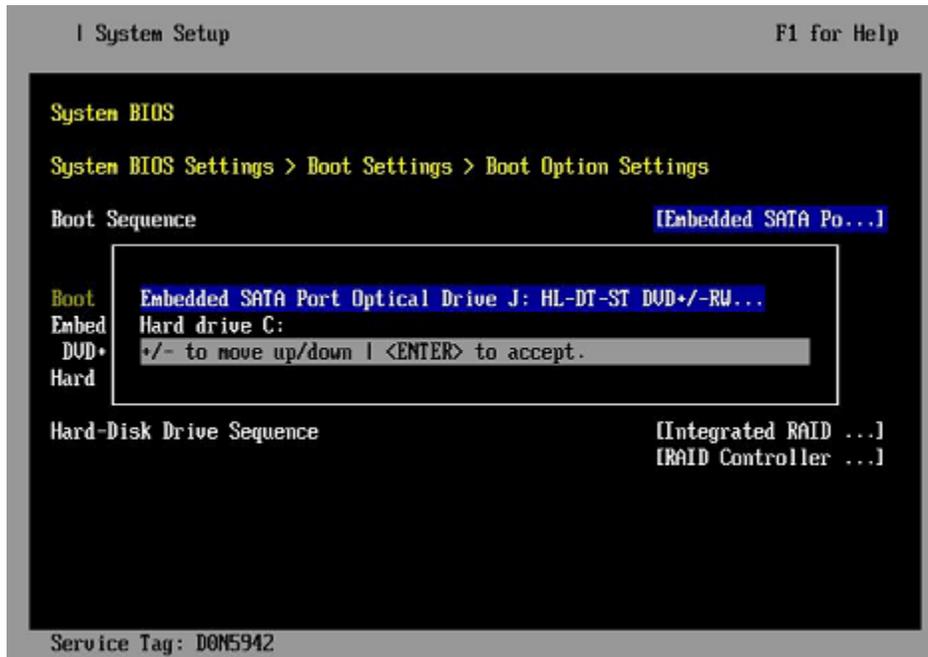
Figure 8. Dell R630 Boot Settings Screen



2. Ensure that **Boot Mode** is **BIOS** and not **UEFI**.
3. Select **Boot Option Settings**, then press **Enter**.

4. Select **Boot Sequence** on the **Boot Option Settings** screen, then press **Enter** to view a pop-up window with the boot sequence.

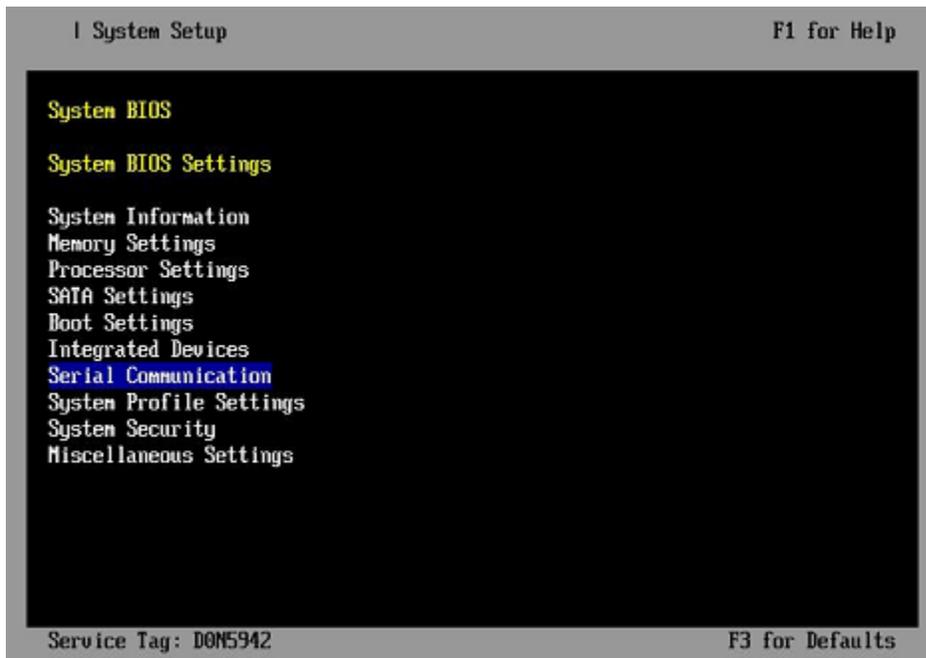
Figure 9. Dell R630 BIOS Boot Sequence



5. Change the boot order in the pop-up window so that the optical drive appears first, then the hard drive. If **Integrated NIC** appears in the list, it should end up below the optical drive and hard drive in the list.

TIP: Use the up-arrow or down-arrow key to highlight or select an item, then use the **+** and **-** keys to move the item up or down.
 6. Select **OK**, then press **Enter** to accept the change.
 7. Click the box next to **Hard drive C:** under the **Boot Option/Enable/Disable** section to enable it. Do the same for the optical drive, if necessary.
 8. Select **integrated NIC**, then press **Enter** to disable it.
 9. Press **Esc** to exit **Boot Option Settings**.
 10. Press **Esc** to exit **Boot Settings** and return to the **System BIOS Settings** screen.
- c. Change Serial Communication Settings.

Figure 10. Dell R630 System BIOS Settings: Serial Communication



1. Select **Serial Communication** on the **System BIOS Settings** screen. The **Serial Communication** screen appears.

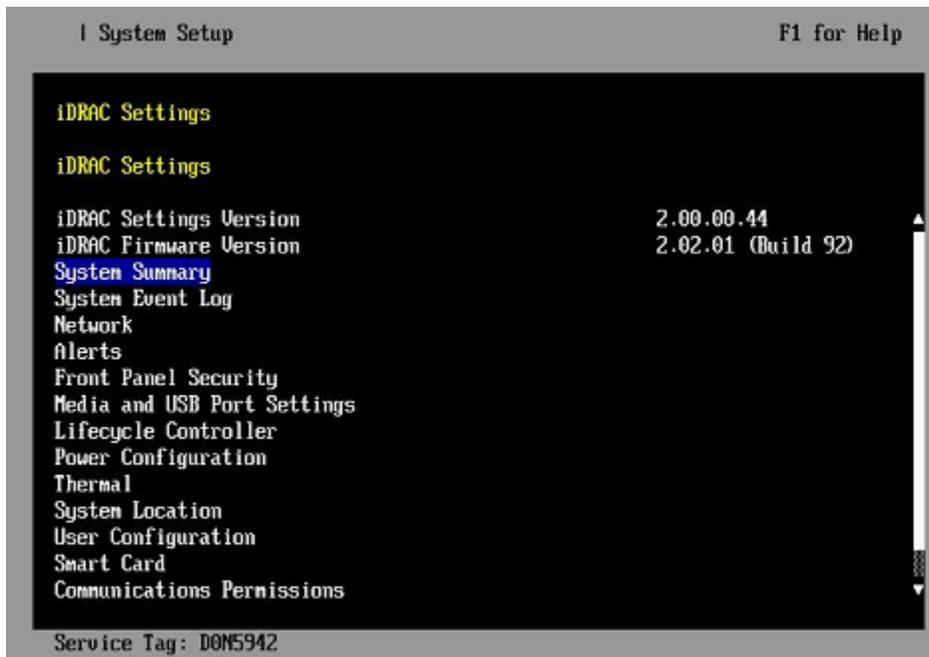
Figure 11. Dell R630 Serial Communication Screen



2. Select **Serial Communication** on the **Serial Communication** screen, then press **Enter**. A pop-up window displays the available options.
3. Select **On with Console Redirection via COM2** in the pop-up window, then press **Enter** to accept the change.

4. Select **Serial Port Address**, then select **Serial Device1=COM1, Serial Device2=COM2**, then press **Enter**.
 5. Select **External Serial Connector**, then press **Enter**. A pop-up window displays the available options.
 6. Select **Remote Access Device** in the pop-up window, then press **Enter** to return to the previous screen.
 7. Select **Failsafe Baud Rate**, then press **Enter**. A pop-up window displays the available options.
 8. Select **115200** in the pop-up window, then press **Enter** to return to the previous screen.
 9. Press the **Esc** key to exit the **Serial Communication** screen.
 10. Press **Esc** to exit the **System BIOS Settings** screen. A "Settings have changed" message appears.
 11. Select **Yes** to save changes. A "Settings saved successfully" message appears.
 12. Select **Ok**.
3. Change the iDRAC (Integrated Dell Remote Access Controller) settings.
Select **iDRAC Settings** on the **System Setup Main Menu**, then press **Enter**.
The **iDRAC Settings** screen appears.

Figure 12. Dell R630 iDRAC Settings Screen



4. Change the iDRAC network.
 - a. Select **Network** to display a long list of network settings.
 - b. Change the DNS DRAC name.
Use the arrow key to scroll down to **DNS DRAC Name**, then enter an iDRAC hostname that is similar to the SMW node hostname (e.g., cray-drac).
 - c. Change the static DNS domain name.

Use the arrow key to scroll down to **Static DNS Domain Name**, then enter the DNS domain name and press **Enter**.

d. Change the IPv4 settings.

Use the arrow key to scroll down to the **IPV4 SETTINGS** list.

1. Ensure that IPv4 is enabled.
 - a. If necessary, select **Enable IPV4**, then press **Enter**.
 - b. Select **<Enabled>** in the pop-up window, then press **Enter** to return to the previous screen.
2. Ensure that DHCP is disabled.
 - a. If necessary, select **Enable DHCP**, then press **Enter**.
 - b. Select **<Disabled>** in the pop-up window, then press **Enter** to return to the previous screen.
3. Change the IP address.
 - a. Select **Static IP Address**.
 - b. Enter the IP address of the iDRAC interface (`ipmi0`) for the SMW, then press **Enter**.
4. Change the gateway.
 - a. Select **Static Gateway**.
 - b. Enter the appropriate value for the gateway of the network to which the iDRAC is connected, then press **Enter**.
5. Change the subnet mask.
 - a. Select **Subnet Mask**.
 - b. Enter the subnet mask for the network to which the iDRAC is connected (such as `255.255.255.0`), then press **Enter**.
6. Change the DNS server settings.
 - a. Select **Static Preferred DNS Server**, enter the IP address of the primary DNS server, then press **Enter**.
 - b. Select **Alternate DNS Server**, enter the IP address of the alternate DNS server, then press **Enter**.

e. Change the IPMI settings.

Change the IPMI settings to enable the Serial Over LAN (SOL) console.

1. Use the arrow key to scroll down to the **IPMI SETTINGS** list.
2. Ensure that **Enable IPMI over LAN** is selected.

TIP: Use the left-arrow or right-arrow to switch between two settings.

3. Ensure that **Channel Privilege Level Limit** is set to **Administrator**.

f. Exit Network screen.

Press the **Esc** key to exit the **Network** screen and return to the **iDRAC Settings** screen.

5. Change hostname in iDRAC LCD display.

Change front panel security to show the hostname in LCD display.

- a. Use the arrow key to scroll down and highlight **Front Panel Security** on the **iDRAC Settings** screen, then press **Enter**.

- b. Select **Set LCD message**, then press **Enter**.
- c. Select **User-Defined String**, then press **Enter**.
- d. Select **User-Defined String**, then enter the SMW hostname and press **Enter**.
- e. Press the **Esc** key to exit the **Front Panel Security** screen.

6. (Optional) Change the iDRAC **System Location** fields.

Change the **System Location** configuration on the **iDRAC Settings** screen to set any of these fields: **Data Center Name**, **Aisle Name**, **Rack Name**, and **Rack Slot**.

7. Configure iDRAC virtual media.

- a. Select **Domain Name**, then press **Enter**.
- b. Select **Virtual Media Configuration**, then press **Enter**.
- c. Select the **Virtual Media** line and press the space key until it indicates **Detached**.
- d. Press **Esc** to exit the **Virtual Media Configuration** menu.

8. Set the password for the iDRAC root account.

- a. Use the arrow key to highlight **User Configuration** on the **iDRAC Settings** screen, then press **Enter**.
- b. Confirm that User Name is root. Select **User Name**, then enter the "root" user name.
- c. Select **Change Password**, then enter a new password.
- d. Reenter the new password in the next pop-up window to confirm it (the default password is "calvin").
- e. Press the **Esc** key to exit the **User Configuration** screen.

9. Exit iDRAC settings.

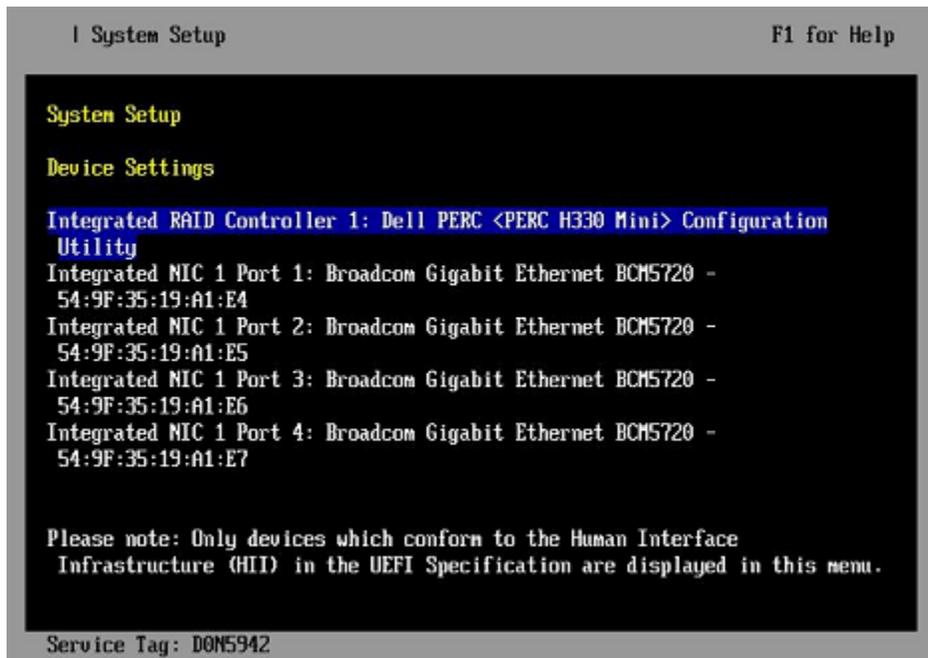
- a. Press the **Esc** key to exit the **iDRAC Settings** screen.
A "Settings have changed" message appears.
- b. Select **Yes**, then press **Enter** to save the changes.
A "Success" message appears.
- c. Select **Ok**, then press **Enter**.
The main screen (**System Setup Main Menu**) appears.

10. Change device settings.

These steps disable an integrated NIC device by changing the setting for the integrated NIC on a port from **PXE** to **None**.

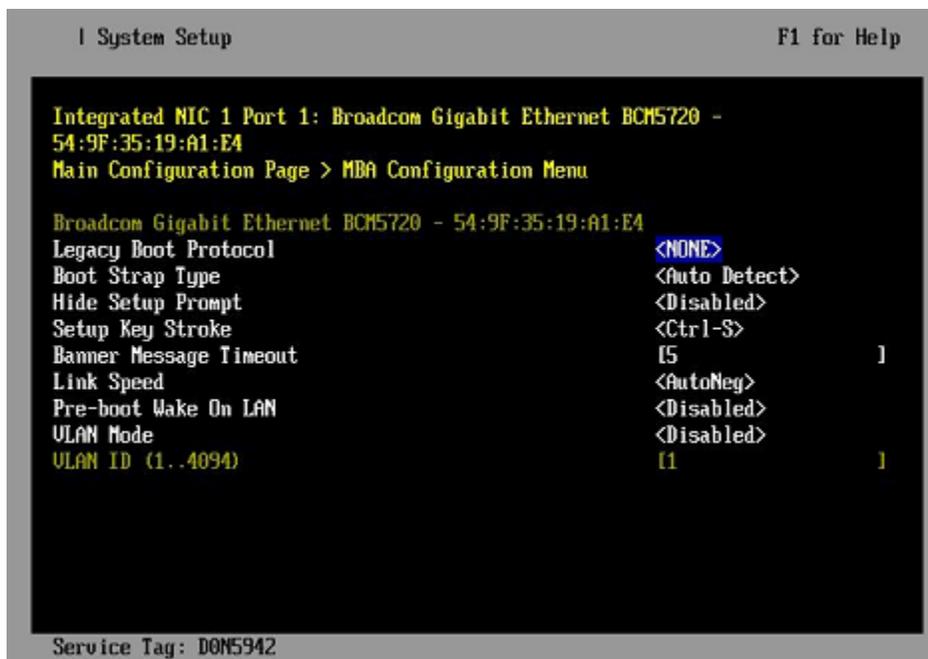
- a. Change Integrated NIC 1 Port 1
 1. Select **Device Settings** on the **System Setup Main Menu**, then press **Enter**. The **Device Settings** screen appears.

Figure 13. Dell R630 Device Settings Screen



2. Select **Integrated NIC 1 Port 1: ...** on the **Device Settings** screen, then press **Enter**.
3. Select **MBA Configuration Menu** on the **Main Configuration Page** screen, then press **Enter**.

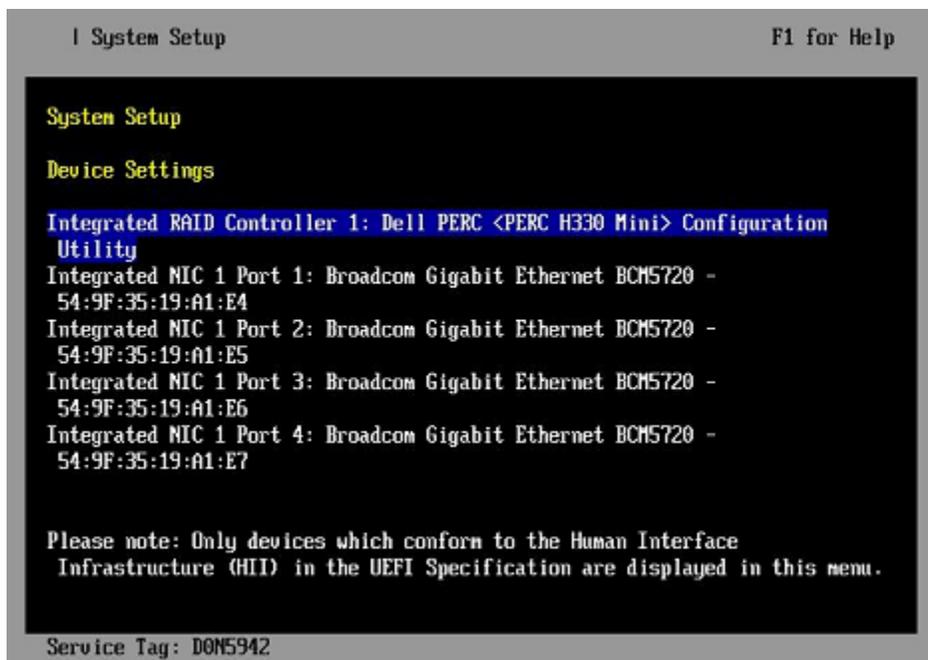
Figure 14. Dell R630 BIOS MBA Configuration Settings



4. Select **Legacy Boot Protocol** on the **MBA Configuration Menu** screen, use the right-arrow or left-arrow key to highlight **None**, then press **Enter**.
5. Press the **Esc** key to exit the **MBA Configuration Menu** screen.

6. Press **Esc** to exit the **Main Configuration Page** screen. A "Warning Saving Changes" message appears.
 7. Select **Yes**, then press **Enter** to save the changes. A "Success" message appears.
 8. Select **OK**, then press **Enter**. The **Device Settings** screen appears.
 9. Press **Esc** to exit the **Device Settings** screen. A "Settings have changed" message appears.
 10. Select **Yes**, then press **Enter** to save the changes. A "Settings saved successfully" message appears.
 11. Select **OK**, then press **Enter**. The main screen (**System Setup Main Menu**) appears.
- b. Change Integrated NIC 1 Port 2
1. Select **Device Settings** on the **System Setup Main Menu**, then press **Enter**. The **Device Settings** screen appears.

Figure 15. Dell R630 Device Settings Screen



2. Select **Integrated NIC 1 Port 2: ...** on the **Device Settings** screen, then press **Enter**.
3. Select **MBA Configuration Menu** on the **Main Configuration Page** screen, then press **Enter**.

Figure 16. Dell R630 BIOS MBA Configuration Settings



4. Select **Legacy Boot Protocol** on the **MBA Configuration Menu** screen, use the right-arrow or left-arrow key to highlight **None**, then press **Enter**.
5. Press the **Esc** key to exit the **MBA Configuration Menu** screen.
6. Press **Esc** to exit the **Main Configuration Page** screen. A "Warning Saving Changes" message appears.
7. Select **Yes**, then press **Enter** to save the changes. A "Success" message appears.
8. Select **OK**, then press **Enter**. The **Device Settings** screen appears.
9. Press **Esc** to exit the **Device Settings** screen. A "Settings have changed" message appears.
10. Select **Yes**, then press **Enter** to save the changes. A "Settings saved successfully" message appears.
11. Select **OK**, then press **Enter**. The main screen (**System Setup Main Menu**) appears.

5.2.2 Install the SLES 12 Base Linux Distribution on the Migration SMW

Prerequisites

This procedure assumes the following:

- The BIOS and iDRAC settings have just been changed on the SMW and it is restarting the boot process.
- All SMW internal disks that are not to receive the operating system are physically ejected from SMW internal disk drive bays.
- All connections to the boot RAID are unplugged so that no disk devices from the boot RAID will inadvertently lose existing data or receive the operating system.

About this task

This procedure describes the base operating system installation process. It provides detailed instructions for installing SLES 12 on the SMW (both Dell R815 and R630 models); configuring the SMW; and performing final steps: reconnect cables, reinsert drives, and reboot the SMW. To install the base operating system, use the DVD labeled Cray-SMWbase12-201511021655, which contains SUSE Linux Enterprise Server version 12 (SLES 12).

Procedure

SLES 12 SOFTWARE PACKAGE INSTALLATION

1. Select one of the **Cray SMW Initial Install** options.

Within 10 to 15 seconds after this **SUSE Linux Enterprise Server** boot menu displays, use the arrow key to scroll down and select one of the install options, then press **Enter**.

```
- Boot from Hard Disk
- Cray SMW Initial Install with software RAID1
- Cray SMW Initial Install without software RAID
- Rescue System
- Check Installation Media
- Firmware Test
- Memory Test
```

Select the option that is best for the SMW model:

For a Dell R815 SMW Select **Cray SMW Initial Install with software RAID1**, a mirrored boot disk option, which creates a software RAID1 mirror on the first two drives. This option is best for a Dell R815 because the R815 should use two disk drives to become the software RAID1 mirror.

For a Dell R630 SMW Select **Cray SMW Initial Install without software RAID**, a non-mirrored boot disk option, for servers with a single disk or virtual disk. This option is best for a Dell R630 because the R630 should have the internal RAID controller configured to present four disk drives as a virtual disk.



WARNING: If the selection is not made in time, the system will boot from the default selection, which is **Boot from Hard Disk**. If that happens, shut down the SMW, then start the power-up sequence again.

Note: The upper left corner of the installation screen has a date/time stamp for when the bootable SLES 12 DVD was created.

As the base installation progresses, the following phases appear on the screen:

```
Starting ... Loading Linux kernel
Initializing
Preparing System for Automated Installation
Initializing the Installation Environment
System Probing
Installation Settings
```

2. Review installation settings while the installation pauses on the **Installation Settings** screen.
3. Confirm the language for the SMW.

English (US) is the primary language by default. To change the primary language:

- a. Select the **Language** heading in the **Installation Settings** screen.
The **Languages** window opens.
 - b. Select a language (or multiple languages) from the drop-down menu, then select **Accept** at the bottom of the window.
4. Begin automated install.
- a. On the **Installation Settings** screen, select **Install**.
The **Confirm Installation** pop-up window appears.
 - b. Select **Install**.

The installation of software packages runs for approximately 20 minutes. The process automatically reboots the SMW from the hard disk, and the installation process continues with system configuration.

SYSTEM CONFIGURATION

5. Log in to SMW as root.
- When the login screen is displayed with the `crayadm` account as the account which will be logged in:
- a. Select **Not listed?**, then enter `root` for the username.
 - b. Either press **Enter** or select **Sign In**.
 - c. Enter the password for root.

6. Change default passwords on the SMW by executing the following commands.

The SMW contains its own `/etc/passwd` file that is separate from the password file for the rest of the CLE system.

```
smw# passwd root
```

```
smw# passwd crayadm
```

```
smw# passwd mysql
```

7. Change the SMW local time zone, if needed.

The default time zone on the SMW is **America/Chicago**. To change it:

- a. Execute this command:

```
smw# yast2 timezone
```

`yast2` opens a new window for changing the time zone, then a pop-up window appears with this message: "file `/etc/ntp.conf` has been changed manually. YaST might lose some of the changes."

- b. Select the **Do not show this message anymore** checkbox, then select **Continue**.
- c. Choose the time zone either by selecting a region on the map or by using the drop-down menus for **Region** and **Time Zone**.
- d. Select **Other Settings** if the time is incorrect, then select the **Manually** radio button and enter **Current Time** and **Current Date**. Select **Accept** when done.

- e. Select **OK** when done with time zone settings.

8. Configure the SMW firewall.

The SUSE firewall settings may need to be adjusted to match site firewall policy and to customize for site IP addresses. These steps enable and configure the firewall.

TIP: It is not necessary to shut down the system before performing this task.

- a. Save the SUSE firewall configuration.

Before modifying the SUSE firewall settings, make a copy of the configuration file.

```
smw# cp -p /etc/sysconfig/SuSEfirewall12 /etc/sysconfig/SuSEfirewall12.orig
```

- b. Check current firewall settings.

Check current firewall settings and change to support any site requirements. During the process of configuring Cray SMW and CLE software, some of the firewall settings may be adjusted. SSH access is one of the protocols permitted through the firewall from the external network to the SMW.

```
smw# iptables -L
smw# vi /etc/sysconfig/SuSEfirewall12
```

- c. Start the firewall immediately.

Invoke the modified configuration.

```
smw# systemctl start SuSEfirewall12_init.service
smw# systemctl start SuSEfirewall12.service
```

- d. Ensure that the firewall will start at next boot.

Execute the following commands to start the firewall at boot time.

```
smw# systemctl enable SuSEfirewall12_init.service
smw# systemctl enable SuSEfirewall12.service
```

- e. Verify firewall changes.

Verify the changes to the `iptables`.

```
smw# iptables -nvL
```

9. Configure LAN on the SMW.

Set network configuration for `eth0` and the hostname for the SMW.

- a. Execute this command:

```
smw# yast2 lan
```

The **Network Settings** screen appears with the **Overview** tab highlighted.

- b. Select the **eth0** line on the **Overview** tab, then select **Edit**.

The **Network Card Setup** screen appears with the **Address** tab highlighted.

- c. Select **Statically Assigned IP address** on the **Address** tab and enter values for IP address, subnet mask, and hostname (including the domain name). Then select **Next**.
- d. Select the **Hostname/DNS** tab on the **Network Settings** screen.

1. For the **Hostname and Domain Name** area, enter Hostname and Domain Name.
 2. For the **Name Servers and Domain Search List**, enter Name Server 1, Name Server 2, Name Server 3, and Domain Search.
- e. Select the **Routing** tab on the **Network Settings** screen, then enter the Default IPv4 Gateway (for the network connected to eth0) and set Device to eth0 using the dropdown menu.
- f. Click **OK** after all of the **Network Settings** have been prepared.

FINAL STEPS

10. Reconnect boot RAID disk cables.

Remove the protective covers from the Fibre Channel or SAS cable connectors, clean the ends of the cable connectors, and reconnect the data cables that connect the SMW to the boot RAID.

11. Reinsert SMW non-boot internal drives.

Reinsert all of the SMW internal disk drives that were removed earlier.

TIP: It is not necessary to turn off the power for the SMW before inserting these drives—the operating system can be in a booted state.

12. Eject the Cray-SMWbase12 DVD.

If the base operating system DVD (Cray-SMWbase12-201511021655) is still in the DVD drive, eject it.

```
smw# eject
```

13. Reboot the SMW.

Reboot the SMW to allow the SMW to discover the drives properly.

```
smw# reboot
```

If the SMW was configured with RAID1, then it may still be synchronizing the data between the two disks in the RAID1 mirror. The resync can take about 30 minutes when SLES 12 is freshly installed. If the SMW is rebooted at this point in the process, that resync will be interrupted. However, that is not a problem because as soon as the SMW is up again, the resync process will continue.

(R815 SMW only) To check the status of any RAID1 resync activities on an R815 SMW, look at `/proc/mdstat`.

In this example, the resync of md127 finishes in 24.3 minutes.

```
smw# cat /proc/mdstat
Personalities : [raid1]
md125 : active raid1 sdc2[1] sda2[0]
      33559424 blocks super 1.0 [2/2] [UU]
      bitmap: 0/1 pages [0KB], 65536KB chunk

md126 : active raid1 sda1[0] sdc1[1]
      4200384 blocks super 1.0 [2/2] [UU]
      bitmap: 0/1 pages [0KB], 65536KB chunk

md127 : active raid1 sda3[0] sdc3[1]
      206437248 blocks super 1.0 [2/2] [UU]
      [=====>.....] resync = 33.7% (69700352/206437248)
      finish=24.3min speed=93748K/sec
      bitmap: 2/2 pages [8KB], 65536KB chunk
```

```
unused devices: <none>
```

- For a stand-alone SMW or the first SMW in an SMW HA system, the next step in the installation and configuration of the SMW base operating system is [Configure Boot RAID Devices](#) on page 51.
- (SMW HA only) For the second SMW in an SMW HA system, there is no need to configure the boot RAID because it is shared with the first SMW and has already been configured. The next step in the process is [Make a Snapshot Manually](#) on page 57.

5.2.3 Configure Boot RAID Devices

In typical system installations, the RAID provides the storage for file systems used by the SMW, boot node, and SDB node. These file systems are prepared from LVM volumes in LVM volume groups using the physical volumes that are created on the RAID LUNs (logical unit numbers) or volumes. RAID units also provide user and scratch space and can be configured to support a variety of file systems. For more information about configuring RAID, see *XC™ Series Lustre® Administration Guide (S-2648)*, which is provided with the CLE release package.

Prerequisites and Assumptions for Configuring the Boot RAID

Sites that require a long distance between the SMW, XC, and the boot RAID will use Fibre Channel (FC) components, while sites that have the SMW, XC, and boot RAID in the same area (within 10 meters) will typically use SAS as the interface for the boot RAID.

- The SMW has an Ethernet connection to the Hardware Supervisory System (HSS) network.
- The SMW has a Fibre Channel (FC) or Serial Attached SCSI (SAS) connection to the boot RAID or to an FC or SAS switch.
- The boot nodes have an FC or SAS connection to the boot RAID or to an FC or SAS switch.
- The SDB nodes have an FC or SAS connection to the boot RAID or to an FC or SAS switch.

Boot RAID Configuration Procedures

Cray provides support for system boot RAID from NetApp, Inc.

NOTE: Cray ships systems with much of this software installed and configured. Performing all of the steps in these boot RAID procedures may not be necessary unless the configuration needs to be changed.

1. Configure the boot RAID for a NetApp, Inc. storage system using the following procedures (reference [Recommended Boot RAID LUN Values](#) on page 51 as needed). The first one installs the SANtricity Storage Manager Utility, which is used to perform the other procedures.
 - a. [Install SANtricity Storage Manager for NetApp, Inc. Devices](#) on page 53
 - b. [Set Up Boot RAID Space for Direct-attached Lustre](#)
 - c. [Create Boot RAID Volume Group and Volumes for NetApp, Inc. devices](#) on page 54
2. [Reboot the SMW and Verify LUNs are Recognized](#) on page 57

5.2.3.1 Recommended Boot RAID LUN Values

The recommended boot RAID LUN configuration is shown in these tables for different sizes of boot RAID: 4.5 TB, 9.0 TB, and 1.5 TB.

Boot RAID with 4.5 TB Available, Non-partitioned System

For a boot RAID with 4.5 TB available, use these values for a non-partitioned system. This is the default configuration installed in the factory.

LUN	Label	Size	Segment Size
0	smw0	3000 GB	256 KB
1	boot0	1000 GB	256 KB
2	sdb0	200 GB	256 KB

Boot RAID with 4.5 TB Available, Multiple Partitions

For a boot RAID with 4.5 TB available, use these values for a system with multiple CLE partitions.

- There must be one SMW LUN for the entire system with a size of at least 1000GB.
- There must be one boot LUN for each partition with a size of at least 500GB.
- There must be one SDB LUN for each partition with a size of at least 100GB.

This table shows example values for three CLE partitions.

LUN	Label	Size	Segment Size
0	smw1	2500 GB	256 KB
1	boot1	500	256 KB
2	sdb1	100 GB	256 KB
3	boot2	500 GB	256 KB
4	sdb2	100 GB	256 KB
5	boot3	500 GB	256 KB
6	sdb3	100 GB	256 KB

Boot RAID with 9.0 TB Available, Non-partitioned System

For a boot RAID with 9.0 TB available, use these values for a non-partitioned system. Values for boot1 and sdb1 LUNs are shown also, because they can be added to volume groups for the boot node volume group and SDB node volume group, if needed. If added, they should be the same size as the boot0 and sdb0.

LUN	Label	Size	Segment Size
0	smw0	4000 GB	256 KB
1	boot0	1000	256 KB
2	sdb0	200 GB	256 KB

LUN	Label	Size	Segment Size
3	boot1	1000 GB	256 KB
4	sdb1	200 GB	256 KB

Boot RAID with 1.5 TB Available, Non-partitioned System

For a boot RAID with only 1.5 TB available, use these values for a non-partitioned system.

LUN	Label	Size	Segment Size
0	smw0	1000 GB	256 KB
1	boot0	400 GB	256 KB
2	sdb0	100 GB	256 KB

5.2.3.2 Install SANtricity Storage Manager for NetApp, Inc. Devices

About this task

The SANtricity Storage Manager software is generally preinstalled and the SANtricity media is shipped with the system. If the SANtricity software is installed, then the `SMclient` executable will be found in `/opt/SMgr/client`. If this Cray system does not have the software installed on the SMW, install it using this procedure.

Procedure

1. Prepare X Windows for NetApp SANtricity Storage Manager.

The NetApp installation software will launch an X Windows application, so an X Windows server must be ready. There are many ways to prepare this: logging into SMW console as root, logging into SMW console as `crayadm` and then becoming root, or logging into SMW from a remote workstation with X Windows port forwarding enabled via `ssh`.

- If already logged in to the SMW as `crayadm`, `su` to root and enable X Windows port forwarding:

```
crayadm@smw> su -
smw# ssh -X localhost
```

- If not already logged on to the SMW, log in and enable X Windows port forwarding like this:

```
user@host> ssh -X root@smw
```

2. Copy NetApp SANtricity Storage Manager installer to SMW.

- If installing from the SANtricity Storage Manager CD, insert it into the SMW CD drive and mount the CD.

```
smw# mount /dev/cdrom /media/cdrom
smw# mkdir -p /tmp/netapp
smw# cp -p /media/cdrom/SMIA-LINUX64-11.25.0A00.0016.bin /tmp/netapp
smw# umount /media/cdrom
smw# eject
```

- If installing from the `SMIA-LINUX64-11.25.0A00.0016.bin` file, copy that file to `/tmp/netapp`.

```
smw# mkdir -p /tmp/netapp
smw# cp ./SMIA-LINUX64-11.25.0A00.0016.bin /tmp/netapp
```

3. Run the NetAPP SANtricity Storage Manager installer.

```
smw# /tmp/netapp/SMIA-LINUX64-11.25.0A00.0016.bin
```

The **SANtricity Storage Manager Introduction** window displays. The following substeps provide guidance through the installation, but the exact steps may differ for newer versions of the NetApp software.

- a. Select **Next** in the **SANtricity Storage Manager Introduction** window.

The **License Agreement** window displays.

- b. Select **I accept the terms of the License Agreement**, then select **Next**.

The **Select Installation Type** window displays.

- c. Select **Typical (Full Installation)**, then select **Next**.

The **Multi-Pathing Driver Warning** window displays.

- d. Select **OK**.

The **Pre-Installation Summary** window displays.

- e. Select **Install**.

The **Installing SANtricity** window displays and shows the installation progress. When the installation completes, an **Install Complete** window displays.

- f. Select **Done** to acknowledge and finish.

The SANtricity client, `SMclient`, is installed in `/opt/SMgr/client`.

4. Enable `crayadm` to run `SMclient`.

To be able to execute `SMclient` from the `crayadm` account, change the ownership and permissions for the executable files. If this step is skipped, only the `root` account will be able to run `SMclient`.

```
smw# chown crayadm /opt/SMgr
smw# chmod 775 /opt/SMgr
smw# chmod 755 /opt/SMgr/client/SMcli /opt/SMgr/client/SMclient
smw# chown -R crayadm:crayadm /var/opt/SM
smw# chmod -R ug+w /var/opt/SM
```

5.2.3.3 Create Boot RAID Volume Group and Volumes for NetApp, Inc. devices

Prerequisites

This procedure assumes the following:

- the SANtricity Storage Manager has been installed
- the user is logged on to the SMW as `crayadm`

About this task

This procedure creates the 8+2 Volume Group and 2 Global Hot Spares for a 4.5 TB Volume Group (the amount of storage for this installation may be different). A standard new boot RAID has 2 hot spares; the number of hot spares depends on the number of available drives left over after configuration of the 8+2 RAID6.

Procedure

1. Start the SANtricity Storage Manager.

```
crayadm@smw> /usr/bin/SMclient
```

The SANtricity Storage Manager window appears.

2. Select a method for adding a volume group.

If the **Select Addition Method** window appears, choose one of the following options. Otherwise, continue with the next step.

- **Automatic.** Select this option if a serial connection was not used to assign IP addresses to the storage array controllers. The SANtricity software automatically detects the available controllers, in-band, using the Fibre Channel link.
- **Manual.** Select this option if IP addresses have already been assigned to the storage array controllers.

3. Create a volume group.

The following substeps apply only if the **Select Addition Method** window did not display or if the **Manual** option was selected.

- a. Double-click the name for the storage array to be configured.
The **Array Management** window displays.
- b. Select the **Logical/Physical** tab.
- c. Right-click **Unconfigured Capacity** and select **Create Volume**.
The **Create Volume** wizard displays.
- d. Select **Next** on the **Introduction (Create Volume)** window.
- e. Select the **Manual** option on the **Specify Volume Group (Create Volume)** window.
- f. Select tray 99, slots 1-10, then select **Add**.
- g. Verify that the RAID level is set to 6.
- h. Select **Calculate Capacity**.
- i. Select **Next** on the **Specify Volume Group (Create Volume)** window.

The **Array Management** window should still be displayed after performing this step.

Create and Configure Volumes

After creating the first volume group, create the first volume when prompted. Configure the boot RAID with enough LUNs to support the various system management file systems (Cray recommends a minimum of three LUNs).

4. Create a volume.
 - a. Enter a new volume capacity. Specify units as GB or MB.
 - b. Enter a name for the volume.
 - c. Select the **Customize Settings** option.
 - d. Select **Next** in the **Specify Capacity/Name (Create Volume)** window.
 - e. Verify the settings on the **Customize Advanced Volume Parameters (Create Volume)** window.

These settings are used for the all of the LUNs.

 - For **Volume I/O characteristics type**, verify that **File System** is selected.
 - For **Preferred Controller Ownership**, verify that **Slot A** is selected. This places the LUN on the A Controller.
 - f. Select **Next** in the **Customize Advanced Volume Parameters (Create Volume)** window.
 - g. Select the **Default** mapping option in the **Specify Volume to LUN Mapping** window.
 - h. For **Host type**, select **Linux** from the drop-down menu.
 - i. Select **Finish** in the **Specify Volume to LUN Mapping** window.
 - j. Select **Yes** when prompted to create more LUNs in the **Creation Successful (Create Volume)** window, unless this is the last volume to be created. If this is the last volume, select **No** and continue with the next step (skipping the rest of these substeps).
 - k. Verify that **Free Capacity** is selected on **Volume Group 1 (RAID 5)** in the **Allocate Capacity (Create Volume)** window.
 - l. Select **Next** in the **Allocate Capacity (Create Volume)** window.
 - m. Repeat step 4 to create all of the volumes (applicable to this system) described in [Recommended Boot RAID LUN Values](#) on page 51
 5. Indicate that volume creation and configuration is complete.

Select **OK** in the **Completed (Create Volume)** window.
 6. Create a hot spare.

The hot spare provides a ready backup if any of the drives in the volume group fail.

 - a. Right-click on the last drive in the slot 12 icon on the right portion of the window and select **Hot Spare Coverage**.
 - b. Select the **Manually Assign Individual Drives** option.
 - c. Select **OK**.
 - d. Select **Close**.
 7. Exit the tool.
-

8. (optional) Configure remote logging of the boot RAID messages.

The NetApp, Inc. storage system uses SNMP to provide boot RAID messages. Cray does not provide a procedure for this; see [NetApp, Inc. Storage System documentation](#) for information about how to configure remote logging.

5.2.3.4 Reboot the SMW and Verify LUNs are Recognized

About this task

Use this procedure to make the SMW rediscover the LUNs (logical unit numbers) and zones that were created.

Procedure

1. Log on as the `root` user.

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

2. Reboot the SMW to ensure that the LUNs are recognized.

```
smw# reboot
```



CAUTION: Failure to reboot the SMW at this point could produce unexpected results later on.

3. When the SMW has finished rebooting, log on as the `root` user.

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

4. Execute the `lsscsi` command to verify that the LUNs (volumes) have been rediscovered.

```
smw# lsscsi
```

5. List the disk devices by using the `fdisk` command to verify that the LUNs (volumes) are configured according to the boot LUN configuration table in [Recommended Boot RAID LUN Values](#) on page 51.

```
smw# fdisk -l
```

5.2.4 Make a Snapshot Manually

Prerequisites

This procedure assumes that the SLES 12 base operating system has been installed on the SMW and boot RAID devices have been configured, but no other software has been installed yet.

About this task

A Btrfs snapshot of the SMW should be created immediately after SLES 12 has been installed and before any files or directories have been modified by Cray's installation software or the rest of the installation process. With this snapshot, it will be possible to revert to this point if an initial/fresh install is repeated.

Snapshots are usually made using the `snaputil` program, but that program has not been installed at this point in the installation process. `snaputil` will be installed to the SMW with other Cray RPMs for the SMW and will be used for all Btrfs snapshot manipulations after this point.

Note that snapshots capture content in these three directories: `/`, `/var/lib/mysql`, and `/var/opt/cray/repos`. Used in conjunction with backups of config sets, they provide enough information to be able to re-create the state of the system at a particular time.

Procedure

1. Determine the root subvolume.

It will be the string starting with "UUID." In this example it is "UUID=ffb0b613-2033-4835-87b5-6ca8ff1bacde."

```
smw# grep " / " /etc/fstab
UUID=ffb0b613-2033-4835-87b5-6ca8ff1bacde /          btrfs
defaults          0 0
```

2. Mount the root subvolume.

Substitute the correct subvolume string for the example string shown in this command.

```
smw# mount -o subvol=@ UUID=ffb0b613-2033-4835-87b5-6ca8ff1bacde /mnt
```

3. Create a subvolume for snapshots (if `/mnt/snapshots` does not already exist).

```
smw# btrfs sub create /mnt/snapshots
```

4. Create the snapshot (if `/mnt/snapshots/SLES12` does not already exist).

```
smw# btrfs sub snap / /mnt/snapshots/SLES12
```

5. Unmount the snapshot.

```
smw# umount /mnt
```

6. Make a new `/media/root-sv` directory.

```
smw# mkdir -p /media/root-sv
```

7. Mount root subvolume under `/media/root-sv` instead of `/mnt` as was used above.

Substitute the correct subvolume string for the example string shown in this command.

```
smw# mount -o subvol=@ UUID=ffb0b613-2033-4835-87b5-6ca8ff1bacde /media/root-sv
```

A "SLES12" snapshot has been made. Reboot to this snapshot whenever it is necessary to restart a fresh software installation from this point.

5.3 Install the SMW and CLE Software on the Migration SMW

To install the SMW and CLE software on the migration SMW, use the following procedures in the order listed.

1. [Start a Typescript File](#) on page 59
2. [Prepare to Bootstrap the SMW Installation](#) on page 60
3. [Determine the Persistent Device Name for a LUN](#) on page 62
4. [RAID Disk Space Requirements](#) on page 63
5. [Bootstrap the SMW Installation](#) on page 65
6. [Provision SMW Storage](#) on page 71
7. [Run the Installer for an Initial Installation](#) on page 72
8. Think you know how to boot an SMW? Don't miss the extra, crucial step in this procedure: [Set Default Snapshot and Boot the SMW](#) on page 73

5.3.1 Start a Typescript File

About this task

Sites can make as few or as many typescripts as they deem useful. Cray recommends starting a typescript file at these milestones:

- just before extracting and archiving current configuration information during a software migration
- just before installing a new software release
- just before configuring the newly installed software

Procedure

1. Log in as root to the SMW.
2. (First time only) Create a release directory for the typescript file.

```
smw# mkdir -p /var/adm/cray/release
```

3. Change to the release directory.

```
smw# cd /var/adm/cray/release
```

4. Set a variable equal to today's date.

```
smw# export TODAY=`date +%Y%m%d`  
smw# echo $TODAY
```

5. Start a typescript file.

```
smw# script -af ${TODAY}.suffix
```

For *suffix*, substitute a unique string to distinguish among typescript files, such as `install.1` or `update.2`.

6. Change prompt to include a timestamp.

```
smw# PS1="\u@\h:\w \t# "
```

5.3.2 Prepare to Bootstrap the SMW Installation

Prerequisites

This procedure assumes that the base operating system has been installed on the SMW and that the boot RAID has been set up.

About this task

The installer requires several ISO files to be available for setting up and installing packages from SLE repositories. The names of these ISOs are hard-coded in the installer configuration, but the containing directory can be anywhere that makes sense.

IMPORTANT: The default location for these ISO files is `/root/isos`. The `--iso-dir` argument must be specified for `SMWinstall` if this is not the correct location for the ISO files on this system.

The Cray release distribution media consists of one DVD and several other pieces of media that may be on DVDs or furnished as ISO files. These ISO files are available for download at CrayPort (<https://crayport.cray.com>).

Procedure

COLLECT ISO FILES

1. Make a directory on the SMW to hold the ISO files, if one does not already exist.

Instead of placing the ISOs directly in `/root/isos`, use these two commands to place that directory into the `btrfs` subvolume `/var/adm/cray`, which is exempt from snapshots. This prevents the large ISO files from unnecessarily increasing the size of snapshots.

```
smw# mkdir -p /var/adm/cray/release/isos
smw# ln -s /var/adm/cray/release/isos /root/isos
```

2. Download the SLES 12 distribution ISOs to the ISO directory on the SMW.
 - `SLE-12-Module-Legacy-DVD-x86_64-GM-DVD1.iso`
 - `SLE-12-Module-Public-Cloud-DVD-x86_64-GM-DVD1.iso`
 - `SLE-12-SDK-DVD-x86_64-GM-DVD1.iso`
 - `SLE-12-Server-DVD-x86_64-GM-DVD1.iso`
 - `SLE-12-WE-DVD-x86_64-GM-DVD1.iso`
3. Download the CentOS 6.5 distribution ISO (`CentOS-6.5-x86_64-bin-DVD1.iso`) to the ISO directory on the SMW.
4. Download CLE 6.0 and SMW 8.0 ISOs to the ISO directory on the SMW.
 - SMW release: `smw-8.0.3075-201701182038.iso`
 - CLE release: `cle-6.0.3074-201701182038.iso`
5. Download the SLES 12 security updates ISO (`sleupdate-12sp0+161026-201611021158.iso`) to the ISO directory on the SMW.

6. Make a directory on the SMW (if it does not already exist) to hold any patches that may be available on CrayPort.

```
smw# mkdir -p /var/adm/cray/release/patchsets
```

7. Download SMW and CLE patches to the patchset directory on the SMW, as described in the release notes.

MOUNT MEDIA

8. Mount SMW media.

- a. Confirm that this is the right SMW media.

```
smw# ls -l /root/isos/smw*iso
-rw-r--r-- 1 root root 427184128 Jan 18 21:42 smw-8.0.3075-201701182038.iso
```

- b. Set environment variables for the SMW media.

Use the release string (actually, the build ID) and the date-time stamp for the SMW media as the values for `SMW_RELEASE` and `SMW_SOFTWARE`, as shown in this example.

```
smw# export SMW_RELEASE=8.0.3075
smw# echo $SMW_RELEASE

smw# export SMW_SOFTWARE=201701182038
smw# echo $SMW_SOFTWARE
```

- c. Mount the SMW release media.

```
smw# mkdir -p /media/SMW
smw# mount -o loop,ro /root/isos/smw-${SMW_RELEASE}-${SMW_SOFTWARE}.iso \
/media/SMW
```

9. Mount CLE media.

- a. Confirm that this is the right CLE media.

```
smw# ls -l /root/isos/cle*iso
-rw-r--r-- 1 root root 1146388480 Jan 18 20:38 cle-6.0.3074-201701182038.iso
```

- b. Set environment variables for the CLE media.

Use the release string and the date-time stamp for the CLE media as the values for `CLE_RELEASE` and `CLE_SOFTWARE`, as shown in this example.

```
smw# export CLE_RELEASE=6.0.3074
smw# echo $CLE_RELEASE

smw# export CLE_SOFTWARE=201701182038
smw# echo $CLE_SOFTWARE
```

- c. Mount the CLE release media.

```
smw# mkdir -p /media/CLE
smw# mount -o loop,ro /root/isos/cle-${CLE_RELEASE}-${CLE_SOFTWARE}.iso \
/media/CLE
```

10. Set an environment variable for the SLES 12 security updates media.

Use the entire name of the SLES 12 security updates media as the environment variable. This will be used when installing SMW and CLE software and SLES 12 security updates together later in the process.

```
smw# export SLE_SOFTWARE=sleupdate-12sp0+161026-201611021158
smw# echo $SLE_SOFTWARE
```

COPY THE INSTALL CONFIGURATION FILE

11. Copy `install.cle.conf`.

The `install.cle.conf` file contains configuration that controls the installer's image building behavior.

Copy `install.cle.conf.example` from the CLE installation media to `/var/adm/cray/install.cle.conf` and modify it if necessary.

```
smw# cp -p /media/CLE/products/cle/install.cle.conf.example \
/var/adm/cray/install.cle.conf
```

At this point there is nothing in this file that should be changed for a fresh install. Later this will be changed for updates to CLE.

12. Unmount CLE media.

```
smw# umount /media/CLE
```

5.3.3 Determine the Persistent Device Name for a LUN

About this task

After initial partitioning of the boot RAID, always address the storage via its persistent `/dev/disk/by-id/` name. Do not use the short `/dev/sdxx` name, which cannot uniquely identify the disk between reboots.

Use this procedure to determine the persistent device name from the LUN number on the boot RAID.

Procedure

1. Use `lsscsi` to show the `/dev/sd*` device name associated with a LUN number.

In the first column of the output, the LUN is the final number in the `[n:n:n:n]` value. In this example, LUN 15 is associated with `/dev/sdo`.

```
crayadm@smw1> lsscsi
[0:0:0:0]    disk    ATA      TOSHIBA MK1661GS ME0D  /dev/sda
[0:0:1:0]    disk    ATA      ST91000640NS    AA03  /dev/sdb
[0:0:2:0]    disk    ATA      TOSHIBA MK1661GS ME0D  /dev/sdc
.
.
.
[5:0:0:15]   disk    LSI      INF-01-00      0786  /dev/sdo
[5:0:0:16]   disk    LSI      INF-01-00      0786  /dev/sdp
[5:0:0:17]   disk    LSI      INF-01-00      0786  /dev/sdq
[5:0:0:18]   disk    LSI      INF-01-00      0786  /dev/sdr
```

If multipathing is used, this command may show more than one line (device name) for a single LUN, making it difficult to know which is the correct one. In this case, try using the `SMdevices` command to see the volume

labels assigned to each LUN. That aids in the process of matching the LUN to the Linux device and ensuring that the intended function of the LUN matches the volume name assigned using the SANtricity Storage Manager software.

2. Use `ls -l` to map the `/dev/sd*` device name to the persistent device name.

To display the persistent device name for only one LUN, use `grep`. This example displays the persistent device name for `/dev/sdo` (that is, LUN 15).

```
crayadm@smw1> ls -l /dev/disk/by-id | grep sdo
lrwxrwxrwx 1 root root 10 Sep  4 00:56 scsi-360080e500037667a000003a2519e3ff2 -
> ../../sdo
lrwxrwxrwx 1 root root 10 Sep  4 00:56 wwn-0x60080e500037667a000003a2519e3ff2 -
> ../../sdo
```

There are two results for LUN 15. The one with prefix "scsi" is the one to use, so the persistent device name for LUN 15 is `scsi-360080e500037667a000003a2519e3ff2`.

3. Record the LUNs and corresponding persistent (by-id) device names for the following devices in preparation for bootstrapping the SMW installation.
 - Disk devices on the boot RAID that can be used for boot node persistent storage
 - Disk devices on the boot RAID that can be used for SDB node persistent storage
 - Disk devices on the boot RAID that can be used for SMW persistent storage

5.3.4 RAID Disk Space Requirements

The SMW, the boot node, and the SDB node all use space on the boot RAID. Here are the recommended sizes for the RAID LUNs, or LVM volume groups, based on the file systems for each. This information will be needed to bootstrap the SMW installation, which is next in the installation process.

SMW File Systems

On the boot RAID, the LVM volume group for the SMW will have the file systems listed in this table in the Mount Point column. The third column shows the recommended LUN size for each file system assuming a standard 4.5 TB RAID. For sites with storage constraints or extra storage, the fourth and fifth columns show suggested LUN sizes.

IMPORTANT: The volume for the `/var/opt/cray/imps` file system on the SMW should be significantly larger than the volume for the `/var/opt/cray/imps` file system on the boot node. This is because that file system on the SMW contains boot images, config sets, and image roots, while that file system on the boot node contains only a subset of the image roots on the SMW. The boot node does an NFS mount of the SMW boot images, so no local space is needed for those.

Table 6. SMW RAID Requirements

Mount Point	FS Type	Recommended Size for 4.5 TB RAID	Minimum Size	Size for 9.0 TB RAID	Description
/home	xfs	200 GB	40 GB	200 GB	Home directories on SMW

Mount Point	FS Type	Recommended Size for 4.5 TB RAID	Minimum Size	Size for 9.0 TB RAID	Description
/var/lib/mysql	btrfs	10 GB	10 GB	10 GB	HSS database
/var/opt/cray/disk/1	xfs	1000 GB	400 GB	2000 GB	logs, debug, dumps
/var/opt/cray/imps	xfs	1000 GB	400 GB	1000 GB	IMPS data
/var/opt/cray/repos	btrfs	200 GB	100 GB	200 GB	IMPS repos

CLE File Systems

On the boot RAID, storage for the boot node and SDB node is defined in the CLE storage set. Within that storage set, storage for the boot node is in the boot node LVM volume group, and storage for the SDB node is in the SDB node LVM volume group. The file systems for those nodes are listed in the tables below in the Mount Point column. The fourth column shows the recommended LUN size for each file system assuming a standard 4.5 TB RAID. For sites with storage constraints, the fifth column shows suggested LUN sizes.

Note that for partitioned systems, the requirements for LUN size apply to the boot node and SDB node in each partition.

Expanding storage space. The LUN sizes for the `/cray_home` and `/non_volatile` file systems may need to be adjusted depending on site usage of those file systems. For example, workload managers, DataWarp, and any node that needs permanent storage can store information in `/non_volatile`, so it may need to be larger than the suggested size. If size adjustment is not made at install time, it can be made later. See *XC™ Series System Administration Guide (S-2393)* for instructions on how to expand storage in a file system, volume, or volume group.

Table 7. Boot Node RAID Requirements

Owning Node	Mount Point	FS Type	Recommended Size for 4.5 TB RAID	Minimum Size	Description
boot	/cray_home	xfs	50 GB	50 GB	Home directories on CLE
boot	/var/opt/cray/imps	btrfs	250 GB	250 GB	IMPS data for PE image roots and for Netroot compute-large and login-large image roots
boot	/non_volatile	xfs	200 GB	50 GB	persistent data, including <code>/var</code> if necessary, for service nodes provided from boot node

Table 8. SDB Node RAID Requirements

Owning Node	Mount Point	FS Type	Recommended Size for 4.5 TB RAID	Minimum Size	Description
sdb	/var/lib/mysql	xfs	20 GB	10 GB	SDB database
sdb	/alps_shared	xfs	20 GB	10 GB	ALPS data

5.3.5 Bootstrap the SMW Installation

Prerequisites

The following information must be gathered before running the installer in bootstrap mode. To find the persistent devices names for these devices, see [Determine the Persistent Device Name for a LUN](#) on page 62. For typical file system sizes, see [RAID Disk Space Requirements](#) on page 63.

- Disk devices on the boot RAID that can be used for boot node persistent storage
- Disk devices on the boot RAID that can be used for SDB node persistent storage
- Disk devices on the boot RAID that can be used for SMW persistent storage
- Size of file systems to be created within LVM volumes within LVM volume groups

NOTE: Check that these file system sizes do not exceed the total size of the volume group containing them. Adjust file system sizes, if needed.

About this task

This procedure runs `SMWinstall` in bootstrap mode, which installs IMPS and Ansible on the SMW, along with some of the global configuration templates. The `SMWinstall` command also invokes the configurator to prepare the storage set configuration. The configurator initiates an interactive session to gather the necessary information, unless the storage configuration template is supplied as a command-line argument, in which case no interactive session is needed. This configuration can be updated later by running the configurator manually.

Configurator navigation hints:

- To get context-sensitive command help, enter `?`.
- To add a single value, enter the data and press **Enter**.
- To add a list, enter `+`, enter each list item on its own line, and press **Ctrl-d** when finished entering the entire list of items.
- To correct an error in a previous setting, press the `<` key to go back to the previous setting, correct it, then continue forward. Use `<` to back up several settings, if needed.

Procedure

1. Start the multipath daemon.

Start `multipathd` but do not enable it on the command line. The multipath service needs to be *started* so that it can display path information needed for some config set settings, but the multipath service must not be

enabled at this point in the process. Because of a SLES bug involving multipath and swap, the multipath service must not be enabled before the first time the new snapshot is booted. After that snapshot is booted for the first time, the Ansible multipath play will enable and start multipathd and will create an `/etc/multipath.conf` file. This file will ensure that on subsequent restarts of multipath or reboots of the SMW, multipathd will do the right thing and ignore swap.

```
smw# systemctl start multipathd
```

2. Install in bootstrap mode.

- Method 1: Provide storage configuration information interactively.

```
smw# /media/SMW/SMWinstall --mode bootstrap
```

- Method 2: Provide storage configuration information using an existing storage configuration **template** (the `_config.yaml`, not the `_worksheet.yaml`).

```
smw# /media/SMW/SMWinstall --mode bootstrap --storage-config \
/path/to/cray_bootraid_config.yaml
```

Trouble? If ERROR and WARNING messages appear shortly after running the installer with the `--storage-config` option, and they complain of template syntax and/or schema errors, first check to see if the right file was provided in the command line. It must be the template (a `_config.yaml` file, also known as the *config file*), NOT the worksheet (a `_worksheet.yaml` file). Note that this contrasts with the way the `cfgset` command works: when configuration information is provided using `cfgset` with the `-w` or `--worksheet-path` option, the file provided on the command line must be a worksheet.

If Method 1 used, continue to step [3](#) on page 66. If Method 2 used, skip to step [11](#) on page 70.

3. Ensure that `cray_bootraid.enabled` is set to `true` to enable the storage service.

CONFIGURE THE CLE DEFAULT STORAGE SET (`cledefault`) VOLUME GROUPS

The configurator now shows the settings for a `storage_set` entry named `cledefault`, within which are three `volume_groups` entries:

- `boot_node_vg`
- `sdb_node_vg`
- `compute_node_local`

The full name of settings within each volume group looks like `cray_bootraid.settings.storage_sets.data.cledefault.volume_groups`, followed by `<volume group name>.<field name>`. For brevity, the next steps show only the volume group name and field name of each setting.

4. Configure the boot node volume group (`boot_node_vg`).

- a. Set the owner of the boot node volume group.

Ensure that `boot_node_vg.owner` is set to "boot" rather than a cname. For a partitioned system, include the partition name (e.g., "boot-p2" for partition p2).

- b. Add entries for the physical volumes (disk devices) that are going to be part of the boot node LVM volume group.

This setting is a list. To add list data, enter **+** at the prompt for `boot_node_vg.devices` to enter list entry mode. Add persistent device names such as `/dev/disk/by-id/scsi-360080e50002f7160000014905640c0c4` for each physical volume. Do not use short names like `/dev/sdn`, which may vary from node to node (SMW, boot node, and SDB node), and may vary from boot to boot of a particular node.

Press **Enter** after each list entry, and when done adding entries, press **Ctrl-d** to exit list entry mode. Remember to press **Enter** again to set the list entries.

- c. For each volume of the boot node volume group, change file system size to match the recommended values in the Boot Node RAID Requirements table in [RAID Disk Space Requirements](#) on page 63.

The `home` volume corresponds to the `/cray_home` file system in the table, the `imps` volume corresponds to `/var/opt/cray/imps`, and the `nvolatile` volume corresponds to `/non_volatile`.

At the following prompt, enter these values to get to the `fs_size` setting for each of the file systems.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg.volumes
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $
```

To get to this setting	enter this at prompt
<code>boot_node_vg.volumes.home.fs_size</code>	1d*
<code>boot_node_vg.volumes.imps.fs_size</code>	2d*
<code>boot_node_vg.volumes.nvolatile.fs_size</code>	3d*

Then at the prompt for that setting, enter a new file system size to change the value, if needed. Accept the current or newly entered value by pressing **Enter**.

- d. When done with the last volume, press **Enter** to set the `boot_node_vg` "volumes" entries.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg.volumes
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $ <cr>
```

5. Configure the SDB node volume group (`sdb_node_vg.owner`).

- a. Set the owner of the SDB node volume group.

Ensure that `sdb_node_vg.owner` is set to "sdb" rather than a cname. For a partitioned system, include the partition name (e.g., "sdb-p2" for partition p2).

- b. Add entries for the physical volumes (disk devices) that are going to be part of the SDB node LVM volume group.

This setting is a list. To add list data, enter **+** at the prompt for `sdb_node_vg.devices` to enter list entry mode. Add persistent device names such as `/dev/disk/by-id/scsi-360080e50002f7160000014925640c108` for each physical volume. Do not use short names like `/dev/sdn`, which may vary from node to node (SMW, boot node, and SDB node), and may vary from boot to boot of a particular node.

Press **Enter** after each list entry, and when done adding entries, press **Ctrl-d** to exit list entry mode. Remember to press **Enter** again to set the list entries.

- c. For each volume of the SDB node volume group, change file system size to match the recommended values in the SDB Node RAID Requirements table in [RAID Disk Space Requirements](#) on page 63.

The `db` volume corresponds to the `/var/lib/mysql` file system in the table, and the `alps` volume corresponds to `/alps_shared`.

At the following prompt, enter these values to get to the `fs_size` setting for each of the file systems.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.sdb_node_vg.volumes
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $
```

To get to this setting	enter this at prompt
<code>sdb_node_vg.volumes.db.fs_size</code>	1d*
<code>sdb_node_vg.volumes.alps.fs_size</code>	2d*

Then at the prompt for that setting, enter a new file system size to change the value, if needed. Accept the current or newly entered value by pressing **Enter**.

- d. When done with the last volume, press **Enter** to set the `sdb_node_vg` "volumes" entries.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.sdb_node_vg.volumes
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <cr>
```

6. Set the compute node volume group (`compute_node_local`) owner setting.

The compute node volume group is not needed for systems that do not use compute nodes with onboard SSDs. Set the owner field to null to ensure that this volume group is not used but is preserved in case this site decides to add compute nodes with SSDs to the system later.

Use the **>** key to skip the other `compute_node_local` settings when presented.

```
compute_node_local.owner: null
compute_node_local.devices: >
```

Set the `compute_node_local` "volumes" entries.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.compute_node_local.volumes
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $ <cr>
```

7. Set the `cledefault` "volume groups" entries.

Review the list of `cledefault` volume groups (enter ***** to see the full list if not all volume groups are displayed), then at the prompt below, enter press **Enter** to set the entries.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $ <cr>
```

CONFIGURE THE SMW DEFAULT STORAGE SET (`smwdefault`) VOLUME GROUPS

The configurator now shows the settings for a `storage_set` entry named `smwdefault`, within which is one `volume_groups` entry: `smw_node_vg`

The full name of settings within each volume group looks like

`cray_bootraid.settings.storage_sets.data.smwdefault.volume_groups.` followed by `<volume group name>.<field name>`. For brevity, the next steps show only the volume group name and field name of each setting.

8. Configure the SMW node volume group (`smw_node_vg`).

- a. Set the owner of the SMW node volume group.

Ensure that `smw_node_vg.owner` is set to "smw." It could also be set to the hostname of the SMW, such as "orion-smw," but there is greater portability of the config set if "smw" is used.

- b. Add entries for the physical volumes (disk devices) that are going to be part of the SMW node LVM volume group.

This setting is a list. To add list data, enter **+** at the prompt for `smw_node_vg.devices` to enter list entry mode. Add persistent device names such

as `/dev/disk/by-id/scsi-360080e50002f889c0000a0654e32232` for each physical volume. Do not use short names like `/dev/sdn`, which may vary from node to node (SMW, boot node, and SDB node), and may vary from boot to boot of a particular node.

Press **Enter** after each list entry, and when done adding entries, press **Ctrl-d** to exit list entry mode. Remember to press **Enter** again to set the list entries.

- c. For each volume of the SMW node volume group, change file system size to match the recommended values in the SMW RAID Requirements table in [RAID Disk Space Requirements](#) on page 63.

The `home` volume corresponds to the `/home` file system in the table, the `db` volume corresponds to `/var/lib/mysql`, the `log` volume corresponds to `/var/opt/cray/disk/1`, the `imps` volume corresponds to `/var/opt/cray/imps`, and the `repos` volume corresponds to `/var/opt/cray/repos`.

At the following prompt, enter these values to get to the `fs_size` setting for each of the file systems.

```
cray_bootraid.settings.storage_sets.data.smwdefault.volume_groups.smw_node_vg.volumes
[<cr>=set 5 entries, +=add an entry, ?=help, @=less] $
```

To get to this setting	enter this at prompt
<code>smw_node_vg.volumes.home.fs_size</code>	1d*
<code>smw_node_vg.volumes.db.fs_size</code>	2d*
<code>smw_node_vg.volumes.log.fs_size</code>	3d*
<code>smw_node_vg.volumes.imps.fs_size</code>	4d*
<code>smw_node_vg.volumes.repos.fs_size</code>	5d*

Then at the prompt for that setting, enter a new file system size to change the value, if needed. Accept the current or newly entered value by pressing **Enter**.

- d. When done with the last volume, press **Enter** to set those `smw_node_vg` "volumes" entries.

```
cray_bootraid.settings.storage_sets.data.smwdefault.volume_groups.smw_node_vg.volumes
[<cr>=set 5 entries, +=add an entry, ?=help, @=less] $ <cr>
```

9. Set the `smwdefault` "volume groups" entries.

Review the list of `smwdefault` volume groups, then at the prompt below, enter press **Enter** to set the entries.

```
cray_bootraid.settings.storage_sets.data.smwdefault.volume_groups
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <cr>
```

10. Set the boot RAID "storage sets" entries.

Review the storage sets. Press **Enter** (`<cr>`) to set the `cledefault` and `smwdefault` storage sets, unless this system has partitions. If configuring a partitioned system, enter **+** to add another CLE storage set. A separate storage set is needed for each partition.

```
cray_bootraid.settings.storage_sets
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $
```

Trouble? If `SMWinstall` fails during the installation, it is because `cfgset` failed, which was invoked by `SMWinstall` to gather configuration information. That failure may be due to missing information. Do not try running `SMWinstall --mode bootstrap` again.

Try one of these options instead:

Option	Description
Run the configurator manually	<ol style="list-style-type: none"> 1. Enable the <code>cfgset</code> command. <pre>smw# . /opt/modules/default/etc/modules.sh smw# module use /opt/cray/ari/modulefiles smw# module load imps</pre> 2. Use <code>cfgset</code> to invoke the configurator in interactive mode to make any needed changes to the <code>cray_bootraid</code> configuration service in the global config set. <pre>smw# cfgset update -m interactive -s cray_bootraid global</pre> 3. Run the installer in bootstrap mode again. <pre>smw# /media/SMW/SMWinstall --mode bootstrap</pre>
Run <code>SMWinstall</code> with the reconfigure option	<ol style="list-style-type: none"> 1. Run <code>SMWinstall</code> in bootstrap mode with the <code>reconfigure</code> option, which invokes the configurator in interactive mode. <pre>smw# /media/SMW/SMWinstall --mode bootstrap --reconfigure</pre>

11. Display `cray_bootraid` information.

```
smw# . /opt/modules/default/etc/modules.sh
smw# module use /opt/cray/ari/modulefiles
smw# module load imps
smw# cfgset search -s cray_bootraid -l basic global
smw# cfgset search -s cray_bootraid -l advanced global
```

12. (SMW HA only) Copy the storage configuration template.

If this is the primary/first SMW installed of an SMW HA pair, save the storage configuration template to another system not on this SMW for fast and consistent system bootstrapping when installing the secondary SMW.

```
smw# scp -p /var/opt/cray/imps/config/sets/global/config/\  
cray_bootraid_config.yaml user@host:~/
```

Note that it is the **template** (the `_config.yaml` file), not the worksheet (the `_worksheet.yaml` file) that must be copied.

13. Remove existing volume groups, as needed.

If doing a fresh install onto a system, and there is a desire to reuse the storage in any existing LVM volume groups for SMW, boot node, and SDB node, then run these commands to remove the volume groups with storage to be reused.

- a. Use `cfgset search` to find the names of all of the volume groups defined in the storage configuration template.

```
smw# cfgset search -s cray_bootraid global |awk -F'.' '{print $7}' |sort -u
boot_node_vg
boottestlvg
sdb_node_vg
sdbtestlvg
smw_node_vg
smwtestlvg
```

- b. Display the volume groups that exist.

```
smw# vgdisplay
```

Alternative (more concise):

```
smw# vgs
```

- c. Remove the volume groups with storage to be reused (in this example, the test volume groups).

```
smw# vgremove -f smwtestlvg
smw# vgremove -f boottestlvg
smw# vgremove -f sdbtestlvg
```

The system is now ready for the provisioning of boot RAID LVM volumes.

5.3.6 Provision SMW Storage

About this task

The provision-storage mode of `SMWinstall` can be run at any time. It uses the boot RAID configuration template (`cray_bootraid_config.yaml`) to provision persistent storage on the boot RAID by creating LVM volume groups and LVM volumes. This is a non-interactive procedure if `--mode bootstrap` was used to bootstrap the installation earlier in the process. Otherwise, it will gather the necessary site-specific configuration information interactively.

Procedure

1. Provision storage for the default SMW storage set.

Use this command only if using an SMW storage set called "smwdefault," which is the default.

```
smw# /media/SMW/SMWinstall --mode=provision-storage
```

If no errors reported, proceed to step 2 on page 72.

Trouble? If errors are reported, review the boot RAID configuration settings using one of these methods. Both methods run the installer in provision-storage mode again after reviewing the settings and making changes. Note that when the installer is run again, it will ask ALL storage configuration questions, and the defaults will be prefilled with existing data.

- Error recovery method 1: Modify using the configurator, then run installer again.

```
smw# cfgset update -s cray_bootraid -m interactive global
```

```
smw# /media/SMW/SMWinstall --mode=provision-storage
```

- Error recovery method 2: Modify manually, then run installer again.

```
smw# vi /var/opt/cray/imps/config/sets/global/config/\
cray_bootraid_config.yaml

smw# /media/SMW/SMWinstall --mode=provision-storage
```

2. View the new volumes.

```
smw# lvs
```

When the provision-storage installer mode completes successfully, the system is ready for the installation of SMW and CLE software.

5.3.7 Run the Installer for an Initial Installation

Prerequisites

This procedure assumes that ISOS for SLES 12 and CentOS 6.5 have been downloaded as described in [Prepare to Bootstrap the SMW Installation](#) on page 60 and SMW storage has been successfully configured.

About this task

This procedure installs SMW and CLE software together to ensure that there is a matched set of software and configuration.

NOTE: Do NOT run the installer from the `/root/isos` directory. Instead, run it from a directory that is not included in any snapshot, such as `/var/adm/cray/release`.

Procedure

1. Set variable for snapshot name.

Setting a variable here enables better command substitution in later commands dealing with snapshots. The SMW will boot from this snapshot.

```
smw# ls -l1st /root/isos
smw# export SNAPSHOT=smw-${SMW_RELEASE}_cle-${CLE_RELEASE}.${TODAY}
smw# echo $SNAPSHOT
```

(SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

2. Install SMW and CLE software and security updates together.

It is possible to install both SMW media and CLE media with a single command to create a unified "release" that is tagged as a snapshot on the SMW system. Run the `SMWinstall` program and tell it where the CLE media is. This invocation creates the "target" snapshot, which was named in step 1, and then installs into that target snapshot (note that in the absence of an existing target snapshot, the installer creates one from the current running snapshot by default). The installer assumes that all of the SLES 12 ISOs are in `/root/isos`.

IMPORTANT: The SLE media must be specified before the CLE media on the command line so that SUSE security updates are installed before the CLE software is installed.

```
smw# /media/SMW/SMWinstall \
--plus-media=/root/isos/${SLE_SOFTWARE}.iso \
--plus-media=/root/isos/cle-${CLE_RELEASE}-${CLE_SOFTWARE}.iso \
--target=${SNAPSHOT}
```

It can take from 10 to 25 minutes to run a combined installation of SMW, CLE, and security updates for the first time on the SMW. The output of `SMWinstall` provides several command hints, including these three:

- snaputil default** The first command hint (`snaputil default`) is used to ensure that the SMW is booted from the correct (new) snapshot, which is essential to a successful reboot.
- snaputil chroot** The second command hint (`snaputil chroot`) is used in the software update process and may be used at other times to look around inside the snapshot.
- snaputil delete** The third command hint (`snaputil delete`) should be used only if this site needs to remove the newly created snapshot for any reason.

Logs will be in `/var/adm/cray/logs/install` for each invocation of `SMWinstall`.

3. Check new snapshot software versions.

When `SMWinstall` completes, check the snapshot details for the expected SMW and CLE release versions (note that the actual output of this command will show different release versions than this example output).

```
smw# /media/SMW/snaputil show ${SNAPSHOT}
active_maps      :
  p0:/var/opt/cray/imps/config/sets/global/nims/maps/p0
boot menu       : False
booted          : True
btrfs_object_id : 365
cle_version     : 6.0.96
created        : 2016-05-23 11:28:01
default        : True
initrd         : initrd-3.12.51-52.39-default
kernel        : vmlinuz-3.12.51-52.39-default
name           : smw-8.0.96_cle-6.0.96.20160523
parent         : @
path           : /media/root-sv/snapshots/smw-8.0.96_cle-6.0.96.20160523
read-only      : False
smw_version    : 8.0.96
smwha_version  : None
storage_set    : smwdefault
subvolumes     :
  /var/lib/mysql:smw-8.0.96_cle-6.0.96.20160523
  /var/opt/cray/repos:smw-8.0.96_cle-6.0.96.20160523
total size     : 27658.24 MB
unshared size  : 1505.28 MB
updated       : 2016-05-23 11:51:01.188064
```

The SMW is now ready to reboot, which starts with setting the default snapshot to boot from. Trying to boot the SMW without first setting the default snapshot will result in an unbootable SMW.

5.3.8 Set Default Snapshot and Boot the SMW

Prerequisites

This procedure assumes that the snapshot variable has been set and the SMW and CLE software has been installed.

About this task

When the `SMWinstall` command was invoked in the previous procedure, it provided several suggested `snaputil` commands. The one used in this procedure ensures that the snapshot target is set as the default snapshot for the next boot of the SMW.

Procedure

1. Set the release snapshot as the default.

IMPORTANT: Do not skip this step. If the SMW is rebooted without first setting the default snapshot, the SMW becomes unbootable.

```
smw# /media/SMW/snaputil default ${SNAPSHOT}
```

2. Verify that the correct snapshot is the default.

```
smw# /media/SMW/snaputil list
```

3. Reboot the SMW to switch to the new release.

```
smw# reboot
```

5.4 Configure Other Features and Services

At this stage in the migration process, the basic SMW and CLE software has been installed and configured on the migration SMW, and the SMW has been booted. These procedures continue the configuration of a functional system.

1. [Set or Change the HSS Data Store \(MariaDB\) Root Password](#) on page 75
2. [Start a Typescript File](#)
3. [Make a Post-install Snapshot using snaputil](#) on page 77
4. [Update install.cle.conf for Software Updates](#) on page 77
5. [Configure Power Management](#) on page 78
6. (recommended) [Reduce Impact of Btrfs Periodic Maintenance on SMW Performance](#) on page 82
7. (optional) [Configure the Simple Event Correlator \(SEC\)](#) on page 82
8. (optional) [Prevent Unintentional Re-creation of Mail Configuration Files](#) on page 83
9. (optional) [Install the Dell Systems Management Tools and Documentation DVD](#) on page 83

5.4.1 Set or Change the HSS Data Store (MariaDB) Root Password

About this task

The method for setting or changing the HSS data store (database) root password has changed with the release of CLE 6.0. By default, MariaDB is installed with no password set up for the root account. Cray strongly recommends adding a password as part of the fresh install procedure.

Old The HSS database was implemented with MySQL. After initial installation, its root password was changed from the initial default empty string to a user-defined value by the `SMWconfig` script, which was run after `SMWinstall` and the initial discovery of the system.

New The HSS database is now implemented with MariaDB, a MySQL work-alike database with identically named commands. As before, the initial default root password is the empty string; however, the `SMWconfig` script is no longer used to set it after installation. The administrator must use the following procedure to set the root password to a user-defined value.

After the MariaDB root password has been set, it must be placed in `/root/.my.cnf`, a file readable only by root that has the format shown in step 2. This file is the mechanism by which the installer and the `snaptutil` command obtain the root password when they access MariaDB as root. If the file does not exist or it has no `password=` line, the system will attempt to access MariaDB using the default empty-string password, which will fail once the password has been changed.

- Create `/root/.my.cnf` the first time the root password is set to a user-defined value.
- Update `/root/.my.cnf` to match the MariaDB root password whenever it is changed.

IMPORTANT: For an SMW HA system, record the new MySQL root password. It will need to be changed on the second SMW later (by editing `/root/.my.cnf`). After the SMW HA cluster has been configured, the MySQL root password needs to be reset with `mysqladmin` on only one SMW, because the MySQL database is shared between both SMWs in the HA cluster.

Procedure

1. Set or change the MariaDB root password.

```
smw# mysqladmin -uroot password -p
```

Do one of the following at the prompt:

- To **set** the root password for fresh installs or after the database has been reinitialized, press **Enter** to enter an empty string, the default initial password.

```
Enter password: <cr>
```

- To **change** the root password, enter the existing password.

```
Enter password: existing_password
```

At these prompts, enter the new root password, and then enter it again.

```
New password:
Confirm new password:
```

2. Ensure that the root password in the `/root/.my.cnf` file matches the new root password.

If this file does not yet exist, create it and add the lines shown in the example, substituting the new password for the placeholder `MariaDB-password`.

```
smw# vi /root/.my.cnf
[client]
user=root
password=MariaDB-password
```

3. Ensure that only root can see or write to the `/root/.my.cnf` file.

```
smw# chmod 600 /root/.my.cnf
```

5.4.2 Start a Typescript File

About this task

Sites can make as few or as many typescripts as they deem useful. Cray recommends starting a typescript file at these milestones:

- just before extracting and archiving current configuration information during a software migration
- just before installing a new software release
- just before configuring the newly installed software

Procedure

1. Log in as root to the SMW.
2. (First time only) Create a release directory for the typescript file.

```
smw# mkdir -p /var/adm/cray/release
```

3. Change to the release directory.

```
smw# cd /var/adm/cray/release
```

4. Set a variable equal to today's date.

```
smw# export TODAY=`date +%Y%m%d`
smw# echo $TODAY
```

5. Start a typescript file.

```
smw# script -af ${TODAY}.suffix
```

For `suffix`, substitute a unique string to distinguish among typescript files, such as `install.1` or `update.2`.

6. Change prompt to include a timestamp.

```
smw# PS1="\u@\h:\w \t# "
```

5.4.3 Make a Post-install Snapshot using snaputil

About this task

This procedure uses `snaputil` to make an archival snapshot of the system after installing a new software release (fresh install or software update) and before configuring the SMW for CLE system hardware.

Best Practice. Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

For more information, see [About Snapshots and Config Set Backups during a Migration](#) on page 424.

Procedure

1. List the available snapshots on the system.

```
smw# snaputil list
```

2. Set the `SNAPSHOT` environment variable using the currently booted snapshot name.

```
smw# export SNAPSHOT=$(snaputil list |grep ^cur| awk '{print $2}')
smw# echo $SNAPSHOT
```

Setting a variable for the snapshot name enables better command substitution in later commands dealing with snapshots.

(SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

3. Create a new snapshot.

```
smw# snaputil create ${SNAPSHOT}.postinstall
```

5.4.4 Update `install.cle.conf` for Software Updates

Prerequisites

This procedure assumes that the installer will not be run again at this point in the installation and configuration process.

About this task

The `/var/adm/cray/install.cle.conf` file contains configuration that controls the image building behavior of the installer. Changing this file now will make later updates of CLE software easier.

Procedure

1. Edit the configuration file.

```
smw# vi /var/adm/cray/install.cle.conf
```

2. (For all systems) Change `build_images` to `yes` to enable the CLE installer to build IMPS images as part of the install process. The remaining options determine what to do if `build_images` is set to `yes`.

```
build_images: yes
```

3. (For partitioned systems only) Uncomment the `map_partition` line and specify the system partitions.

```
map_partition: ['p1', 'p2']
```

5.4.5 Configure Power Management

Prerequisites

This is a required step in bringing up a Cray XC system with releases later than CLE 6.0 UP01 / SMW 8.0 UP01. The PostgreSQL database on the SMW is needed even if a site will be using a remote (off-SMW) database node to store Power and SEDC data.

NOTE: (SMW HA only) Skip this procedure if doing a migration of the first SMW in an SMW HA system, and the Cray SMWHA software will be installed immediately afterwards, because power management for the SMW HA system will be configured later in the HA fresh install process.

This procedure assumes that a disk drive is available for use as a dedicated drive for the PMDB. The drive should be physically located within the SMW at slot 4. On a Dell PowerEdge™ R815 Rack Server, the device for PMDISK is `/dev/disk/by-path/pci-0000:05:00.0-sas-0x4433221103000000-lun-0`. On a Dell PowerEdge™ R630 Rack Server, the device for PMDISK

is `/dev/disk/by-path/pci-0000:03:00.0-scsi-0:0:4:0`.

To determine which physical slot maps to a drive (in case the SMW at this site uses a slot or device name different than those listed above), use this command:

```
smw# smwmapdrives
List of SMW-installed disk drives
-----
Physical slot 0:
  /dev/sdbu
  /dev/disk/by-id/
  /dev/disk/by-id/scsi-SATA_ST9500620NS_9XF3BGQ5
  /dev/disk/by-path/pci-0000:05:00.0-sas-phy7-0x4433221107000000-lun-0
Physical slot 1:
  /dev/sdbx
  /dev/disk/by-id/
  /dev/disk/by-id/scsi-SATA_ST9500620NS_9XF3BGWA
  /dev/disk/by-path/pci-0000:05:00.0-sas-phy6-0x4433221106000000-lun-0
<snip>
```

The system cannot be in use during this procedure. If this is not a fresh install, CLE must be shut down.

About this task

Power Management allows Cray® XC Series™ systems to operate more efficiently. By monitoring, profiling, and limiting power usage administrators can:

- Increase system stability by reducing heat dissipation
- Reduce system cooling requirements
- Reduce site cooling requirements
- Reduce utility costs by minimizing power usage when rates are the highest
- Respond to external environmental conditions and prevent power outages
- Calculate the actual power cost for individual users and/or jobs



CAUTION: Do not use this procedure in preparation for setting up an SMW HA system. As part of the HA configuration the `SMWHAconfig` copies the contents of the PMDB to a shared power management RAID disk. For more information see *XC™ Series SMW HA Installation Guide (S-0044)*.

These steps are performed as `root`.

Procedure

1. Verify that the PMDISK is inserted into the SMW by entering the correct device name. This example, and the ones that follow, are for a Dell R815.

```
smw# fdisk -l /dev/disk/by-path/pci-0000:05:00.0-sas-0x4433221103000000-lun-0

Disk /dev/disk/by-path/pci-0000:05:00.0-sas-0x4433221103000000-lun-0: 931.5 GiB,
1000204886016 bytes, 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
Disklabel type: dos
Disk identifier: 0x10692081
```

2. Create a new primary partition for the PMDISK, and write it to the partition table. If there are any existing partitions on this disk, manually delete them first.

```
smw# fdisk /dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0
Welcome to fdisk (util-linux 2.25).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
  Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-1953525167, default 2048): [press return]
Last sector, +sectors or +size{K,M,G,T,P} (2048-1953525167, default 1953525167): [press
return]

Created a new partition 1 of type 'Linux' and of size 931.5 GiB.

Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

3. Verify that the partition has been created. On a Dell R815 this should be device `/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1`. On a Dell R630 this should be `/dev/disk/by-path/pci-0000:03:00.0-scsi-0:0:4:0-part1`.

```
smw# fdisk -l \
/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0

Disk /dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0: 931.5 GiB, 1000204886016
bytes, 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
Disklabel type: dos
Disk identifier: 0x96c1b0f0

Device                                                    Boot Start      End
Sectors   Size Id Type
/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1      2048 1953525167
1953523120 931.5G 83 Linux
```

4. Create an ext4 file system on the PMDISK partition.

```
smw# mkfs.ext4 \
/dev/disk/by-path/pci-0000:05:00.0-sas-0x4433221103000000-lun-0-part1

Creating filesystem with 244190390 4k blocks and 61054976 inodes
Filesystem UUID: 6d791409-e327-4620-a80c-2933271b3eec
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000, 214990848

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

5. Stop the RSMS services.

```
smw# systemctl stop rsms
smw# systemctl status rsms
rsms.service - hss daemon control
   Loaded: loaded (/usr/lib/systemd/system/rsms.service; enabled)
   Active: inactive (dead) since Wed 2015-11-04 15:42:04 CST; 19s ago
     Process: 5471 ExecStop=/opt/cray/hss/default/bin/hssctl stop (code=exited, status=0/
SUCCESS)
     Process: 30305 ExecStart=/opt/cray/hss/default/bin/hssctl start (code=exited, status=0/
SUCCESS)

Nov 03 16:01:43 smw hssctl[30305]: Starting daemons: erd erdh state_man...md
Nov 04 15:42:04 smw hssctl[5471]: Stopping daemons: sec_cmd boot_cmds ca...rd
Hint: Some lines were ellipsized, use -l to show in full.
```

6. Verify that the RSMS services are stopped. While the RSMS services are stopped, the system may continue to run applications, however the high-speed network will be throttled until RSMS is restarted.

```
smw# rsms status
```

PID	DAEMON	STATE	UPTIME
	erd	stopped	
	erdh	stopped	
	state_manager	stopped	
	nid_mgr	stopped	
	bootmanager	stopped	

```

sedc_manager      stopped
xtpmd             stopped
erfsd            stopped
xtremoted         stopped
xtpowerd         stopped
nimsd            stopped
xtsnmpd          stopped
xtdiagd          stopped

```

7. Run the xtmvpmdb script.

```

smw# xtmvpmdb /dev/disk/by-path/pci-0000:05:00.\
0-sas-0x4433221103000000-lun-0-part1 ext4
- Checking userid
- Checking destination directory name
- Checking destination directory existence

Move database to: /dev/disk/by-path/pci-0000:05:00.0-sas-0x4433221103000000-lun-0-part1
[y/n] [y]: y
- Checking current PM database directory existence
- Checking for booted system
- Checking for rsms daemons
- Creating directory /media/temp_pgsql_data
Dir: /media/temp_pgsql_data created
- Checking status of PM database process
Checking for PostgreSQL 9.3.8: ..running
postgresql.service - LSB: Start the PostgreSQL master daemon
  Loaded: loaded (/etc/init.d/postgresql)
  Active: active (exited) since Tue 2015-11-03 15:38:45 CST; 24h ago
  Process: 16633 ExecReload=/etc/init.d/postgresql reload (code=exited, status=0/SUCCESS)
  Process: 16255 ExecStart=/etc/init.d/postgresql start (code=exited, status=0/SUCCESS)

- Stopping PM database
- Copy contents of /var/lib/pgsql to /media/temp_pgsql_data
- This may take a few minutes to complete.
- Rename previous DB directory from: /var/lib/pgsql to: /var/lib/pgsql.
11-04-2015t15:43:04
- Unmount device from temporary mount point: /media/temp_pgsql_data
- Unmount btrfs subvolume: /var/lib/pgsql
- Mount device at permanent mount point: /var/lib/pgsql
- Add mount point to /etc/fstab
- Start PM database

- Transfer of PM database complete.

```

8. Restart the RSMS services and verify that the daemons are starting.

```

smw# systemctl start rsms
smw# systemctl status rsms
rsms.service - hss daemon control
  Loaded: loaded (/usr/lib/systemd/system/rsms.service; enabled)
  Active: active (exited) since Wed 2015-11-04 15:44:24 CST; 9s ago
  Process: 5471 ExecStop=/opt/cray/hss/default/bin/hssctl stop (code=exited, status=0/SUCCESS)
  Process: 9227 ExecStart=/opt/cray/hss/default/bin/hssctl start (code=exited, status=0/SUCCESS)

Nov 04 15:44:24 smw hssctl[9227]: Starting daemons: erd erdh state_manag...md
Hint: Some lines were ellipsized, use -l to show in full.

```

9. Verify that certain RSMS services are running.

The erd, erdh, state_manager, nid_mgr and bootmanager RSMS services should be verified as running. All other services should be stopped.

```
smw# rsms status
PID          DAEMON          STATE          UPTIME
75876        erd              running        Tue 2017-02-21 13:16:17 CST
76000        erdh             running        Tue 2017-02-21 13:16:19 CST
76124        state_manager   running        Tue 2017-02-21 13:16:20 CST
76251        nid_mgr         running        Tue 2017-02-21 13:16:21 CST
76372        bootmanager     running        Tue 2017-02-21 13:16:22 CST
             xtpmd           stopped
             erfsd           stopped
             xtremoted       stopped
             xtpowerd        stopped
             nimsd          stopped
             xtsnmpd        stopped
             xtdiagd        stopped
```

5.4.6 Reduce Impact of Btrfs Periodic Maintenance on SMW Performance

About this task

Btrfs (B-tree file system) runs periodic maintenance. The weekly and monthly maintenance scripts, which include balance, trim, and scrub actions, can consume large amounts of compute resource. This can impact a site's ability to use the SMW for normal operations, even using SSH to log into nodes. This procedure describes how to reduce the impact to SMW performance by controlling when these scripts are run.

Procedure

1. Create a file `/etc/cron.d/cray_btrfs.cron`.

The new cron file needs to be in `/etc/cron.d` because the btrfs RPM installs links to maintenance scripts into the `/etc/cron.{weekly,monthly}` directories.

```
smw# vi /etc/cron.d/cray_btrfs.cron
```

Add these lines to the new file. Adjust as needed for this site.

```
# Control when btrfs maintenance scripts run by deleting the corresponding
# 'lastrun' files at a predetermined time.  Caveat, this affects all of the
# scripts in the corresponding cron directories (/etc/cron.{weekly,monthly})

# Run weekly on Saturday at 2 AM as root
0 2 * * 6 root rm -f /var/spool/cron/lastrun/cron.weekly
# Run monthly on the first Sunday of the month at 2 AM as root
0 2 * * 0 root [ $(date +%d) -le 07 ] && rm -f /var/spool/cron/lastrun/cron.monthly
```

2. Set ownership of the new cron file to root,root with permissions 644.

```
smw# chown root:root /etc/cron.d/cray_btrfs.cron
smw# chmod 644 /etc/cron.d/cray_btrfs.cron
```

5.4.7 Configure the Simple Event Correlator (SEC)

The Simple Event Correlator (SEC) is an SMW utility that parses every line being appended to system log files, watching for specific strings that represent the occurrence of significant system events. When a specified string is detected, SEC sends notification that this has happened, either by email, IRC, writing to a file, or some user-configurable combination of all three.

SEC is enabled by default, and by default is configured to generate email notifications to `crayadm`. The types of notifications generated and the recipients to whom notifications are sent are defined in the SEC configuration file, `/etc/opt/cray/cray_sec_actions_config`.

The System Management Workstation (SMW) release includes `sec-2.7.6` and an SEC support package, `cray-sec-8.0.0`. The SEC support package contains control scripts to manage the starting and stopping of SEC around a Cray mainframe boot session, in addition to other utilities and a rule set designed for Cray systems.

For configuration procedures, see *XC™ Series SEC Configuration Guide (S-2542)* for release CLE 6.0.UP03.

For a migration, use the following files from the CLE 5.2 / SMW 7.2 system for reference while configuring SEC for CLE 6.0 / SMW 8.0. They contain site-customized data for which SEC rules to run, any site-local SEC rules (`/opt/cray/sec/default/rules/local`), site host name, serial number of this XC system, email list, scheduler path, and so forth.

```
/etc/aliases
/etc/opt/cray/cray_sec_actions_config
/opt/cray/sec/default/SEC_VARIABLES*
/opt/cray/sec/default/SHELL_VARIABLES*
/opt/cray/sec/default/rules
/opt/cray/sec/default/bin/check_xt.sh
/opt/cray/sec/default/bin/check_xt_wrapper.ex
```

5.4.8 Prevent Unintentional Re-creation of Mail Configuration Files

This procedure is optional. It applies to systems where postfix or sendmail are configured on the SMW.

To prevent the `master.cf` and `main.cf` postfix configuration files from being re-created during software updates or fixes, edit the `/etc/sysconfig/mail` file on the SMW and ensure that the `MAIL_CREATE_CONFIG` setting is set to "no."

```
smw# vi /etc/sysconfig/mail
```

```
MAIL_CREATE_CONFIG="no"
```

5.4.9 Install the Dell Systems Management Tools and Documentation DVD

About this task

This procedure installs the OpenManage Server Administrator (OMSA) software from the Dell Systems Management Tools and Documentation DVD, which is shipped with the SMW. This software enables advanced control over the Integrated Dell Remote Access Controller (iDRAC) and provides features such as Automatic Recovery (automatic system boot after a power event).

Visit the Dell OpenManage Linux Repository to view the Dell OpenManage Server Administrator documentation:

<http://linux.dell.com/wiki/index.php/Repository/OMSA>

Procedure

1. Obtain the Dell System Management Tools and Documentation DVD.
2. Log on to the SMW as `root`.
3. Mount the DVD.

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

4. Go to the location of the installation scripts.

```
smw# cd /media/cdrom/SYSMGMT/srvadmin/linux/supportscripts
```

5. Execute the script to install the software.

```
smw# sh srvadmin-install.sh --express
```

6. Start the Server Administrator services.

```
smw# sh srvadmin-services.sh start
```

7. Double-click the icon named **Launch Server Administrator** on the SMW screen.
8. Enter the SMW user name `root`.
9. Enter the SMW `root` account password.

The system can now be managed for Properties, Shutdown, Logs, Alert Management, and Session Management.

6 Preparation of Configuration Data and Software Images on the Migration SMW

To minimize the downtime required to switch an XC system from CLE 5.2 / SMW 7.2 to the new CLE 6.0 / SMW 8.0 release, configuration data and software images can be prepared ahead of time. This phase of the migration process requires access to a migration SMW with the new software already installed, which is used to stage the changes for configuration and image management. Note that the migration SMW and boot RAID are not yet connected to XC system hardware, which affects some of these procedures.

1. Perform these procedures on the system running CLE 5.2 / SMW 7.2:
 - a. Ensure either access to the physical keyboard, mouse, and monitor of the original SMW (running CLE 5.2 / SMW 7.2) or connection over iDRAC (if that SMW is not physically present) before trying to perform the migration.
 - b. [Start a Typescript File](#) to capture commands and output.
 - c. [Extract Configuration Data from the CLE 5.2 / SMW 7.2 System](#) on page 86 (running CLE 5.2 / SMW 7.2).
2. Perform these procedures on the migration SMW, which is running CLE 6.0 / SMW 8.0:
 - a. [Read Man Pages for New Commands](#) on page 98 to increase familiarity with the new CLE 6.0 / SMW 8.0 commands.
 - b. [Transfer Configuration Data to Configuration Worksheets](#) on page 98.
 - c. [Load and Validate Configuration Data on the Migration SMW](#) on the migration SMW.
 - d. [Update Non-config-set Configuration Files on the Migration SMW](#).
 - e. [Choose Image Recipes to Build](#) on page 323.
 - f. [Build Image Roots and Boot Images from Recipes](#) on page 328.
 - g. [Assign Kernel Parameters to Nodes](#) on page 341.
 - h. [Check NIMS Information](#) on page 342.
 - i. [Identify and Port Site-local Scripts](#) on page 343 (done on CLE 5.2 / SMW 7.2 system + migration SMW).
 - j. [Install Cray Programming Environment \(PE\) Software](#) on page 344.

6.1 Start a Typescript File

About this task

Sites can make a few or as many typescripts as they deem useful. Cray recommends starting a typescript file at these milestones:

- just before extracting and archiving current configuration information during a software migration
- just before installing a new software release
- just before configuring the newly installed software

Procedure

1. Log in as root to the SMW.
2. (First time only) Create a release directory for the typescript file.

```
smw# mkdir -p /var/adm/cray/release
```

3. Change to the release directory.

```
smw# cd /var/adm/cray/release
```

4. Set a variable equal to today's date.

```
smw# export TODAY=`date +%Y%m%d`  
smw# echo $TODAY
```

5. Start a typescript file.

```
smw# script -af ${TODAY}.suffix
```

For *suffix*, substitute a unique string to distinguish among typescript files, such as `install.1` or `update.2`.

6. Change prompt to include a timestamp.

```
smw# PS1="\u@\h:\w \t# "
```

6.2 Extract Configuration Data from the CLE 5.2 / SMW 7.2 System

About this task

Use [Migration Checklist 2.1: Extract Configuration Data](#) on page 435 to track progress while performing this procedure.

This procedure gathers configuration data from the CLE 5.2 / SMW 7.2 system. It includes saving configuration files from selected nodes and actively probing the running system for configuration settings. It is important to probe the running system because any site-local scripts could have overridden data from configuration files, which could have overridden data from installer files. The order of precedence should be that any data from a probe should override data from configuration files, which should override data from installer files.

Procedure

1. Save configuration files from the sharedroot.

It is important to save all of the specialized `/etc` files (default, class, node) in a tar archive for later analysis. These files contain the customizations for the service nodes and, if using the DSL (Dynamic Shared Libraries) feature, the compute nodes using the `cnos` node class. There may be other site-modified files in other locations of the sharedroot that this site wishes to save, but beware that the sharedroot will also contain many gigabytes of data from old versions of the programming environment (PE) software.

- a. Determine which files in the sharedroot default, class, or node views have been modified from the default.

Save the output from the `xtoprdump` command.

```
boot# xtopview
default:/:/ # xtoprdump -m
class:cnos:/etc/netconfig:1.2:*
class:cnos:/etc/opt/cray/modules/Base-opts.default.local:1.2:*
class:dvs:/etc/hosts.deny:1.2:*
class:lnet:/etc/modprobe.conf.local:1.3:*
class:lnet:/etc/sysconfig/infiniband:1.2:*
class:login:/etc/HOSTNAME:1.2:*
class:login:/etc/fstab:1.2:*
class:login:/etc/fstab.old:1.1:*
class:login:/etc/opt/cray/modules/Base-opts.default.local:1.2:*
class:login:/etc/sysconfig/network/routes:1.2:*
class:postproc:/etc/opt/cray/alps/nolaunch:1.1:*
class:postproc:/etc/opt/cray/modules/Base-opts.default.local:1.2:*
class:postproc:/etc/ssh/sshd_config:1.2:*
class:postproc:/etc/sysctl.conf:1.2:*
default:./etc/auto.master:1.2:*
default:./etc/group:1.5:*
default:./etc/group.old:1.4:*
default:./etc/hosts:1.3:*
default:./etc/init.d/.depend.boot:1.5:*
default:./etc/init.d/.depend.start:1.26:*
default:./etc/init.d/.depend.stop:1.25:*
default:./etc/init.d/opensmd:1.2:*
default:./etc/ld.so.cache:1.23:*
default:./etc/ldap.conf:1.2:*
default:./etc/mcelog/mcelog.conf:1.2:*
default:./etc/modprobe.conf.local:1.4:*
default:./etc/modprobe.d/dvs:1.2:*
default:./etc/modprobe.d/iomemory-vs14.conf:1.2:*
default:./etc/modprobe.d/unsupported-modules:1.2:*
default:./etc/motd:1.2:*
default:./etc/nsswitch.conf:1.2:*
default:./etc/ntp.conf:1.3:*
default:./etc/openldap/ldap.conf:1.2:*
default:./etc/opt/cray/alps/alps.conf:1.2:*
default:./etc/opt/cray/alps/alps.conf.unmerged:1.2:*
default:./etc/opt/cray/alps/alps.families:1.2:*
default:./etc/opt/cray/cnrte/roots.conf:1.2:*
default:./etc/opt/cray/hosts/service_alias.conf:1.2:*
default:./etc/opt/cray/llm/sessionid:1.50:*
default:./etc/opt/cray/nodehealth/nodehealth.conf:1.3:*
default:./etc/opt/cray/pdsh/machines:1.2:*
default:./etc/opt/cray/release/manifests/5.2.82:1.21:*
default:./etc/opt/cray/sdb/class_roles:1.2:*
default:./etc/opt/cray/sdb/node_classes:1.2:*
default:./etc/opt/cray/sdb/processor:1.2:*
default:./etc/pam.d/common-account-pc:1.3:*
default:./etc/pam.d/common-auth-pc:1.3:*
default:./etc/pam.d/common-password-pc:1.2:*
```

```
default::/etc/pam.d/common-session-pc:1.3:*
default::/etc/pam.d/sshd:1.2:*
default::/etc/passwd:1.6:*
default::/etc/passwd.old:1.5:*
default::/etc/resolv.conf:1.2:*
default::/etc/shadow:1.5:*
default::/etc/shadow.old:1.5:*
default::/etc/sysconfig/iomemory-vsl4:1.2:*
default::/etc/sysconfig/ldap:1.2:*
default::/etc/sysconfig/network/config:1.2:*
default::/etc/sysconfig/xt:1.2:*
node:10:/etc/sysconfig/infiniband:1.3:*
node:10:/etc/sysconfig/network/ifcfg-ib0:1.2:*
node:10:/etc/sysconfig/opensm:1.2:*
node:13:/etc/fstab:1.2:*
node:13:/etc/sysconfig/infiniband:1.3:*
node:13:/etc/sysconfig/network/ifcfg-bond0:1.4:*
node:13:/etc/sysconfig/network/ifcfg-eth0:1.1:*
node:13:/etc/sysconfig/network/ifcfg-ib0:1.3:*
node:13:/etc/sysconfig/network/ifcfg-ib1:1.3:*
node:13:/etc/sysconfig/network/routes:1.2:*
node:13:/etc/sysconfig/opensm:1.2:*
node:14:/etc/opt/cray/rsipd/rsipd.conf:1.1:*
node:14:/etc/sysconfig/infiniband:1.3:*
node:14:/etc/sysconfig/network/ifcfg-bond0:1.4:*
node:14:/etc/sysconfig/network/ifcfg-eth0:1.2:*
node:14:/etc/sysconfig/network/ifcfg-ib0:1.3:*
node:14:/etc/sysconfig/network/ifcfg-ib1:1.3:*
node:14:/etc/sysconfig/network/routes:1.2:*
node:14:/etc/sysconfig/opensm:1.2:*
node:14:/etc/sysctl.conf:1.2:*
node:14:/etc/udev/rules.d/77-network.rules:1.2:*
node:2:/etc/nginx/conf.d/dwrest.conf:1.1:*
node:2:/etc/opt/cray/dws/login.crt:1.1:*
node:2:/etc/opt/cray/dws/login.key:1.1:*
node:2:/etc/sysconfig/network/ifcfg-eth0:1.1:*
node:25:/etc/modprobe.d/dvs:1.1:*
node:25:/etc/sysconfig/irqbalance:1.1:*
node:25:/etc/sysconfig/network/ifcfg-bond0:1.2:*
node:25:/etc/sysconfig/network/ifcfg-eth0:1.1:*
node:25:/etc/sysconfig/network/ifcfg-eth1:1.1:*
node:25:/etc/sysconfig/network/ifcfg-eth2:1.1:*
node:25:/etc/sysconfig/network/ifcfg-eth3:1.1:*
node:25:/etc/sysctl.conf:1.1:*
node:26:/etc/modprobe.d/dvs:1.1:*
node:26:/etc/sysconfig/irqbalance:1.1:*
node:26:/etc/sysconfig/network/ifcfg-bond0:1.2:*
node:26:/etc/sysconfig/network/ifcfg-eth0:1.1:*
node:26:/etc/sysconfig/network/ifcfg-eth1:1.1:*
node:26:/etc/sysconfig/network/ifcfg-eth2:1.2:*
node:26:/etc/sysconfig/network/ifcfg-eth3:1.2:*
node:26:/etc/sysctl.conf:1.1:*
node:5:/etc/sysconfig/network/ifcfg-ib0:1.1:*
node:6:/etc/exports:1.2:*
node:6:/etc/fstab:1.4:*
node:6:/etc/fstab.old:1.4:*
node:6:/etc/init.d/boot.local:1.1:*
node:6:/etc/opt/cray/dws/dwsd.yaml:1.1:*
node:6:/etc/sysconfig/network/ifcfg-eth0:1.1:*
node:6:/etc/sysconfig/syslog:1.1:*
node:6:/etc/syslog-ng/syslog-ng.conf:1.2:*
```

```
node:9:/etc/nginx/conf.d/dwrest.conf:1.1:*
node:9:/etc/opt/cray/dws/nid00009.crt:1.1:*
node:9:/etc/opt/cray/dws/nid00009.key:1.1:*
node:9:/etc/sysconfig/infiniband:1.4:*
node:9:/etc/sysconfig/network/ifcfg-eth0:1.1:*
node:9:/etc/sysconfig/network/ifcfg-ib0:1.2:*
node:9:/etc/sysconfig/network/ifcfg-ib1:1.1:*
node:9:/etc/sysconfig/opensm:1.2:*
```

- b. For any file that was modified, check the reason logged when `xtopview` was exited.

The `xtoprlog` command displays the RCS log with any comments about why the file was changed.

This example looks at the login class `/etc/fstab` file. Version 1.1 was the initial one, and version 1.2 has this comment "Initial installation of 5.2.82 20160916."

```
default:// # xtoprlog -c login /etc/fstab

RCS file: /.shared/base/class/login/etc/RCS/fstab,v
Working file: /.shared/base/class/login/etc/fstab
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2;      selected revisions: 2
description:
-----
revision 1.2
date: 2016/09/16 18:18:49; author: root; state: Exp; lines: +1 -1
Initial installation of 5.2.82 20160916
-----
revision 1.1
date: 2016/09/16 18:18:49; author: root; state: Exp;
Initial revision
=====
```

- c. Create a tar archive file of the entire sharedroot directory structure.

Several Linux files have changed between SLES 11 SP3 and SLES 12, so the SLES 11 SP3 default Linux files may not be appropriate to be copied into a SLES 12 system.

NOTE: The sharedroot specialized `/etc` is stored underneath `/rr/current/.shared` (from the boot node) or `.shared` when inside the default `xtopview`. Capture all of these files into the tar archive.

```
boot# xtopview
default:// # cd .shared
default://.shared # tar -zcf /software/sharedroot_etc.tar.tgz .
default:// # exit
boot# ls -l /rr/current/software/sharedroot_etc.tgz
-rw-r--r-- 1 root root 229744640 Nov 24 05:11 /rr/current/software/
sharedroot_etc.tar
```

- d. Look in the `.shared` directory structure.

The directory structure underneath `.shared` has several layers. The files from the default view are under `base/default`. For class specialized files, look under `base/class` and then the name of the node class. For node specialized files, look under `base/node` and then the `nid` number of the node—not the `cname`.

```

boot# xtopview
default:// # cd .shared
default://.shared # ls -l
total 224924
drwxr-xr-x  5 root root      4096 Sep 16 13:17 base
drwxr-xr-x 16 root root      4096 Sep 16 13:18 class
drwxr-xr-x  3 root root      4096 Sep 16 13:17 default
-rw-r--r--  1 root root 288374 Nov 23 08:22 log
drwxr-xr-x 18 root root      4096 Sep 16 13:18 node
-rw-r--r--  1 root root 36421 Sep 16 13:17 var-skel.tgz

default://.shared # ls -l base
total 12
drwxr-xr-x  6 root root 4096 Oct 10 13:56 class
drwxr-xr-x  3 root root 4096 Sep 16 13:17 default
drwxr-xr-x 11 root root 4096 Oct 10 13:48 node

```

- e. Examine the symbolic links under the default, class, and node directories.

The symbolic links can indicate whether a node had a link to the `base/default` version of the file, the `base/class` version of the file, or the `base/node` version of the file.

This example shows the links for different `/etc/fstab` files. Any node in the login class will use the `base/class/login` version of `/etc/fstab`. In this example, node 9 shows that it uses the default version of the file in `base/default/etc/fstab`, node 2 uses the login class specialized version of the file in `base/class/login/etc/fstab`, and node 6 uses a node-specialized version of the file in the `base/node/6/etc/fstab`.

```

boot# xtopview
default://.shared # ls -l default/etc/fstab
lrwxrwxrwx 1 root root 31 Sep 16 13:18 default/etc/fstab -> /.shared/base/
default/etc/fstab
default://.shared # ls -l class/login/etc/fstab
lrwxrwxrwx 1 root root 35 Sep 16 13:18 class/login/etc/fstab -> /.shared/
base/class/login/etc/fstab
default://.shared # ls -l node/9/etc/fstab
lrwxrwxrwx 1 root root 31 Sep 16 13:18 node/9/etc/fstab -> /.shared/base/
default/etc/fstab
default://.shared # ls -l node/2/etc/fstab
lrwxrwxrwx 1 root root 35 Sep 16 13:18 node/2/etc/fstab -> /.shared/base/
class/login/etc/fstab
default://.shared # ls -l node/6/etc/fstab
lrwxrwxrwx 1 root root 30 Sep 16 13:18 node/6/etc/fstab -> /.shared/base/
node/6/etc/fstab

```

- f. Save RUR (resource utilization reporting) plugin scripts, if any.

IMPORTANT: If RUR was configured in the CLE 5.2 / SMW 7.2 system with site-local data plugins or output plugins, then these RUR plugin scripts must be saved so they can be copied to the CLE 6.0 / SMW 8.0 system. They may need to be ported from SLES 11 SP3 to SLES 12 or otherwise changed for CLE 6.0 / SMW 8.0.

There may be other files from RUR customized plugins in the `sharedroot /opt/cray/rur/default/bin` directory. See the translation tables for RUR plugins in [Update cray_rur Worksheet](#) on page 273 to learn where these site-local RUR plugins are defined to get the pointer to their location on the sharedroot or other file system. Ensure that these RUR plugin scripts are saved in the archive so that they can be transferred to the CLE 6.0 / SMW 8.0 system.

2. Save configuration files from the SMW.

a. Archive these files and directories.

Archive all of `/etc`, all of `/opt/cray/hss/default/etc`, and all of `/opt/cray/sec/default`. The files and directories listed below are of particular interest because they contain information necessary for configuring the system in the CLE 6.0.UP03 / SMW 8.0.UP03 release.

```
/etc
  /etc/aliases
  /etc/bash.bashrc.local
  /etc/csh.cshrc.local
  /etc/dhcpd.conf
  /etc/group
  /etc/hosts
  /etc/hosts.allow
  /etc/hosts.deny
  /etc/nsswitch.conf
  /etc/ntp.conf
  /etc/opt/cray/admin
  /etc/opt/cray/capmc
  /etc/opt/cray/config_set
  /etc/opt/cray/cray_sec_actions_config
  /etc/opt/cray/dumpsys/plugins
  /etc/opt/cray/global
  /etc/opt/cray/imps
  /etc/opt/cray/llm
  /etc/opt/cray/modules
  /etc/multipath.conf
  /etc/opt/cray/share
  /etc/opt/cray/share/p0/lustre/.lctrl
  /etc/passwd
  /etc/resolv.conf
  /etc/rootkey.yaml
  /etc/rsyslog.conf
  /etc/shadow
  /etc/ssh
  /etc/ssl
  /etc/sysctl.conf
  /etc/sysconfig

/etc/opt/cray/llm/xttrim.conf

/home/crayadm
  /home/crayadm/auto*
  /home/crayadm/.xtdumpsys-plugin

/opt/cray/hss/default/etc
  /opt/cray/hss/default/etc/*.ini
  /opt/cray/hss/default/etc/*.conf
  /opt/cray/hss/default/etc/auto.*
  /opt/cray/hss/default/etc/bios_settings
  /opt/cray/hss/default/etc/phy_cmp_offset
  /opt/cray/hss/default/etc/pre_nodeup_mmrs
  /opt/cray/hss/default/etc/snowbush_phy_workaround1.mmrs
  /opt/cray/hss/default/etc/cab_json.sedc
  /opt/cray/hss/default/etc/blade_json.sedc

/opt/cray/hss/default/pm

/opt/cray/sec/default
  /opt/cray/sec/default/bin/check_xt.sh
```

```

/opt/cray/sec/default/bin/check_xt_wrapper.ex
/opt/cray/sec/default/rules
/opt/cray/sec/default/SEC_VARIABLES*
/opt/cray/sec/default/SHELL_VARIABLES*

/opt/xt-images/templates

/root/.ssh

/var/spool/rsyslog/local-rules

```

Note that the `/opt/cray/hss/default/etc/cab_json.sedc` and `/opt/cray/hss/default/etc/blade_json.sedc` files (in above list) need to be saved only if this site has customized them. Because the CLE 6.0 / SMW 8.0 versions of those files have better defaults, it is likely they will not require customization.

- b. Archive SuSEfirewall2 firewall settings and defined services in `/etc/sysconfig/SuSEfirewall2` and `/etc/sysconfig/SuSEfirewall2.d/services/`.

These SuSEfirewall2 firewall files are included in the archive of `/etc/sysconfig` in the previous substep. They are explicitly mentioned here because they will be needed for some comparison and analysis later in the migration process in [Check for Site Modifications to SMW Firewall and IP Tables](#) on page 322, along with the iptables that will be captured from the SMW probe (`iptables-save`) later in this procedure (step 6 on page 94).

The SuSEfirewall2 firewall files will be archived again just before the "Shutdown and Switch" phase to capture any changes that may have occurred since those files were saved in this procedure. At that time, further comparison will be needed to determine if those changes, if any, need to be migrated to CLE 6.0 / SMW 8.0.

- c. Save any HSS/CRMS files that were changed or added.

It will be useful to know which of the HSS/CRMS files in `/opt/cray/hss/7.2.0/etc` have been changed from what was delivered in the SMW 7.2 `lsb-cray-hss-crms` RPM and what extra files were added.

This `rpm` query looks for files that have been changed from the delivered RPM. In this example two files have been changed.

```

smw# rpm -Vvv lsb-cray-hss-crms 2>&1 | grep /opt/cray/hss/7.2.0/etc \
| grep " c " | egrep -v "\.\.\.\.\.\.\.\."
S.5....T c /opt/cray/hss/7.2.0/etc/auto.xtshutdown
S.5....T c /opt/cray/hss/7.2.0/etc/xtremoted/xtremoted.ini

```

The output format begins with an eight-character code followed by an attribute marker and then the file name. See the `rpm(8)` man page for the `-V` (`verify`) option. Each of the eight characters denotes the result of a comparison of attribute(s) of the file to the value of those attribute(s) recorded in the database. A single "." (period) means the test passed, while a single "?" (question mark) means the test could not be performed (e.g., file permissions prevent reading). Otherwise, these characters denote failure of the corresponding `rpm --verify` test:

- S: file **S**ize differs
- M: **M**ode differs (includes permissions and file type)
- 5: **M**D5 sum differs
- D: **D**evice major/minor number mismatch
- L: **r**ead**L**ink(2) path mismatch
- U: **U**ser ownership differs

G: Group ownership differs
T: mTime differs

Possible attribute markers are:

c: %config configuration file
d: %doc documentation file
g: %ghost file (i.e., the file contents are not included in the package payload)
l: %license license file
r: %readme readme file

An alternative approach is to see which files in `/opt/cray/hss/7.2.0/etc` are not from an RPM.

```
smw# rpm -q --whatprovides $(for i in /opt/cray/hss/default/etc/*; \
do readlink -f $i; done) | grep "is not owned" | grep -v "pid"
file /opt/cray/hss/7.2.0/etc/XT5m is not owned by any package
file /opt/cray/hss/7.2.0/etc/auto.pluto is not owned by any package
file /opt/cray/hss/7.2.0/etc/bc_coeff_file_timestamp is not owned by any
package
file /opt/cray/hss/7.2.0/etc/routing is not owned by any package
file /opt/cray/hss/7.2.0/etc/routing.prev is not owned by any package
file /opt/cray/hss/7.2.0/etc/rtr.discovery is not owned by any package
file /opt/cray/hss/7.2.0/etc/sm.ini is not owned by any package
file /opt/cray/hss/7.2.0/etc/xtbootsys-xtconsumer-args is not owned by any
package
file /opt/cray/hss/7.2.0/etc/xtbounce.ini is not owned by any package
file /opt/cray/hss/7.2.0/etc/xtccsetpoint.ini is not owned by any package
file /opt/cray/hss/7.2.0/etc/xtdiscover-bounce-cmd is not owned by any
package
file /opt/cray/hss/7.2.0/etc/xtdiscover-config-changes.diff is not owned by
any package
file /opt/cray/hss/7.2.0/etc/xtdiscover.ini is not owned by any package
```

Every non-RPM-owned file should be specifically evaluated to determine if it should be moved forward during the migration. Some systems have had changes made to various tuning settings or timeouts. Others may have had a workaround applied for a nonstandard routing configuration.

If `xtbounce.ini` exists, then the system has special settings required by `xtbounce` on SMW 7.2. These may need to be carried forward to SMW 8.0 to get clean bounces.

Check with Cray service personnel about the contents of these files.

3. Save installer files from the SMW.

Archive the files listed here. Note that the installer provides the flexibility to put installer files anywhere, so these files may be located elsewhere for this site.

```
/etc/sysset.conf
/home/crayadm/SMWinstall.conf
/home/crayadm/install.5.2.*/CLEinstall.conf* (from last software update)
/root/pe/install-cdt.yaml (from PE software install)
```

4. Save configuration files from the boot node.

Archive all of `/etc` and all of `/root`. The files and directories listed below are of particular interest because they contain information necessary for configuring the system in the CLE 6.0.UP03 / SMW 8.0.UP03 release.

```
/etc
/etc/bash.bashrc.local
```

```

/etc/csh.cshrc.local
/etc/dhcpd.conf
/etc/group
/etc/hosts
/etc/hosts.allow
/etc/hosts.deny
/etc/modprobe.conf
/etc/modprobe.conf.local
/etc/multipath.conf
/etc/my.cnf
/etc/nsswitch.conf
/etc/ntp.conf
/etc/opt/cray
/etc/resolv.conf
/etc/rsyslog.conf
/etc/shadow
/etc/ssh
/etc/ssl
/etc/sysctl.conf
/etc/sysconfig
/etc/xhostname
/root
/root/.my.cnf
/root/.odbc.ini.root
/root/.ssh

```

5. Save files from the user file system (ufs) node.

The ufs node, the syslog node, and the SDB node are often the same node. Archive the files and directories listed here.

```

/ufs/home/crayadm
/ufs/alps_shared/apschedNextId
/ufs/alps_shared/apschedPDomain

```

- Other files in `/ufs/home` may be archived as well, at the discretion of the site system administrator.
- The `/ufs/alps_shared/apschedNextId` file can be saved to ensure that apids and resids continue from the highest number in the CLE 5.2 / SMW 7.2 system rather than restarting from 1. If saved, this file will put on the `/alps_shared` file system mounted by the SDB node on the CLE 6.0 / SMW 8.0 system later in the migration process.
- The `/ufs/alps_shared/apschedPDomain` file can be saved to preserve protection domains. If saved, this file will be put on the `/alps_shared` file system mounted by the SDB node on the CLE 6.0 / SMW 8.0 system later in the migration process.

6. Actively probe the running SMW.

- a. Use the suggested commands to collect actual setting values such as network settings, because some original settings might have been overridden by site-local scripts.

The first two commands start a typescript just for the following commands, because it is easier to search through this typescript file than to search through the one for the entire archival process.

```

smw# export TODAY=`date +%Y%m%d`
smw# script -af ${TODAY}.migration.probe.smw
smw#
    date
    ifconfig -a
    netstat -rn

```

```

hostid
hostname
dmidecode
lspci
lsscsi
fdisk -l
SMdevices
df -k
sysctl -a
cat /proc/cmdline
sshd -T
module list
xtcli part_cfg show
xtcli status s0
xthwinv s0
xthwinv -X s0
xtalive
xtshow_alert
xtshow_class
xtshow_compute
xtshow_diag
xtshow_disabled
xtshow_empty
xtshow_error
xtshow_halt
xtshow_network
xtshow_noflags
xtshow_not_empty
xtshow_off
xtshow_on
xtshow_ready
xtshow_reserve
xtshow_service
xtshow_standby
xtshow_topology
xtshow_warn
chkconfig -l
xtdaemonconfig | grep stonith
ls -l /opt/cray/cdt
iptables-save
smw# exit

```

b. Check PE software versions.

The above list of commands includes the command to check which versions of the PE software release have been installed on the CLE 5.2 / SMW 7.2 system. Any version installed on the CLE 5.2 / SMW 7.2 system prior to CDT 15.09, released in September 2015, cannot be installed on CLE 6.0 / SMW 8.0. However all of the newer releases can be. This example output shows 12 monthly releases of PE software have been installed on the CLE 5.2 / SMW 7.2 system.

```

smw# ls -l /opt/cray/cdt
total 48
drwxr-xr-x 3 root root 4096 Nov 20 2015 15.11
drwxr-xr-x 3 root root 4096 Nov 20 2015 15.12
drwxr-xr-x 3 root root 4096 Dec 18 2015 16.01
drwxr-xr-x 3 root root 4096 Jan 29 2016 16.02
drwxr-xr-x 3 root root 4096 Feb 12 2016 16.03
drwxr-xr-x 3 root root 4096 Apr 6 2016 16.04
drwxr-xr-x 3 root root 4096 May 26 2016 16.06
drwxr-xr-x 3 root root 4096 Jul 8 01:07 16.07

```

```
drwxr-xr-x 3 root root 4096 Jul 22 20:09 16.08
drwxr-xr-x 3 root root 4096 Aug 23 23:50 16.09
drwxr-xr-x 3 root root 4096 Sep 29 01:21 16.10
drwxr-xr-x 3 root root 4096 Oct 13 02:14 16.11
```

7. Actively probe the boot node of the running system.

Use the suggested commands to collect actual setting values such as network settings, since some original settings might have been overridden by site-local scripts.

```
smw# export TODAY=`date +%Y%m%d`
smw# script -af ${TODAY}.migration.probe.boot
smw# ssh boot
boot#
    date
    hostname
    ifconfig -a
    netstat -rn
    lspci
    lsscsi
    fdisk -l
    df -k
    chkconfig -l
    lsblk
    sysctl -a
    cat /proc/cmdline
    sshd -T
    module list
    ls -l /bin/ping
    iptables-save
boot# exit
smw# exit
```

8. Actively probe all non-boot service nodes of the running system.

Use the suggested commands to collect actual setting values such as network settings, since some original settings might have been overridden by site-local scripts.

This can be done via a script run on the boot node. For this example, the two nodes being excluded from the machines file are the alternate boot and alternate SDB nodes, which are in standby mode and are unable to respond to `ssh`.

```
smw# export TODAY=`date +%Y%m%d`
smw# script -af ${TODAY}.migration.probe.servicenodes
smw# ssh boot
boot# for i in $(cat /etc/opt/cray/pdsh/machines | egrep -v "c0-0c0s4n1|
c0-0c0s5n1")
do
    echo "Probing $i"
    echo date
    ssh $i date
    echo hostname
    ssh $i hostname
    echo ifconfig -a
    ssh $i ifconfig -a
    echo netstat -rn
    ssh $i netstat -rn
    echo lspci
    ssh $i lspci
    echo lsscsi
```

```

ssh $i lsscsi
echo fdisk -l
ssh $i fdisk -l
echo df -k
ssh $i df -k
echo chkconfig -l
ssh $i chkconfig -l
echo lsblk
ssh $i lsblk
echo sysctl -a
ssh $i sysctl -a
echo cat /proc/cmdline
ssh $i cat /proc/cmdline
echo sshd -T
ssh $i sshd -T
echo module list
ssh $i module list
echo ls -l /bin/ping
ssh $i ls -l /bin/ping
echo ls -l /opt/cray/nodehealth/default/bin/pcmd
ssh $i ls -l /opt/cray/nodehealth/default/bin/pcmd
echo iptables-save
ssh $i iptables-save

done
boot# exit
smw# exit

```

9. (SMW HA sites only) Probe the active SMW and record the cluster configuration.

Save the following information about the CLE 5.2 / SMW 7.2 configuration. This should be run from the active SMW.

- a. Start a typescript for the SMW HA probe output.

```

smw# export TODAY=`date +%Y%m%d`
smw# script -af ${TODAY}.migration.probe.smwha

```

- b. Display the virtual host name.

```

smw# crm resource param fsync show virtual_hostname
stplabha-smw

```

- c. Display the virtual IP address.

```

smw# crm resource param ClusterIP show ip
172.31.73.165

```

- d. Display the iDRAC6 IP addresses.

```

smw# crm resource param stonith-1 show ipaddr
172.31.73.142

smw# crm resource param stonith-2 show ipaddr
172.31.73.77

```

- e. (DRBD PMDB sites only) Display the DRBD configuration.

```

smw# cat /etc/drbd.d/r0.res
(...)

```

- f. Display the file synchronization list.

```
smw# cat /etc/csync2/csync2_cray.cfg  
(...)
```

- g. Exit the SMW HA probe typescript.

```
smw# exit
```

6.3 Read Man Pages for New Commands

The migration SMW has the CLE 6.0.UP03 / SMW 8.0.UP03 release installed on it. On the migration SMW, read the following man pages to increase familiarity with the new commands related to IMPS (Image Management and Provisioning System), CMF (Configuration Management Framework), NIMS (Node Image Mapping Service), and others.

- IMPS: recipe, pkgcoll, repo, image
- CMF: cfgset, cray-ansible, ansible_cfg_search
- NIMS: cmap, cnode
- General: imgbuilder, snaputil, cnat

6.4 Transfer Configuration Data to Configuration Worksheets

This part of the migration process takes the configuration information extracted from the CLE 5.2 / SMW 7.2 system (in [Extract Configuration Data from the CLE 5.2 / SMW 7.2 System](#) on page 86) and enters it into configuration worksheets on the migration SMW, which is running CLE 6.0 / SMW 8.0.

How to Interpret the Level Assigned to a Configuration Variable

The procedures that follow provide instructions on how to enter configuration data in each configuration service worksheet. An explanation of the purpose of each variable, its type (string, boolean, list, multival, etc.), allowed or recommended values, and what regular expression (regex) is used to validate that setting can be found in each worksheet. The worksheet also indicates the configuration *level* assigned to each variable, which is one of these three values:

- required** Settings that must be set or the system will not function. This level is used primarily for services with settings that are not provided with a default value by the configuration template, usually because no reasonable default value exists. The config set will not validate if any required settings are skipped (i.e., left unset).
- basic** Settings that are likely to be used by most sites. If a **basic** setting is left unset, the template-provided default is used.
- advanced** Optional settings that are likely to be used only by advanced users to tune a service. If an **advanced** setting is left unset, the template-provided default is used.

How to Interpret Variable Names

In the configurator and configuration worksheets, variable names can be quite long because they are composed of a data structure hierarchy. Each variable name begins with the name of the service to which it belongs. The next part of each name is always 'settings' to indicate that what follows is a *service setting*, one of the available settings for that service. After 'settings' comes the name of the setting, which could be a simple data type (string, boolean, integer, etc.) or a more complex data type (list, multival, etc.). The next part after the name of the setting is always 'data' to indicate that what follows is one of the fields of that setting. For a full description of data types, see *XC™ Series Configurator User Guide (S-2560)*.

For example, here is the variable for the IP address of the high-speed network (HSN), one of several networks.

```
cray_net.settings.networks.data.hsn.ipv4_network
```

This variable belongs to the `cray_net` service and the `networks` setting of that service. The `networks` setting is of type multival, which means it can have multiple entries, and each entry can have multiple fields to set. This variable targets the `ipv4_network` field of the `hsn` network entry.

This example shows the variable for the IP address of the HSN SDB node alias interface (one of several interfaces) of the SDB node (one of several hosts).

```
cray_net.settings.hosts.data.sdbnode.interfaces.hsn_sdb_alias.ipv4_address
```

This variable belongs to the `cray_net` service and the `hosts` setting of that service. The `hosts` setting is of type multival, and this variable belongs to the `sdbnode` host entry. The `sdb_node` host has a field `interfaces`, which is also of type multival. This variable targets the `ipv4_address` field of the `hsn_sdb_alias` interface entry.

How to Interpret Translation Tables

Most of these procedures provide a translation table, which lists the variables that must be set and how to find the information needed for that variable on the currently running CLE 5.2 / SMW 7.2 system. If a procedure does not have a translation table, it means that the variables in that configuration service have no analog in the CLE 5.2 / SMW 7.2 software.

Title. Each translation table has a title. Because variable names can be too long to fit neatly within a table column, the title shows the portion of the full variable name that is common to each variable in the table. The portion of the variable name that is unique to that variable is shown in the first column of the table.

Columns. Each translation table has these five columns:

Setting/ Field Name	The portion of the variable name that is unique to this variable. Append this to the common portion of the variable name shown in the table title to reconstruct the full name of the variable in the CLE 6.0 / SMW 8.0 configuration service.
Default	Default value for the variable. Note that empty brackets mean that the default value is an empty list.
Level	Assigned configuration level of the variable (required, basic, or advanced).
Probe	A value ("probe (<i>n</i>)" or N/A) that indicates whether there is a suggested command to use on the CLE 5.2 / SMW 7.2 system to probe for the information needed to set the value of the CLE 6.0 / SMW 8.0 variable. If "probe (<i>n</i>)" appears in this column, find the command for that probe in the <i>n</i> th footnote of the table. The command prompt will indicate the host name where the command should be run.

Files/ Installer	<p>File(s) and/or installer on the currently running system that have the information needed to set the value of the CLE 6.0 / SMW 8.0 variable.</p> <p>Information about files begins with the context of the file (SMW, bootroot, default sharedroot, class sharedroot, or node sharedroot) followed by the file path. After the file path, a variable name within the file may be included, shown in parentheses. If no context is indicated, then it is assumed to be the SMW.</p> <p>Information about the installer configuration file begins with which installer (<code>SMWinstall.conf</code> or <code>CLEinstall.conf</code>) to look in. This is followed by the variable within that file, shown in parentheses, that has the information needed.</p>
-----------------------------	---

6.4.1 Prepare Global Worksheets for Migration

Prerequisites

This procedure assumes that SMW 8.0.UP03 and CLE 6.0.UP03 software has been installed. Patches for the base release will be applied later in the migration process.

About this task

The global config set must be updated with site-specific information about several services. This procedure prepares the global configuration worksheets for editing (in subsequent procedures) to include site-specific configuration data.

When editing configuration worksheets, a general rule is to uncomment all settings that are marked `level=basic` and modify values as needed. All settings that remain commented are considered unconfigured. Some settings are already uncommented in the original worksheet; Cray recommends not modifying those preconfigured settings because they are needed for proper configuration of the system. For more information about editing configuration worksheets and updating config sets, see *XC™ Series Configurator User Guide (S-2560)*.

NOTE: (SMW HA only) For SMW HA systems, config set operations need to be performed on only one SMW because the config sets are shared between both SMWs in the SMW HA pair.

Procedure

1. Generate configuration worksheets for the global config set using prepare mode and the no-scripts option. In this example, the config set is named `global_example`.

```
smw# cfgset create -m prepare -t global --no-scripts global_example
```

2. Save a copy of original global worksheets.

Copy the original configuration worksheets into a new directory to preserve them in case they are needed later for comparison.

```
smw# ls -l /var/opt/cray/imps/config/sets/global_example/worksheets
```

```
smw# cp -a /var/opt/cray/imps/config/sets/global_example/worksheets \  
/var/opt/cray/imps/config/sets/global_example/worksheets.orig
```

3. Make a work area for global worksheets.

Make a work area and copy the global configuration worksheets to that work area for editing. The worksheets should not be edited in their original location for two reasons: (1) the configurator will not permit updating a config set from worksheets within that config set, and (2) edits would be overwritten when the config set is updated.

```
smw# cp -a /var/opt/cray/imps/config/sets/global/worksheets \
/var/adm/cray/release/global_worksheet_workarea
```

4. Change to the work area directory to simplify the editing commands in the following procedures.

```
smw# cd /var/adm/cray/release/global_worksheet_workarea
```

5. Edit and update the global configuration worksheets using the procedures that follow.

Many procedures provide translation tables, which list the variables that can be set and how to find the information needed for that variable on the currently running CLE 5.2 / SMW 7.2 system. Some procedures do not have translation tables because the variables in those configuration services have no analog in the CLE 5.2 / SMW 7.2 software. All of the worksheets in the global config set must be edited and updated for a successful migration to the new system software.

NOTE: Skip `cray_network_boot_packages_worksheet.yaml`, which is enabled by default and has no variables that need to be changed.

Use [Migration Checklist 2.2: Transfer Global Configuration Data](#) on page 435 to track progress updating the worksheets.

6.4.1.1 Update `cray_bootraid` Worksheet in Global Config Set

Prerequisites

This procedure assumes that a work area has been set up for editing global configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/global_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Cray boot RAID service configures storage for the boot RAID. Until the SMW is connected to the boot RAID, the actual values needed for the devices in the LVM volume groups for the SMW, boot node, and SDB node cannot be added to this worksheet. However, all of the other settings can be prepared.

There are two storage sets to configure: a CLE default storage set (`cledefault`) and an SMW default storage set (`smwdefault`). Each storage set may have multiple volume groups defined for the node (owner) that will mount the file systems in volumes of the volume group.

Procedure

1. Edit the `cray_bootraid` configuration worksheet.

```
smw# vi cray_bootraid_worksheet.yaml
```

2. Enable the `cray_bootraid` service.

Uncomment `cray_bootraid.enabled` and set it to `true`.

CONFIGURE THE CLE DEFAULT STORAGE SET (`cledefault`) VOLUME GROUPS

The full name of settings within each CLE default volume group looks like

`cray_bootraid.settings.storage_sets.data.cledefault.volume_groups`, followed by `<volume group name>.<field name>`. For brevity, the steps in the `cledefault` section show only the volume group name and field name of each setting.

3. Configure the boot node volume group (`boot_node_vg`).

- a. Set the owner of the boot node volume group.

Uncomment `boot_node_vg.owner` and ensure that it is set to "boot" rather than a cname. For a partitioned system, include the partition name (e.g., "boot-p2" for partition p2).

- b. Uncomment `boot_node_vg.devices` (physical volumes) that are going to be part of the boot node LVM volume group.

Add a list of persistent device names such

as `/dev/disk/by-id/scsi-360080e50002f7160000014905640c0c4` for each physical volume.

Do not use short names like `/dev/sdn`, which may vary from node to node (SMW, boot node, and SDB node), and may vary from boot to boot of a particular node.

- c. For each volume of the boot node volume group, change file system size to meet site needs.

The `home` volume corresponds to the `/cray_home` file system in the table, the `imps` volume corresponds to `/var/opt/cray/imps`, and the `nvolatile` volume corresponds to `/non_volatile`. These are the minimum values for file system size. When space is available, Cray recommends increasing `nvolatile.fs_size` to 200.

```
boot_node_vg.volumes.home.fs_size: '50'
boot_node_vg.volumes.imps.fs_size: '250'
boot_node_vg.volumes.nvolatile.fs_size: '50'
```

Expanding storage space. The LUN sizes for the `/cray_home` and `/non_volatile` file systems may need to be adjusted depending on site usage of those file systems. For example, workload managers, DataWarp, and any node that needs permanent storage can store information in `/non_volatile`, so it may need to be larger than the suggested size. If size adjustment is not made at install time, it can be made later. See *XC™ Series System Administration Guide (S-2393)* for instructions on how to expand storage in a file system, volume, or volume group.

Table 9. Boot Node RAID Requirements

Owning Node	Mount Point	FS Type	Recommended Size for 4.5 TB RAID	Minimum Size	Description
boot	/cray_home	xfs	50 GB	50 GB	Home directories on CLE
boot	/var/opt/cray/imps	btrfs	250 GB	250 GB	IMPS data for PE image roots and for Netroot compute-large and login-large image roots
boot	/non_volatile	xfs	200 GB	50 GB	persistent data, including /var if necessary, for service nodes provided from boot node

4. Configure the SDB node volume group (`sdb_node_vg`).

a. Set the owner of the SDB node volume group.

Uncomment `sdb_node_vg.owner` and ensure that it is set to "sdb" rather than a cname. For a partitioned system, include the partition name (e.g., "sdb-p2" for partition p2).

b. Uncomment `sdb_node_vg.devices` (physical volumes) that are going to be part of the SDB node LVM volume group.

Add a list of persistent device names such

as `/dev/disk/by-id/scsi-360080e50002f7160000014905640c0c4` for each physical volume.

Do not use short names like `/dev/sdn`, which may vary from node to node (SMW, boot node, and SDB node), and may vary from boot to boot of a particular node.

c. For each volume of the SDB node volume group, change file system size to meet site needs.

The `db` volume corresponds to the `/var/lib/mysql` file system in the table, and the `alps` volume corresponds to `/alps_shared`. These are the minimum settings. When space is available, Cray recommends increasing the size for both of these file systems to 20.

```
sdb_node_vg.volumes.db.fs_size: '10'
sdb_node_vg.volumes.alps.fs_size: '10'
```

Table 10. SDB Node RAID Requirements

Owning Node	Mount Point	FS Type	Recommended Size for 4.5 TB RAID	Minimum Size	Description
sdb	/var/lib/mysql	xfs	20 GB	10 GB	SDB database
sdb	/alps_shared	xfs	20 GB	10 GB	ALPS data

5. Configure the compute node local volume group (`compute_node_local_vg`).

This volume group is needed only for systems that use compute nodes with onboard SSDs. CLE 5.2 / SMW 7.2 systems being migrated do not have such compute nodes, so this step simply sets the owner field to null. That ensures that this volume group is not used but is preserved in case this site decides to add compute nodes with SSDs to the system later.

Use the > key to skip the other `compute_node_local` settings.

```
compute_node_local.owner: null
compute_node_local.devices: >
```

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.compute_node_local.volumes
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $ <cr>
```

6. Configure the compute node volume group (`compute_node_local`).

This is the third of three predefined volume groups in the `cledefault` storage set. This volume group is needed only for systems that use compute nodes with onboard SSDs. Systems running CLE 5.2 / SMW 7.2 do not have such compute nodes, so this step simply sets the owner field to null. That ensures that this volume group is not used but is preserved in case this site decides to add compute nodes with SSDs to the CLE 6.0 / SMW 8.0 system after migration is completed.

Find the `compute_node_local.owner` setting.

```
#cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.compute_node_local.owner: ''
```

Uncomment it and set it to null.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.compute_node_local.owner: null
```

CONFIGURE THE SMW DEFAULT STORAGE SET (`smwdefault`) VOLUME GROUPS

The full name of settings within each SMW default volume group looks like

`cray_bootraid.settings.storage_sets.data.smwdefault.volume_groups.` followed by `<volume group name>.<field name>`. For brevity, the steps in the `smwdefault` section show only the volume group name and field name of each setting.

7. Configure the SMW node volume group (`smw_node_vg`).

a. Set the owner of the SMW node volume group.

Ensure that `smw_node_vg.owner` is set to "smw." It could also be set to the host name of the SMW, such as "orion-smw," but there is greater portability of the config set if "smw" is used.

b. Uncomment `smw_node_vg.devices` (physical volumes) that are going to be part of the SMW node LVM volume group.

Leave this variable set to an empty list for now. Later in the process, after the SMW is connected to the boot RAID and the LUNs have been created on the boot RAID, add a list of persistent device names such as `/dev/disk/by-id/scsi-360080e50002f7160000014905640c0c4` for each physical volume. Do not use short names like `/dev/sdn`, which may vary from node to node (SMW, boot node, and SDB node), and may vary from boot to boot of a particular node.

c. For each volume of the SMW node volume group, change file system size to meet site needs.

The `home` volume corresponds to the `/home` file system in the table, the `db` volume corresponds to `/var/lib/mysql`, the `log` volume corresponds to `/var/opt/cray/disk/1`, the `imps` volume corresponds to `/var/opt/cray/imps`, and the `repos` volume corresponds to `/var/opt/cray/repos`. These are the minimum settings. When space is available, Cray recommends increasing the file system sizes.

```
smw_node_vg.volumes.home.fs_size: '40'
smw_node_vg.volumes.db.fs_size: '10'
smw_node_vg.volumes.log.fs_size: '400'
smw_node_vg.volumes.imps.fs_size: '400'
smw_node_vg.volumes.repos.fs_size: '100'
```

The volume for the `/var/opt/cray/imps` file system on the SMW should be significantly larger than the volume for the `/var/opt/cray/imps` file system on the boot node. This is because that file system on the SMW contains boot images, config sets, and image roots, while that file system on the boot node contains only a subset of the image roots on the SMW. The boot node does an NFS mount of the SMW boot images, so no local space is needed for those.

Table 11. SMW RAID Requirements

Mount Point	FS Type	Recommended Size for 4.5 TB RAID	Minimum Size	Size for 9.0 TB RAID	Description
<code>/home</code>	xfs	200 GB	40 GB	200 GB	Home directories on SMW
<code>/var/lib/mysql</code>	btrfs	10 GB	10 GB	10 GB	HSS database
<code>/var/opt/cray/disk/1</code>	xfs	1000 GB	400 GB	2000 GB	logs, debug, dumps
<code>/var/opt/cray/imps</code>	xfs	1000 GB	400 GB	1000 GB	IMPS data
<code>/var/opt/cray/repos</code>	btrfs	200 GB	100 GB	200 GB	IMPS repos

6.4.1.2 Update `cray_firewall` Worksheet in Global Config Set

Prerequisites

This procedure assumes that a work area has been set up for editing global configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/global_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The global Cray firewall service is a centralized mechanism for restricting packet traffic from various networks. This procedure enables this service using the global `cray_firewall` configuration worksheet.

Procedure

1. Edit `cray_firewall_worksheet.yaml`.

```
smw# vi cray_firewall_worksheet.yaml
```

2. Uncomment `cray_firewall.enabled` and set it to `true`.

6.4.1.3 Update `cray_global_net` Worksheet in Global Config Set

Prerequisites

This procedure assumes that a work area has been set up for editing global configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/global_worksheet_workarea
```

About this task

The global networking configuration worksheet defines key network attributes within the system. It is necessary to define these attributes in order to produce a functional system. This procedure enables the service and configures some settings using the `cray_global_net` configuration worksheet. It also provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the global config set for CLE 6.0 / SMW 8.0.

There are two major sections to `cray_global_net`. One describes the networks to which the SMW is connected and the other describes the hosts (`primary_smw` is the only host) and the network interfaces on `primary_smw` that are on those networks.

Procedure

1. Edit `cray_global_net_worksheet.yaml`.

```
smw# vi cray_global_net_worksheet.yaml
```

2. Uncomment `cray_global_net.enabled` and ensure that it is set to `true`.

MIGRATE DATA FOR THE NETWORK SETTINGS

3. Prepare to migrate the networks configuration data.

Search in the file for `'networks' DATA`, then uncomment all of the lines below it that begin with `cray_global_net.settings.networks` so that those settings will be applied and marked as configured. They define four networks: `admin`, `SMW failover`, `HSS`, and `management`.

NOTE: Do NOT uncomment the similar lines under this heading, because they are examples only and are not configured for these four networks.

```
# ** EXAMPLE 'networks' VALUE (with current defaults) **
```

Other notes:

- If entering a value for a string setting that currently is set to `' '` (empty string), remove the quotes before entering the new value. For example, `ipv4_network: ' '` becomes `ipv4_network: 10.1.0.0`. In cases where the string value might be interpreted as a number, retain the single quotes. For example, a string setting with value `'512'` needs quotes.
- If entering one or more values for a list setting that is currently set to `[]` (empty list), remove the brackets and add each entry on a separate line, beginning with `"- "` (a dash and a space). For example, the `dns_servers`, `dns_search`, and `ntp_servers` settings are lists that can have multiple entries, and they should look like this:

```
cray_global_net.settings.networks.data.management.dns_servers:
- 172.31.84.40
- 172.30.84.40
- 172.28.84.40
```

- Do NOT change or remove the null value in lines like this that appear at the beginning of each set of network definitions. This line sets the key for that network definition, which in this example is "management."

```
cray_global_net.settings.networks.data.name.management: null
```

4. Migrate data for the admin network.

Enter SMW-specific or site-specific values for these admin network items.

```
cray_global_net.settings.networks.data.admin.ipv4_network:
cray_global_net.settings.networks.data.admin.ipv4_netmask:
cray_global_net.settings.networks.data.admin.ipv4_broadcast:
cray_global_net.settings.networks.data.admin.ipv4_gateway:
cray_global_net.settings.networks.data.admin.dns_servers:
cray_global_net.settings.networks.data.admin.dns_search:
cray_global_net.settings.networks.data.admin.ntp_servers:
```

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to the admin network settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Note that for global configuration services, no context is provided in the Files/Installer column because it is assumed to be the SMW.

Table 12. Variables beginning with `cray_global_net.settings.networks.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
name.admin	null	required	N/A	N/A
admin.description	Network that connects the SMW, boot and SDB nodes.	basic	N/A	N/A
admin.ipv4_network	10.3.0.0	required	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth3
admin.ipv4_netmask	255.255.0.0	required	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth3
admin.ipv4_broadcast		advanced	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth3
admin.ipv4_gateway		basic	N/A	N/A
admin.dns_servers	[]	basic	N/A	N/A
admin.dns_search	[]	basic	N/A	N/A
admin.ntp_servers	[]	basic	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
admin.fw_external	false	basic	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) smw# ifconfig eth3				

5. Migrate data for the SMW failover network.

Enter SMW-specific or site-specific values for these SMW failover network items.

```
cray_global_net.settings.networks.data.smw_failover.ipv4_network:
cray_global_net.settings.networks.data.smw_failover.ipv4_netmask:
cray_global_net.settings.networks.data.smw_failover.ipv4_broadcast:
cray_global_net.settings.networks.data.smw_failover.ipv4_gateway:
cray_global_net.settings.networks.data.smw_failover.dns_servers:
cray_global_net.settings.networks.data.smw_failover.dns_search:
cray_global_net.settings.networks.data.smw_failover.ntp_servers:
```

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to the SMW failover network settings.

Table 13. Variables beginning with `cray_global_net.settings.networks.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
name.smw_failover	null	required	N/A	N/A
smw_failover.description	Network that connects the HA smw pair.	basic	N/A	N/A
smw_failover.ipv4_network	10.2.0.0	required	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth2
smw_failover.ipv4_netmask	255.255.0.0	required	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth2
smw_failover.ipv4_broadcast		advanced	N/A	N/A
smw_failover.ipv4_gateway		basic	N/A	N/A
smw_failover.dns_servers	[]	basic	N/A	N/A
smw_failover.dns_search	[]	basic	N/A	N/A
smw_failover.ntp_servers	[]	basic	N/A	N/A
smw_failover.fw_external	false	basic	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) smw# ifconfig eth2				

6. Migrate data for the HSS network.

Enter SMW-specific or site-specific values for these HSS network items.

```
cray_global_net.settings.networks.data.hss.ipv4_network:
cray_global_net.settings.networks.data.hss.ipv4_netmask:
cray_global_net.settings.networks.data.hss.ipv4_broadcast:
cray_global_net.settings.networks.data.hss.ipv4_gateway:
cray_global_net.settings.networks.data.hss.dns_servers:
cray_global_net.settings.networks.data.hss.dns_search:
cray_global_net.settings.networks.data.hss.ntp_servers:
```

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to the HSS network settings.

Table 14. Variables beginning with `cray_global_net.settings.networks.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
name.hss	null	required	N/A	N/A
hss.description	Network connecting the SMW to the controllers.	basic	N/A	N/A
hss.ipv4_network	10.1.0.0	required	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth1
hss.ipv4_netmask	255.255.0.0	required	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth1
hss.ipv4_broadcast		advanced	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth1
hss.ipv4_gateway		basic	N/A	N/A
hss.dns_servers	[]	basic	N/A	N/A
hss.dns_search	[]	basic	N/A	N/A
hss.ntp_servers	[]	basic	N/A	N/A
hss.fw_external	false	basic	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) <code>smw# ifconfig eth1</code>				

7. Migrate data for the management network.

Enter SMW-specific or site-specific values for these management network items.

NOTE: If this site does not use DNS search but does use DNS domain in `/etc/resolv.conf`, then adding a single entry to the `dns_search` setting is functionally equivalent to setting the DNS domain.

```

cray_global_net.settings.networks.data.management.ipv4_network:
cray_global_net.settings.networks.data.management.ipv4_netmask:
cray_global_net.settings.networks.data.management.ipv4_broadcast:
cray_global_net.settings.networks.data.management.ipv4_gateway:
cray_global_net.settings.networks.data.management.dns_servers:
cray_global_net.settings.networks.data.management.dns_search:
cray_global_net.settings.networks.data.management.ntp_servers:

```

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to the management network settings.

NOTE: If CLE 5.2 / SMW 7.2 values for the management/customer network will be used on the physical migration SMW, then either that SMW or the CLE 5.2 / SMW 7.2 must be disconnected from the network. They cannot both have the same IP address on a shared network.

Table 15. Variables beginning with `cray_global_net.settings.networks.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
name.management	null	required	N/A	N/A
management.description	Customer network connected to the SMW. In some cases the same as the login network.	basic	N/A	N/A
management.ipv4_network		required	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth0
management.ipv4_netmask		required	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth0
management.ipv4_broadcast		advanced	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth0
management.ipv4_gateway		basic	<i>probe (2)</i>	files: /etc/sysconfig/network/routes (default)
management.dns_servers	[]	basic	N/A	files: /etc/resolv.conf (nameserver) /etc/sysconfig/network/config (NETCONFIG_DNS_STATIC_SERVERS)
management.dns_search	[]	basic	N/A	files: /etc/resolv.conf (search)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/sysconfig/network/config (NETCONFIG_DNS_STATIC_SEARCHLIST)
management.ntp_servers	[]	basic	N/A	<i>files:</i> /etc/ntp.conf (servers) <i>installer:</i> SMWinstall.conf (NTPServers)
management.fw_external	false	basic	N/A	<i>files:</i> /etc/sysconfig/network/config (FIREWALL) /etc/sysconfig/SuSEfirewall2 (FW_DEV_EXT (if eth0 is in this variable))
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) smw# ifconfig eth0 (2) smw# netstat -rn				

Finally, set the management network external firewall to true.

```
cray_global_net.settings.networks.data.management.fw_external: true
```

MIGRATE DATA FOR THE HOST SETTINGS AND HOST NETWORK INTERFACE SETTINGS

- Prepare to migrate the host and host network interface configuration data.

Search in the file for 'hosts' DATA, then uncomment all of the lines that begin with `cray_global_net.settings.hosts` so that those settings will be applied and marked as configured. They define a host called "primary_smw" and two interfaces for it: one that connects to the customer management network and one that connects to admin nodes, such as the boot and SDB nodes.

- Migrate data for the host.

Enter SMW-specific or site-specific values for these items.

```
cray_global_net.settings.hosts.data.primary_smw.aliases:
cray_global_net.settings.hosts.data.primary_smw.hostid:
cray_global_net.settings.hosts.data.primary_smw.hostname:
```

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings.

NOTE: Cray recommends using a different host name and host ID for the physical migration SMW, therefore do NOT copy the following settings from the SMW running CLE 5.2 / SMW 7.2 to the migration SMW:

```
cray_global_net.settings.hosts.data.primary_smw.hostid
```

```
cray_global_net.settings.hosts.data.primary_smw.hostname
```

Table 16. Variables beginning with `cray_global_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>name.primary_smw</code>	<code>null</code>	required	N/A	
<code>primary_smw.description</code>		basic	N/A	
<code>primary_smw.aliases</code>	<code>[]</code>	basic	N/A	<i>files:</i> <code>/etc/hosts</code>
<code>primary_smw.roles</code>		basic	N/A	N/A
<code>primary_smw.hostid</code>		basic	<i>probe (1)</i>	N/A
<code>primary_smw.host_type</code>	<code>management</code>	basic	N/A	N/A
<code>primary_smw.hostname</code>		basic	<i>probe (2)</i>	<i>files:</i> <code>/etc/HOSTNAME</code> (without the domain name)
<code>primary_smw.standby_node</code>	<code>false</code>	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) <code>smw# hostid</code>				
(2) <code>smw# hostname</code>				

10. Migrate data for the host (`primary_smw`) customer Ethernet network interface.

Enter SMW-specific or site-specific values for these items.

```
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.aliases:
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.network:
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.ipv4_address:
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.mac:
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.startmode:
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.bootproto:
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.mtu:
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.extra_attributes:
cray_global_net.settings.hosts.data.primary_smw.interfaces.admin_interface.unmanaged_interface:
```

Set the `unmanaged_interface` field to `true`. This applies to both stand-alone SMWs and SMW HA systems. In the case of an SMW that is or will be configured for an SMW HA system, this prevents Ansible from managing `eth0` and `eth3` before the SMW HA cluster has been configured.

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings.

NOTE: Cray recommends using different Ethernet and mac addresses for the physical migration SMW, therefore do NOT copy the following settings from the SMW running CLE 5.2 / SMW 7.2 to the migration SMW:

```
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.ipv4_address
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.mac
```

Table 17. Variables beginning with `cray_global_net.settings.hosts.data.primary_smw.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.customer_ethernet</code>	null	required	N/A	N/A
<code>customer_ethernet.name</code>	eth0	required	N/A	N/A
<code>customer_ethernet.description</code>	Interface connecting to the customer management network.	basic	N/A	N/A
<code>customer_ethernet.aliases</code>	[]	basic	N/A	files: /etc/hosts
<code>customer_ethernet.network</code>	management	basic	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth0
<code>customer_ethernet.ipv4_address</code>		basic	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth0
<code>customer_ethernet.mac</code>	false	advanced	<i>probe (1)</i>	N/A
<code>customer_ethernet.startmode</code>	auto	advanced	N/A	files: /etc/sysconfig/network/ifcfg-eth0 (STARTMODE)
<code>customer_ethernet.bootproto</code>	static	basic	N/A	files: /etc/sysconfig/network/ifcfg-eth0 (BOOTPROTO)
<code>customer_ethernet.mtu</code>			<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth0 (MTU)
<code>customer_ethernet.extra_attributes</code>	[]	advanced	N/A	files: /etc/sysconfig/network/ifcfg-eth0
<code>customer_ethernet.module</code>		advanced	N/A	N/A
<code>customer_ethernet.params</code>		advanced	N/A	N/A
<code>customer_ethernet.unmanaged_interface</code>	false	advanced	N/A	N/A

Commands for probing the CLE 5.2 / SMW 7.2 system:

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
(1) smw# <code>ifconfig eth0</code>				

Note that if the customer Ethernet IP address changes, the output from the `hostid` command will be different. After changing the Ethernet setting

```
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.ipv4_address
```

ensure that this setting (the SMW host ID) is set to the output of the `hostid` command.

```
cray_global_net.settings.hosts.data.primary_smw.hostid
```

11. Migrate data for the host (primary_smw) admin network interface.

Enter SMW-specific or site-specific values for these items.

```
cray_global_net.settings.hosts.data.primary_smw.interfaces.admin_interface.aliases:
cray_global_net.settings.hosts.data.primary_smw.interfaces.admin_interface.network:
cray_global_net.settings.hosts.data.primary_smw.interfaces.admin_interface.ipv4_address:
cray_global_net.settings.hosts.data.primary_smw.interfaces.admin_interface.mac:
cray_global_net.settings.hosts.data.primary_smw.interfaces.admin_interface.startmode:
cray_global_net.settings.hosts.data.primary_smw.interfaces.admin_interface.bootproto:
cray_global_net.settings.hosts.data.primary_smw.interfaces.admin_interface.mtu:
cray_global_net.settings.hosts.data.primary_smw.interfaces.admin_interface.extra_attributes:
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.unmanaged_interface:
```

Set the `unmanaged_interface` field to `true`. This applies to both stand-alone SMWs and SMW HA systems. In the case of an SMW that is or will be configured for an SMW HA system, this prevents Ansible from managing `eth0` and `eth3` before the SMW HA cluster has been configured.

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings.

Table 18. Variables beginning with `cray_global_net.settings.hosts.data.primary_smw.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.admin_interface</code>	null	required	N/A	N/A
<code>admin_interface.name</code>	<code>eth3</code>	required	N/A	N/A
<code>admin_interface.description</code>	Interface connecting to the admin nodes such as boot and sdb.	basic	N/A	N/A
<code>admin_interface.aliases</code>	[]	basic	N/A	files: /etc/hosts
<code>admin_interface.network</code>	admin	basic	<i>probe (1)</i>	files: /etc/sysconfig/network/ifcfg-eth3 (NETWORK)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
admin_interface.ipv4_address	10.3.1.1	basic	<i>probe (1)</i>	<i>files:</i> /etc/sysconfig/network/ifcfg-eth3
admin_interface.mac	false	advanced	<i>probe (1)</i>	N/A
admin_interface.startmode	auto	advanced	N/A	<i>files:</i> /etc/sysconfig/network/ifcfg-eth3 (STARTMODE)
admin_interface.bootproto	static	basic	N/A	<i>files:</i> /etc/sysconfig/network/ifcfg-eth3 (BOOTPROTO)
admin_interface.mtu			<i>probe (1)</i>	<i>files:</i> /etc/sysconfig/network/ifcfg-eth3 (MTU)
admin_interface.extra_attributes	[]	advanced	N/A	<i>files:</i> /etc/sysconfig/network/ifcfg-eth3
admin_interface.module		advanced	N/A	N/A
admin_interface.params		advanced	N/A	N/A
admin_interface.unmanaged_interface	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) smw# ifconfig eth3				

6.4.1.4 Update cray_ipforward Worksheet in Global Config Set

Prerequisites

This procedure assumes that a work area has been set up for editing global configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/global_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The global Cray IP Forwarding service enables centralized IP forwarding between service nodes and the SMW. This procedure enables the service using the `cray_ipforward` configuration worksheet.

Procedure

1. Edit `cray_ipforward_worksheet.yaml`.

```
smw# vi cray_ipforward_worksheet.yaml
```

2. Uncomment `cray_ipforward.enabled` and ensure that it is set to `true`.

6.4.1.5 Update `cray_liveupdates` Worksheet in Global Config Set

Prerequisites

This procedure assumes that a work area has been set up for editing global configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/global_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The live updates service enables package manager (e.g., zypper, yum) actions (e.g., install, search, upgrade) on CLE nodes using repositories shared from the SMW to those nodes. This procedure enables the `cray_liveupdates` service. There are no other settings that can be changed.

Procedure

1. Edit `cray_liveupdates_worksheet.yaml`.

```
smw# vi cray_liveupdates_worksheet.yaml
```

2. Uncomment `cray_liveupdates.enabled` and ensure that it is set to `true`.

6.4.1.6 Update `cray_logging` Worksheet in Global Config Set

Prerequisites

This procedure assumes that a work area has been set up for editing global configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/global_worksheet_workarea
```

About this task

The global Cray logging service provides centralized logging for the system. This procedure enables the service and configures some settings using the `cray_logging` configuration worksheet. It also provides a translation table for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the global config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_logging_worksheet.yaml`.

```
smw# vi cray_logging_worksheet.yaml
```

2. Uncomment `cray_logging.enabled` and ensure that it is set to `true`.
3. Migrate logging configuration data.

Uncomment these settings and replace the defaults with SMW-specific or site-specific values, as needed. Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings.

- a. Uncomment `cray_logging.settings.global_options.data.raid` and change its value if the boot RAID for this system has a non-standard IP address.
- b. Uncomment `cray_logging.settings.site_loghost.data.name` and change its value if this site has a site log host.
- c. Uncomment `cray_logging.settings.site_loghost.data.syslog_format` and change its value if needed.

If in the CLE 5.2 / SMW 7.2 system, `/etc/init.d/cray-syslog sitecompatmode` is set to "yes," then set this variable to "non_rfc5424." If `sitecompatmode` is unset or set to "no," then it used rfc 5424, so leave this variable set to "rfc5424" (the default value).

This site may uncomment and change other settings as well. The worksheet provides guidance on each setting.

For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Note that for global configuration services, no context is provided in the Files/Installer column because it is assumed to be the SMW.

Table 19. Variables beginning with `cray_logging.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>global_options.data.global_log_level</code>	4	advanced	N/A	N/A
<code>global_options.data.raid</code>	10.1.0.	basic	N/A	<i>files:</i> /etc/init.d/cray-syslog (llm_raid_ip) <i>installer:</i> SMWinstall.conf (llm_raid_ip)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
site_loghost.data.name		basic	N/A	<i>files:</i> /etc/init.d/cray-syslog (siteloghost) <i>installer:</i> SMWinstall.conf (LLM_siteloghost)
site_loghost.data.ip_protocol	tcp	advanced	N/A	N/A
site_loghost.data.ip_port	514	advanced	N/A	N/A
site_loghost.data.syslog_format	rfc5424	advanced	N/A	<i>files:</i> /etc/init.d/cray-syslog (sitecompatmode not yes) <i>installer:</i> SMWinstall.conf (LLM_sitecompatmode)
direct_delivery.data.default	514	advanced	N/A	N/A
direct_delivery.data.console	5150	advanced	N/A	N/A
direct_delivery.data.hss_nlrd	5151	advanced	N/A	N/A
direct_delivery.data.hss_erd	5152	advanced	N/A	N/A
direct_delivery.data.consumer	5153	advanced	N/A	N/A
direct_delivery.data.netwatch	5154	advanced	N/A	N/A
direct_delivery.data.hss_pcimon	5155	advanced	N/A	N/A
direct_delivery.data.dumpd	5156	advanced	N/A	N/A
direct_delivery.data.hss_syslog	5171	advanced	N/A	N/A
direct_delivery.data.hss_bios	5172	advanced	N/A	N/A
direct_delivery.data.hss_coldstart	5172	advanced	N/A	N/A
direct_delivery.data.apollo_syslog	5173	advanced	N/A	N/A
direct_delivery.data.bbs	5174	advanced	N/A	N/A
direct_delivery.data.dws	5174	advanced	N/A	N/A
direct_delivery.data.hss_diagd_history	5175	advanced	N/A	N/A
direct_delivery.data.hss_diagd_ssd	5176	advanced	N/A	N/A
direct_delivery.data.hss_xtremoted	5177	advanced	N/A	N/A
direct_delivery.data.xtremoted_audit	5178	advanced	N/A	N/A
direct_delivery.data.hss_cmdlog	5179	advanced	N/A	N/A
direct_delivery.data.athena_syslog	5180	advanced	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
rsyslog.data.input_modules	(see worksheet)	advanced	N/A	N/A
rsyslog.data.main_msg_queue	(see worksheet)	advanced	N/A	N/A
rsyslog.data.action_queue	(see worksheet)	advanced	N/A	N/A
rsyslog.data.maxopenfiles	8192	advanced	N/A	files: /etc/init.d/cray-syslog (maxfiles)
journald.data.journald_conf	(see worksheet)	advanced	N/A	N/A
systemd.data.systemd_conf	(see worksheet)	advanced	N/A	N/A
systemd.data.systemd_user_conf	(see worksheet)	advanced	N/A	N/A
dirs.data.user	root	advanced	N/A	N/A
dirs.data.group	crayadm	advanced	N/A	N/A
dirs.data.mode	'0775'	advanced	N/A	N/A
rsyslog_conf.data.use	root	advanced	N/A	N/A
rsyslog_conf.data.group	root	advanced	N/A	N/A
rsyslog_conf.data.mode	'0644'	advanced	N/A	N/A
logs.data.rotation	day	advanced	N/A	N/A
logs.data.consolidated	true	advanced	N/A	N/A
logs.data.debugraw	false	advanced	N/A	N/A
logs.data.debugmax	false	advanced	N/A	N/A
logs.data.user	root	advanced	N/A	N/A
logs.data.group	crayadm	advanced	N/A	N/A
logs.data.mode	'0664'	advanced	N/A	N/A

4. Determine whether data for log rotation needs to be migrated.

If this site has not configured log rotation on the CLE 5.2 / SMW 7.2 system in the `/etc/rsyslog.conf` file, skip this step and proceed to step 5 on page 120. If this site may have configured log rotation, use this command to find out.

```
smw# grep file-console /etc/rsyslog.conf
```

If the output looks like the following example, with just "file-console" in the "then" clause, then it is similar to the default CLE 6.0 / SMW 8.0 worksheet setting of `cray_logging.settings.logs.data.rotation`:
day.

```
if $structured-data == '[console@34]' then -?file-console;format-  
message_time_host
```

If the output looks like the following instead, with "file-console-hour" in the "then" clause, then change `cray_logging.settings.logs.data.rotation` to something other than "day." The choices include one log per day; three logs per day; and one log per hour, quarter hour, and minute (see the `cray_logging_worksheet.yaml` worksheet guidance for this setting to see the corresponding values to use).

```
if $structured-data == '[console@34]' then -?file-console-hour;format-  
message_time_host
```

5. Determine whether site-local rsyslog rules exist that should be migrated.

If this site has customized the rsyslog configuration on the CLE 5.2 / SMW 7.2 system with local rules on the SMW in `/var/spool/rsyslog/local-rules`, look for any site-specific content in those locations. If it exists, those file(s) could be brought forward, if desired. Those site-local hooks should still work under CLE 6.0 / SMW 8.0, although Cray recommends getting the system working with the default rsyslog configuration before attempting to make site-specific changes.

6.4.1.7 Update `cray_multipath` Worksheet in Global Config Set

Prerequisites

This procedure assumes that a work area has been set up for editing global configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/global_worksheet_workarea
```

About this task

The Cray multipath service provides a means to support redundant paths to a device for failover or performance reasons. If multipath configuration is desired on the management node (SMW) as well as CLE nodes, Cray recommends enabling this service in the global config set and configuring it for both SMW and CLE nodes. The multipath service in the CLE config sets should inherit the global configuration data.

Multipath does NOT need to be fully cabled to be used. The multipath driver can handle using one path or many.

NOTE: (SMW HA only) Cray recommends configuring multipath before configuring and enabling HA. If HA is configured and enabled first, then additional precautions must be taken when enabling multipath, as documented in *XC™ Series SMW HA Installation Guide*.

This procedure enables or disables the service and configures some settings (if enabled) using the `cray_multipath` configuration worksheet.

Procedure

1. Edit `cray_multipath_worksheet.yaml`.

```
smw# vi cray_multipath_worksheet.yaml
```

2. Enable multipath, as needed.

Choose one of the following options, depending on whether this site intends to use multipath.

Will multipath be used?

- If no, then uncomment `cray_multipath.enabled` and ensure that it is set to `false`. Skip the rest of this procedure.
- If yes, then uncomment `cray_multipath.enabled` and set it to `true`. Continue with the following steps.

3. Update `cray_multipath`.

For a migration, the CLE 5.2 / SMW 7.2 multipath configuration file is located on the SMW at `/etc/opt/cray/share/pN/dist.d/multipath.conf.cray`, where `pN` is the partition. Check this file for any site modifications that would cause it to deviate from the Cray default values, which have not changed for this release. The CLE 6.0 / SMW 8.0 defaults match the CLE 5.2 / SMW 7.2 defaults. A table showing all of the CLE 6.0 / SMW 8.0 multipath variables, their defaults, and many pre-populated device settings is included at the end of this step.

a. Enter the list of multipath nodes.

Uncomment `cray_multipath.settings.multipath.data.node_list`, remove the `[]` (denotes empty list), and add a list of nodes (by cname or host ID) in this system that have multipath devices and need to have multipath configured. For sites with boot node failover and/or SDB node failover, Cray recommends adding both the active and passive (failover) nodes to this list.

This example shows a list of three nodes: an SMW with host ID `1eac4e0c`, a boot node with cname `c0-0c0s4n1`, and an SDB node with cname `c0-0c0s3n1`.

```
cray_multipath.settings.multipath.data.node_list:
- 1eac4e0c
- c0-0c0s4n1
- c0-0c0s3n1
```

b. Configure enabled devices.

Cray has provided a number of enabled devices with pre-populated data under `# ** 'enabled_devices' DATA **`. These storage devices are the devices that will be whitelisted, which means they will be listed as exceptions to the blacklist. The settings for these devices have default values provided by the device vendors and do not need to be changed. If this site intends to configure a multipath device that does not appear in this group of enabled devices, contact a Cray representative for help.

c. (Optional) Configure aliases for the multipath devices.

This is the equivalent of adding aliases to the multipaths section of the `multipath.conf` file.

In the worksheet, copy the two lines below `# ** EXAMPLE 'aliases' VALUE` (with current defaults) `**` and paste them below `# NOTE: Place additional 'aliases' setting entries here, if desired.`

```
# ** EXAMPLE 'aliases' VALUE (with current defaults) **
#   cray_multipath.settings.aliases.data.wwid.sample_key_a: null <-- setting a multival key
#   cray_multipath.settings.aliases.data.sample_key_a.alias: ''
#
```

Uncomment the lines, replace `sample_key_a` with the World Wide Identifier (WWID) of the device to be aliased (60080e50002e203c00002a085551b2c8 in this example) in all lines, and remove the `<--` setting a multival key text at the end of the first line (note that the null value is required; do not remove or change it). Finally, add the alias for this device (`smw_node_pv1` in this example). Repeat this substep for each device, as needed.

```
# NOTE: Place additional 'aliases' setting entries here, if desired.
cray_multipath.settings.aliases.data.wwid.60080e50002e203c00002a085551b2c8: null
cray_multipath.settings.aliases.data.60080e50002e203c00002a085551b2c8.alias: smw_node_pv1
#***** END Service Setting: aliases *****
```

- d. Correct the default values for three pre-populated device settings.

The default values of the following variables are incorrect in `cray_multipath_worksheet.yaml` for this release (they are correct in the table below). In the worksheet, find these variables and change their values as indicated.

```
enabled_devices.data.DDN_SFA12K_20.product: SFA12K-20

enabled_devices.data.DDN_SFA12K_40.product: SFA12K-40|SFA12KX*

enabled_devices.data.DDN_EF3015.path_grouping_policy: group_by_prio
```

This table shows all of the CLE 6.0 / SMW 8.0 multipath variables and defaults. For an explanation of variable names and table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Table 20. Translation Table: Variables beginning with `cray_multipath.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>multipath.data.node_list</code>	<code>[]</code>	basic	NA	NA
<code>defaults.data.polling_interval</code>	10	advanced	NA	NA
<code>defaults.data.path_selector</code>	round-robin 0	advanced	NA	NA
<code>defaults.data.path_grouping_policy</code>	multibus	advanced	NA	NA
<code>defaults.data.uid_attribute</code>	ID_SERIAL	advanced	NA	NA
<code>defaults.data.getuid_callout</code>	<code>/lib/udev/scsi_id -g -u -d /dev/%n</code>	advanced	NA	NA
<code>defaults.data.prio</code>	const	advanced	NA	NA
<code>defaults.data.path_checker</code>	directio	advanced	NA	NA
<code>defaults.data.max_fds</code>	8192	advanced	NA	NA
<code>defaults.data.rr_weight</code>	priorities	advanced	NA	NA
<code>defaults.data.failback</code>	immediate	advanced	NA	NA
<code>defaults.data.no_path_retry</code>	30	advanced	NA	NA
<code>defaults.data.user_friendly_names</code>	yes	advanced	NA	NA
<code>defaults.data.multipath_dir</code>		advanced	NA	NA
<code>defaults.data.verbosity</code>		advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
defaults.data.find_multipaths		advanced	NA	NA
defaults.data.features		advanced	NA	NA
defaults.data.rr_min_io		advanced	NA	NA
defaults.data.rr_min_io_rq		advanced	NA	NA
defaults.data.queue_without_daemon		advanced	NA	NA
defaults.data.flush_on_last_del		advanced	NA	NA
defaults.data.checker_timeout		advanced	NA	NA
defaults.data.fast_io_fail_tmo		advanced	NA	NA
defaults.data.dev_loss_tmo		advanced	NA	NA
defaults.data.log_checker_err		advanced	NA	NA
defaults.data.hwtable_regex_match		advanced	NA	NA
defaults.data.reservation_key		advanced	NA	NA
defaults.data.retain_attached_hw_handler		advanced	NA	NA
defaults.data.detect_prio		advanced	NA	NA
blacklist_devnodes.data.devnodes	^(ram raw loop fd md dm- sr scd st)[0-9]*, ^hd[a-z], ^cciss!c[0-9]d[0-9]*	advanced	NA	NA
blacklist_devices.data.all.vendor	*	advanced	NA	NA
blacklist_devices.data.all.product	*	advanced	NA	NA
blacklist_wwids.data.wwids	[]	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.vendor	LSI	basic	NA	NA
enabled_devices.data.LSI_INF-01-00.product	INF-01-00	basic	NA	NA
enabled_devices.data.LSI_INF-01-00.revision		advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.path_grouping_policy	group_by_prio	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.prio	rdac	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.uid_attribute	ID_SERIAL	advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.LSI_INF-01-00.getuid_callout	/lib/udev/scsi_id -g -u -d /dev/%n	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.path_checker	rdac	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.path_selector	round-robin 0	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.hardware_handler	1 rdac	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.failback	immediate	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.rr_weight	priorities	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.features	2 pg_init_retries 50	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.no_path_retry	30	advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.rr_min_io		advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.fast_io_fail_tmo		advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.flush_on_last_del		advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.LSI_INF-01-00.detect_prio		advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.vendor	NETAPP	basic	NA	NA
enabled_devices.data.NETAPP_INF-01-00.product	INF-01-00	basic	NA	NA
enabled_devices.data.NETAPP_INF-01-00.revision		advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.path_grouping_policy	group_by_prio	advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.NETAPP_INF-01-00.prio	rdac	advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.uid_attribute	ID_SERIAL	advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.getuid_callout	/lib/udev/scsi_id -g -u -d /dev/%n	advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.path_checker	rdac	advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.path_selector	round-robin 0	advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.hardware_handler	1 rdac	advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.failback	immediate	advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.rr_weight	priorities	advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.features	2 pg_init_retries 50	advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.no_path_retry	30	advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.rr_min_io		advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.fast_io_fail_tmo		advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.flush_on_last_del		advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.NETAPP_INF-01-00.detect_prio		advanced	NA	NA
enabled_devices.data.ENGENIO_INF-01-00.vendor	ENGENIO	basic	NA	NA
enabled_devices.data.ENGENIO_INF-01-00.product	INF-01-00	basic	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.ENGENIO_IN F-01-00.revision		advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.path_grouping_policy	group_by_prio	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.prio	rdac	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.uid_attribute	ID_SERIAL	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.getuid_callout	/lib/udev/scsi_id -g -u -d /dev/%n	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.path_checker	rdac	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.path_selector	round-robin 0	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.hardware_handler	1 rdac	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.failback	immediate	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.rr_weight	priorities	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.features	2 pg_init_retries 50	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.no_path_retry	30	advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.rr_min_io		advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.fast_io_fail_tmo		advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.flush_on_last_del		advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.ENGENIO_IN F-01-00.detect_prio		advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.DDN_EF3010.vendor	DDN	basic	NA	NA
enabled_devices.data.DDN_EF3010.product	EF3010	basic	NA	NA
enabled_devices.data.DDN_EF3010.revision		advanced	NA	NA
enabled_devices.data.DDN_EF3010.path_grouping_policy	multibus	advanced	NA	NA
enabled_devices.data.DDN_EF3010.prio	alua	advanced	NA	NA
enabled_devices.data.DDN_EF3010.uid_attribute	ID_SERIAL	advanced	NA	NA
enabled_devices.data.DDN_EF3010.getuid_callout	/lib/udev/scsi_id --whitelisted --device=/dev/%n	advanced	NA	NA
enabled_devices.data.DDN_EF3010.path_checker	tur	advanced	NA	NA
enabled_devices.data.DDN_EF3010.path_selector		advanced	NA	NA
enabled_devices.data.DDN_EF3010.hardware_handler	0	advanced	NA	NA
enabled_devices.data.DDN_EF3010.failback	immediate	advanced	NA	NA
enabled_devices.data.DDN_EF3010.rr_weight	uniform	advanced	NA	NA
enabled_devices.data.DDN_EF3010.features		advanced	NA	NA
enabled_devices.data.DDN_EF3010.no_path_retry	fail	advanced	NA	NA
enabled_devices.data.DDN_EF3010.rr_min_io	100	advanced	NA	NA
enabled_devices.data.DDN_EF3010.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.DDN_EF3010.fast_io_fail_tmo		advanced	NA	NA
enabled_devices.data.DDN_EF3010.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.DDN_EF3010.flush_on_last_del		advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.DDN_EF3010.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.DDN_EF3010.detect_prio		advanced	NA	NA
enabled_devices.data.DDN_EF3015.vendor	DDN	basic	NA	NA
enabled_devices.data.DDN_EF3015.product	EF3015	basic	NA	NA
enabled_devices.data.DDN_EF3015.revision		advanced	NA	NA
enabled_devices.data.DDN_EF3015.path_grouping_policy	group_by_prio	advanced	NA	NA
enabled_devices.data.DDN_EF3015.prio	alua	advanced	NA	NA
enabled_devices.data.DDN_EF3015.uid_attribute	ID_SERIAL	advanced	NA	NA
enabled_devices.data.DDN_EF3015.getuid_callout	/lib/udev/scsi_id --whitelisted --device=/dev/%n	advanced	NA	NA
enabled_devices.data.DDN_EF3015.path_checker	tur	advanced	NA	NA
enabled_devices.data.DDN_EF3015.path_selector	round-robin 0	advanced	NA	NA
enabled_devices.data.DDN_EF3015.hardware_handler	0	advanced	NA	NA
enabled_devices.data.DDN_EF3015.failback	immediate	advanced	NA	NA
enabled_devices.data.DDN_EF3015.rr_weight	uniform	advanced	NA	NA
enabled_devices.data.DDN_EF3015.features		advanced	NA	NA
enabled_devices.data.DDN_EF3015.no_path_retry	18	advanced	NA	NA
enabled_devices.data.DDN_EF3015.rr_min_io	100	advanced	NA	NA
enabled_devices.data.DDN_EF3015.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.DDN_EF3015.fast_io_fail_tmo		advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.DDN_EF3015.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.DDN_EF3015.flush_on_last_del		advanced	NA	NA
enabled_devices.data.DDN_EF3015.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.DDN_EF3015.detect_prio		advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.vendor	DDN	basic	NA	NA
enabled_devices.data.DDN_S2A_6620.product	S2A 6620	basic	NA	NA
enabled_devices.data.DDN_S2A_6620.revision		advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.path_grouping_policy	group_by_prio	advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.prio	alua	advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.uid_attribute	ID_SERIAL	advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.getuid_callout	/lib/udev/scsi_id --page=0x83 --whitelisted --device=/dev/%n	advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.path_checker	tur	advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.path_selector		advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.hardware_handler		advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.failback	immediate	advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.rr_weight		advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.features		advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.no_path_retry	12	advanced	NA	NA
enabled_devices.data.DDN_S2A_6620.rr_min_io		advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.DDN_S2A_66 20.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.DDN_S2A_66 20.fast_io_fail_tmo		advanced	NA	NA
enabled_devices.data.DDN_S2A_66 20.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.DDN_S2A_66 20.flush_on_last_del		advanced	NA	NA
enabled_devices.data.DDN_S2A_66 20.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.DDN_S2A_66 20.detect_prio		advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.vendor	DDN	basic	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.product	S2A 9[579]*	basic	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.revision		advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.path_grouping_policy	group_by_prio	advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.prio	alua	advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.uid_attribute	ID_SERIAL	advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.getuid_callout	/lib/udev/scsi_id --page=0x80 -- whitelisted --device=/dev/%n	advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.path_checker	tur	advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.path_selector		advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.hardware_handler		advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.failback	immediate	advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.rr_weight		advanced	NA	NA
enabled_devices.data.DDN_S2A_95 _97_99.features		advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.DDN_S2A_95_97_99.no_path_retry	fail	advanced	NA	NA
enabled_devices.data.DDN_S2A_95_97_99.rr_min_io		advanced	NA	NA
enabled_devices.data.DDN_S2A_95_97_99.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.DDN_S2A_95_97_99.fast_io_fail_tmo		advanced	NA	NA
enabled_devices.data.DDN_S2A_95_97_99.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.DDN_S2A_95_97_99.flush_on_last_del		advanced	NA	NA
enabled_devices.data.DDN_S2A_95_97_99.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.DDN_S2A_95_97_99.detect_prio		advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.vendor	DDN	basic	NA	NA
enabled_devices.data.DDN_SFA_10000.product	SFA 10000	basic	NA	NA
enabled_devices.data.DDN_SFA_10000.revision		advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.path_grouping_policy	group_by_prio	advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.prio	sfa	advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.uid_attribute	ID_SERIAL	advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.getuid_callout	/lib/udev/scsi_id --page=0x83 --whitelisted --device=/dev/%n	advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.path_checker	tur	advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.path_selector	round-robin 0	advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.hardware_handler		advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.failback	2	advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.DDN_SFA_10000.rr_weight	uniform	advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.features		advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.no_path_retry	12	advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.rr_min_io		advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.fast_io_fail_tmo		advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.flush_on_last_del		advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.DDN_SFA_10000.detect_prio		advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.vendor	DDN	basic	NA	NA
enabled_devices.data.DDN_SFA_12000.product	SFA 12000	basic	NA	NA
enabled_devices.data.DDN_SFA_12000.revision		advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.path_grouping_policy	group_by_prio	advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.prio	sfa	advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.uid_attribute	ID_SERIAL	advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.getuid_callout	/lib/udev/scsi_id --page=0x83 --whitelisted --device=/dev/%n	advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.path_checker	tur	advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.path_selector	round-robin 0	advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.DDN_SFA_12000.hardware_handler		advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.failback	2	advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.rr_weight	uniform	advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.features		advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.no_path_retry	12	advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.rr_min_io		advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.fast_io_fail_tmo		advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.flush_on_last_del		advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.DDN_SFA_12000.detect_prio		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.vendor	DDN	basic	NA	NA
enabled_devices.data.DDN_SFA12K_20.product	SFA12K-20	basic	NA	NA
enabled_devices.data.DDN_SFA12K_20.revision		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.path_grouping_policy	group_by_prio	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.prio	sfa	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.uid_attribute	ID_SERIAL	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.getuid_callout	/lib/udev/scsi_id --page=0x83 --whitelisted --device=/dev/%n	advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.DDN_SFA12K_20.path_checker	tur	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.path_selector	round-robin 0	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.hardware_handler		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.failback	2	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.rr_weight	uniform	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.features		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.no_path_retry	12	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.rr_min_io		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.fast_io_fail_tmo		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.flush_on_last_del		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_20.detect_prio		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.vendor	DDN	basic	NA	NA
enabled_devices.data.DDN_SFA12K_40.product	SFA12K-40 SFA12KX*	basic	NA	NA
enabled_devices.data.DDN_SFA12K_40.revision		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.path_grouping_policy	group_by_prio	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.prio	sfa	advanced	NA	NA

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled_devices.data.DDN_SFA12K_40.uid_attribute	ID_SERIAL	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.getuid_callout	/lib/udev/scsi_id --page=0x83 --whitelisted --device=/dev/%n	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.path_checker	tur	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.path_selector	round-robin 0	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.hardware_handler		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.failback	2	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.rr_weight	uniform	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.features		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.no_path_retry	12	advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.rr_min_io		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.rr_min_io_rq		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.fast_io_fail_tmo		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.dev_loss_tmo		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.flush_on_last_del		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.retain_attached_hw_handler		advanced	NA	NA
enabled_devices.data.DDN_SFA12K_40.detect_prio		advanced	NA	NA

6.4.1.8 Update cray_time Worksheet in Global Config Set

Prerequisites

This procedure assumes that a work area has been set up for editing global configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/global_worksheet_workarea
```

About this task

The Cray Time service configures the time zone and several advanced features, such as the minimum poll interval for NTP messages. This procedure enables the service and configures some settings using the `cray_time` configuration worksheet. It also provides a translation table for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the global config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_time_worksheet.yaml`.

```
smw# vi cray_time_worksheet.yaml
```

2. Uncomment `cray_time.enabled` and ensure that it is set to `true`.
3. Migrate time configuration data.

Uncomment these settings and replace the defaults with SMW-specific or site-specific values. The worksheet provides guidance on each setting.

```
#cray_time.settings.service.data.timezone: US/Central
#cray_time.settings.clock.data.hwclock: --utc
#cray_time.settings.clock.data.systohc: true
#cray_time.settings.clock.data.utc: true
#cray_time.settings.ntp.data.iburst_enable: true
#cray_time.settings.ntp.data.minpoll: 4
#cray_time.settings.ntp.data.monitor_allow: false
```

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98. Note that for global configuration services, no context is provided in the Files/Installer column because it is assumed to be the SMW.

Notes:

- For `cray_time.settings.clock.data.hwclock`, if the CLE 5.2 / SMW 7.2 `/etc/sysconfig/clock` file has "HWCLOCK=-u" or "HWCLOCK=--utc" then set the CLE 6.0 / SMW 8.0 variable to "--utc" to match.
- For `cray_time.settings.clock.data.systohc`, if the CLE 5.2 / SMW 7.2 `/etc/sysconfig/clock` file has "SYSTOHC =yes" then set the CLE 6.0 / SMW 8.0 variable to "true" to match.

Table 21. Variables beginning with `cray_time.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>service.data.timezone</code>	US/Central	basic	N/A	<i>files:</i>

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/sysconfig/clock (DEFAULT_TIMEZONE)
clock.data.hwclock	--utc	advanced	N/A	files: /etc/sysconfig/clock (HWCLOCK)
clock.data.systohc	true	advanced	N/A	files: /etc/sysconfig/clock (SYSTOHC)
clock.data.utc	true	advanced	N/A	files: /etc/sysconfig/clock (UTC (if present))
ntp.data.iburst_enable	true	advanced	N/A	files: /etc/ntp.conf ("iburst" on server line)
ntp.data.minpoll	4	advanced	N/A	files: /etc/ntp.conf ("minpoll 4" on server line)
ntp.data.monitor_allow	false	advanced	N/A	files: /etc/ntp.conf ("disable monitor")

6.4.2 Prepare CLE Worksheets for Migration

Prerequisites

This procedure assumes that SMW 8.0.UP03 and CLE 6.0.UP03 software has been installed. Patches for the base release will be applied later in the migration process.

About this task

The Cray XC system stores configuration information used to boot and customize the CLE system in the p0 config set, or if the system is partitioned, in config set p1 for partition p1 and config set p2 for partition p2, and so forth. This procedure prepares the CLE configuration worksheets, which are then edited to include site-specific configuration data.

When editing configuration worksheets, a general rule is to uncomment all settings that are marked level=basic and modify values as needed. All settings that remain commented are considered unconfigured. Some settings are already uncommented in the original worksheet; Cray recommends not modifying those preconfigured settings because they are needed for proper configuration of the system. For more information about editing configuration worksheets and updating config sets, see *XC™ Series Configurator User Guide (S-2560)*.

NOTE: (SMW HA only) For SMW HA systems, config set operations need to be performed on only one SMW because the config sets are shared between both SMWs in the SMW HA pair.

Procedure

1. Generate configuration worksheets for a CLE config set using prepare mode and the no-scripts option.

In this example, the config set is named `p0_example`.

```
smw# cfgset create -m prepare -t cle --no-scripts p0_example
```

2. Save a copy of original worksheets.

Make a copy of the original CLE configuration worksheets directory to preserve the worksheets in case they are needed for comparison later.

```
smw# ls -l /var/opt/cray/imps/config/sets/p0_example/worksheets
```

```
smw# cp -a /var/opt/cray/imps/config/sets/p0_example/worksheets \
/var/opt/cray/imps/config/sets/p0_example/worksheets.orig
```

3. Copy the CLE worksheets to a work area.

Make a copy of the CLE configuration worksheets directory outside the config set to be used as a work area for editing. The worksheets should not be edited in their original location for two reasons: (1) the configurator will not permit updating a config set from worksheets within that config set, and (2) edits would be overwritten when the config set is updated.

REMEMBER: For partitioned systems, each partition generally has its own config set and associated configuration worksheets. Copy the CLE configuration worksheets to a separate work area for each partition.

```
smw# cp -a /var/opt/cray/imps/config/sets/p0_example/worksheets \
/var/adm/cray/release/p0_worksheet_workarea
```

4. Change to the work area directory to simplify the editing commands in the following procedures.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

5. Edit and update the CLE configuration worksheets using the procedures that follow.

Many procedures provide translation tables, which list the variables that can be set and how to find the information needed for that variable on the currently running CLE 5.2 / SMW 7.2 system. Some procedures do not have translation tables because the variables in those configuration services have no analog in the CLE 5.2 / SMW 7.2 software. All of the worksheets in the CLE config set must be edited and updated for a successful migration to the new system software.

TIP: Update the node groups worksheet (`cray_node_groups_worksheet.yaml`) first. Many configuration worksheets use node groups, and it will be much easier to update those worksheets if the necessary node groups are already defined.

Use [Migration Checklist 2.3: Transfer CLE Configuration Data](#) on page 436 to track progress updating the worksheets.

6.4.2.1 Update `cray_alps` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

Cray ALPS (Application Level Placement Scheduler) is the Cray-supported mechanism for placing and launching applications on Cray system compute nodes. ALPS provides application placement, launch, and management functions and cooperates closely with third-party workload managers (WLM) for application scheduling across Cray systems. The third-party WLMs make policy and scheduling decisions, whereas ALPS provides a mechanism to place and launch the applications contained within batch jobs. ALPS also supports interactive application placement and launch.

This procedure enables the `cray_alps` service and configures some settings in the `cray_alps` configuration worksheet to add site-specific data. The MIGRATE CONFIGURATION DATA section of this procedure provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_alps_worksheet.yaml`.

```
smw# vi cray_alps_worksheet.yaml
```

2. Uncomment `cray_alps.enabled` and ensure that it is set to `true`.
3. Uncomment `cray_alps.settings.common.data.xthostname` and set it to the name of this Cray system.
4. Configure ALPS node groups.

If there are service nodes other than login nodes and the ALPS master node (the SDB node) that need to run ALPS commands, add them to a node group by editing `cray_node_groups_worksheet.yaml`. That node group should include the workload manager (WLM) server and MOM (machine-oriented miniserver) nodes.

Uncomment `cray_alps.settings.common.data.alps_node_groups`, remove the empty list (`[]`), and add that node group (and any other node groups, as needed) on a separate line prefixed by a hyphen and space (`-`).

```
cray_alps.settings.common.data.alps_node_groups:  
- NODE_GROUP_1  
- NODE_GROUP_2
```

5. (Optional) If DRC (dynamic RDMA credentials) will be used in a large system, uncomment `cray_alps.settings.apshed.data.pDomainMax` and set it to 256.

If the maximum number of user protection domains is not increased from its default value of 10 to something like 256, DRC might exhaust all of the domains, which could cause problems for sites with larger, more complex systems.

6. Uncomment `cray_alps.settings.apsys.data.prologPath` and `cray_alps.settings.apsys.data.epilogPath`, even if they are assigned a null value.
7. (Optional) If RUR (resource utilization reporting) will be used at this site, set the `prologPath` and `epilogPath` settings (from the previous step) to these paths.

```
cray_alps.settings.apsys.data.prologPath: /opt/cray/rur/default/bin/rur_prologue.py
cray_alps.settings.apsys.data.epilogPath: /opt/cray/rur/default/bin/rur_epilogue.py
```

Also, ensure that the `cray_rur` service is enabled. See [Update cray_rur Worksheet](#) on page 273.

MIGRATE CONFIGURATION DATA

Use the translation tables below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

8. Migrate common configuration settings, as needed.

Table 22. Variables beginning with `cray_alps.settings.common.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
xhostname	cray	basic	N/A	files: default sharedroot /etc/xhostname installer: CLEinstall.conf (xhostname)
master_node	sdb	advanced	N/A	files: default sharedroot /etc/sysconfig/alps (ALPS_MASTER_NODE) installer: CLEinstall.conf (alps_master_node)
bridge_node	boot	advanced	N/A	files: default sharedroot /etc/sysconfig/alps (ALPS_BRIDGE_NODE) installer: CLEinstall.conf (alps_bridge_node)
alps_node_groups	[]	basic	N/A	N/A
nidorder	-On	advanced	N/A	files: default sharedroot /etc/sysconfig/alps (ALPS_NIDORDER) installer: CLEinstall.conf (alps_nidorder)
apwatch_erd	smw	advanced	N/A	files: default sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/sysconfig/alps (APWATCH_ERD) installer: CLEinstall.conf (alps_apwatch_erd)
apevent_sync_secs	300	advanced	N/A	files: default sharedroot /etc/sysconfig/alps (APEVENT_SYNC_SECS)
sharedDir	/alps_shared	advanced	N/A	N/A

9. Migrate logging configuration settings.

Note that the wildcardBind variable, which was set to 1 or 0 in CLE 5.2 / SMW 7.2, is set to true or false in CLE 6.0 / SMW 8.0.

Table 23. Variables beginning with `cray_alps.settings.logging.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
logMethod	3	advanced	N/A	files: default sharedroot /etc/opt/cray/alps/alps.conf ([logging] logMethod)
timeFormat	'%F-%T'	advanced	N/A	N/A
wildcardBind	true	advanced	N/A	files: default sharedroot /etc/opt/cray/alps/alps.conf ([logging] wildcardBind)

10. Migrate apsched configuration settings.

The Files/Installer column indicates where to look for the CLE 5.2 / SMW 7.2 values to migrate. Here is the portion of the `default sharedroot /etc/opt/cray/alps/alps.conf` file that contains the `apsched` stanza. Note the indented attributes within the stanza.

```
apsched
  fanout 32
  debug 0
/apsched
```

The `fanout` and `debug` variables in the CLE 6.0 / SMW 8.0 system (`cray_alps.settings.apsched.data.fanout` and `cray_alps.settings.apsched.data.debug`) are the first two rows of the following translation table.

Note that the following variables, which were set to 1 or 0 in CLE 5.2 / SMW 7.2, are set to true or false in CLE 6.0 / SMW 8.0.

```
lustreFlush
memoryCompact
resFullNode
```

suspendResume
noNetwork

Table 24. Variables beginning with `cray_alps.settings.apsched.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
fanout	32	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] fanout)
debug	'0x0000'	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] debug)
cpuAffinity	cpu	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] cpuAffinity)
lustreFlush	true	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] lustreFlush)
memoryCompact	true	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] memoryCompact)
maxResv	4096	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] maxResv)
batchCPCU	1	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] batchCPCU)
interactiveCPCU	1	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] interactiveCPCU)
resFullNode	true	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] resFullNode)
claimsPerRes	1000	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] claimsPerRes)
suspendResume	false	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] suspendResume)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
loadLimit	2	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] loadLimit)
srRetry	5	advanced	N/A	N/A
srRetryDelay	5	advanced	N/A	N/A
srLogDelay	30	advanced	N/A	N/A
fakeNumaNodes		advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] fakeNumaNodes)
noNetwork	false	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] noNetwork)
noNetworkAppLimit	1	advanced	N/A	N/A
pKeyFreeDelay	30	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] .pKeyFreeDelay)
pDomainMax	10	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsched] pDomainMax)
cleanSharedEnable	true	advanced	N/A	N/A
cleanSharedFrequency	hourly	advanced	N/A	N/A
cleanSharedAge	1800	advanced	N/A	N/A
timeFormat		advanced	N/A	N/A

11. Migrate apsys configuration settings.

Table 25. Variables beginning with `cray_alps.settings.apsys.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
debug	'0x0001'	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsys] debug)
prologPath		basic	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsys] prologPath)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
epilogPath		basic	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsys] epilogPath)
prologTimeout	300	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsys] prologTimeout)
epilogTimeout	300	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsys] epilogTimeout)
prologPathCCM	/opt/cray/ccm/default/etc/ ccm-prologue	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsys] prologPathCCM)
epilogPathCCM	/opt/cray/ccm/default/etc/ ccm-epilogue	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsys] epilogPathCCM)
prologTimeoutCCM	120	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsys] prologTimeoutCCM)
epilogTimeoutCCM	120	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([apsys] epilogTimeoutCCM)
timeFormat		advanced	N/A	N/A

12. Migrate aprun configuration settings.

Note that the last five variables in the aprun translation table below are used by Slurm. Different workload managers may use different configuration variables. The Files/Installer column for those variables indicates where to look for the CLE 5.2 / SMW 7.2 values to migrate. Here is the portion of the default sharedroot /etc/opt/cray/alps/alps.conf file that contains four of those variables, showing the Slurm values assigned:

```
# defineNid          SLURM_NODEID
# defineEachID      SLURM_PROCID
# defineNPPN        SLURM_TASKS_PER_NODE
# defineLocalEnt    SLURM_LOCALID,PBS_TASKNUM
```

Table 26. Variables beginning with *cray_alps.settings.aprun.data*.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
defineNodeCount		advanced	N/A	N/A
defineWorldSize		advanced	N/A	N/A
defineNid		advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/alps/alps.conf ([aprun] defineNid)
defineEachID		advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/alps/alps.conf ([aprun] defineEachID)
defineNodeList		advanced	N/A	N/A
defineNPPN		advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/alps/alps.conf ([aprun] defineNPPN)
defineLocalEnt		advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/alps/alps.conf ([aprun] defineLocalEnt)

13. Migrate apstat configuration settings.

Table 27. Variables beginning with *cray_alps.settings.apstat.data*.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
nodeTable		advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/alps/alps.conf ([apstat] nodeTable)
appsTable		advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/alps/alps.conf ([apstat] appsTable)
resvTable		advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/alps/alps.conf ([apstat] resvTable)
pendingAppsTable		advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/alps/alps.conf ([apstat] pendingAppsTable)
gpuTable		advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/alps/alps.conf ([apstat] gpuTable)
pDomainTable		advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/alps/alps.conf ([apstat] pDomainTable)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
memoryTable		advanced	N/A	N/A
memPrefix		advanced	N/A	N/A

14. Migrate application cleanup configuration settings.

Note that the variables 'configured' and 'reports,' which were set to on or off in CLE 5.2 / SMW 7.2, are set to true or false in CLE 6.0 / SMW 8.0.

Table 28. Variables beginning with `cray_alps.settings.application_cleanup.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
configured	true	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup application] configured)
reports	true	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup application] reports)
reportWait	2000	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup application] reportWait)
iterationSleep	1000	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup application] iterationSleep)
iterationMax	10	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup application] iterationMax)
iterationNHC	3	advanced	N/A	N/A
connectTimeout	1000	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup application] connectTimeout)
connectAttempts	5	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup application] connectAttempts)
waitMin	5000	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup application] waitMin)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
pokeMin	5	advanced	N/A	N/A
pokeMax	50	advanced	N/A	N/A
pokeImpatient	3	advanced	N/A	N/A
pokeStartDelay	1000	advanced	N/A	N/A
pokeWaitMax	1000	advanced	N/A	N/A
queryWait	3000	advanced	N/A	N/A

15. Migrate reservation cleanup configuration settings.

Note that the variables 'configured' and 'reports,' which were set to on or off in CLE 5.2 / SMW 7.2, are set to true or false in CLE 6.0 / SMW 8.0.

Table 29. Variables beginning with `cray_alps.settings.reservation_cleanup.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
configured	true	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup reservation] configured)
reports	true	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup reservation] reports)
reportWait	2000	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup reservation] reportWait)
iterationSleep	1000	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup reservation] iterationSleep)
iterationMax	10	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup reservation] iterationMax)
iterationNHC	3	advanced	N/A	N/A
connectTimeout	1000	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup reservation] connectTimeout)
connectAttempts	5	advanced	N/A	<i>files:</i> default sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/opt/cray/alps/alps.conf ([cleanup reservation] connectAttempts)
waitMin	5000	advanced	N/A	files: default sharedroot /etc/opt/cray/alps/alps.conf ([cleanup reservation] waitMin)
pokeMin	5	advanced	N/A	N/A
pokeMax	50	advanced	N/A	N/A
pokeImpatient	3	advanced	N/A	N/A
pokeStartDelay	1000	advanced	N/A	N/A
pokeWaitMax	1000	advanced	N/A	N/A
queryWait	3000	advanced	N/A	N/A

16. Migrate log rotation configuration settings.

Table 30. Variables beginning with `cray_alps.settings.logrotate.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
compress	true	advanced	N/A	N/A
frequency	daily	advanced	N/A	N/A
rotations	365	advanced	N/A	N/A

6.4.2.2 Update `cray_auth` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray Authentication configuration service provides a way to list the authentication domains that should govern how users of the system are identified and authenticated. Authentication domains include LDAP, NIS, and Active Directory.

This procedure configures some settings in the `cray_auth` configuration worksheet to add site-specific data. For examples of modifying a config set for use with an authentication method other than the default LDAP setup, see

[Modify a Config Set for use with Advanced Authentication Configurations](#). The last step in this procedure provides a translation table for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_auth_worksheet.yaml`.

```
smw# vi cray_auth_worksheet.yaml
```

2. Uncomment `cray_auth.enabled` and set it to `true`.
3. Review the `nsswitch_sources` service setting of the worksheet.

```
#***** START Service Setting: nsswitch_sources *****
```

This service setting controls the settings in the `nsswitch.conf` file. Add, delete, or change these settings to modify the `nsswitch.conf` file, as needed.

4. Review the `common_ldap_options` service setting of the worksheet, especially if LDAP will NOT be used at this site.

```
#***** START Service Setting: common_ldap_options *****
```

This is an advanced level setting that has several pre-populated values for common LDAP configuration options.

- Sites NOT using LDAP for part or all of the authentication must change some settings in this section (for example, to use Kerberos or Active Directory).
- Sites using LDAP may need to add, change, or delete options in this section.

To add a `common_ldap_options` stanza to the worksheet, copy the two lines below `# ** EXAMPLE` `'common_ldap_options'` VALUE (with current defaults) `**` and paste them below `# NOTE:` Place additional `'common_ldap_options'` setting entries here, if desired.

```
# ** EXAMPLE 'common_ldap_options' VALUE (with current defaults) **
# cray_auth.settings.common_ldap_options.data.option.sample_key_a: null <-- setting a multival key
# cray_auth.settings.common_ldap_options.data.sample_key_a.value: ''
# ** 'common_ldap_options' FIELD SPECIFICATION -- MULTIVAL KEY FIELD **
```

Uncomment the lines, replace `sample_key_a` in all lines with the LDAP option to be specified, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, add the value of the option in the second line. Repeat this step for each option/value pair to be specified.

```
# NOTE: Place additional 'common_ldap_options' setting entries here, if desired.
cray_auth.settings.common_ldap_options.data.option.sample_key_a: null
cray_auth.settings.common_ldap_options.data.sample_key_a.value: ''
#***** END Service Setting: common_ldap_options *****
```

5. (If using NIS) Review the `common_nis_options` service setting of the worksheet and configure these settings if this site wishes to use NIS.

```
#***** START Service Setting: common_nis_options *****
```

This is an advanced level setting that has several pre-populated values for common NIS configuration options. Add, change, or delete options if this site has special authentication needs, such as when using Kerberos (not common).

In the worksheet, copy the two lines below # ** EXAMPLE 'common_nis_options' VALUE (with current defaults) ** and paste them below # NOTE: Place additional 'common_nis_options' setting entries here, if desired.

```
# ** EXAMPLE 'common_nis_options' VALUE (with current defaults) **
# cray_auth.settings.common_nis_options.data.option.sample_key_a: null <-- setting a multival key
# cray_auth.settings.common_nis_options.data.sample_key_a.value: ''
```

Uncomment the lines, replace `sample_key_a` in all lines with the NIS option to be specified, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, add the value of the option in the second line. Repeat this step for each option/value pair to be specified.

```
# NOTE: Place additional 'common_nis_options' setting entries here, if desired.
cray_auth.settings.common_nis_options.data.option.sample_key_a: null
cray_auth.settings.common_nis_options.data.sample_key_a.value: ''

***** END Service Setting: common_nis_options *****
```

6. Review the domain service setting of the worksheet and configure settings, as needed.

```
***** START Service Setting: domain *****
```

a. (If using LDAP) Configure LDAP domains to connect to LDAP servers

In the worksheet, copy the four lines below # ** EXAMPLE 'domain' VALUE (with current defaults) ** and paste them below # NOTE: Place additional 'domain' setting entries here, if desired.

```
# ** EXAMPLE 'domain' VALUE (with current defaults) **
# cray_auth.settings.domain.data.reference.sample_key_a: null <-- setting a multival key
# cray_auth.settings.domain.data.sample_key_a.servers: []
# cray_auth.settings.domain.data.sample_key_a.schema: rfc2307
# cray_auth.settings.domain.data.sample_key_a.aux_settings: []
```

Uncomment the lines, replace `sample_key_a` in all lines with some unique authentication domain identifier, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, add values in accordance with site requirements. For settings that are lists, remove the empty brackets and add each list element on a separate line prefixed by a hyphen and space (-).

```
# NOTE: Place additional 'domain' setting entries here, if desired.
cray_auth.settings.domain.data.reference.<ldap_domain_name>: null
cray_auth.settings.domain.data.<ldap_domain_name>.servers: []
cray_auth.settings.domain.data.<ldap_domain_name>.schema: rfc2307
cray_auth.settings.domain.data.<ldap_domain_name>.aux_settings: []
***** END Service Setting: domain *****
```

b. Configure non-LDAP domains, as needed.

As in the previous substep, copy and paste the four-line stanza for a domain setting, but instead of uncommenting all four lines, leave commented the `servers` variable and the `schema` variable, which are specific to LDAP domains.

```
# NOTE: Place additional 'domain' setting entries here, if desired.
cray_auth.settings.domain.data.reference.<domain_name>: null
#cray_auth.settings.domain.data.<domain_name>.servers: []
#cray_auth.settings.domain.data.<domain_name>.schema: rfc2307
cray_auth.settings.domain.data.<domain_name>.aux_settings: []
***** END Service Setting: domain *****
```

7. (If using NIS) Review the `nis` service setting of the worksheet and configure these settings if this site wishes to use NIS.

```
#***** START Service Setting: nis *****
```

- a. Enable NIS (the `nis.data.enabled` setting).

Uncomment `cray_auth.settings.nis.data.enabled` and set it to `true`.

- b. Configure the domain name (the `nis.data.domainname` setting).

Uncomment `cray_auth.settings.nis.data.domainname` and set it to the domain name that was configured on the NIS server (must match). Check for the CLE 5.2 / SMW 7.2 value of this variable in the translation tables in step 10 on page 153.

- c. Configure the servers (the `nis.data.servers` setting).

Uncomment `cray_auth.settings.nis.data.servers: []`, remove the empty brackets, and add a list of NIS server host names or IP addresses. Check for the CLE 5.2 / SMW 7.2 value of this variable in the translation tables in step 10 on page 153.

```
cray_auth.settings.nis.data.servers:
- 172.32.3.4
- 172.32.4.55
```

8. Review the `access` service setting of the worksheet and configure these settings, as needed.

```
#***** START Service Setting: access *****
```

- a. Set the access policy (the `access.data.policy` setting).

Whether using NIS or LDAP, ensure that `cray_auth.settings.access.data.policy` is uncommented and set it to the list shown here. At a minimum, these values are recommended to ensure that root and crayadm are using the local passwd entries and not ones from the authentication service. Check for the CLE 5.2 / SMW 7.2 value of this variable in the translation tables in step 10 on page 153.

NOTICE: The initial `-` (hyphen and space) at the beginning of each list element is part of the YAML syntax. The access policy data, which begins with either a `-` or `+`, starts after that.

```
cray_auth.settings.access.data.policy:
- +:root:LOCAL
- +:crayadm:LOCAL
```

- b. Configure access to compute nodes (the `access.data.config_computes` setting).

Uncomment `cray_auth.settings.access.data.config_computes` and set in accordance with site requirements. For most systems, set this variable to `false`.

Set this variable to `true` for any of these conditions:

- This site wants to allow compute nodes to use network lookup services to identify users (setting `config_computes` to `true` does not mean that users will be allowed to log into compute nodes directly).
- This site is using Slurm and network authentication.
- This site is using cluster compatibility mode (CCM) with LDAP accounts (`ccmrun` will work regardless, but `ccmlogin` will work only if `config_computes` is set to `true`).

IMPORTANT: If `cray_auth.settings.access.data.config_computes` is set to `true`, ensure that:

- RSIP is configured to enable the compute nodes to contact the LDAP server.

- Network user lookup servers are equipped to handle the volume of requests made by the compute nodes.
- c. Configure node groups to recognize user IDs provided by off-node identification services, if needed (the `access.data.config_id_service_groups` setting).

If there are any non-login nodes that may need to identify users without allowing user access, such as DAL (direct-attached Lustre) MDS nodes or MOM nodes, add their crames to a node group by editing `cray_node_groups_worksheet.yaml`, and then add that node group to the list of `config_id_service_groups`. Nodes within these groups should be provided with a network path to the relevant servers.

Uncomment `cray_auth.settings.access.data.config_id_service_groups`, remove the empty list (`[]`), and add that node group (and any other node groups, as needed) on a separate line prefixed by a hyphen and space (`-`).

```
cray_auth.settings.access.data.config_id_service_groups:
- NODE_GROUP_1
- NODE_GROUP_2
```

9. Review the `section` service setting and the `options` embedded service setting (they go together) of the worksheet and configure these settings, as needed.

The "section" service setting refers to the section of the `sssd.conf` file, and "options" are the multiple entries within a "section" that an administrator can specify. This enables the administrator to override settings in any section of the `sssd.conf` file.

In the worksheet, copy the three lines below `# ** EXAMPLE 'section' VALUE` (with current defaults) `**` and paste them below `# NOTE: Place additional 'section' setting entries here, if desired`.

```
# ** EXAMPLE 'section' VALUE (with current defaults) **
# cray_auth.settings.section.data.section_name.sample_key_a: null <-- setting a multival key
# cray_auth.settings.section.data.sample_key_a.options.option_name.sample_key_b: null <-- setting a multival key
# cray_auth.settings.section.data.sample_key_a.options.sample_key_b.value: ''
```

Uncomment the lines, replace `sample_key_a` in all lines with a unique section identifier and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Replace `sample_key_b` in the second and third lines with a unique option identifier. Finally, add values in accordance with site requirements.

Here is an example that adds a `"debug_level = 7"` line to the otherwise unnamed/unused "[pam]" section in the `sssd.conf` file.

```
# NOTE: Place additional 'section' setting entries here, if desired.
cray_auth.settings.section.data.section_name.pam: null
cray_auth.settings.section.data.pam.options.option_name.debug_level: null
cray_auth.settings.section.data.pam.options.debug_level.value: 7
***** END Service Setting: section *****
```

Here is the resulting section of the `sssd.conf` file.

```
[nss]
filter_users = root, crayadm

[pam]
debug_level = 7 <-----

[domain/crayit]
```

10. Migrate other configuration settings, as needed.

Use the translation tables below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

This table contains three sets of variables for common_ldap_options (id_provider, auth_provider, ldap_tls_reqcert) and four sets of variables for common_nis_options (id_provider, auth_provider, proxy_lib_name, and proxy_pam_target).

Table 31. Variables beginning with `cray_auth.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
nsswitch_sources.data.database.passwd	null	advanced	N/A	N/A
nsswitch_sources.data.passwd.sources	compat sss	advanced	N/A	login class sharedroot /etc/nsswitch.conf (passwd)
nsswitch_sources.data.database.group	null	advanced	N/A	N/A
nsswitch_sources.data.group.sources	compat sss	advanced	N/A	login class sharedroot /etc/nsswitch.conf (group)
nsswitch_sources.data.database.netgroup	null	advanced	N/A	N/A
nsswitch_sources.data.netgroup.sources	files sss	advanced	N/A	login class sharedroot /etc/nsswitch.conf (netgroup)
nsswitch_sources.data.database.passwd_compat	null	advanced	N/A	N/A
nsswitch_sources.data.passwd_compat.sources	nis	advanced	N/A	login class

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				sharedroot /etc/nsswitch.conf (passwd_compat)
nsswitch_sources.data.database.group_compat	null	advanced	N/A	N/A
nsswitch_sources.data.group_compat.sources	nis	advanced	N/A	login class sharedroot /etc/nsswitch.conf (group_compat)
common_ldap_options.data.option.id_provider	null	advanced	N/A	N/A
common_ldap_options.data.id_provider.value	ldap	advanced	N/A	N/A
common_ldap_options.data.option.auth_provider	null	advanced	N/A	N/A
common_ldap_options.data.auth_provider.value	ldap	advanced	N/A	N/A
common_ldap_options.data.option.ldap_tls_reqcert	null	advanced	N/A	N/A
common_ldap_options.data.ldap_tls_reqcert.value	allow	advanced	N/A	N/A
common_nis_options.data.option.id_provider	null	advanced	N/A	N/A
common_nis_options.data.id_provider.value	proxy	advanced	N/A	N/A
common_nis_options.data.option.auth_provider	null	advanced	N/A	N/A
common_nis_options.data.auth_provider.value	proxy	advanced	N/A	N/A
common_nis_options.data.option.proxy_lib_name	null	advanced	N/A	N/A
common_nis_options.data.proxy_lib_name.value	nis	advanced	N/A	N/A
common_nis_options.data.option.proxy_pam_target	null	advanced	N/A	N/A
common_nis_options.data.proxy_pam_target.value	sssdnisproxy	advanced	N/A	N/A

This table shows variables that configure NIS, which is disabled by default. Look in the suggested CLE 5.2 / SMW 7.2 files for information about whether/how NIS was configured at this site.

Table 32. Variables beginning with `cray_auth.settings.nis.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
enabled	false	advanced	N/A	<code>files: login class sharedroot</code> <code>/etc/yp.conf</code>
domainname		advanced	<i>probe (1)</i>	<code>files: login class sharedroot</code> <code>/etc/yp.conf</code> <code>/etc/sysconfig/network/config</code> (NETCONFIG_NIS_STATIC_DOMAIN)
servers	[]	advanced	N/A	<code>files: login class sharedroot</code> <code>/etc/yp.conf</code> <code>/etc/sysconfig/network/config</code> (NETCONFIG_NIS_STATIC_SERVERS)
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) <code>login# domainname</code>				

This table show variables that configure the access policy. Look in the suggested CLE 5.2 / SMW 7.2 files for information about how access policy was configured at this site.

Table 33. Variables beginning with `cray_auth.settings.access.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
policy	- +:root:LOCAL - +:crayadm:LOCAL	basic	N/A	<code>files: login class sharedroot</code> <code>/etc/security/access.conf</code>
restrictive	true	advanced	N/A	<code>files: login class sharedroot</code> <code>/etc/security/access.conf</code> (:ALL:ALL)
config_computes	false	basic	N/A	N/A
config_id_service_groups	[]	basic	N/A	N/A

NOTE: The `cray_auth.settings.access.data.config_id_service_groups` variable refers to a list of node groups containing non-login service nodes that may need to identify users without allowing user access, for example, DAL MDS nodes, or MOM nodes. Nodes within these groups should be provided with a network path to the relevant servers.

This table shows an example of a domain controller, which is a multival field that is needed when using LDAP but is not needed for NIS. An example domain controller name (key) of "site_DC" is used. If using LDAP, ensure that `cray_auth.settings.nis.data.enabled` is set to false.

Table 34. Example Domain (variables beginning with `cray_auth.settings.domain.data`.)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files
reference.site_DC	null	basic	N/A	N/A
site_DC.servers	- 172.32.3.4 - 172.32.4.55	basic	N/A	files: login node sharedroot /etc/ldap.conf (host)
site_DC.schema	rfc2307	advanced	N/A	files: login node sharedroot /etc/ldap.conf (nss_schema)
site_DC.aux_settings	- ldap_search_base=dc=datacenter,dc=site,dc=com - ldap_netgroup_search_base=dc=datacenter,dc=site,dc=com	advanced	N/A	files: login node sharedroot /etc/ldap.conf (base) /etc/ldap.conf (nss_base_netgroup)

6.4.2.3 Update `cray_batchlimit` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray `batchlimitd` daemon is used to limit the number of processes that a batch job can create, thereby avoiding a potentially harmful proliferation of processes. Linux limits the total number of processes on a per UID basis, but `batchlimitd` introduces process and thread creation limits on a per-cpuset basis. In CLE 5.2 / SMW 7.2, the `batchlimitd` daemon was supported only with workload managers (WLM) that were compiled with cpuset support, such as Moab/TORQUE 4.x or later.

This procedure enables or disables the `cray_batchlimit` configuration service and, if enabled, configures some settings using the `cray_batchlimit` configuration worksheet. The last step in this procedure provides a translation table for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_batchlimit_worksheet.yaml`.

```
smw# vi cray_batchlimit_worksheet.yaml
```

2. Enable `cray_batchlimit`, as needed.

Uncomment `cray_batchlimit.enabled` and do one of the following:

- If `batchlimit` NOT used at this site, set it to `false` to disable the service. No other settings are needed for a fresh install.
- If `batchlimit` is or will be used at this site, set it to `true` and configure the advanced settings to values appropriate for this site.

3. Configure advanced settings, as needed.

Check the files in the translation table below to determine whether this service is in use at this site. If it is, use this table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Table 35. Variables beginning with `cray_batchlimit.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>base.data.LLM</code>	'on'	advanced	N/A	<i>files:</i> MOM node sharedroot <code>/etc/opt/cray/batchlimit/batchlimit.conf (LLM)</code>
<code>base.data.LLM_debug</code>	'off'	advanced	N/A	<i>files:</i> MOM node sharedroot <code>/etc/opt/cray/batchlimit/batchlimit.conf (LLM_debug)</code>
<code>base.data.log_file</code>	<code>/var/log/batchlimitd.log</code>	advanced	N/A	<i>files:</i> MOM node sharedroot <code>/etc/opt/cray/batchlimit/batchlimit.conf (log_file)</code>
<code>base.data.self_nice</code>	'-1'	advanced	N/A	<i>files:</i> MOM node sharedroot <code>/etc/opt/cray/batchlimit/batchlimit.conf (self_nice)</code>
<code>base.data.self_oom</code>	'-1000'	advanced	N/A	<i>files:</i> MOM node sharedroot <code>/etc/opt/cray/batchlimit/batchlimit.conf (self_oom)</code>
<code>base.data.cpuset_dir</code>	<code>/dev/cpuset/torque</code>	advanced	N/A	<i>files:</i> MOM node sharedroot <code>/etc/opt/cray/batchlimit/batchlimit.conf (cpuset_dir)</code>
<code>base.data.set_oom_appkill</code>	'on'	advanced	N/A	<i>files:</i> MOM node sharedroot <code>/etc/opt/cray/batchlimit/batchlimit.conf (set_oom_appkill)</code>
<code>base.data.enforce_cpu_values</code>	'on'	advanced	N/A	<i>files:</i> MOM node sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/opt/cray/batchlimit/ batchlimit.conf (enforce_cpu_values)
base.data.max_processes	'32'	advanced	N/A	files: MOM node sharedroot /etc/opt/cray/batchlimit/ batchlimit.conf (max_processes)
base.data.address_space_size	no_change	advanced	N/A	files: MOM node sharedroot /etc/opt/cray/batchlimit/ batchlimit.conf (address_space_size)
base.data.limits_string	Max file locks=unlimited: unlimited	advanced	N/A	files: MOM node sharedroot /etc/opt/cray/batchlimit/ batchlimit.conf (limits_string)
base.data.oom_score	500	advanced	N/A	files: MOM node sharedroot /etc/opt/cray/batchlimit/ batchlimit.conf (oom_score)

6.4.2.4 Update cray_boot Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Cray boot configuration service provides a way to specify which nodes will act as boot nodes on the high-speed network (HSN). This service must be enabled for the system to function properly. This procedure configures some basic settings in the cray_boot configuration worksheet.

Procedure

1. Edit `cray_boot_worksheet.yaml`.

```
smw# vi cray_boot_worksheet.yaml
```

2. Uncomment `cray_boot.enabled` and ensure that it is set to `true`.
3. Configure the boot groups setting.

This setting specifies a list of node groups whose members will act as boot nodes.

Uncomment `cray_boot.settings.node_groups.data.boot_groups` and the line immediately following it. By default, the `boot_nodes` node group is the first node group in the list of boot groups. To use other nodes as boot nodes on the HSN, add one or more node groups to this list.

IMPORTANT: Any node group added to `boot_groups` must first be defined in `cray_node_groups_worksheet.yaml`.

Because this is a list setting, each node group must be on a separate line prefixed by a hyphen and space (-).

```
cray_boot.settings.node_groups.data.boot_groups:
- boot_nodes
```

6.4.2.5 Update `cray_ccm` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

Cray cluster compatibility mode (CCM) enables users to run independent software vendor (ISV) applications without modification. Supported workload managers (WLM) include PBS, Moab/TORQUE, Slurm, and LSF.

This procedure disables the `cray_ccm` service because it should be disabled until a WLM is installed. If this is a migration, and this service is currently in use at this site, some advanced settings in the `cray_ccm` configuration worksheet can be set at this time, even with the service disabled. The last step in this procedure provides a translation table for migrating site-specific CCM configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_ccm_worksheet.yaml`.

```
smw# vi cray_ccm_worksheet.yaml
```

2. Uncomment `cray_ccm.enabled` and set it to `false`.

This disables this service. When a service is disabled, the values of all other settings in the service are ignored when configuration is applied. This worksheet contains additional configuration settings that are level advanced, but this service and its advanced settings are not needed for a fresh install.

For a migration, even if this service is disabled because a WLM has not been installed yet, the advanced settings can still be set to the values used on the CLE 5.2 / SMW 7.2 system.

3. Migrate configuration settings, as needed.

Check the files in the translation table below to determine whether this service is in use at this site. If it is, use this table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

These two variables must be adjusted to match the WLM that will be used. See the guidance in the worksheet.

```
cray_ccm.settings.base.data.ccm_wlm
cray_ccm.settings.base.data.cray_batch_var
```

If using Slurm, then this variable must be set as well (in CLE 5.2 / SMW 7.2, this variable was set for PBS also).

```
cray_ccm.settings.base.data.ccm_queues
```

Table 36. Variables beginning with `cray_ccm.settings.base.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
ccm_debug	'no'	advanced	N/A	files: sharedroot /etc/opt/cray/ccm/ccm.conf (CCM_DEBUG)
ccm_enable_rsh	'yes'	advanced	N/A	files: sharedroot /etc/opt/cray/ccm/ccm.conf (CCM_ENABLELSH)
ccm_enable_nis	'no'	advanced	N/A	files: sharedroot /etc/opt/cray/ccm/ccm.conf (CCM_ENABLENIS)
ccm_wlm	pbs	advanced	N/A	files: sharedroot /etc/opt/cray/ccm/ccm.conf (CCM_WLM)
ccm_queues		advanced	N/A	files: sharedroot /etc/opt/cray/ccm/ccm.conf (CCM_QUEUES)
cray_batch_var	/var/spool/ PBS	advanced	N/A	files: sharedroot /etc/opt/cray/ccm/ccm.conf (CRAY_BATCH_VAR)
ccm_ssh_max_connection_timeout	20	advanced	N/A	files: sharedroot /etc/opt/cray/ccm/ccm.conf (SSH_MAX_CONNECTION_TIMEOUT)
ccm_connectionattempts	300	advanced	N/A	N/A

6.4.2.6 Update `cray_cnat` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Cray Compute Node Administrative Tool (CNAT) is a mechanism for submitting and monitoring the execution of batch scripts; it requires a workload manager (WLM) to function. This procedure disables the Cray CNAT configuration service because CNAT is not needed for a first-time boot of CLE. It can be enabled and configured at a later time when a WLM is installed.

For one use of CNAT, see "Apply Rolling Patches to Compute Nodes" in *XC™ Series System Administration Guide (S-2393)*.

Procedure

1. Edit `cray_cnat_worksheet.yaml`.

```
smw# vi cray_cnat_worksheet.yaml
```

2. Uncomment `cray_cnat.enabled` and set it to `false`.

This disables this service. When a service is disabled, the values of all other settings in the service are ignored when configuration is applied.

6.4.2.7 About Configuring Cray Dynamic RDMA Credentials (DRC)

Dynamic RDMA Credentials (DRC) is a new XC system service that enables shared network access between different user applications. DRC enables user applications to request managed network credentials, which can be shared with other users, groups, or jobs. Access to a credential is governed by the application and DRC to provide authorized and protected sharing of network access between applications. DRC extends the existing protection domain functionality provided by ALPS without exposing application data to unauthorized applications. DRC can also be used with other batch systems, such as Slurm, without any loss of functionality.

Trouble? Do not use DRC with VMDH (virtual memory domain handle). DRC does not use VMDH or limit its use; however, in a MAMU (multiple application multiple user) scenario, the use of VMDH by an application that is also using DRC could cause problems for other applications using VMDH on the same node, resulting in the failure of one or more of those processes.

When configuring Cray DRC, using the default values of the following settings will be sufficient for most cases. There are two required settings that must be configured with site-specific information however: `server_cname` and `cookie_provider`, which are both DRC server settings. Those must be assigned non-null values to complete the configuration process.

DRC Client (DRCC) Settings

None of the DRCC settings are required.

socket_location	Location of the DRCC UNIX domain socket. This location should allow read-write access for any user, because libDRC must be able to write to the socket to make any necessary requests. Default value: <code>/tmp/drcc.sock</code>
logging_directory	Storage location for DRCC logs. This can be located anywhere convenient, as long as the directory is: <ul style="list-style-type: none">• (required) writeable by root• (recommended) persistent between reboots so that the log file can be retrieved in a node-down event Default value: <code>/tmp</code>
logging_filename	Name of the log file for DRCC. This name can be anything except a null value. Default value: <code>drcc.log</code>
logging_level	Verbosity of the DRCC logger. Possible values in order of increasing verbosity: <code>critical</code> , <code>error</code> , <code>warning</code> , <code>info</code> , and <code>debug</code> . Default value: <code>error</code>
requests_log_level	Verbosity of the python-requests logger. Possible values in order of increasing verbosity: <code>critical</code> , <code>error</code> , <code>warning</code> , <code>info</code> , and <code>debug</code> . Default value: <code>error</code>
llm_log_enabled	If enabled, DRCC will log messages to the lightweight log management (LLM) service. Default value: <code>true</code>
llm_log_level	Verbosity of DRCC log messages to the LLM service. Possible values in order of increasing verbosity: <code>critical</code> , <code>error</code> , <code>warning</code> , <code>info</code> , and <code>debug</code> . Default value: <code>error</code>

DRC Server (DRCS) Settings

Two of the DRCS settings are required: `server_cname` and `cookie_provider`.

server_cname (REQUIRED)	The cname of the node where DRCS will reside (e.g., <code>c0-0c1s4n0</code>). DRCS can reside on a login node or any unspecialized service node, but NOT on any boot or SDB nodes. This cname is also used as the value for the clients setting when preparing a persistent mount for DRC in <code>cray_persistent_data_worksheet.yaml</code> (see the <code>database_directory</code> setting below). Because this is a required field and no default is provided, a value must be entered.
logging_directory	Storage location for DRCS logs. This can be located anywhere convenient as long as the directory is: <ul style="list-style-type: none">• (required) writeable by root

- (recommended) persistent between reboots so that the log file can be retrieved in a node-down event

Default value: `/tmp`

logging_filename	Name of the log file for DRCS. This name can be anything except a null value. Default value: <code>drcc.log</code>
logging_level	Verbosity of the DRCS logger. Possible values in order of increasing verbosity: <code>critical</code> , <code>error</code> , <code>warning</code> , <code>info</code> , and <code>debug</code> . Default value: <code>error</code>
port	TCP port on which the DRC server will listen to requests. Do not assign this port to any other TCP service. Default value: <code>4000</code>
use_ssl	Should the DRCS server use SSL? This additional layer of security is not necessary but is recommended. Default value: <code>true</code>
rpc_uri	Remote procedure call (RPC) URI used by both client and server to correctly address DRCS services. Default value: <code>json-rpc</code>
werkzeug_log_level	Verbosity of the python-werkzeug logger. Possible values in order of increasing verbosity: <code>critical</code> , <code>error</code> , <code>warning</code> , <code>info</code> , and <code>debug</code> . Default value: <code>error</code>
jsonrpc_log_level	Verbosity of the python-jsonrpc logger. Possible values in order of increasing verbosity: <code>critical</code> , <code>error</code> , <code>warning</code> , <code>info</code> , and <code>debug</code> . Default value: <code>error</code>
requests_log_level	Verbosity of the python-requests logger. Possible values in order of increasing verbosity: <code>critical</code> , <code>error</code> , <code>warning</code> , <code>info</code> , and <code>debug</code> . Default value: <code>error</code>
authorized_uids	List of UIDs that are allowed to interface directly with DRCS through DRCC, DRCCLI, and DRCJEDi (DRC job expiration director). If DRCC, DRCCLI, or DRCJEDi is run under a UID that is not in this list, any request made by that user will be rejected. Default value: <code>['0']</code>
admin_uids	List of UIDs that are allowed to run DRCCLI. At present, this is limited to the values in the <code>authorized_uids</code> list. Default value: <code>['0']</code>
cookie_provider (REQUIRED)	A string that indicates which workload manager binary DRCS should contact for cookies. Possible values: <code>apmgr</code> For systems running ALPS (Application Level Placement Service) <code>ncmd</code> For systems running Slurm (the native workload manager) Because this is a required field and no default is provided, a value must be entered.

llm_log_enabled	If enabled, DRCS will log messages to the lightweight log management (LLM) service. Default value: <code>true</code>
llm_log_level	Verbosity of DRCS log messages to the LLM service. Possible values in order of increasing verbosity: <code>critical</code> , <code>error</code> , <code>warning</code> , <code>info</code> , and <code>debug</code> . Default value: <code>error</code>
database_directory	Storage location for the credential database. This needs to be persistent <code>/var/</code> so that the database can support node restart features. This is the same path name used in <code>cray_persistent_data_worksheet.yaml</code> to set up a persistent mount for DRC (see Update cray_persistent_data Worksheet on page 266). Default value: <code>/var/opt/cray/rdma-credentials</code>
database_filename	Name of the credential database file. Default value: <code>drc.db</code>

6.4.2.8 Update `cray_drc` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Cray dynamic RDMA credentials (DRC) service configures dynamic RDMA (remote direct memory access) credentials, which are secure network credentials that can be shared between user applications to achieve intercommunication between applications running in different job reservations. This procedure configures some basic settings in the `cray_drc` configuration worksheet to add site-specific data.

This service is disabled by default. For additional information about Cray DRC to help decide whether to enable it and know what configuration parameters are available, see [About Configuring Cray Dynamic RDMA Credentials \(DRC\)](#) on page 161. To enable and use this service, follow these steps.

NOTICE: Do not use DRC with VMDH (virtual memory domain handle).

Note that the database directory for DRC must be persistent storage to have persistent credentials and support node restart features. The server `cname` (step 3) and directory path (step 5) specified in this worksheet will be used in `cray_persistent_data_worksheet.yaml` to set up a client mount for persistent storage.

Procedure

1. Edit `cray_drc_worksheet.yaml`.

```
smw# vi cray_drc_worksheet.yaml
```

2. Uncomment `cray_drc.enabled`.

- To disable this service, set to `false` and skip the rest of the procedure.
- To enable this service, set to `true` and continue to the next step.

3. Uncomment `cray_drc.settings.server.data.server_cname` and set it to the `cname` of the service node that should be running the DRC server.

This `cname` will also be used as the value for the `clients` setting when preparing a persistent mount for DRC in the Cray persistent data worksheet (`cray_persistent_data_worksheet.yaml`).

4. Uncomment `cray_drc.settings.server.data.cookie_provider` and set it to one of these values:

- `apmgr` if using an ALPS (Application Level Placement Scheduler) workload manager (WLM)
- `ncmd` if using the Slurm WLM

5. Uncomment `cray_drc.settings.server.data.database_directory` and set it to `/var/opt/cray/rdma-credentials`.

This needs to be persistent storage so that the database can have persistent credentials and support node restart features. This is the same path name that will be used as the mount point when setting up a persistent mount for DRC in `cray_persistent_data_worksheet.yaml` (see [Update cray_persistent_data Worksheet](#) on page 266).

6. Go back and uncomment the following settings, and set them in accordance with site preferences.

Cray recommends configuring these settings so that diagnostic information is available if needed. Using persistent storage for the logging directories is best; however that depends on available storage space.

```
cray_drc.settings.client.data.logging_directory
cray_drc.settings.client.data.logging_filename
cray_drc.settings.server.data.logging_directory
cray_drc.settings.server.data.logging_filename
```

If this system uses ALPS, Cray recommends increasing the maximum number of user protection domains when DRC is in use, especially for large systems. That parameter is set in the Cray ALPS service with the `pDomainMax` field. See [Update cray_alps Worksheet](#) on page 138.

6.4.2.9 Update `cray_dvs` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

Cray DVS (Data Virtualization Service) is a distributed network service that projects local file systems resident on I/O nodes or remote file servers to compute and service nodes within the Cray system. DVS provides a highly scalable mechanism to share file systems to a large number of client nodes using a fanout tree as configured in `cray_scalable_services`. This service must be enabled if Programming Environment (PE) software is to be used

on compute and login nodes. It is also required if Netroot image roots will be used on compute and login nodes (Netroot is a mechanism that enables nodes booted with a minimal, local in-memory file system to execute within the context of a larger, full-featured root file system.).

This procedure enables the `cray_dvs` configuration service. For a migration, this procedure also includes steps and translation tables to migrate site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_dvs_worksheet.yaml`.

```
smw# vi cray_dvs_worksheet.yaml
```

2. Uncomment `cray_dvs.enabled` and set it to `true`.

MIGRATE CONFIGURATION DATA -----

Use the translation tables below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

3. Migrate kernel parameters, as needed.

These are tuning settings that will probably not need to be changed.

Table 37. Variables beginning with `cray_dvs.settings.kernel_param.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>dvsipc_heartbeat_timeout</code>	60	advanced	N/A	<i>files:</i> default sharedroot /etc/modprobe.d/dvs (<code>dvsipc_heartbeat_timeout</code>)
<code>dvs_debug_mask</code>	0	advanced	N/A	<i>files:</i> default sharedroot /etc/modprobe.d/dvs (<code>dvs_debug_mask</code>)

4. On the SMW running CLE 5.2 / SMW 7.2, find out if any DVS-projected file systems are in `/etc/fstab` for compute nodes.

```
smw# grep dvs /opt/xt-images/templates/default/etc/fstab
/ufs /ufs dvs path=/ufs,nodename=c0-0c0s3n1
/cray/css /cray/css dvs path=/cray/
css,nodename=c0-0c0s3n1:c0-0c0s4n1,loadbalance,cache,ro,noauto,attrcache_timeout=14400
```

In the example output, there are two DVS mount points. The `/ufs` file system is projected from a single DVS server, `c0-0c0s3n1`. The `/cray/css` file system is projected from a pair of DVS servers, `c0-0c0s3n1` and `c0-0c0s4n1`.

- The first projects `/ufs` so that the home directories under `/ufs/home` will be the same on the login nodes and the compute nodes. The `/ufs` file system is NFS-mounted on the DVS server `c0-0c0s3n1` from the boot node.
- The second projects `/cray/css`, which is a file system external to the XC system that has been mounted on the DVS servers `c0-0c0s3n1` and `c0-0c0s4n1` in a load balancing, readonly way with an attribute cache timeout of 14400. This external file system that is mounted on the DVS server may be any network-based file system, such as NFS or GPFS.

5. Configure any mount points found in step 4 on page 166 using migrated configuration data.

The following translation table shows how these two example file systems would be configured in the `cray_dvs` configuration worksheet and where to find the data to migrate.

The `/ufs/home` from CLE 5.2 / SMW 7.2 has become `/cray_home` in CLE 6.0 / SMW 8.0 and will be called `ComputeHome`. This file system can be DVS-projected by the tier2 servers. The `/cray/css` file system will be projected by the same service nodes as were used on the CLE 5.2 / SMW 7.2 system, `c0-0c0s3n1` and `c0-0c0s4n1`, and will be called `CSS`. The nodes `c0-0c0s3n1` and `c0-0c0s4n1` need to be added to a new `node_group` called "css_dvs_servers" in `cray_node_groups`.

Both of these file systems were mounted on all compute nodes in CLE 5.2 / SMW 7.2 in the example. In CLE 6.0 / SMW 8.0, the client nodes that will mount the DVS-projected file system from the DVS server are in the `client_groups` setting. If no node groups are specified, then every suitable compute node will perform the given mount. Both of these file systems have been given `client_groups` that are an empty list of node groups.

Table 38. Variables beginning with `cray_dvs.settings.client_mount.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
reference.ComputeHome	null	basic	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
ComputeHome.mount_point	'/cray_home'	basic	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
ComputeHome.spath	'/cray_home'	basic	N/A	files: DVS node sharedroot /etc/fstab
ComputeHome.server_groups	tier2_nodes	basic	N/A	N/A
ComputeHome.client_groups	[]	basic	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
ComputeHome.loadbalance	false	advanced	N/A	files: SMW

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/opt/xt-images/ templates/default/etc/ fstab
ComputeHome.attrcache_timeout	0	advanced	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
ComputeHome.readonly	false	basic	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
ComputeHome.options	'maxnodes=1'	advanced	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
reference.CSS	null	basic	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
CSS.mount_point	'/cray/css'	basic	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
CSS.spath	'/cray/css'	basic	N/A	files: DVS node sharedroot /etc/fstab
CSS.server_groups	css_dvs_servers	basic	N/A	N/A
CSS.client_groups	[]	basic	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
CSS.loadbalance	true	advanced	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
CSS.attrcache_timeout	14400	advanced	N/A	files: SMW

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/opt/xt-images/ templates/default/etc/ fstab
CSS.readonly	true	basic	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab
CSS.options		advanced	N/A	files: SMW /opt/xt-images/ templates/default/etc/ fstab

a. Configure the ComputeHome client mount.

In the worksheet, copy the nine lines below # ** EXAMPLE 'client_mount' VALUE (with current defaults) ** and paste them below # NOTE: Place additional 'client_mount' setting entries here, if desired.

```
# ** EXAMPLE 'client_mount' VALUE (with current defaults) **
#  cray_dvs.settings.client_mount.data.reference.sample_key_a: null  <-- setting a multival key
#  cray_dvs.settings.client_mount.data.sample_key_a.mount_point:
#  cray_dvs.settings.client_mount.data.sample_key_a.spath:
#  cray_dvs.settings.client_mount.data.sample_key_a.server_groups: []
#  cray_dvs.settings.client_mount.data.sample_key_a.client_groups: []
#  cray_dvs.settings.client_mount.data.sample_key_a.loadbalance: false
#  cray_dvs.settings.client_mount.data.sample_key_a.attrcache_timeout: 14400
#  cray_dvs.settings.client_mount.data.sample_key_a.readonly: true
#  cray_dvs.settings.client_mount.data.sample_key_a.options:
```

Uncomment the lines, replace `sample_key_a` in all lines with a string that identifies that mount point, and remove the `<-- setting a multival key` text at the end of the first line (note that the null value is required; do not remove or change it). Finally, modify the values as shown below.

```
# NOTE: Place additional 'client_mount' setting entries here, if desired.
cray_dvs.settings.client_mount.data.reference.ComputeHome: null
cray_dvs.settings.client_mount.data.ComputeHome.mount_point: /cray_home
cray_dvs.settings.client_mount.data.ComputeHome.spath: /cray_home
cray_dvs.settings.client_mount.data.ComputeHome.server_groups:
- tier2_nodes
cray_dvs.settings.client_mount.data.ComputeHome.client_groups: []
cray_dvs.settings.client_mount.data.ComputeHome.loadbalance: false
cray_dvs.settings.client_mount.data.ComputeHome.attrcache_timeout: 0
cray_dvs.settings.client_mount.data.ComputeHome.readonly: false
cray_dvs.settings.client_mount.data.ComputeHome.options: maxnodes=1

#***** END Service Setting: client_mount *****
```

b. Configure the CSS client mount.

In the worksheet, copy the nine lines below # ** EXAMPLE 'client_mount' VALUE (with current defaults) **, as in the previous substep (for the ComputeHome client mount), and paste them below the lines added in that substep.

Uncomment the lines, replace `sample_key_a` in all lines with a string that identifies that mount point, and remove the `<-- setting a multival key text` at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, modify the values as shown below.

```
# NOTE: Place additional 'client mount' setting entries here, if desired.
cray_dvs.settings.client_mount.data.reference.CSS: null
cray_dvs.settings.client_mount.data.CSS.mount_point: /cray/css
cray_dvs.settings.client_mount.data.CSS.spath: /cray/css
cray_dvs.settings.client_mount.data.CSS.server_groups:
- css_dvs_servers
cray_dvs.settings.client_mount.data.CSS.client_groups: []
cray_dvs.settings.client_mount.data.CSS.loadbalance: true
cray_dvs.settings.client_mount.data.CSS.attrcache_timeout: 14400
cray_dvs.settings.client_mount.data.CSS.readonly: true
cray_dvs.settings.client_mount.data.CSS.options: maxnodes=1

#***** END Service Setting: client_mount *****
```

6. Add any needed node groups in the `cray_node_groups` configuration worksheet.

For the CSS file system of the previous example, this entry would be added to the `cray_node_groups` worksheet to create the `css_dvs_servers` node group.

```
cray_node_groups.settings.groups.data.group_name.css_dvs_servers: null
cray_node_groups.settings.groups.data.css_dvs_servers.description: Node group
that contains
  all the DVS servers projecting the CSS file system
cray_node_groups.settings.groups.data.css_dvs_servers.members:
- c0-0c0s3n1
- c0-0c0s4n1
```

The DVS servers might mount the remote file system via NFS or GPFS directly in `/etc/fstab` or they might use the automounter to mount it upon request. Either way, some files will need to be put in the Simple Sync directory structure later to distribute either `/etc/fstab` or the `/etc/auto*` files. See [Configure Simple Sync for DVS Server Nodes](#) on page 307.

DVS uses the LNet (Lustre networking) networking layer, so ensure that `cray_lnet` is enabled as well. See [Update cray_lnet Worksheet](#) on page 180.

6.4.2.10 Update `cray_dw_wlm` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The `cray_dw_wlm` service is an interface used by Workload Managers (WLM) to interact with Cray DataWarp. It should be enabled on every system with Cray DataWarp. This procedure enables `dw_wlm`, but no other settings are changed at this point in the fresh install process. See *XC™ Series DataWarp™ Installation and Administration Guide* for information about how to use this configuration service to set limits on what options users can add to DataWarp commands in their job scripts.

Procedure

1. Edit `cray_dw_wlm_worksheet.yaml`.
2. Uncomment `cray_dw_wlm.enabled` and set it to `true`.

No other settings need to be changed for a fresh install.

6.4.2.11 Update `cray_dws` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray DataWarp Service (DWS) provides access to SSD (solid state device) storage for high bandwidth application I/O.

This procedure disables the `cray_dws` configuration service or enables and configures it depending on whether this site currently has DataWarp.

Is DataWarp enabled in CLE 5.2?

- If NO, then disable the service and skip the rest of the configuration.
- If YES, then enable and configure the service using this procedure. The last step provides translation tables for migrating site-specific DataWarp configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Notes on DataWarp SSDs:

- If this site has not reformatted/over-provisioned Intel P3608 SSD cards as directed in FN6121a *Datawarp - Performance Issues*, then these Intel P3608 SSD cards must be reformatted. This will be done during the "Shutdown and Switch" phase of the migration process.
- DataWarp Fusion IO SSDs that are ioMemory3 (for example, SX300) are supported in the CLE 6.0.UP03 / SMW 8.0.UP03 release, but no other models from Fusion IO are supported. The SLES 12 version of SanDisk/ Fusion driver (VSL4.2.5) requires firmware version 8.9.5. Sites may need to update (flash) the driver firmware to 8.9.5. However, once updated, the firmware cannot be reverted to the previous version. **DO NOT UPDATE FIRMWARE NOW.** That will be done during the "Shutdown and Switch" phase of the migration process.



CAUTION: Once updated, the firmware revision cannot be reverted to the previous version, so the SSDs will NOT be usable in a CLE 5.2 / SMW 7.2 system.

- DataWarp SanDisk Fusion ioScale2 SSD PCIe boards are no longer supported with CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_dws_worksheet.yaml`.

```
smw# vi cray_dws_worksheet.yaml
```

2. (Sites with DataWarp NOT enabled) Uncomment `cray_dws.enabled` and set it to `false`.

Save and exit the `cray_dws` worksheet and skip the rest of this procedure.

DataWarp can be enabled and configured any time after the migration is completed. For this post-migration scenario, follow the procedures in *XC™ Series DataWarp™ Installation and Administration Guide (S-2564)*.

3. (Sites with DataWarp enabled) Uncomment `cray_dws.enabled` and ensure that it is set to `true`.

MIGRATE CONFIGURATION DATA -----

Migrate CLE 5.2 / SMW 7.2 DataWarp configuration information using the steps below. After CLE has been booted, there are additional steps to be done on the DataWarp nodes with SSDs beyond what is in this `cray_dws` configuration worksheet. Those will be addressed later in the migration process.

4. Set the DataWarp managed nodes variable.

Uncomment `#cray_dws.settings.service.data.managed_nodes_groups` and set it to a list of node groups that contains only a new DataWarp node group that will be defined for this system in a later step.

```
cray_dws.settings.service.data.managed_nodes_groups:
- datawarp_nodes
```

5. Set the DataWarp API gateways variable.

In CLE 5.2 / SMW 7.2, the `CLEinstall.conf` file could have the CLE nodes listed in `datawarp_api_gateways` as `cname`, `nidXXXXX`, or `hostname`. By contrast, in CLE 6.0 / SMW 8.0, the similar variable (`cray_dws.settings.service.data.api_gateway_nodes_groups`) must reference a node group that can contain only `cnames`.

Uncomment `cray_dws.settings.service.data.api_gateway_nodes_groups` and set it to a list of node groups that will contain the DataWarp gateway nodes. Typically this is the set of login nodes, but sites can define a subset of login nodes as gateway nodes. This example assumes that all nodes in the predefined `login_nodes` node group are gateway nodes.

```
cray_dws.settings.service.data.api_gateway_nodes_groups
- login_nodes
```

6. Define the DataWarp node group and customize the predefined node group for login nodes.

- a. Edit `cray_node_groups_worksheet.yaml`.

```
smw# vi cray_node_groups_worksheet.yaml
```

- b. Define the new DataWarp node group.

In the worksheet, copy the three lines below `# ** EXAMPLE 'groups' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'groups' setting entries here, if desired`.

```
# ** EXAMPLE 'groups' VALUE (with current defaults) **
# NOTE: Place additional 'groups' setting entries here, if desired.
#   cray_node_groups.settings.groups.data.group_name.sample_key_a: null    <-- setting a multival
key
#   cray_node_groups.settings.groups.data.sample_key_a.description: ''
#   cray_node_groups.settings.groups.data.sample_key_a.members: []
```

Uncomment the lines, replace `sample_key_a` with `datawarp_nodes` in all lines, and remove the `<--` setting a multival key text at the end of the first line (note that the null value is required; do not remove or change it). For the node group members, substitute the correct `cname(s)` for this system. Use the translation table in the last step of this procedure for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

```
# NOTE: Place additional 'groups' setting entries here, if desired.
cray_node_groups.settings.groups.data.group_name.datawarp_nodes: null
cray_node_groups.settings.groups.data.datawarp_nodes.description:
  Node group that contains all DataWarp managed nodes with SSDs
cray_node_groups.settings.groups.data.datawarp_nodes.members:
- c0-0c0s5n1
- c0-0c1s6n1
- c0-1c2s4n0
```

- c. Customize the predefined `login_nodes` node group for DataWarp gateway node(s).
 - Use the translation table in the last step of this procedure to see what nodes were defined as `datawarp_api_gateways` in `CLEinstall.conf` for CLE 5.2 / SMW 7.2 and verify that they are in the `login_nodes` node group. If all the DataWarp api gateway nodes are equivalent to those in the `login_nodes` node group, then use the already defined `login_nodes` group. It may be necessary to define a DataWarp-specific login node group. In that case, use the instructions in the previous step to define it.
 - If the predefined `login_nodes` node group has NOT been customized for this system, complete this step.

In the worksheet, find the definition for `login_nodes` under `# ** 'groups' DATA **`.

```
cray_node_groups.settings.groups.data.group_name.login_nodes: null
cray_node_groups.settings.groups.data.login_nodes.description: Default node
group
  which contains the login nodes for the configured system.
#cray_node_groups.settings.groups.data.login_nodes.members: []
```

Uncomment `cray_node_groups.settings.groups.data.login_nodes.members`, remove the empty brackets, and add the `cnames` of login nodes on this system on separate lines prefixed by a hyphen and space (`-`). Use the translation table in the last step of this procedure for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

```
cray_node_groups.settings.groups.data.login_nodes.members:
- c0-0c0s3n2
- c0-0c0s1n1
```

Save and exit the `cray_node_groups` worksheet.

7. Migrate remaining configuration data, as needed.

Use the translation tables below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to CLE 6.0 / SMW 8.0 settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Notes about migrating DataWarp settings:

- This setting should be set to the empty list because it is not yet fully functional.

```
cray_dws.settings.service.data.external_api_gateway_hostnames: []
```

- `dwrest_cacheroot_whitelist` and `dwrest_cacheroot_whitelist` are new configuration variables for the CLE 6.0 release. Sites that want to allow users to mount cache file systems must specify those PFS (parallel

file system) paths allowed for user cache file system mounts. See *XC™ Series DataWarp™ Installation and Administration Guide (S-2564)* section 2.3 "Use the Configurator for Initial DataWarp Setup" steps 9 and 10 for more information about these variables and how to set them.

Table 39. Variables beginning with `cray_dws.settings.service.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
managed_nodes_groups	[]	required	N/A	<i>installer:</i> CLEinstall.conf (datawarp_manager_nodes)
api_gateway_nodes_groups	[]	required	N/A	<i>installer:</i> CLEinstall.conf (datawarp_api_gateways)
external_api_gateway_hostnames	[]	basic	N/A	N/A
dwrest_cacheroot_whitelist	[]	required	N/A	N/A
dwrest_cachemount_whitelist	[]	required	N/A	N/A
allow_dws_cli_from_computes	false	required	N/A	N/A
lvm_issue_discards	0	advanced	N/A	N/A

Table 40. Variables beginning with `cray_dws.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
dwmd.data.dwmd_conf	- 'iscsi_initiator_cred_path: / etc/opt/cray/dws/ iscsi_target_secret' - 'iscsi_target_cred_path: / etc/opt/cray/dws/ iscsi_initiator_secret' - 'capmc_os_cacert: /etc/ pki/trust/anchors/ certificate_authority.pem'	advanced	N/A	N/A
dwsd.data.dwsd_conf	- 'log_mask: 0x7' - 'instance_optimization_def ault: bandwidth' - 'scratch_limit_action: 0x3'	advanced	N/A	<i>files:</i> sdb node sharedroot /etc/opt/cray/dws/ dwsd.yaml (log_mask instance_optimization_d efault scratch_limit_action)
dwrest.data.dwrest_conf	- 'port: 2015'	advanced	N/A	<i>files:</i> API gateway login node sharedroot /etc/opt/cray/dws/ dwrest.yaml (port)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
dwrestgun.data.dwrestgun_conf	- max_requests=1024	advanced	N/A	N/A

6.4.2.12 Update Cray eLogin Service Worksheets

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software. Note that eLogin nodes were formerly called CDL nodes or esLogin nodes.

This procedure temporarily disables these eLogin (external login) configuration services at this point in the migration process. They will be enabled at the appropriate place in the eLogin migration process, which is documented in *XC™ Series esLogin to eLogin Migration Guide S-2584 Rev A*.

- Cray eLogin LNet service** LNet (Lustre networking) is needed by any system that has external login nodes that mount Lustre file systems.
- Cray eLogin MOTD service** Generates the `/etc/motd` file for the eLogin nodes specified in the configuration set.
- Cray eLogin Networking service** Defines the number of eLogin nodes connected to the Cray system and their key network attributes.
- Cray eswrap service** `eswrap` wraps several XT, ALPS (Application Level Placement Scheduler), and WLM (workload manager) commands on eLogin nodes and executes them on the Cray login gateway.

Procedure

1. Disable the Cray eLogin LNet service.
 - a. Edit `cray_elogin_lnet_worksheet.yaml`.

```
smw# vi cray_elogin_lnet_worksheet.yaml
```
 - b. Uncomment `cray_elogin_lnet.enabled` and set it to `false`.

2. Disable the Cray eLogin MOTD service.
 - a. Edit `cray_elogin_motd_worksheet.yaml`.

```
smw# vi cray_motd_worksheet.yaml
```

- b. Uncomment `cray_elogin_motd.enabled` and set it to `false`.
3. Disable the Cray eLogin Networking service.
 - a. Edit `cray_elogin_networking_worksheet.yaml`.

```
smw# vi cray_elogin_networking_worksheet.yaml
```
 - b. Uncomment `cray_elogin_networking.enabled` and set it to `false`.
4. Disable the Cray eswrap service.
 - a. Edit `cray_eswrap_worksheet.yaml`.

```
smw# vi cray_eswrap_worksheet.yaml
```
 - b. Uncomment `cray_eswrap.enabled` and set it to `false`.

6.4.2.13 Update `cray_firewall` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Cray firewall service is a mechanism for restricting packet traffic from various networks. This procedure configures the inherit and enable settings in the `cray_firewall` configuration worksheet. This service has a global template as well as a CLE template, and therefore it can be configured to inherit settings from the global config set.

Procedure

1. Edit `cray_firewall_worksheet.yaml`.

```
smw# vi cray_firewall_worksheet.yaml
```
2. Uncomment `cray_firewall.inherit` and ensure that it is set to `false`.

This means that firewall settings in the CLE config set will be used instead of firewall settings in the global config set.
3. Uncomment `cray_firewall.enabled` and set it to `true`.

6.4.2.14 Update `cray_image_binding` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Cray image binding service is a mechanism for mixing image content between booted IMPS (Image Management and Provisioning System) images and IMPS images that are projected onto a running system. This is a common scenario with the Cray Programming Environment (PE), which is installed into an IMPS image, pushed to the CLE boot node, then projected to compute nodes using DVS (Data Virtualization Service). The diagnostics (diag) image root is also pushed to the boot node and projected by DVS.

This procedure enables the `cray_image_binding` service and configures the PE and diag bind mount profiles.

Procedure

1. Edit `cray_image_binding_worksheet.yaml`.

```
smw# vi cray_image_binding_worksheet.yaml
```

2. Uncomment `cray_image_binding.enabled` and set it to `true`.

3. Configure the PE bind mount profile.

- a. Uncomment the commented PE bind mount settings.

```
cray_image_binding.settings.profiles.data.profile_name.PE: null
#cray_image_binding.settings.profiles.data.PE.image:
pe_compute_cle_6.0up03_sles_12
cray_image_binding.settings.profiles.data.PE.bind_directories: []
#cray_image_binding.settings.profiles.data.PE.callbacks:
#- opt/cray/pe/bin/pe_postmount_callback.sh
#cray_image_binding.settings.profiles.data.PE.enabled: false
```

Note that the name of the image for the PE profile will be used later to create an image root from the PE recipe, and then the image root will be pushed to the boot node.

- b. Enable the PE profile.

```
cray_image_binding.settings.profiles.data.PE.enabled: true
```

4. Configure the diags bind mount profile.

- a. Uncomment the commented diags bind mount settings.

```
cray_image_binding.settings.profiles.data.profile_name.diags: null
#cray_image_binding.settings.profiles.data.diags.image:
```

```
diags_cle_6.0up03_sles_12_x86-64
cray_image_binding.settings.profiles.data.diags.bind_directories:
- /opt/cray/diag
#cray_image_binding.settings.profiles.data.diags.callbacks: []
#cray_image_binding.settings.profiles.data.diags.enabled: false
```

Note that the name of the diags image will be used to create an image root as specified in `cray_image_groups.yaml` in the global config set directory when the `imgbuilder` command is run. That image root will be pushed to the boot node later in the process.

- b. Enable the diags profile.

```
cray_image_binding.settings.profiles.data.diags.enabled: true
```

6.4.2.15 Update `cray_ipforward` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Cray IP Forwarding service enables IP forwarding between service nodes and the SMW. This procedure configures the inherit and enable settings in the `cray_ipforward` configuration worksheet. This service has a global template as well as a CLE template, and therefore it can be configured to inherit settings from the global config set.

Procedure

1. Edit `cray_ipforward_worksheet.yaml`.

```
smw# vi cray_ipforward_worksheet.yaml
```

2. Uncomment `cray_ipforward.inherit` and set it to `true`.

This means that IP forwarding settings in the global config set will be used instead of IP forwarding settings in the CLE config set.

3. Uncomment `cray_ipforward.enabled` and ensure that it is set to `true`.

6.4.2.16 Update `cray_liveupdates` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The live updates service enables package manager (e.g., zypper, yum) actions (e.g., install, search, upgrade) on CLE nodes using repositories shared from the SMW to those nodes. This procedure sets the CLE `cray_liveupdates` service to inherit from the global `cray_liveupdates` service. There are no other settings that can be changed.

Procedure

1. Edit `cray_liveupdates_worksheet.yaml`.

```
smw# vi cray_liveupdates_worksheet.yaml
```

2. Uncomment `cray_liveupdates.inherit` and set it to `true`.

6.4.2.17 Update `cray_lmt` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray Lustre Monitoring Tool (LMT) monitors Lustre servers using the Cerebro monitoring system. An LMT Cerebro module collects stats published in `/proc/fs/lustre` on the Lustre servers, and pushes them to the LMT server, which is co-located with the Lustre Management Server (MGS). A Cerebro module on the LMT server collects the statistics and stores them in a MySQL database. Lustre clients are not monitored.

This procedure disables the `cray_lmt` configuration service. If this is a migration, and the CLE 5.2 / SMW 7.2 system currently uses DAL (direct-attached Lustre) and LMT, then some settings in the `cray_lmt` configuration worksheet can be set at this time, even with the service disabled.

Procedure

1. Edit `cray_lmt_worksheet.yaml`.

```
smw# vi cray_lmt_worksheet.yaml
```

2. Uncomment `cray_lmt.enabled` and ensure that it is set to `false`.

This disables this service. When a service is disabled, the values of all other settings in the service are ignored when configuration is applied. This worksheet contains additional configuration settings that are level advanced, but this service and its advanced settings are not needed for a fresh install.

6.4.2.18 Update `cray_lnet` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

LNet (Lustre networking) is the networking layer used by Lustre and DVS. The Cray LNet configuration service must be configured on any systems that use DVS to mount external file systems or have Lustre clients and/or servers.

This procedure configures some basic settings in the `cray_lnet` configuration worksheet to add site-specific data. For a migration, this procedure also provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_lnet_worksheet.yaml`.

```
smw# vi cray_lnet_worksheet.yaml
```

2. Uncomment `cray_lnet.enabled` and do one of the following:

- Set it to `true` if this system has external Lustre or DAL (direct-attached Lustre) or will use DVS to mount external file systems.
- Set it to `false` otherwise.

For systems with external Lustre, continue to the next step. If this is a migration and `cray_lnet` was enabled, then use the `cray_lnet` translation tables for help migrating site-specific LNet configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

THE REMAINING STEPS ARE ONLY FOR SYSTEMS WITH EXTERNAL LUSTRE *****

See also the *XC™ Series Lustre® Administration Guide (S-2648)*.

3. Configure these settings.

These settings are commonly configured with site-specific data when the system has external Lustre. Uncomment and set them as appropriate for this site.

```
cray_lnet.settings.ko2iblnd.data.peer_credits
cray_lnet.settings.ko2iblnd.data.concurrent_sends
```

`cray_lnet.settings.local_lnet.data.lnet_name` (set to something like `gni`, `gni1`, `gni2`, `gni3`)
`cray_lnet.settings.local_lnet.data.ip_wildcard` (change from default on a partitioned system or any system that changes the HSN (high speed network) address range)

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these and other settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Things to note:

- Some sites have node specialized files for the LNet router nodes, but others may have made changes in a `node_class` called `lnet` or another name. The examples below refer to the "lnet node sharedroot" files, so that may be a node specialized file or a class specialized file.
- The CLE 5.2 / SMW 7.2 values of some of the general LNet settings can be found in `/etc/modprobe.conf.local` (on lnet node sharedroot) on lines that start with "options *name*." For example, the value to migrate to the CLE 6.0 / SMW 8.0 variable `cray_lnet.settings.lnet.data.check_routers_before_use` can be found in the first line of this example modprobe file, where *name* is "lnet":

```
options lnet check_routers_before_use=1
options lnet large_router_buffers=1024 small_router_buffers=16384
```

The table entry looks like this:

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>lnet.data.check_routers_before_use</code>	false	advanced	N/A	<i>files</i> : lnet node sharedroot <code>/etc/modprobe.conf.local</code> (lnet <code>check_routers_before_use</code>)

- The local LNet name on this system, `cray_lnet.settings.local_lnet.data.lnet_name`, will be of the form 'gniX' where 'X' is a small nonnegative number or omitted. Some systems will have this set in the modprobe file with the "options lnet networks=" line, which shows "gni," and others with the "options lnet ip2nets=" line, which shows "gni2." Only one of these lines would have been uncommented on the CLE 5.2 / SMW 7.2 system.

```
smw# grep gni /opt/xt-images/templates/default/etc/modprobe.conf
#options lnet networks=gni
options lnet ip2nets="gni2 10.128.*.*; o2ib 10.149.*.*"
```

- If the HSN has been changed from the default of 10.128.0.0 with netmask of 255.252.0.0, then `cray_lnet.settings.local_lnet.data.ip_wildcard` will need to be adjusted to match.
- The `cray_lnet.settings.local_lnet.data.ip_wildcard` setting may have been `10.128.*.*` in CLE 5.2 / SMW 7.2, but the format for this setting should be changed so that it matches the IP addresses of all interfaces on the local LNet. For example, if the local HSN interfaces are all on the network 10.128.0.0 with netmask 255.252.0.0, then the IP wildcard matching all local interfaces would be `"10.[128-131].*.*"`.

```
smw# grep gni /opt/xt-images/templates/default/etc/modprobe.conf
options lnet ip2nets="gni2 10.128.*.*; o2ib 10.149.*.*"
```

- If a local `ip2nets` file was created with some external process or tool, such as `clcv`, for the CLE 5.2 / SMW 7.2 system, then that file should be moved a location in the CLE 6.0 / SMW 8.0 config set in

smw:/var/opt/cray/imps/config/sets/<config_set>/files/roles/lnet/ with the file name set in `cray_lnet.settings.local_lnet.data.ip2nets_file`.

- Several settings that were 1/0 or on/off in CLE 5.2 / SMW 7.2, such as those shown below, are now true/false in CLE 6.0 / SMW 8.0.

```
cray_lnet.settings.kgnilnd.data.peer_health
cray_lnet.settings.lnet.data.check_routers_before_use
```

Table 41. Translation Table: Variables beginning with `cray_lnet.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>kgnilnd.data.credit</code>	2048	advanced	N/A	<i>files:</i> lnet node sharedroot <code>/etc/modprobe.conf.local</code> (kgnilnd credits)
<code>kgnilnd.data.peer_health</code>	true	advanced	N/A	<i>files:</i> lnet node sharedroot <code>/etc/modprobe.conf.local</code> (kgnilnd peer_health)
<code>ko2iblnd.data.timeout</code>	10	advanced	N/A	<i>files:</i> lnet node sharedroot <code>/etc/modprobe.conf.local</code> (ko2iblnd timeout)
<code>ko2iblnd.data.peer_timeout</code>	40	advanced	N/A	<i>files:</i> lnet node sharedroot <code>/etc/modprobe.conf.local</code> (ko2iblnd peer_timeout)
<code>ko2iblnd.data.credits</code>	2048	advanced	N/A	<i>files:</i> lnet node sharedroot <code>/etc/modprobe.conf.local</code> (ko2iblnd credits)
<code>ko2iblnd.data.ntx</code>	2048	advanced	N/A	<i>files:</i> lnet node sharedroot <code>/etc/modprobe.conf.local</code> (ko2iblnd ntx)
<code>ko2iblnd.data.peer_credits</code>	126	basic	N/A	<i>files:</i> lnet node sharedroot <code>/etc/modprobe.conf.local</code> (ko2iblnd peer_credits)
<code>ko2iblnd.data.concurrent_sends</code>	63	basic	N/A	<i>files:</i> lnet node sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/modprobe.conf.local ko2iblnd concurrent_sends)
ko2iblnd.data.peer_buffer_credits	128	advanced	N/A	files: Inet node sharedroot /etc/modprobe.conf.local (ko2iblnd peer_buffer_credits
Inet.data.check_routers_before_use	false	advanced	N/A	files: Inet node sharedroot /etc/modprobe.conf.local (Inet check_routers_before_u se)
Inet.data.router_ping_timeout	50	advanced	N/A	files: Inet node sharedroot /etc/modprobe.conf.local (Inet router_ping_timeout)
Inet.data.dead_router_check_interval	60	advanced	N/A	files: Inet node sharedroot /etc/modprobe.conf.local (Inet dead_router_check_inter val)
Inet.data.live_router_check_interval	60	advanced	N/A	files: Inet node sharedroot /etc/modprobe.conf.local (Inet live_router_check_interv al)
Inet.data.large_router_buffers	1024	advanced	N/A	files: Inet node sharedroot /etc/modprobe.conf.local (Inet large_router_buffers)
Inet.data.small_router_buffers	16384	advanced	N/A	files: Inet node sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/modprobe.conf.local (Inet small_router_buffers)
local_inet.data.inet_name	gni	basic	<i>probe (1)</i>	files: Inet node sharedroot /etc/modprobe.conf.local
local_inet.data.ip_wildcard	10.[128-131].*.*	basic	<i>probe (1)</i>	files: Inet node sharedroot /etc/modprobe.conf.local
local_inet.data.ip2nets_file		advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) smw# grep gni /opt/xt-images/templates/default/etc/modprobe.conf				

4. Configure the following group of settings if this system uses flat routes to an external Lustre file system. Repeat this step for each external Lustre file system.

Enter all external LNetS that will be reached via flat routing. The information entered for each of these flat LNetS will be used to set up ip2nets on the routers and routes to reach the external LNetS through the routers on the clients.

In the worksheet, copy the six lines below `# ** EXAMPLE 'flat_routes' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'flat_routes' setting entries here, if desired.`

```
# ** EXAMPLE 'flat_routes' VALUE (with current defaults) **
# cray_inet.settings.flat_routes.data.dest_inet.sample_key_a: null <-- setting a multival key
# cray_inet.settings.flat_routes.data.sample_key_a.dest_inet_ip_wildcard: ''
# cray_inet.settings.flat_routes.data.sample_key_a.router_groups: []
# cray_inet.settings.flat_routes.data.sample_key_a.src_inet: ''
# cray_inet.settings.flat_routes.data.sample_key_a.ko2iblnd_peer_credits: 126
# cray_inet.settings.flat_routes.data.sample_key_a.ko2iblnd_concurrent_sends: 63
#
# ** 'flat_routes' FIELD SPECIFICATION -- MULTIVAL KEY FIELD **
```

Uncomment the lines, replace `sample_key_a` with the name of the LNet on the external Lustre file system (`o2ib` in this example) in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, modify the values as appropriate for this site. For a migration, use the translation table below.

This example uses "o2ib" as the name for this destination LNet and has a `dest_inet_ip_wildcard` of `10.149.*.*`.

IMPORTANT: The settings for `peer_credits` and `concurrent_sends` must match between the external Lustre server and `cray_inet.settings.flat_routes.data.o2ib.ko2iblnd_peer_credits` and `cray_inet.settings.flat_routes.data.o2ib.ko2iblnd_concurrent_sends`.

Note that the `cray_inet.settings.flat_routes.data.o2ib.dest_inet_ip_wildcard` setting is the IP address wildcard that matches the IP addresses of all router interfaces for this flat route. For example, for a flat route from CLE clients to an external Lustre file system, the destination LNet might be 'o2ib', and the wildcard '10.149.*.*' would match the IB interfaces on the router nodes that are to be on the 'o2ib' LNet.

```
# NOTE: Place additional 'flat_routes' setting entries here, if desired.
cray_lnet.settings.flat_routes.data.dest_lnet.o2ib: null
cray_lnet.settings.flat_routes.data.o2ib.dest_lnet_ip_wildcard: 10.149.*.*
cray_lnet.settings.flat_routes.data.o2ib.router_groups:
- lnet_flat_routers
cray_lnet.settings.flat_routes.data.o2ib.src_lnet: gni2
cray_lnet.settings.flat_routes.data.o2ib.ko2iblnd_peer_credits: 63
cray_lnet.settings.flat_routes.data.o2ib.ko2iblnd_concurrent_sends: 63
#***** END Service Setting: flat_routes *****
```

For the flat routes router_groups setting, if there are no existing node groups that contain the router nodes for this site, create one or more node groups for this purpose (*lnet_flat_routers* in this example) using the procedure in [Update cray_node_groups Worksheet](#) on page 236 and reference the node group(s) in `cray_lnet.settings.flat_routes.data.o2ib.router_groups`.

For a migration, use this translation table to help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to flat route settings.

This example shows using the suggested probe to find that "o2ib" is "10.149.*.*" and the gni is "gni2" rather than the default "gni." In this case, the value of the CLE 6.0 / SMW 8.0 variable, `cray_lnet.settings.flat_routes.data.o2ib.src_lnet`, would need to be changed to gni2.

```
smw# grep o2ib /opt/xt-images/templates/default/etc/modprobe.conf
options lnet ip2nets="gni2 10.128.*.*; o2ib 10.149.*.*"
```

Table 42. Translation Table: Variables beginning with `cray_lnet.settings.flat_routes.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
dest_lnet.o2ib	null	basic	N/A	N/A
o2ib.dest_lnet_ip_wildcard		basic	probe (1)	files: lnet node sharedroot /etc/modprobe.conf.local
o2ib.router_groups	[]	basic	N/A	N/A
o2ib.src_lnet	gni	basic	probe (2)	files: lnet node sharedroot /etc/modprobe.conf.local
o2ib.ko2iblnd_peer_credits	126	basic	N/A	files: lnet node sharedroot /etc/modprobe.conf.local (ko2iblnd peer_credits)
o2ib.ko2iblnd_concurrent_sends	63	basic	N/A	files: lnet node sharedroot /etc/modprobe.conf.local (ko2iblnd concurrent_sends)
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) smw# grep o2ib /opt/xt-images/templates/default/etc/modprobe.conf				
(2) smw# grep gni /opt/xt-images/templates/default/etc/modprobe.conf				

- Configure the following group of settings if this system uses fine-grained routing (FGR) to an external Lustre file system. Repeat this step for each external Lustre file system.

Enter all external LNETs that will be reached via FGR. The information entered for each of these FGR routes will be used to set up ip2nets on the routers and routes to reach the external LNETs through the routers on the clients.

In the worksheet, copy the six lines below # ** EXAMPLE 'fgr_routes' VALUE (with current defaults) ** and paste them below # NOTE: Place additional 'fgr_routes' setting entries here, if desired.

```
# ** EXAMPLE 'fgr_routes' VALUE (with current defaults) **
# cray_lnet.settings.fgr_routes.data.dest_name.sample_key_a: null <-- setting a multival key
# cray_lnet.settings.fgr_routes.data.sample_key_a.router_groups: []
# cray_lnet.settings.fgr_routes.data.sample_key_a.ip2nets_file: ''
# cray_lnet.settings.fgr_routes.data.sample_key_a.routes_file: ''
# cray_lnet.settings.fgr_routes.data.sample_key_a.ko2iblnd_peer_credits: 126
# cray_lnet.settings.fgr_routes.data.sample_key_a.ko2iblnd_concurrent_sends: 63
#
# ** 'fgr_routes' FIELD SPECIFICATION -- MULTIVAL KEY FIELD **
```

Uncomment the lines, replace `sample_key_a` with the name of the external Lustre file system to which you are routing (**sonexion** in this example) in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, modify the values as appropriate for this site. For a migration, use the translation table below.

This example uses "sonexion" as the name for this destination LNET.

IMPORTANT: The settings for `peer_credits` and `concurrent_sends` must match between the external Lustre server and `cray_lnet.settings.fgr_routes.data.sonexion.ko2iblnd_peer_credits` and `cray_lnet.settings.fgr_routes.data.sonexion.ko2iblnd_concurrent_sends`.

```
# NOTE: Place additional 'fgr_routes' setting entries here, if desired.
cray_lnet.settings.fgr_routes.data.dest_name.sonexion: null
cray_lnet.settings.fgr_routes.data.sonexion.router_groups:
- lnet_fgr_routers
cray_lnet.settings.fgr_routes.data.sonexion.ip2nets_file: 'ip2nets.conf'
cray_lnet.settings.fgr_routes.data.sonexion.routes_file: 'routes.conf'
cray_lnet.settings.fgr_routes.data.sonexion.ko2iblnd_peer_credits: 126
cray_lnet.settings.fgr_routes.data.sonexion.ko2iblnd_concurrent_sends: 63
#***** END Service Setting: fgr_routes *****
```

To use fine grained routing, the `ip2nets.conf` and `routes.conf` configuration files should be moved a location in the CLE 6.0 / SMW 8.0 config set in

`smw:/var/opt/cray/imps/config/sets/p0/files/roles/lnet` (for p0 config set), and then set the file name in `cray_lnet.settings.fgr_routes.data.sonexion.ip2nets_file` and `cray_lnet.settings.fgr_routes.data.sonexion.routes_file`. These files can be generated with `clcvrt(8)`.

Note that `clcvrt` on the CLE 5.2 / SMW 7.2 system could have generated entries in either the `/etc/modprobe.conf.local` notation and format or the external file format (`ip2nets.conf` and `routes.conf`). If these entries were put into `/etc/modprobe.conf.local` format, then they should be converted to the format for `ip2nets` and `routes` files to use fine-grained routing. Here is an example of using direct entries in `/etc/modprobe.conf.local` for the LNET router on the shared root:

```
options lnet ip2nets="gni0 10.128.*.*;\
o2ib4000(ib0) 10.157.13.1;\
o2ib4000(ib2) 10.157.13.4;\
o2ib4002(ib0) 10.157.9.[2,3,4];\
o2ib4003(ib2) 10.157.11.[1,2,3]"

options lnet routes="o2ib4000 1 [201,206]@gni0;\
o2ib4002 1 [202,205,206]@gni0;\
o2ib4003 1 [201,202,205]@gni0"
```

Here is an example of using file references in `/etc/modprobe.conf.local` for the login node on the shared root to files generated by `clcvrt(8)`.

```
options lnet ip2nets="/etc/lnet/ip2nets.conf"
options lnet routes="/etc/lnet/routes.conf"
```

For the fine-grained routes router_groups setting, if there are no existing node groups that contain the router nodes for this site, create one or more node groups for this purpose (*lnet_fgr_routers* in this example) using the procedure in [Update cray_node_groups Worksheet](#) on page 236 and reference the node group(s) in `cray_lnet.settings.fgr_routes.data.sonexion.router_groups`.

Use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to flat route settings. This table continues the example, using "sonexion" as the name for this destination LNet.

Table 43. Translation Table: Variables beginning with `cray_lnet.settings.fgr_routes.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>dest_name.sonexion</code>	null	basic	N/A	N/A
<code>sonexion.router_groups</code>	[]	basic	N/A	N/A
<code>sonexion.ip2nets_file</code>		basic	N/A	<i>files:</i> lnet node sharedroot /etc/modprobe.conf.local
<code>sonexion.routes_file</code>		basic	N/A	<i>files:</i> lnet node sharedroot /etc/modprobe.conf.local
<code>sonexion.ko2iblnD_peer_credits</code>	126	basic	N/A	<i>files:</i> lnet node sharedroot /etc/modprobe.conf.local (ko2iblnD peer_credits)
<code>sonexion.ko2iblnD_concurrent_sends</code>	63	basic	N/A	<i>files:</i> lnet node sharedroot /etc/modprobe.conf.local (ko2iblnD concurrent_sends)

There may be additional settings that should be set for sites with external Lustre servers. Seek advice from the site Lustre server administrator.

6.4.2.19 Update `cray_local_users` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray Local Users Configuration Service defines local user accounts such as root and crayadm. At a minimum, the root user account must be defined in order to log into the system.

Most sites use an external LDAP or NIS service for account management. The accounts listed in this configuration service are local accounts with entries in the `/etc/passwd` and `/etc/group` files on CLE nodes. Their home directories can be on the boot RAID file system `/cray_home`, such as for `crayadm`, or on an external file system. Most accounts using LDAP or NIS will have an external home directory mounted on a service node (or nodes) that are DVS-projected to the login and compute nodes.

This procedure configures some basic settings in the `cray_local_users` configuration worksheet to add site-specific data. The MIGRATE CONFIGURATION DATA section of this procedure provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_local_users_worksheet.yaml`.

```
smw# vi cray_local_users_worksheet.yaml
```

2. Ensure that `cray_local_users.enabled` is uncommented and set to `true` (it should be by default).
3. If using local home directories (most sites mount an external home file system instead), configure the home directory location in two places in this worksheet and one place in `cray_bootraid_worksheet.yaml`.
 - a. Uncomment this line and replace `/cray_home` with the local home directory for this site.

```
#cray_local_users.settings.directories.data.home: /cray_home
```

- b. Change the home directory for `crayadm` users.

Look for this line in the worksheet:

```
# ** 'users' DATA **
```

Underneath, there are pre-populated 'users' settings for `crayadm` and `root`.

Change the value of the 'crayadm' home directory. Replace `/cray_home/crayadm` with the `crayadm` home directory for this site.

```
cray_local_users.settings.users.data.crayadm.home: /cray_home/crayadm
```

- c. Change the home directory in `cray_bootraid_worksheet.yaml`.

On the boot RAID, the boot node volume group has a file system for home directories mounted by default on `/cray_home`. If this site does not use `cray_home` as the base for local account home directories, then change this setting in `cray_bootraid_worksheet.yaml` in the global config set to match what was set in substep a. The two settings should be in agreement about where local accounts have their home directory.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg.volumes.home.fs_mount_point: /cray_home
```

4. Do nothing to the `crayadm` and `root` accounts "crypt" settings (in the pre-populated 'users' data section).

Do not set the crayadm and root accounts "crypt" settings in the pre-populated 'users' data section, which must be an encrypted string. Later in this process, all of the configuration worksheets will be imported into the new CLE config set, and the config set will be updated. During the update, the configurator will prompt for the crayadm and root "crypt" settings. Because they are encrypted, the configurator will ask for the password, ask a second time to verify that they match, and then put an encrypted form of that password into the config set. Attempting to place an encrypted string into this worksheet manually is prone to error and could result in accounts that cannot be accessed.

5. Ensure that the root domain groups are uncommented.

This parameter is also in the pre-populated 'users' settings under this line in the worksheet:

```
# ** 'users' DATA **

cray_local_users.settings.users.data.root.domain_groups
- all_nodes
```

Make sure both lines are uncommented.

MIGRATE CONFIGURATION DATA -----

Use the translation tables below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to CLE 6.0 / SMW 8.0 configuration settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

6. Migrate other configuration settings, as needed.

Table 44. Translation Table: Variables beginning with `cray_local_users.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
directories.data.home	/cray_home	advanced	N/A	N/A
groups.data.name.crayadm	null	basic	N/A	N/A
groups.data.crayadm.gid	'14901'	basic	N/A	files: default sharedroot /etc/group
groups.data.crayadm.description	Default Cray administrative group	basic	N/A	files: default sharedroot /etc/group
groups.data.crayadm.deleted	false	advanced	N/A	N/A
groups.data.crayadm.domain_groups	- all_nodes	advanced	N/A	N/A
cray_local_users.settings.users.data.name.crayadm	null	basic	N/A	N/A
users.data.crayadm.uid	'12795'	basic	N/A	files: default sharedroot /etc/passwd
users.data.crayadm.group	crayadm	basic	N/A	files: default sharedroot /etc/passwd
users.data.crayadm.crypt		basic	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
users.data.crayadm.other_groups	[]	basic	N/A	files: default sharedroot /etc/group
users.data.crayadm.description	default cray administrative user	basic	N/A	files: default sharedroot /etc/passwd
users.data.crayadm.shell	/bin/bash	basic	N/A	files: default sharedroot /etc/passwd
users.data.crayadm.home	/cray_home/ crayadm	basic	N/A	files: default sharedroot /etc/passwd
users.data.crayadm.system	true	basic	N/A	N/A
users.data.crayadm.deleted	false	advanced	N/A	N/A
users.data.crayadm.domain_groups	- all_nodes	basic	N/A	N/A
users.data.name.root	null	basic	N/A	N/A
users.data.root.uid	'0'	basic	N/A	N/A
users.data.root.group	root	basic	N/A	N/A
users.data.root.crypt		basic	N/A	N/A
users.data.root.other_groups	[]	basic	N/A	files: default sharedroot /etc/group
users.data.root.description	super user	basic	N/A	files: default sharedroot /etc/passwd
users.data.root.shell	/bin/bash	basic	N/A	files: default sharedroot /etc/passwd
users.data.root.home	/root	basic	N/A	files: default sharedroot /etc/passwd
users.data.root.system	true	basic	N/A	N/A
users.data.root.deleted	false	advanced	N/A	N/A
users.data.root.domain_groups	- all_nodes	basic	N/A	files: default sharedroot /etc/passwd

7. Configure additional local accounts.

The `cray_local_users` configuration worksheet will create both the root and crayadm accounts on the CLE 6.0 / SMW 8.0 system. If any site-local accounts exist in the CLE 5.2 / SMW 7.2

sharedroot /etc/passwd file or site local groups exist in the sharedroot /etc/group, then they need to be added to this CLE 6.0 / SMW 8.0 worksheet.

NOTE: In CLE 5.2 / SMW 7.2, the local accounts like crayadm had directories under /ufs/home, but in CLE 6.0 / SMW 8.0 the path has been changed to /cray_home. This enables /home to be used for an external network file system with home directories to be mounted on /home without obscuring the home directory for crayadm.

To add other local accounts (in addition to the pre-populated root and crayadm accounts), copy these two stanzas of information, one for a new account group and one for a new user account.

- Group entry (example shows a group named "Employee group"):

```
# NOTE: Place additional 'groups' setting entries here, if desired.

cray_local_users.settings.groups.data.name.employee: null
cray_local_users.settings.groups.data.employee.gid: '14901'
cray_local_users.settings.groups.data.employee.description: Employee group
cray_local_users.settings.groups.data.employee.deleted: false
cray_local_users.settings.groups.data.employee.domain_groups:
- all_nodes
```

- User account entry (example shows a user account for user "Bob Name"):

```
# NOTE: Place additional 'users' setting entries here, if desired.

cray_local_users.settings.users.data.name.bob: null
cray_local_users.settings.users.data.bob.uid: '0'
cray_local_users.settings.users.data.bob.group: employee
#cray_local_users.settings.users.data.bob.crypt:
cray_local_users.settings.users.data.bob.other_groups: []
cray_local_users.settings.users.data.bob.description: Bob Name
cray_local_users.settings.users.data.bob.shell: /bin/bash
cray_local_users.settings.users.data.bob.home: /cray_home/bob
cray_local_users.settings.users.data.bob.system: true
cray_local_users.settings.users.data.bob.deleted: false
cray_local_users.settings.users.data.bob.domain_groups:
- all_nodes
```

IMPORTANT: The old encrypted passwords can be copied from CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0. They are in the default sharedroot /etc/shadow file.

Use this translation table to find the relevant CLE 5.2 / SMW 7.2 data to be migrated to these configuration settings, continuing the example for Employee group and Bob Name begun above.

Table 45. Translation Table: Variables beginning with cray_local_users.settings.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
groups.data.name.employee	null	basic	N/A	N/A
groups.data.employee.gid	'12345'	basic	N/A	files: default sharedroot /etc/group
groups.data.employee.description	Employee group	basic	N/A	files: default sharedroot /etc/group
groups.data.employee.deleted	false	advanced	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
groups.data. employee.domain_groups	- all_nodes	advanced	N/A	N/A
users.data.name.bob	null	basic	N/A	N/A
users.data.bob.uid	'52121'	basic	N/A	files: default sharedroot /etc/passwd
users.data.bob.group	employee	basic	N/A	files: default sharedroot /etc/passwd
users.data.bob.crypt		basic	N/A	files: default sharedroot /etc/shadow
users.data.bob.other_groups	[]	basic	N/A	files: default sharedroot /etc/group
users.data.bob.description	Bob Name	basic	N/A	files: default sharedroot /etc/passwd
users.data.bob.shell	/bin/bash	basic	N/A	files: default sharedroot /etc/passwd
users.data.bob.home	/cray_home/bob	basic	N/A	files: default sharedroot /etc/passwd
users.data.bob.system	true	basic	N/A	N/A
users.data.bob.deleted	false	advanced	N/A	N/A
users.data.bob.domain_groups	- all_nodes	basic	N/A	N/A

6.4.2.20 Update cray_logging Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

This procedure configures the inherit and enable settings in the Cray Logging service configuration worksheet. This service has a global template as well as a CLE template, and therefore it can be configured to inherit settings from the global config set.

Procedure

1. Edit `cray_logging_worksheet.yaml`.

```
smw# vi cray_logging_worksheet.yaml
```

2. Uncomment `cray_logging.inherit` and set it to `true`.

This means that logging settings in the global config set will be used instead of logging settings in the CLE config set. If `cray_logging.inherit` is set to `false`, then other settings may need to be changed.

6.4.2.21 Update `cray_login` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Login service provides information and listings for login nodes, which are used by users to access the Cray system. Also, the "nologin" feature is configured in this service. This procedure configures some basic settings in the Cray Login service configuration worksheet to add site-specific data. The last step of this procedure provides a translation table for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_login_worksheet.yaml`.
2. Uncomment `cray_login.enabled` and set it to `true`.
3. Enter the node group (or groups) of the login nodes on this system.

Cray has provided a pre-populated node group called "login_nodes" to contain the login nodes (by cname) for the system. If that node group has not yet been customized for this site, see [Update `cray_node_groups` Worksheet](#) on page 236.

Uncomment `cray_login.settings.login_nodes.data.member_groups`, remove the empty list (`[]`), and add that node group (and any other node groups, as needed) on a separate line prefixed by a hyphen and space (`-`).

```
cray_login.settings.login_nodes.data.member_groups:  
- login_nodes
```

4. Uncomment `cray_login.settings.login_nodes.data.login_prohibited_after_boot` and do one of the following:
 - Set it to `false` to have the `/etc/nologin` file removed automatically on each node in the list of login node groups (set in step 3) as it completes its boot.

- Set it to `true` to require a system administrator to remove `/etc/nologin` on each node by running a command like the following after all of the CLE nodes have been booted and the system is ready for users to log in. This command could be added to the boot automation file.

```
sdb# pcmd -r -n ALL_SERVICE "rm /etc/nologin"
```

Check the boot automation file on the CLE 5.2 / SMW 7.2 system for an entry like this, which removes the `/etc/nologin` file after all nodes have booted:

```
lappend actions { crms_exec_on_bootnode "root" "xtunspec -r /rr/current -d /etc/nologin" }
```

In CLE 6.0 / SMW 8.0, this is the equivalent entry for the boot automation file as long as it appears after the compute nodes have all booted:

```
lappend actions { crms_exec_via_bootnode "sdb" "root" "pcmd -r -n ALL_SERVICE 'rm /etc/nologin'" }
```

Table 46. Translation Table: Variables beginning with `cray_login.settings.login_nodes.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>member_groups</code>	<code>[]</code>	basic	N/A	N/A
<code>login_prohibited_after_boot</code>	<code>false</code>	basic	N/A	<i>files:</i> SMW boot automation file

6.4.2.22 Update `cray_lustre_client` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray Lustre Client configuration service is used to configure Lustre clients on an XC system. This procedure configures some basic settings in the `cray_lustre_client` configuration worksheet to add site-specific data. This procedure also provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_lustre_client_worksheet.yaml`.

```
smw# vi cray_lustre_client_worksheet.yaml
```

2. Uncomment `cray_lustre_client.enabled` and do one of the following:

- Set it to `false` for systems that are NOT a Lustre client of either an external Lustre server or direct-attached Lustre (DAL). Skip the rest of the procedure.
- Set it to `true` for systems that are a Lustre client of either an external Lustre server or DAL. Proceed to the next step.

3. Migrate settings for Lustre module parameters.

These general settings apply to all Lustre file systems. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Table 47. Translation Table: Variables beginning with `cray_lustre_client.settings.module_params.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>libcfs_panic_on_lbug</code>	<code>true</code>	advanced	probe (1)	N/A
<code>ptlrpc_at_min</code>	40	advanced	N/A	N/A
<code>ptlrpc_at_max</code>	400	advanced	N/A	N/A
<code>ptlrpc_ldlm_enqueue_min</code>	260	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) <code>smw# grep libcfs_panic_on_lbug /opt/xt-images/templates/default/etc/modprobe.conf</code>				

4. Configure a client mount for each Lustre file system that will be mounted.

Repeat this step for each client mount.

In the worksheet, copy the lines below `# ** EXAMPLE 'client_mounts' VALUE` (with current defaults) `**` and paste them below `# NOTE: Place additional 'client_mounts' setting entries here, if desired.`

```
# ** EXAMPLE 'client_mounts' VALUE (with current defaults) **
# cray_lustre_client.settings.client_mounts.data.fs_name.sample_key_a: null <-- setting a multival
key
# cray_lustre_client.settings.client_mounts.data.sample_key_a.lustre_fs_name: ''
# cray_lustre_client.settings.client_mounts.data.sample_key_a.mount_point: ''
# cray_lustre_client.settings.client_mounts.data.sample_key_a.mgs_lnet_nids: []
# cray_lustre_client.settings.client_mounts.data.sample_key_a.mount_options: rw,flock,lazystatfs
# cray_lustre_client.settings.client_mounts.data.sample_key_a.mount_at_boot: true
# cray_lustre_client.settings.client_mounts.data.sample_key_a.client_groups:
# - login_nodes
# - compute_nodes
# - elogin_nodes
#
```

Uncomment the lines, replace `sample_key_a` with a string that identifies that mount (`snx11023` in the example below), then remove the `<-- setting a multival key` text at the end of the first line in each set (note that the `null` value is required; do not remove or change it). Finally, modify the values as needed for this site. For a migration, use the translation tables provided in the examples below.

Example of mounting an external Lustre file system.

This example uses a Sonexion with a file system called `snx11023` that is mounted on `/lus/snx11023` by three node groups: login nodes, compute nodes, and eLogin nodes. All of them mount the file system as the node is booting.

```
# NOTE: Place additional 'client_mount' setting entries here, if desired.
cray_lustre_client.settings.client_mounts.data.fs_name.snx11023: null
cray_lustre_client.settings.client_mounts.data.snx11023.lustre_fs_name: snx11023
cray_lustre_client.settings.client_mounts.data.snx11023.mount_point: /lus/snx11023
cray_lustre_client.settings.client_mounts.data.snx11023.mgs_lnet_nids:
- 10.149.4.3@o2ib
- 10.149.4.4@o2ib
cray_lustre_client.settings.client_mounts.data.snx11023.mount_options: rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.snx11023.mount_at_boot: true
cray_lustre_client.settings.client_mounts.data.snx11023.client_groups:
- login_nodes
- compute_nodes
- elogin_nodes

#***** END Service Setting: client_mounts *****
```

Use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. This table continues the external Lustre file system example with snx11023.

The table indicates that values for many of these variables can be found in /opt/xt-images/templates/default/etc/fstab on the SMW. Here is an example of a relevant entry in that file:

```
smw# grep snx11023 /opt/xt-images/templates/default/etc/fstab
10.149.4.3@o2ib:10.149.4.4@o2ib:/snx11023 /lus/snx11023 lustre
rw,flock,lazystatfs
```

Table 48. Translation Table: Variables beginning with cray_lustre_client.settings.client_mounts.data.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
fs_name.snx11023	null	basic	N/A	N/A
snx11023.lustre_fs_name	snx11023	basic	probe (1)	files: SMW /opt/xt-images/templates/default/etc/fstab
snx11023.mount_point	/lus/snx11023	basic	probe (1)	files: SMW /opt/xt-images/templates/default/etc/fstab
snx11023.mgs_lnet_nids	- 10.149.4.3@o2ib - 10.149.4.4@o2ib	basic	probe (1)	files: SMW /opt/xt-images/templates/default/etc/fstab
snx11023.mount_options	rw,flock,lazystatfs	basic	probe (1)	files: SMW /opt/xt-images/templates/default/etc/fstab
snx11023.mount_at_boot	true	basic	N/A	N/A
snx11023.client_groups	- login_nodes - compute_nodes - elogin_nodes	basic	N/A	N/A

Commands for probing the CLE 5.2 / SMW 7.2 system:

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
(1) login# mount grep lustre				

Example of mounting an internal Lustre file system (DAL).

The following example shows values for two DAL client mounts: one for login nodes (first set of lines) and one for compute nodes (second set of lines).

The Lustre file system is started via `lustre_control` in the boot automation file after all service nodes have booted. The boot automation file then has a step to mount the Lustre file system on any service nodes that need to mount it. So those service nodes cannot have "mount_at_boot" set to true. However, the compute nodes are booted after the DAL file system has been started, so they can have "mount_at_boot" set to true. This example uses a DAL server with a file system called `dal` that is mounted on `/lus/dal` by three node groups: login nodes, eLogin nodes, and compute nodes. Only the compute nodes mount the file system as the node is booting (`mount_at_boot=true`).

```
# NOTE: Place additional 'client_mount' setting entries here, if desired.
cray_lustre_client.settings.client_mounts.data.fs_name.dal_login: null
cray_lustre_client.settings.client_mounts.data.dal_login.lustre_fs_name: dal
cray_lustre_client.settings.client_mounts.data.dal_login.mount_point: /lus/dal
cray_lustre_client.settings.client_mounts.data.dal_login.mgs_lnet_nids:
- 27@gni
- 29@gni
cray_lustre_client.settings.client_mounts.data.dal_login.mount_options: rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.dal_login.mount_at_boot: false
cray_lustre_client.settings.client_mounts.data.dal_login.client_groups:
- login_nodes
- elogin_nodes

cray_lustre_client.settings.client_mounts.data.fs_name.dal_compute: null
cray_lustre_client.settings.client_mounts.data.dal_compute.lustre_fs_name: dal
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_point: /lus/dal
cray_lustre_client.settings.client_mounts.data.dal_compute.mgs_lnet_nids:
- 27@gni
- 29@gni
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_options: rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_at_boot: true
cray_lustre_client.settings.client_mounts.data.dal_compute.client_groups:
- compute_nodes

#***** END Service Setting: client_mounts *****
```

This translation table continues the internal Lustre file system example with DAL. The table indicates that values for many of these variables can be found in `/opt/xt-images/templates/default/etc/fstab` on the SMW. Here is an example of a relevant entry in that file:

```
smw# grep dal /opt/xt-images/templates/default/etc/fstab
27@gni:29@gni:/dal /lus/dal rw,flock,user_xattr 0 0
```

Table 49. Translation Table: Variables beginning with `cray_lustre_client.settings.client_mounts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
fs_name.dal_login	null	basic	N/A	N/A
dal_login.lustre_fs_name	dal	basic	<i>probe (1)</i>	files: SMW

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/opt/xt-images/templates/default/etc/fstab
dal_login.mount_point	/lus/dal	basic	<i>probe (1)</i>	files: SMW /opt/xt-images/templates/default/etc/fstab
dal_login.mgs_inet_nids	- 27@gni - 29@gni	basic	<i>probe (1)</i>	files: SMW /opt/xt-images/templates/default/etc/fstab
dal_login.mount_options	rw,flock,lazystatfs	basic	<i>probe (1)</i>	files: SMW /opt/xt-images/templates/default/etc/fstab
dal_login.mount_at_boot	false	basic	N/A	N/A
dal_login.client_groups	- login_nodes - elogin_nodes	basic	N/A	N/A
fs_name.dal_login	null	basic	N/A	N/A
dal_login.lustre_fs_name	dal	basic	<i>probe (1)</i>	files: SMW /opt/xt-images/templates/default/etc/fstab
dal_login.mount_point	/lus/dal	basic	<i>probe (1)</i>	files: SMW /opt/xt-images/templates/default/etc/fstab
dal_login.mgs_inet_nids	- 27@gni - 29@gni	basic	<i>probe (1)</i>	files: SMW /opt/xt-images/templates/default/etc/fstab
dal_login.mount_options	rw,flock,lazystatfs	basic	<i>probe (1)</i>	files: SMW /opt/xt-images/templates/default/etc/fstab
dal_login.mount_at_boot	true	basic	N/A	N/A
dal_login.client_groups	- compute_nodes	basic	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) login# mount grep lustre				

5. Verify that the node groups referenced in step 4 on page 195 have been accurately defined for this site.

To verify, edit `cray_node_groups_worksheet.yaml` and search for these node groups:

```
login_nodes
compute_nodes
elogin_nodes
```

DAL servers also need to be configured. See [Update cray_lustre_server Worksheet](#) on page 199. Further configuration of DAL occurs later in the installation process.

6.4.2.23 Update cray_lustre_server Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray Lustre server configuration service should be enabled and configured only if this system uses direct-attached Lustre (DAL). It enables configuration of Lustre-server-related kernel module parameters. This procedure configures some basic settings in the `cray_lustre_server` configuration worksheet to add site-specific data.

Procedure

1. Edit `cray_lustre_server_worksheet.yaml`.

```
smw# vi cray_lustre_server_worksheet.yaml
```

2. Uncomment `cray_lustre_server.enabled` and do one of the following:

- Set it to `false` for system that do not use DAL (direct-attached Lustre). Skip the remaining steps.
- Set it to `true` for systems that do use DAL. Proceed to the next step.

3. (Only for systems with DAL) Enter the node group that contains the Lustre Management Server (MGS) node on this system.

To see which node group contains the MGS node (by `cname`) or to create such a node group for this system (`MGS_NODE_GROUP` in this example), edit `cray_node_groups_worksheet.yaml`. For a migration, try looking in `CLEinstall.conf` (`direct_attached_lustre`) for the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

Uncomment `cray_lustre_server.settings.lustre_servers.data.mgs_group`, remove the empty list (`[]`), and add that node group on a separate line prefixed by a hyphen and space (`-`).

```
cray_lustre_server.settings.lustre_servers.data.mgs_group:
- MGS_NODE_GROUP
```

4. (Only for systems with DAL) Enter the node group(s) that contain the Lustre MetaData Server (MDS) nodes on this system.

To see which node group(s) contain the MDS nodes (by cname) or to create that node group(s) for this system (`MDS_NODE_GROUP_1` and `MDS_NODE_GROUP_2` in this example), edit `cray_node_groups_worksheet.yaml`. For a migration, try looking in `CLEinstall.conf` (`direct_attached_lustre`) for the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

Uncomment `cray_lustre_server.settings.lustre_servers.data.mds_groups`, remove the empty list (`[]`), and add the node group(s) on a separate line prefixed by a hyphen and space (`-`).

```
cray_lustre_server.settings.lustre_servers.data.mds_groups:
- MDS_NODE_GROUP_1
- MDS_NODE_GROUP_2
```

5. (Only for systems with DAL) Enter the node group(s) that contain the Lustre Object Storage Server (OSS) nodes on this system.

To see which node group(s) contain the OSS nodes (by cname) or to create that node group(s) for this system (`OSS_NODE_GROUP_1` and `OSS_NODE_GROUP_2` in this example), edit `cray_node_groups_worksheet.yaml`. For a migration, try looking in `CLEinstall.conf` (`direct_attached_lustre`) for the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

Uncomment `cray_lustre_server.settings.lustre_servers.data.oss_groups`, remove the empty list (`[]`), and add the node group(s) on a separate line prefixed by a hyphen and space (`-`).

```
cray_lustre_server.settings.lustre_servers.data.oss_groups:
- OSS_NODE_GROUP_1
- OSS_NODE_GROUP_2
```

6. (Only for systems with DAL) Set Lustre kernel module parameters, as needed.

This worksheet contains additional settings that tune the Lustre kernel modules. Seek advice from the site Lustre server administrator before changing them.

6.4.2.24 Update `cray_multipath` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray multipath service provides a means to support redundant paths to a device for failover or performance reasons. Multipath does NOT need to be fully cabled to be used. The multipath driver can handle using one path or many.

NOTE: (SMW HA only) Cray recommends configuring multipath before configuring and enabling HA. If HA is configured and enabled first, then additional precautions must be taken when enabling multipath, as documented in *XC™ Series SMW HA Installation Guide*.

The multipath configuration service has a global template as well as a CLE template, and therefore the service can be configured to inherit settings from the global config set or use settings from the CLE config set(s), if there is a need to have it configured differently in different config sets. If multipath configuration is desired on the management node (SMW) as well as CLE nodes, Cray recommends enabling this service in the global config set

and configuring it there for both SMW and CLE nodes. The multipath service in the CLE config sets would then inherit the global configuration data.

This procedure configures the inherit setting and possibly some other settings in the Cray multipath service configuration worksheet in a CLE config set.

Procedure

1. Edit `cray_multipath_worksheet.yaml`.
2. Uncomment `cray_multipath.inherit` and set it to one of the following values:
 - Set it to `true` to manage multipath settings in the global config set instead of in the CLE config set. If this option is chosen, skip the rest of the steps.
 - Set it to `false` to manage multipath settings in one or more CLE config sets instead of in the global config set. If this option is chosen, continue to the next step.
3. (If `inherit` set to `false`) Uncomment `cray_multipath.enabled`.
Set it to `true` if this site desires to use multipath, otherwise set it to `false`. If enabling this service, continue to the next step.
4. (If `enabled` set to `true`) Complete the configuration of multipath.
 - a. Enter the list of multipath nodes.

Uncomment `cray_multipath.settings.multipath.data.node_list`, remove the `[]` (denotes empty list), and add a list of nodes (by cname or host ID) in this system that have multipath devices and need to have multipath configured. For sites with boot node failover and/or SDB node failover, Cray recommends adding both the active and passive (failover) nodes to this list.

This example shows a list of three nodes: an SMW with host ID `1eac4e0c`, a boot node with cname `c0-0c0s4n1`, and an SDB node with cname `c0-0c0s3n1`.

```
cray_multipath.settings.multipath.data.node_list:
- 1eac4e0c
- c0-0c0s4n1
- c0-0c0s3n1
```

- b. Configure enabled devices.

Cray has provided a number of enabled devices with pre-populated data under `# ** 'enabled_devices' DATA **`. These storage devices are the devices that will be whitelisted, which means they will be listed as exceptions to the blacklist. The settings for these devices have default values provided by the device vendors and do not need to be changed. If this site intends to configure a multipath device that does not appear in this group of enabled devices, contact a Cray representative for help.

- c. (Optional) Configure aliases for the multipath devices.

This is the equivalent of adding aliases to the multipaths section of the `multipath.conf` file.

In the worksheet, copy the two lines below `# ** EXAMPLE 'aliases' VALUE` (with current defaults) `**` and paste them below `# NOTE: Place additional 'aliases' setting entries here, if desired.`

```
# ** EXAMPLE 'aliases' VALUE (with current defaults) **
#   cray_multipath.settings.aliases.data.wwid.sample_key_a: null <-- setting a multival key
```

```
#   cray_multipath.settings.aliases.data.sample_key_a.alias: ''
#
```

Uncomment the lines, replace `sample_key_a` with the World Wide Identifier (WWID) of the device to be aliased (60080e50002e203c00002a085551b2c8 in this example) in all lines, and remove the `<--` setting a multival key text at the end of the first line (note that the null value is required; do not remove or change it). Finally, add the alias for this device (`smw_node_pv1` in this example). Repeat this substep for each device, as needed.

```
# NOTE: Place additional 'aliases' setting entries here, if desired.
cray_multipath.settings.aliases.data.wwid.60080e50002e203c00002a085551b2c8: null
cray_multipath.settings.aliases.data.60080e50002e203c00002a085551b2c8.alias: smw_node_pv1
#***** END Service Setting: aliases *****
```

- d. Correct the default values for three pre-populated device settings.

The default values of the following variables are incorrect in `cray_multipath_worksheet.yaml` for this release (they are correct in the table below). In the worksheet, find these variables and change their values as indicated.

```
enabled_devices.data.DDN_SFA12K_20.product: SFA12K-20
enabled_devices.data.DDN_SFA12K_40.product: SFA12K-40|SFA12KX*
enabled_devices.data.DDN_EF3015.path_grouping_policy: group_by_prio
```

6.4.2.25 Update `cray_munge` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

Cray MUNGE is an authentication service that creates and validates credentials. It is required by the DataWarp service (`cray_dws`), Slurm (a workload manager), and Dynamic RDMA Credentials (`cray_drc`). This procedure enables/disables the `cray_munge` service.

Procedure

1. Edit `cray_munge_worksheet.yaml`.

```
smw# vi cray_munge_worksheet.yaml
```

2. Uncomment `cray_munge.enabled` and do one of the following:

- Set it to `true` only if this site wishes to enable the DataWarp service or Slurm while doing a fresh install of SMW/CLE software or if this site is using Dynamic RDMA Credentials.

- Set it to `false` otherwise.

If the MUNGE service was disabled in this step, it can be enabled later when configuring DataWarp or Slurm.

6.4.2.26 Update `cray_net` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray Networking configuration service defines all network information for CLE nodes, which is necessary for a functional system. This procedure configures some basic settings in the `cray_net` configuration worksheet to add site-specific data. This procedure also provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

REMEMBER: For partitioned systems, each partition generally has its own config set and associated configuration worksheets. Follow this procedure to make changes to the `cray_net_worksheet.yaml` for each partition. Some steps call out what settings should be different for different partitions.

There are two major sections to `cray_net`. One describes the networks to which the CLE nodes are connected, and the other describes the hosts and the network interfaces on each host that are on those networks.

Networks. All networks for CLE nodes must be defined here. The high speed network (HSN) will be connected to the `ipogif0` interface on each CLE node. The login network will be used by the internal login (or network gateway) nodes to an external-to-XC network. Any additional number of networks can be added or described using unique network names, such as for an InfiniBand network or a 40GigEthernet network.

Hosts and network interfaces. Host entries in `cray_net` are used to describe specific information about a host that has network interfaces or to make a host name alias in `/etc/hosts`. Every CLE node does not need to be listed here because the IP address, `nid` name, and `cname` entry in the `/etc/hosts` file will be generated based on the address range of the HSN.

Notes on editing a configuration worksheet:

- To enter a value for a string that currently is set to `' '` (empty string), replace the quotes with the new value. For example, `ipv4_network: ' '` becomes `ipv4_network: 10.1.0.0`. In cases where the string value might be interpreted as a number, retain the single quotes. For example, a string setting with value `'512'` needs quotes.
- To enter one or more values for a list that is currently set to `[]` (empty list), remove the brackets and add each entry on a separate line, preceded by a dash and a space (`-`).
- Do NOT change or remove the null value in lines like this that appear at the beginning of each set of network, host, or host interface definitions. This line sets the key, or identifier, for that definition. In this example, `"hsn"` is the identifier for the HSN network definition.

```
cray_net.settings.networks.data.name.hsn: null
```

Procedure

1. Edit `cray_net_worksheet.yaml`.

```
smw# vi cray_net_worksheet.yaml
```

2. Uncomment `cray_net.enabled` and ensure that it is set to `true`.
3. Uncomment these two settings for the HSN (high speed network).

If this is a partitioned system, then enter different values for these settings. Partitions p1 and p2 will not have the same `ipv4` network, but will have similar `ipv4_netmask` (though different from the full machine).

```
# ** 'networks' DATA **
cray_net.settings.networks.data.hsn.ipv4_network: 10.128.0.0
cray_net.settings.networks.data.hsn.ipv4_netmask: 255.252.0.0
```

For a migration, use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Table 50. Translation Table: Variables beginning with `cray_net.settings.networks.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>name.hsn</code>	null	required	N/A	N/A
<code>hsn.description</code>	The Cray high-speed network	basic	N/A	N/A
<code>hsn.ipv4_network</code>	10.128.0.0	required	<i>probe (1)</i>	<i>installer: CLEinstall.conf (HSN_byte1 HSN_byte2)</i>
<code>hsn.ipv4_netmask</code>	255.252.0.0	required	<i>probe (1)</i>	<i>installer: CLEinstall.conf (bootimage_bootifnetmask)</i>
<code>hsn.ipv4_broadcast</code>		advanced	N/A	N/A
<code>hsn.ipv4_gateway</code>		basic	N/A	N/A
<code>hsn.dns_servers</code>	[]	basic	N/A	N/A
<code>hsn.dns_search</code>	[]	basic	N/A	N/A
<code>hsn.ntp_servers</code>	[]	basic	N/A	N/A
<code>hsn.fw_external</code>	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) <code>boot# ifconfig ipogif0</code>				

DEFINE NETWORKS -----

IMPORTANT:

- Add values for the `dns_servers` and `dns_search` settings to the login network only, not to any other network.

- DO NOT add a value for the `ntp_servers` setting for any network used for CLE nodes, because CLE nodes must source their time/NTP settings from the SMW rather than try to contact NTP servers on the login network.

4. Configure a login network and add the information for the "Customer network" to which the login nodes connect.

Scroll down to the pre-populated network settings below the `# ** 'networks' DATA **` line and find the login network definition. Uncomment the commented lines and modify the values as needed for this site's internal systems. Note that in the first line, the `null` value is required; do not remove or change it.

NOTE: If this site does not use DNS search but does use DNS domain in `/etc/resolv.conf`, then adding a single entry to the `dns_search` setting is functionally equivalent to setting the DNS domain.

```
# ** 'networks' DATA **
...
cray_net.settings.networks.data.name.login: null
cray_net.settings.networks.data.login.description: Customer network
cray_net.settings.networks.data.login.ipv4_network: 172.30.48.0
cray_net.settings.networks.data.login.ipv4_netmask: 255.255.240.0
cray_net.settings.networks.data.login.ipv4_broadcast: ''
cray_net.settings.networks.data.login.ipv4_gateway: 172.30.48.1
cray_net.settings.networks.data.login.dns_servers:
- 172.30.84.40
- 172.31.84.40
- 172.28.84.40
cray_net.settings.networks.data.login.dns_search:
- us.cray.com
- americas.cray.com
- cray.com
cray_net.settings.networks.data.login.ntp_servers: []
cray_net.settings.networks.data.login.fw_external: false
```

IMPORTANT: If the login network should be treated as an external network for the firewall, then set `cray_net.settings.networks.data.login.fw_external` (the last line in the example) to `true`.

For a migration, use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings.

Table 51. Translation Table: Variables beginning with `cray_net.settings.networks.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>name.login</code>	<code>null</code>	required	N/A	N/A
<code>login.description</code>	Customer network	basic	N/A	N/A
<code>login.ipv4_network</code>		required	<i>probe (1)</i>	N/A
<code>login.ipv4_netmask</code>		required	<i>probe (1)</i>	<code>files: login node sharedroot</code> <code>/etc/sysconfig/network/ifcfg-eth0</code>
<code>login.ipv4_broadcast</code>		advanced	<i>probe (1)</i>	N/A
<code>login.ipv4_gateway</code>		basic	N/A	<code>files: login class sharedroot</code>

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/sysconfig/network/routes
login.dns_servers	[]	basic	N/A	files: login class sharedroot /etc/resolv.conf (nameserver) /etc/sysconfig/network/config (NETCONFIG_DNS_STATIC_SERVERS)
login.dns_search	[]	basic	N/A	files: login node sharedroot /etc/resolv.conf search /etc/sysconfig/network/config (NETCONFIG_DNS_STATIC_SEARCHLIST)
login.ntp_servers	[]	basic	N/A	files: login node /etc/ntp.conf (servers) (see "About migrating NTP servers on the login network" below)
login.fw_external	false	advanced	N/A	files: login node /etc/sysconfig/SuSEfirewall2 (FW_DEV_EXT)
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) login# ifconfig eth0				

About migrating NTP servers on the login network. Cray discourages sites from allowing CLE nodes to source time/NTP settings from NTP servers on the login network instead of from the SMW. However, if this site has a CLE 5.2 / SMW 7.2 system with the login node class of the sharedroot using NTP servers outside of the SMW/CLE system, and this site wishes to migrate that arrangement to CLE 6.0 / SMW 8.0, then Cray recommends the following:

1. Define one network for the internal login nodes to use that does not have any NTP servers (`ntp_servers: []`).
 2. Define a different network for the external login nodes to use that does have NTP servers (the list of `ntp_servers` is not empty).
5. Configure additional networks, as needed for this system.

In the worksheet, copy the ten lines below `# ** EXAMPLE 'networks' VALUE` (with current defaults) `**` and paste one set for each network below the line `# NOTE: Place additional 'networks' setting entries here, if desired.`

```
# ** EXAMPLE 'networks' VALUE (with current defaults) **
# cray_net.settings.networks.data.name.sample_key_a: null <-- setting a multival key
# cray_net.settings.networks.data.sample_key_a.description: ''
# cray_net.settings.networks.data.sample_key_a.ipv4_network: ''
# cray_net.settings.networks.data.sample_key_a.ipv4_netmask: ''
# cray_net.settings.networks.data.sample_key_a.ipv4_broadcast: ''
# cray_net.settings.networks.data.sample_key_a.ipv4_gateway: ''
```

```
# cray_net.settings.networks.data.sample_key_a.dns_servers: []
# cray_net.settings.networks.data.sample_key_a.dns_search: []
# cray_net.settings.networks.data.sample_key_a.ntp_servers: []
# cray_net.settings.networks.data.sample_key_a.fw_external: false

# ** 'networks' FIELD SPECIFICATION -- MULTIVAL KEY FIELD **
```

Uncomment the lines, replace `sample_key_a` with an identifier for the network (lnet and ethernet40gig in the example below) in all lines, and remove the `<-- setting a multival key text at the end of the first line` (note that the null value is required; do not remove or change it). Finally, modify the values as needed for this site.

The following example shows two additional networks. The first is a single Infiniband network (lnet) used for the LNet router nodes. Sites that have more than one InfiniBand network will need to add more networks like this first one. The second network has been defined for nodes that have 40GigEthernet interfaces (ethernet40gig), and for such networks, the `fw_external` variable must be set to true.

```
# NOTE: Place additional 'networks' setting entries here, if desired.
cray_net.settings.networks.data.name.lnet: null
cray_net.settings.networks.data.lnet.description: The InfiniBand network for
LNet router nodes to external Lustre server
cray_net.settings.networks.data.lnet.ipv4_network: 10.150.0.0
cray_net.settings.networks.data.lnet.ipv4_netmask: 255.255.0.0
cray_net.settings.networks.data.lnet.ipv4_broadcast: ''
cray_net.settings.networks.data.lnet.ipv4_gateway: ''
cray_net.settings.networks.data.lnet.dns_servers: []
cray_net.settings.networks.data.lnet.dns_search: []
cray_net.settings.networks.data.lnet.ntp_servers: []
cray_net.settings.networks.data.lnet.fw_external: false

cray_net.settings.networks.data.name.ethernet40gig: null
cray_net.settings.networks.data.ethernet40gig.description:
Network for 40GigEthernet
cray_net.settings.networks.data.ethernet40gig.ipv4_network: 138.55.19.0
cray_net.settings.networks.data.ethernet40gig.ipv4_netmask: 255.255.255.0
cray_net.settings.networks.data.ethernet40gig.ipv4_broadcast: ''
cray_net.settings.networks.data.ethernet40gig.ipv4_gateway: ''
cray_net.settings.networks.data.ethernet40gig.dns_servers: []
cray_net.settings.networks.data.ethernet40gig.dns_search: []
cray_net.settings.networks.data.ethernet40gig.ntp_servers: []
cray_net.settings.networks.data.ethernet40gig.fw_external: true
#***** END Service Setting: networks *****
```

For a migration, use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. This table continues the example: an Infiniband network (lnet) and a network for nodes with 40GigEthernet interfaces.

Table 52. Translation Table: Variables beginning with `cray_net.settings.networks.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
name.lnet	null	required	N/A	N/A
lnet.description	The InfiniBand network for LNet router nodes to	basic	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
	external Lustre server			
Inet.ipv4_network		required	<i>probe (1)</i>	N/A
Inet.ipv4_netmask		required	<i>probe (1)</i>	<i>files:</i> Inet node sharedroot /etc/sysconfig/network/ifcfg-ib0
Inet.ipv4_broadcast		advanced	N/A	N/A
Inet.ipv4_gateway		basic	N/A	N/A
Inet.dns_servers	[]	basic	N/A	N/A
Inet.dns_search	[]	basic	N/A	N/A
Inet.ntp_servers	[]	basic	N/A	N/A
Inet.fw_external	false	advanced	N/A	N/A
name.ethernet40gig	null	required	N/A	N/A
ethernet40gig.description	Network for 40Gig Ethernet	basic	N/A	N/A
ethernet40gig.ipv4_network		required	<i>probe (2)</i>	N/A
ethernet40gig.ipv4_netmask		required	<i>probe (2)</i>	<i>files:</i> 40gignetwork node sharedroot /etc/sysconfig/network/ifcfg-eth0
ethernet40gig.ipv4_broadcast		advanced	<i>probe (2)</i>	N/A
ethernet40gig.ipv4_gateway		basic	N/A	<i>files:</i> 40gignetwork class sharedroot /etc/sysconfig/network/routes
ethernet40gig.dns_servers	[]	basic	N/A	<i>files:</i> 40gignetwork class sharedroot /etc/resolv.conf (nameserver) /etc/sysconfig/network/config (NETCONFIG_DNS_STATIC_SERVERS)
ethernet40gig.dns_search	[]	basic	N/A	<i>files:</i> 40gignetwork node sharedroot /etc/resolv.conf (search) /etc/sysconfig/network/config (NETCONFIG_DNS_STATIC_SEARCHLIST)
ethernet40gig.ntp_servers	[]	basic	N/A	<i>files:</i> 40gignetwork node /etc/ntp.conf (servers)
ethernet40gig.fw_external	false	advanced	N/A	<i>files:</i> 40gignetwork node

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/sysconfig/SuSEfirewall2 (FW_DEV_EXT)
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) <code>lnet# ifconfig ib0</code> (2) <code>40gignetwork# ifconfig eth0</code>				

DEFINE HOSTS AND THEIR NETWORK INTERFACES -----

6. Configure a host as the boot node.

Cray has defined a default `bootnode` host, which is located under the # `** 'hosts' DATA **` line. Every system has this host.

IMPORTANT: Never set `cray_net.settings.hosts.data.bootnode.aliases` to "boot" because that is a host name alias that belongs to the virtual IP address for the boot node in support of the boot node failover feature.

a. Configure the host ID of the boot node.

Uncomment `cray_net.settings.hosts.data.bootnode.hostid` and set it to the cname of the boot node.

For a migration, use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

b. Configure the host name of the boot node.

Uncomment `cray_net.settings.hosts.data.bootnode.hostname` and set it to the host name of the boot node.

Do not set the host name to "boot" because that name is reserved for the virtual IP address of the boot node, regardless of whether it is the full system or a partitioned system. Choose a name that includes the machine name and "boot" such as "boot-panda," or if this is a partitioned system, then identify the boot node as "boot-p1," "boot-p2," and so forth.

For a migration, use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting. The boot node in this example is called `boot-panda`; substitute the appropriate name for this site, such as `boot-machinename` or `boot-p1` or `boot-p2`.

Table 53. Translation Table: Variables beginning with `cray_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Values	Level	Probe	Files/Installer
<code>common_name.bootnode</code>	null	required	N/A	N/A
<code>bootnode.description</code>	Boot node for the system	basic	N/A	N/A
<code>bootnode.aliases</code>	[]	basic	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Values	Level	Probe	Files/Installer
bootnode.hostid	c0-0c0s0n1	basic	<i>probe (1)</i>	<i>installer:</i> CLEinstall.conf (node_boot_primary)
bootnode.host_type	admin	basic	N/A	N/A
bootnode.hostname	boot-panda	basic	N/A	<i>files:</i> bootroot /etc/HOSTNAME <i>installer:</i> CLEinstall.conf (node_boot_hostname)
bootnode.standby_node	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) <code>smw# xtcli part_cfg show p0 grep boot</code>				

- c. Configure the primary Ethernet interface of the boot node.

Uncomment

`cray_net.settings.hosts.data.bootnode.interfaces.primary_ethernet.ipv4_addresses` and set it as follows. This is on the "admin" network to the SMW.

- 10.3.1.254 for a full system (p0).
- 10.3.1.254 for p1, 10.3.1.252 for p2, and so forth for partitioned systems.

For a migration, use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this and other settings.

Table 54. Translation Table: Variables beginning with `cray_net.settings.hosts.data.bootnode.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
common_name.primary_ethernet	null	required	N/A	N/A
primary_ethernet.name	eth0	required	N/A	N/A
primary_ethernet.description	Ethernet connecting boot node to the SMW.	basic	N/A	N/A
primary_ethernet.aliases	[]	basic	N/A	N/A
primary_ethernet.network	admin	basic	N/A	N/A
primary_ethernet.ipv4_address	10.3.1.254	basic	<i>probe (1)</i>	<i>files:</i> bootroot /etc/sysconfig/network/ifcfg-eth0

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				<i>installer</i> : CLEinstall.conf (bootnode_eth0_IPAddr)
primary_ethernet.mac		advanced	N/A	N/A
primary_ethernet.startmode	auto	advanced	N/A	<i>files</i> : bootroot /etc/sysconfig/network/ifcfg-eth0
primary_ethernet.bootproto	static	basic	N/A	<i>files</i> : bootroot /etc/sysconfig/network/ifcfg-eth0 <i>installer</i> : CLEinstall.conf (bootnode_eth0_bootproto)
primary_ethernet.mtu		basic	N/A	<i>files</i> : bootroot /etc/sysconfig/network/ifcfg-eth0 <i>installer</i> : CLEinstall.conf (bootnode_eth0_mtu)
primary_ethernet.extra_attributes	[]	basic	N/A	<i>files</i> : bootroot /etc/sysconfig/network/ifcfg-eth0
primary_ethernet.module		advanced	N/A	N/A
primary_ethernet.params		advanced	N/A	N/A
primary_ethernet.unmanaged_interface	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) boot# ifconfig eth0				

- d. Configure the HSN boot alias interface of the boot node.

Uncomment

`cray_net.settings.hosts.data.bootnode.interfaces.hsn_boot_alias.ipv4_address` and set it as follows. This is on the HSN and is the "virtual IP address" for the virtual interface ipogif0:1.

- 10.131.255.254 for a full system (p0).
- The highest address possible for a partition's HSN, for partitioned systems. For example, if p1 HSN `ipv4_address=10.128.0.0` with `ipv4_netmask 255.255.0.0`, then set `ipv4_address=10.128.255.254` for p1. If p2 HSN `ipv4_address=10.129.0.0` with `ipv4_netmask 255.255.0.0`, then set `ipv4_address=10.129.255.254` for p2.

For a migration, use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this and other settings.

Table 55. Translation Table: Variables beginning with `cray_net.settings.hosts.data.bootnode.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.hsn_boot_alias</code>	null	required	N/A	N/A
<code>hsn_boot_alias.name</code>	<code>ipogif0:1</code>	required	N/A	N/A
<code>hsn_boot_alias.description</code>	Well known address used for boot node services.	basic	N/A	N/A
<code>hsn_boot_alias.aliases</code>	[]	basic	N/A	N/A
<code>hsn_boot_alias.network</code>	hsn	basic	N/A	N/A
<code>hsn_boot_alias.ipv4_address</code>	10.131.255.254	basic	<i>probe (1)</i>	<i>installer: CLEinstall.conf (bootnode_failover_IPaddr)</i>
<code>hsn_boot_alias.mac</code>		advanced	N/A	N/A
<code>hsn_boot_alias.startmode</code>	auto	advanced	N/A	N/A
<code>hsn_boot_alias.bootproto</code>	static	basic	N/A	N/A
<code>hsn_boot_alias.mtu</code>		basic	N/A	N/A
<code>hsn_boot_alias.extra_attributes</code>	[]	basic	N/A	N/A
<code>hsn_boot_alias.module</code>		advanced	N/A	N/A
<code>hsn_boot_alias.params</code>		advanced	N/A	N/A
<code>hsn_boot_alias.unmanaged_interface</code>	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) <code>boot# ifconfig ipogif0:1</code>				

7. Configure a host as the SDB node.

Cray has defined a default `sdbnode` host, which is located under the `# ** 'hosts' DATA **` line. Every system has this host.

a. Configure the host ID of the SDB node.

Uncomment `cray_net.settings.hosts.data.sdbnode.hostid` and set it to the cname of the SDB node.

For a migration, use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

b. Configure the host name of the SDB node.

Uncomment `cray_net.settings.hosts.data.sdbnode.hostname` and set it to "sdb."

For a migration, use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

Table 56. Translation Table: Variables beginning with `cray_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.sdbnode</code>	null	required	N/A	N/A
<code>sdbnode.description</code>	SDB node for the system	basic	N/A	N/A
<code>sdbnode.aliases</code>	[]	basic	N/A	N/A
<code>sdbnode.hostid</code>	c0-0c0s1n1	basic	<i>probe (1)</i>	<i>installer: CLEinstall.conf (node_sdb_primary)</i>
<code>sdbnode.host_type:</code>	admin	basic	N/A	N/A
<code>sdbnode.hostname</code>	sdb	basic	N/A	<i>files: sdb node sharedroot /etc/HOSTNAME installer: CLEinstall.conf (node_boot_hostname)</i>
<code>sdbnode.standby_node</code>	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) <code>smw# xtcli part_cfg show p0 grep sdb</code>				

- c. Configure the primary Ethernet interface of the SDB node.

Uncomment

`cray_net.settings.hosts.data.sdbnode.interfaces.primary_ethernet.ipv4_address` and set it as follows. This is on the "admin" network to the SMW.

- 10.3.1.253 for a full system (p0).
- 10.3.1.253 for p1, 10.3.1.251 for p2, and so forth for partitioned systems.

For a migration, use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this and other settings.

Table 57. Translation Table: Variables beginning with `cray_net.settings.hosts.data.sdbnode.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.primary_ethernet</code>	null	required	N/A	N/A
<code>primary_ethernet.name</code>	eth0	required	N/A	N/A
<code>primary_ethernet.description</code>	Ethernet connecting SDB node to the SMW.	basic	N/A	N/A
<code>primary_ethernet.aliases</code>	[]	basic	N/A	N/A
<code>primary_ethernet.network</code>	admin	basic	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
primary_ethernet.ipv4_address	10.3.1.253	basic	<i>probe (1)</i>	<i>files:</i> sdb node sharedroot /etc/sysconfig/network/ifcfg-eth0 <i>installer:</i> CLEinstall.conf (sdbnode_eth0_IPaddr)
primary_ethernet.mac		advanced	N/A	N/A
primary_ethernet.startmode	auto	advanced	N/A	<i>files:</i> sdb node sharedroot /etc/sysconfig/network/ifcfg-eth0
primary_ethernet.bootproto	static	basic	N/A	<i>files:</i> sdb node sharedroot /etc/sysconfig/network/ifcfg-eth0 <i>installer:</i> CLEinstall.conf (sdbnode_eth0_bootproto)
primary_ethernet.mtu		basic	N/A	<i>files:</i> sdb node sharedroot /etc/sysconfig/network/ifcfg-eth0 <i>installer:</i> CLEinstall.conf (sdbnode_eth0_mtu)
primary_ethernet.extra_attributes	[]	basic	N/A	<i>files:</i> sdb node sharedroot /etc/sysconfig/network/ifcfg-eth0
primary_ethernet.module		advanced	N/A	N/A
primary_ethernet.params		advanced	N/A	N/A
primary_ethernet.unmanaged_interface	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) sdb# ifconfig eth0				

- d. Configure the HSN SDB alias interface of the SDB node.

Uncomment

`cray_net.settings.hosts.data.sdbnode.interfaces.hsn_sdb_alias.ipv4_address` and set it as follows. This is on the HSN and is the "virtual IP address" for the virtual interface `ipogif0:1`.

- 10.131.255.253 for a full system (p0).
- The highest address possible for a partition's HSN, for partitioned systems.

For example, if p1 HSN `ipv4_address=10.128.0.0` with `ipv4_netmask 255.255.0.0`, then set `ipv4_address=10.128.255.253` for p1. If p2 HSN `ipv4_address=10.129.0.0` with `ipv4_netmask 255.255.0.0`, then set `ipv4_address=10.129.255.253` for p2.

For a migration, use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this and other settings.

Table 58. Translation Table: Variables beginning with `cray_net.settings.hosts.data.sdbnode.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.hsn_sdb_alias</code>	null	required	N/A	N/A
<code>hsn_sdb_alias.name</code>	<code>ipogif0:1</code>	required	N/A	N/A
<code>hsn_sdb_alias.description</code>	Well known address used for SDB node services.	basic	N/A	N/A
<code>hsn_sdb_alias.aliases</code>	[]	basic	N/A	N/A
<code>hsn_sdb_alias.network</code>	hsn	basic	N/A	N/A
<code>hsn_sdb_alias.ipv4_address</code>	10.131.255.253	basic	<i>probe (1)</i>	<i>installer: CLEinstall.conf (sdbnode_failover_IPaddr)</i>
<code>hsn_sdb_alias.mac</code>		advanced	N/A	N/A
<code>hsn_sdb_alias.startmode</code>	auto	advanced	N/A	N/A
<code>hsn_sdb_alias.bootproto</code>	static	basic	N/A	N/A
<code>hsn_sdb_alias.mtu</code>		basic	N/A	N/A
<code>hsn_sdb_alias.extra_attributes</code>	[]	advanced	N/A	N/A
<code>hsn_sdb_alias.module</code>		advanced	N/A	N/A
<code>hsn_sdb_alias.params</code>		advanced	N/A	N/A
<code>hsn_sdb_alias.unmanaged_interface</code>	false	N/A	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) <code>sdb# ifconfig ipogif0:1</code>				

8. Configure a host as the login node.
 - a. Configure the aliases of the login node.

Uncomment `cray_net.settings.hosts.data.login_node.aliases` and set it to a list of aliases, as follows.

- If this site wishes the login node to have a host name alias of "login:"

```
cray_net.settings.hosts.data.login_node.aliases:
- login
```

- If this site has more than one login node, the first one could have aliases of "login" and "login1," and the others would be set to "login2," "login3," and so forth.

```
cray_net.settings.hosts.data.login_node.aliases:
- login
- login1
```

For a migration, use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

- b. Configure the host ID of the login node.

Uncomment `cray_net.settings.hosts.data.login_node.hostid` and set it to the cname of the login node. If this system has more than one login node, set this variable to the first login node.

For a migration, use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

- c. Configure the host name of the login node.

Uncomment `cray_net.settings.hosts.data.login_node.hostname` and set it to the host name.

This could be the machine name, for systems that have only one login node. For example, on a machine known as panda, this would be "panda." For systems with more than one login node, the host name could be "panda1" for the first one, "panda2" for the second one, and so forth.

For a migration, use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting. Every system has this host.

Table 59. Translation Table: Variables beginning with `cray_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.login_node</code>	null	required	N/A	N/A
<code>login_node.description</code>	Login node for the system	basic	N/A	N/A
<code>login_node.aliases</code>	- login	basic	N/A	<i>installer:</i> CLEinstall.conf (node_class[x]=login)
<code>login_node.hostid</code>	"	basic	N/A	<i>installer:</i> CLEinstall.conf (node_loginnode_class[x]=login)
<code>login_node.host_type</code>		basic	N/A	N/A
<code>login_node.hostname</code>	"	basic	N/A	<i>files:</i> login class sharedroot /etc/HOSTNAME <i>installer:</i> CLEinstall.conf (node_class_login_hostname)
<code>login_node.standby_node</code>	false	advanced	N/A	N/A

- d. Configure the login Ethernet interface of the login node.

Uncomment

`cray_net.settings.hosts.data.login_node.interfaces.login_ethernet.ipv4_addresses` and set it to the IP address of the login node's eth0 interface on the "login" network.

For a migration, use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this and other settings.

Table 60. Translation Table: Variables beginning with `cray_net.settings.hosts.data.login_node.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.login_ethernet</code>	null	required	N/A	N/A
<code>login_ethernet.name</code>	eth0	required	N/A	N/A
<code>login_ethernet.description</code>	Ethernet connecting the login node to the customer network.	basic	N/A	N/A
<code>login_ethernet.aliases</code>	[]	basic	N/A	N/A
<code>login_ethernet.network</code>	login	basic	N/A	N/A
<code>login_ethernet.ipv4_address</code>	"	basic	<i>probe (1)</i>	<i>files:</i> login node sharedroot /etc/sysconfig/network/ifcfg-eth0
<code>login_ethernet.mac</code>		advanced	N/A	N/A
<code>login_ethernet.startmode</code>	auto	advanced	N/A	<i>files:</i> login node sharedroot /etc/sysconfig/network/ifcfg-eth0
<code>login_ethernet.bootproto</code>	static	basic	N/A	<i>files:</i> login node sharedroot /etc/sysconfig/network/ifcfg-eth0
<code>login_ethernet.mtu</code>		basic	N/A	<i>files:</i> login node sharedroot /etc/sysconfig/network/ifcfg-eth0
<code>login_ethernet.extra_attributes</code>	[]	basic	N/A	<i>files:</i> login node sharedroot /etc/sysconfig/network/ifcfg-eth0
<code>login_ethernet.module</code>		advanced	N/A	N/A
<code>login_ethernet.params</code>		advanced	N/A	N/A
<code>login_ethernet.unmanaged_interface</code>	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
(1) login# <code>ifconfig eth0</code>				

9. Configure additional hosts, as needed.

If this system has additional service nodes that need to have host name or host name alias or network interface settings, then for each one add a host definition stanza like the following, placing it under `NOTE`: Place additional 'hosts' setting entries here, if desired. The first example shows the host configuration of a DVS node (`dvs_node`) with the host name set to "dvs1," a host name alias of "dvs," and one Ethernet interface connected to the "login" network. For a migration from a CLE 5.2 / SMW 7.2 system, there are additional examples in the translation tables that follow these worksheet examples.

```
cray_net.settings.hosts.data.common_name.dvs_node: null
cray_net.settings.hosts.data.dvs_node.description: DVS node
cray_net.settings.hosts.data.dvs_node.aliases:
- dvs
cray_net.settings.hosts.data.dvs_node.hostid: c0-0c0s0n2
cray_net.settings.hosts.data.dvs_node.host_type: ''
cray_net.settings.hosts.data.dvs_node.hostname: dvs1
cray_net.settings.hosts.data.dvs_node.standby_node: false

cray_net.settings.hosts.data.dvs_node.interfaces.common_name.eth0: null
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.name: eth0
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.description: Ethernet
    connecting the DVS node to the customer network.
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.aliases: []
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.network: login
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.ipv4_address: 172.30.50.128
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.mac: ''
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.startmode: auto
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.bootproto: static
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.mtu: ''
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.extra_attributes: []
#cray_net.settings.hosts.data.dvs_node.interfaces.eth0.module: ''
#cray_net.settings.hosts.data.dvs_node.interfaces.eth0.params: ''
#cray_net.settings.hosts.data.dvs_node.interfaces.eth0.unmanaged_interface: false
```

The second example shows the host configuration for an LNet router node (`clfs_lnet_1`) that has two different InfiniBand interfaces (`ib0` and `ib2`) to connect to two different networks.

NOTICE: In this example, the interface parameter `mtu` for both interfaces is set to a numerical value within single quotes. The quotes are important. The configurator expects a string for this setting, and without the single quotes, it could interpret this value as a number and return an error. The values provided for other parameters of type string do not need single quotes because they would not be interpreted as anything other than strings.

```
cray_net.settings.hosts.data.common_name.clfs_lnet_1: null
cray_net.settings.hosts.data.clfs_lnet_1.description: CLFS router 1 node
cray_net.settings.hosts.data.clfs_lnet_1.aliases: []
cray_net.settings.hosts.data.clfs_lnet_1.hostid: c0-0c1s0n1
cray_net.settings.hosts.data.clfs_lnet_1.host_type: ''
cray_net.settings.hosts.data.clfs_lnet_1.hostname: lnet1
cray_net.settings.hosts.data.clfs_lnet_1.standby_node: false

cray_net.settings.hosts.data.clfs_lnet_1.interfaces.common_name.ib0: null
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.name: ib0
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.description: InfiniBand
    ib0 connecting the CLFS router 1 node to the lnet network.
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.aliases: []
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.network: lnet
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.ipv4_address: 10.150.10.65
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.mac: ''
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.startmode: auto
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.bootproto: static
```

```

cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.mtu: '65520'
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.extra_attributes:
- IPOIB_MODE='connected'
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.module: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.params: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.unmanaged_interface: false

cray_net.settings.hosts.data.clfs_lnet_1.interfaces.common_name.ib2: null
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.name: ib2
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.description: InfiniBand
ib2 connecting the CLFS router 1 node to the lnet1 network.
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.aliases: []
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.network: lnet1
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.ipv4_address: 10.151.10.65
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.mac: ''
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.startmode: auto
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.bootproto: static
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.mtu: '65520'
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.extra_attributes:
- IPOIB_MODE='connected'
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.module: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.params: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.unmanaged_interface: false

```

For migration from a CLE 5.2 / SMW 7.2 system, here are several additional examples of hosts and host interfaces that are commonly defined. Examples include:

- an RSIP host (rsip_node) with one Ethernet interface (primary_ethernet)
- a DVS host (dvs_node) with two Ethernet interfaces (eth0 and eth1)
- an LNet host (lnet_node1) with one InfiniBand interface (ib0)
- an LNet host (lnet_node2) with two InfiniBand interfaces (ib0 and ib2)
- a host (gig40node) with one 40GigEthernet interface (eth0), which needs an additional kernel module loaded
- two hosts (dal_mds and dal_oss) with no network interfaces, which is a way to get a host name alias into the /etc/hosts file

These first two translation tables show an example of an RSIP node with one Ethernet interface.

Table 61. Example Translation Table: Variables beginning with cray_net.settings.hosts.data.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
common_name.rsip_node	null	required	N/A	N/A
rsip_node.description	RSIP server node	basic	N/A	N/A
rsip_node.aliases		basic	N/A	N/A
rsip_node.hostid	c0-0c0s0n0	basic	N/A	installer: CLEinstall.conf (rsip_servicenode_clients)
rsip_node.host_type		basic	N/A	N/A
rsip_node.hostname	rsip1	basic	N/A	N/A
rsip_node.standby_node	false	advanced	N/A	N/A

Table 62. Example Translation Table: Variables beginning with `cray_net.settings.hosts.data.rsip_node.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
<code>common_name.primary_ethernet</code>	null	required	N/A	N/A
<code>primary_ethernet.name</code>	eth0	required	N/A	N/A
<code>primary_ethernet.description</code>	RSIP interface	basic	N/A	N/A
<code>primary_ethernet.aliases</code>	- panda-rsip	basic	N/A	N/A
<code>primary_ethernet.network</code>	login	basic	N/A	N/A
<code>primary_ethernet.ipv4_address</code>	172.30.12.106	basic	N/A	N/A
<code>primary_ethernet.mac</code>		advanced	N/A	N/A
<code>primary_ethernet.startmode</code>		advanced	N/A	N/A
<code>primary_ethernet.bootproto</code>	static	basic	N/A	N/A
<code>primary_ethernet.mtu</code>		basic	N/A	N/A
<code>primary_ethernet.extra_attributes</code>	□	basic	N/A	N/A
<code>primary_ethernet.module</code>		advanced	N/A	N/A
<code>primary_ethernet.params</code>		advanced	N/A	N/A
<code>primary_ethernet.unmanaged_interface</code>	false	advanced	N/A	N/A

These two translation tables show an example of a DVS node with two Ethernet interfaces.

Table 63. Example Translation Table: Variables beginning with `cray_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
<code>common_name.dvs_node</code>	null	required	N/A	N/A
<code>dvs_node.description</code>	DVS node to CSS	basic	N/A	N/A
<code>dvs_node.aliases</code>	- dvs1	basic	N/A	N/A
<code>dvs_node.hostid</code>	c0-0c0s0n3	basic	N/A	<i>installer:</i> CLEinstall.conf (node_class[x]=dvs)
<code>dvs_node.host_type</code>		basic	N/A	N/A
<code>dvs_node.hostname</code>	dvs	basic	N/A	N/A
<code>dvs_node.standby_node</code>	false	basic	N/A	N/A

Table 64. Example Translation Table: Variables beginning with `cray_net.settings.hosts.data.dvs_node.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
<code>common_name.eth0</code>	null	required	N/A	N/A
<code>eth0.name</code>	eth0	required	N/A	N/A
<code>eth0.description</code>	DVS network to CSS	basic	N/A	N/A
<code>eth0.aliases</code>	- panda-dvs	basic	N/A	N/A
<code>eth0.network</code>	dvs-nfsnet	basic	N/A	N/A
<code>eth0.ipv4_address</code>	172.30.12.107	basic	N/A	N/A
<code>eth0.mac</code>		advanced	N/A	N/A
<code>eth0.startmode</code>		advanced	N/A	N/A
<code>eth0.bootproto</code>	static	basic	N/A	N/A
<code>eth0.mtu</code>		basic	N/A	N/A
<code>eth0.extra_attributes</code>	[]	basic	N/A	N/A
<code>eth0.module</code>		advanced	N/A	N/A
<code>eth0.params</code>		advanced	N/A	N/A
<code>eth0.unmanaged_interface</code>	false	advanced	N/A	N/A
<code>common_name.eth1</code>	null	required	N/A	N/A
<code>eth1.name</code>	eth1	required	N/A	N/A
<code>eth1.description</code>	DVS network to GPFS	basic	N/A	N/A
<code>eth1.aliases</code>	- panda-dvs-gpfs	basic	N/A	N/A
<code>eth1.network</code>	dvs-gpfsnet	basic	N/A	N/A
<code>eth1.ipv4_address</code>	172.30.12.107	basic	N/A	N/A
<code>eth1.mac</code>		advanced	N/A	N/A
<code>eth1.startmode</code>		advanced	N/A	N/A
<code>eth1.bootproto</code>	static	basic	N/A	N/A
<code>eth1.mtu</code>		basic	N/A	N/A
<code>eth1.extra_attributes</code>	[]	advanced	N/A	N/A
<code>eth1.module</code>		advanced	N/A	N/A
<code>eth1.params</code>		advanced	N/A	N/A
<code>eth1.unmanaged_interface</code>	false	advanced	N/A	N/A

These two translation tables show an example of an LNet node with one InfiniBand interface.

Table 65. Example Translation Table: Variables beginning with `cray_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
<code>common_name.lnet_node1</code>	null	required	N/A	N/A
<code>lnet_node1.description</code>	LNet router	basic	N/A	N/A
<code>lnet_node1.aliases</code>	- lnet	basic	N/A	N/A
<code>lnet_node1.hostid</code>	c0-0c0s8n3	basic	N/A	<i>installer:</i> CLEinstall.conf (node_class[x] lnet)
<code>lnet_node1.host_type</code>		basic	N/A	N/A
<code>lnet_node1.hostname</code>	lnet1	basic	N/A	N/A
<code>lnet_node1.standby_node</code>	false	advanced	N/A	N/A

Table 66. Example Translation Table: Variables beginning with `cray_net.settings.hosts.data.lnet_node1.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
<code>common_name.ib0</code>	null	required	N/A	N/A
<code>ib0.name</code>	ib0	required	N/A	N/A
<code>ib0.description</code>	InfiniBand to external Lustre	basic	N/A	N/A
<code>ib0.aliases</code>	- panda-lnet1	basic	N/A	N/A
<code>ib0.network</code>	lnet	basic	N/A	N/A
<code>ib0.ipv4_address</code>	10.150.6.45	basic	N/A	N/A
<code>ib0.mac</code>		advanced	N/A	N/A
<code>ib0.startmode</code>	auto	advanced	N/A	N/A
<code>ib0.bootproto</code>	static	basic	N/A	N/A
<code>ib0.mtu</code>	'65520'	basic	N/A	N/A
<code>ib0.extra_attributes</code>	- IPOIB_MODE='connected'	basic	N/A	N/A
<code>ib0.module</code>		advanced	N/A	N/A
<code>ib0.params</code>		advanced	N/A	N/A
<code>ib0.unmanaged_interface</code>	false	advanced	N/A	N/A

These two translation tables show an example of an LNet node with two InfiniBand interfaces.

Table 67. Example Translation Table: Variables beginning with `cray_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
<code>common_name.lnet_node2</code>	null	required	N/A	N/A
<code>lnet_node2.description</code>	LNet router with 2 InfiniBand interfaces	basic	N/A	N/A
<code>lnet_node2.aliases</code>	- lnet	basic	N/A	N/A
<code>lnet_node2.hostid</code>	c0-0c0s8n3	basic	N/A	<i>installer:</i> CLEinstall.conf (node_class[x] lnet)
<code>lnet_node2.host_type</code>		basic	N/A	N/A
<code>lnet_node2.hostname</code>	lnet2	basic	N/A	N/A
<code>lnet_node2.standby_node</code>	false	advanced	N/A	N/A

Table 68. Example Translation Table: Variables beginning with `cray_net.settings.hosts.data.lnet_node2.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
<code>common_name.ib0</code>	null	required	N/A	N/A
<code>ib0.name</code>	ib0	required	N/A	N/A
<code>ib0.description</code>	InfiniBand to external Lustre odd network	basic	N/A	N/A
<code>ib0.aliases</code>	- panda-lnet2-odd	basic	N/A	N/A
<code>ib0.network</code>	lnetodd	basic	N/A	N/A
<code>ib0.ipv4_address</code>	10.151.7.56	basic	N/A	N/A
<code>ib0.mac</code>		advanced	N/A	N/A
<code>ib0.startmode</code>	auto	advanced	N/A	N/A
<code>ib0.bootproto</code>	static	basic	N/A	N/A
<code>ib0.mtu</code>	'65520'	basic	N/A	N/A
<code>ib0.extra_attributes</code>	- IPOIB_MODE='connected'	basic	N/A	N/A
<code>ib0.module</code>		advanced	N/A	N/A
<code>ib0.params</code>		advanced	N/A	N/A
<code>ib0.unmanaged_interface</code>	false	advanced	N/A	N/A
<code>common_name.ib2</code>	null	required	N/A	N/A
<code>ib2.name</code>	ib2	required	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
ib2.description	InfiniBand to external Lustre even network	basic	N/A	N/A
ib2.aliases	- panda-lnet2-even	basic	N/A	N/A
ib2.network	lneteven	basic	N/A	N/A
ib2.ipv4_address	10.149.5.34	basic	N/A	N/A
ib2.mac		advanced	N/A	N/A
ib2.startmode	auto	advanced	N/A	N/A
ib2.bootproto	static	basic	N/A	N/A
ib2.mtu	'65520'	basic	N/A	N/A
ib2.extra_attributes	- IPOIB_MODE='connected'	basic	N/A	N/A
ib2.module		advanced	N/A	N/A
ib2.params		advanced	N/A	N/A
ib2.unmanaged_interface	false	advanced	N/A	N/A

These two translation tables show an example of a node (gig40node) with one 40GigEthernet interface, which needs an additional kernel module loaded. Sites that use special network cards (e.g., Mellanox ConnectX-3) must specify which kernel module is used by those cards. See step 12 on page 233 for instructions on how to do this.

Table 69. Example Translation Table: Variables beginning with `cray_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
common_name.gig40node	null	required	N/A	N/A
gig40node.description	Node with 40 Gig Ethernet	basic	N/A	N/A
gig40node.aliases		basic	N/A	N/A
gig40node.hostid	c0-0c0s0n3	basic	N/A	N/A
gig40node.host_type		basic	N/A	N/A
gig40node.hostname	gig40	basic	N/A	N/A
gig40node.standby_node	false	advanced	N/A	N/A

Table 70. Example Translation Table: Variables beginning with `cray_net.settings.hosts.data.gig40node.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
<code>common_name.eth0</code>	null	required	N/A	N/A
<code>eth0.name</code>	eth0	required	N/A	N/A
<code>eth0.description</code>	40 Gig network	basic	N/A	N/A
<code>eth0.aliases</code>	- panda-gig40	basic	N/A	N/A
<code>eth0.network</code>	ethernet40gig	basic	N/A	N/A
<code>eth0.ipv4_address</code>	172.40.4.10	basic	N/A	N/A
<code>eth0.mac</code>		advanced	N/A	N/A
<code>eth0.startmode</code>		advanced	N/A	N/A
<code>eth0.bootproto</code>	static	basic	N/A	N/A
<code>eth0.mtu</code>		basic	N/A	N/A
<code>eth0.extra_attributes</code>	□	basic	N/A	N/A
<code>eth0.module</code>	mlx4_en	advanced	N/A	N/A
<code>eth0.params</code>		advanced	N/A	N/A
<code>eth0.unmanaged_interface</code>	false	basic	N/A	N/A

This last (for this step) table shows an example of two hosts (`dal_mds` and `dal_oss`) with no network interfaces, which is a way to get a host name alias into the `/etc/hosts` file.

Table 71. Example Translation Table: Variables beginning with `cray_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
<code>common_name.dal_mds</code>	null	required	N/A	N/A
<code>dal_mds.description</code>	DAL MDS node	basic	N/A	N/A
<code>dal_mds.aliases</code>	- dal-mds - dal1 - mds1	basic	N/A	N/A
<code>dal_mds.hostid</code>	c0-0c0s0n2	basic	N/A	N/A
<code>dal_mds.host_type</code>		basic	N/A	N/A
<code>dal_mds.hostname</code>		basic	N/A	N/A
<code>dal_mds.standby_node</code>	false	advanced	N/A	N/A
<code>common_name.dal_oss</code>	null	required	N/A	N/A
<code>dal_oss.description</code>	DAL OSS node	basic	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Example Value	Level	Probe	Files/Installer
dal_oss.aliases	- dal-oss - dal2 - oss1	basic	N/A	N/A
dal_oss.hostid	c0-0c0s1n2	basic	N/A	N/A
dal_oss.host_type		basic	N/A	N/A
dal_oss.hostname		basic	N/A	N/A
dal_oss.standby_node	false	advanced	N/A	N/A

10. Configure a host as the second boot node for boot node failover.

If using the boot node failover feature, then define a backup boot node host with the "standby_node" variable set to true.

Use this command to find out if boot node failover has been enabled on the CLE 5.2 / SMW 7.2 system. If it shows two nodes, then boot node failover has been enabled, and the second node listed is the backup boot node.

```
smw# xtcli part_cfg show p0 | grep boot
[boot]: c0-0c0s0n1:ready,c0-0c0s4n1:standby
```

To migrate configuration values from the CLE 5.2 / SMW 7.2 system, use the translation tables that follow this worksheet example.

NOTE: The host name for the primary and backup boot node should both be set to "boot." The aliases can be different so that the /etc/hosts entry for the cname has the host name alias.

```
cray_net.settings.hosts.data.common_name.backup_bootnode: null
cray_net.settings.hosts.data.backup_bootnode.description: backup Boot node for the system
cray_net.settings.hosts.data.backup_bootnode.aliases:
- cray-boot2
cray_net.settings.hosts.data.backup_bootnode.hostid: c0-0c0s4n1
cray_net.settings.hosts.data.backup_bootnode.host_type: admin
cray_net.settings.hosts.data.backup_bootnode.hostname: boot
cray_net.settings.hosts.data.backup_bootnode.standby_node: true

cray_net.settings.hosts.data.backup_bootnode.interfaces.common_name.hsn_boot_alias: null
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.name: ipogif0:1
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.description: Well known
address used for boot node services.
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.aliases: []
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.network: hsn
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.ipv4_address: 10.131.255.254
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.mac: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.startmode: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.bootproto: static
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.mtu: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.extra_attributes: []
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.module: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.params: ''
#cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.unmanaged_interface: false

cray_net.settings.hosts.data.backup_bootnode.interfaces.common_name.primary_ethernet: null
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.name: eth0
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.description: Ethernet
connecting boot node to the SMW.
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.aliases: []
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.network: admin
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.ipv4_address: 10.3.1.254
```

```

cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.mac: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.startmode: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.bootproto: static
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.mtu: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.extra_attributes: []
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.module: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.params: ''
#cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.unmanaged_interface: false

```

For a migration, use these backup boot node translation tables for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. This first table is for the host definition.

Table 72. Translation Table: Variables beginning with `cray_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.backup_bootnode</code>	null	required	N/A	N/A
<code>backup_bootnode.description</code>	Boot Node Failover	basic	N/A	N/A
<code>backup_bootnode.aliases</code>	[]	basic	N/A	N/A
<code>backup_bootnode.hostid</code>	c0-0c0s4n1	basic	<i>probe (1)</i>	<i>installer:</i> CLEinstall.conf (node_boot_alternate)
<code>backup_bootnode.host_type</code>	admin	basic	N/A	N/A
<code>backup_bootnode.hostname</code>	boot	basic	N/A	<i>files:</i> bootroot /etc/HOSTNAME <i>installer:</i> CLEinstall.conf (node_boot_hostname)
<code>backup_bootnode.standby_node</code>	true	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) <code>smw# xtcli part_cfg show p0 grep boot</code>				

This second backup boot node translation table is for the primary Ethernet interface.

Table 73. Translation Table: Variables beginning with `cray_net.settings.hosts.data.backup_bootnode.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.primary_ethernet</code>	null	required	N/A	N/A
<code>primary_ethernet.name</code>	eth0	required	N/A	N/A
<code>primary_ethernet.description</code>	Ethernet connecting failover boot node to the SMW.	basic	N/A	N/A
<code>primary_ethernet.aliases</code>	[]	basic	N/A	N/A
<code>primary_ethernet.network</code>	admin	basic	N/A	N/A
<code>primary_ethernet.ipv4_address</code>	10.3.1.254	basic	<i>probe (1)</i>	<i>files:</i> bootroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/sysconfig/network/ ifcfg-eth0 installer: CLEinstall.conf (bootnode_eth0_IPAddr)
primary_ethernet.mac		advanced	N/A	N/A
primary_ethernet.startmode	auto	advanced	N/A	files: bootroot /etc/sysconfig/network/ ifcfg-eth0
primary_ethernet.bootproto	static	basic	N/A	files: bootroot /etc/sysconfig/network/ ifcfg-eth0 installer: CLEinstall.conf (bootnode_eth0_bootproto)
primary_ethernet.mtu		basic	N/A	files: bootroot /etc/sysconfig/network/ ifcfg-eth0 installer: CLEinstall.conf (bootnode_eth0_mtu)
primary_ethernet.extra_attributes	[]	basic	N/A	files: bootroot /etc/sysconfig/network/ ifcfg-eth0
primary_ethernet.module		advanced	N/A	N/A
primary_ethernet.params		advanced	N/A	N/A
primary_ethernet.unmanaged_interface	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) boot# ifconfig eth0				

This third backup boot node translation table is for the HSN boot alias interface.

Table 74. Translation Table: Variables beginning with `cray_net.settings.hosts.data.backup_bootnode.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
common_name.hsn_boot_alias	null	required	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
hsn_boot_alias.name	ipogif0:1	required	N/A	N/A
hsn_boot_alias.description	Well known address used for boot node services.	basic	N/A	N/A
hsn_boot_alias.aliases	[]	N/A	N/A	N/A
hsn_boot_alias.network	hsn	basic	N/A	N/A
hsn_boot_alias.ipv4_address	10.131.255.254	basic	<i>probe (1)</i>	<i>installer: CLEinstall.conf (bootnode_failover_IPAddr)</i>
hsn_boot_alias.mac		advanced	N/A	N/A
hsn_boot_alias.startmode	auto	advanced	N/A	N/A
hsn_boot_alias.bootproto	static	basic	N/A	N/A
hsn_boot_alias.mtu		basic	N/A	N/A
hsn_boot_alias.extra_attributes	[]	basic	N/A	N/A
hsn_boot_alias.module		advanced	N/A	N/A
hsn_boot_alias.params		advanced	N/A	N/A
hsn_boot_alias.unmanaged_interface	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) boot# ifconfig ipogif0:1				

11. Configure a host as the second SDB node for SDB node failover.

If using the SDB node failover feature, then define a backup SDB node host with the "standby_node" variable set to true.

Use this command to find out if SDB node failover has been enabled on the CLE 5.2 / SMW 7.2 system. If it shows two nodes, then SDB node failover has been enabled, and the second node listed is the backup SDB node.

```
smw# xtcli part_cfg show p0 | grep sdb
[sdb]: c0-0c0s1n2:ready,c0-0c0s5n1:standby
```

To migrate configuration values from the CLE 5.2 / SMW 7.2 system, use the translation tables that follow this worksheet example.

NOTE: The host name for the primary and backup SDB node should both be set to "sdb." The aliases can be different so that the /etc/hosts entry for the cname has the host name alias.

```
cray_net.settings.hosts.data.common_name.backup_sdbnode: null
cray_net.settings.hosts.data.backup_sdbnode.description: backup SDB node for the system
cray_net.settings.hosts.data.backup_sdbnode.aliases:
- cray-sdb2
cray_net.settings.hosts.data.backup_sdbnode.hostid: c0-0c0s3n1
cray_net.settings.hosts.data.backup_sdbnode.host_type: admin
```

```

cray_net.settings.hosts.data.backup_sdbnode.hostname: sdb
cray_net.settings.hosts.data.backup_sdbnode.standby_node: true

cray_net.settings.hosts.data.backup_sdbnode.interfaces.common_name.hsn_boot_alias: null
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.name: ipogif0:1
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.description: Well known
address used for SDB node services.
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.aliases: []
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.network: hsn
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.ipv4_address: 10.131.255.253
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.mac: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.startmode: auto
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.bootproto: static
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.mtu: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.extra_attributes: []
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.module: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.params: ''
#cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.unmanaged_interface: false

cray_net.settings.hosts.data.backup_sdbnode.interfaces.common_name.primary_ethernet: null
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.name: eth0
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.description: Ethernet
connecting SDB node to the SMW.
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.aliases: []
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.network: admin
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.ipv4_address: 10.3.1.253
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.mac: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.startmode: auto
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.bootproto: static
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.mtu: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.extra_attributes: []
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.module: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.params: ''
#cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.unmanaged_interface: false

```

For a migration, use these backup SDB node translation tables for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. This first table is for the backup SDB host definition.

Table 75. Translation Table: Variables beginning with `cray_net.settings.hosts.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.backup_sdbnode</code>	null	required	N/A	N/A
<code>backup_sdbnode.description</code>	backup SDB node for the system	basic	N/A	N/A
<code>backup_sdbnode.aliases</code>	- cray-sdb2	basic	N/A	N/A
<code>backup_sdbnode.hostid</code>	c0-0c0s5n1	basic	<i>probe (1)</i>	<i>installer:</i> CLEinstall.conf (node_sdb_alternate)
<code>backup_sdbnode.host_type</code>	admin	basic	N/A	N/A
<code>backup_sdbnode.hostname</code>	sdb	basic	N/A	<i>files:</i> sdb node sharedroot /etc/HOSTNAME <i>installer:</i> CLEinstall.conf (node_boot_hostname)
<code>backup_sdbnode.standby_node</code>	true	advanced	N/A	N/A

Commands for probing the CLE 5.2 / SMW 7.2 system:

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
(1) <code>smw# xtcli part_cfg show p0 grep sdb</code>				

This second backup SDB node translation table is for the primary Ethernet interface.

Table 76. Translation Table: Variables beginning with `cray_net.settings.hosts.data.backup_sdbnode.interfaces`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>common_name.primary_ethernet</code>	null	required	N/A	N/A
<code>primary_ethernet.name</code>	eth0	required	N/A	N/A
<code>primary_ethernet.description</code>	Ethernet connecting SDB node to the SMW.	basic	N/A	N/A
<code>primary_ethernet.aliases</code>	[]	basic	N/A	N/A
<code>primary_ethernet.network</code>	admin	basic	N/A	N/A
<code>primary_ethernet.ipv4_address</code>	10.3.1.253	basic	<i>probe (1)</i>	<i>files:</i> sdb node sharedroot <code>/etc/sysconfig/network/ifcfg-eth0</code> <i>installer:</i> CLEinstall.conf (sdbnode_eth0_IPaddr)
<code>primary_ethernet.mac</code>		advanced	N/A	N/A
<code>primary_ethernet.startmode</code>	auto	advanced	N/A	<i>files:</i> sdb node sharedroot <code>/etc/sysconfig/network/ifcfg-eth0</code>
<code>primary_ethernet.bootproto</code>	static	basic	N/A	<i>files:</i> sdb node sharedroot <code>/etc/sysconfig/network/ifcfg-eth0</code> <i>installer:</i> CLEinstall.conf (sdbnode_eth0_bootproto)
<code>primary_ethernet.mtu</code>		basic	N/A	<i>files:</i> sdb node sharedroot <code>/etc/sysconfig/network/ifcfg-eth0</code>

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				<i>installer:</i> CLEinstall.conf (sdbnode_eth0_mtu)
primary_ethernet.extra_attributes	[]	basic	N/A	<i>files:</i> sdb node sharedroot /etc/sysconfig/network/ ifcfg-eth0
primary_ethernet.module		advanced	N/A	N/A
primary_ethernet.params		advanced	N/A	N/A
primary_ethernet.unmanaged_interface	false	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) sdb# ifconfig eth0				

This third backup SDB node translation table is for the HSN SDB alias interface.

Table 77. Translation Table: Variables beginning with *cray_net.settings.hosts.data.backup_sdbnode.interfaces*.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
common_name.hsn_sdb_alias	null	required	N/A	N/A
hsn_sdb_alias.name	ipogif0:1	required	N/A	N/A
hsn_sdb_alias.description	Well known address used for SDB node services.	basic	N/A	N/A
hsn_sdb_alias.aliases	[]	basic	N/A	N/A
hsn_sdb_alias.network	hsn	basic	N/A	N/A
hsn_sdb_alias.ipv4_address	10.131.255.253	basic	<i>probe (1)</i>	<i>installer:</i> CLEinstall.conf (sdbnode_failover_IPaddr)
hsn_sdb_alias.mac		advanced	N/A	N/A
hsn_sdb_alias.startmode	auto	advanced	N/A	N/A
hsn_sdb_alias.bootproto	static	basic	N/A	N/A
hsn_sdb_alias.mtu		basic	N/A	N/A
hsn_sdb_alias.extra_attributes	[]	advanced	N/A	N/A
hsn_sdb_alias.module		advanced	N/A	N/A
hsn_sdb_alias.params		advanced	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
hsn_sdb_alias.unmanaged_interface	false	false	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) sdb# ifconfig ipogif0:1				

12. Set the module and params settings for any hosts that are network nodes and use special network cards.

Sites that use special network cards (e.g., Mellanox ConnectX-3) must specify which kernel module is used by those cards. For each host that uses such a card, uncomment (if commented out) the following setting, then replace the empty string with the kernel module name.

This example specifies `m1x4_en`, the module for Mellanox ConnectX-3 cards.

```
cray_net.settings.hosts.data.network_node.interfaces.eth0.module: m1x4_en
```

Any kernel module parameters that need to be set for that module can be specified by uncommenting the following setting, then replacing the empty string with "parameter=value" pairs (pairs separated by spaces). This is not common; no parameters need to be specified for the `m1x4_en` module. Note that the = syntax may vary by kernel module; consult the documentation of the kernel module being used.

```
cray_net.settings.hosts.data.networknode.interfaces.eth0.params: param1=200 param2=30
```

6.4.2.27 About Configuring Netroot Preload

Netroot is a feature that enables nodes booted with a minimal, local in-memory file system to execute within the context of a larger, full-featured root file system. Netroot uses the Data Virtualization Service (DVS) to access the remote root content. While DVS has data and attribute caching features that minimize the impact of most remote references, files that are referenced frequently may still incur an undesirable performance penalty.

The Netroot Preload feature mitigates that performance penalty by copying specified remote files from the Netroot to a node-local in-memory file system early in the node boot process. All future references to those files will be serviced by the local file system rather than requiring remote data and/or metadata DVS operations. This improves system and application performance. However, as a consequence, the amount of memory available on the node is reduced by the cumulative size of all files copied into its memory.

Netroot Preload can be enabled, disabled, and customized using the configurator on the SMW or by editing the configuration worksheets on the SMW.

Netroot Preload Configuration Settings

Netroot Preload configuration consists of defining one or more "loads," or sets of data to be preloaded on specified nodes. The load setting has the following fields:

- label** A convenient, descriptive label for a particular load.
- targets** A list of node groups that reference the nodes that will be preloaded with files on their local file systems. Must provide at least one node group.
- content_lists** Content lists are relative paths to files within the config set. These files contain file paths that are copied into the node-local memory by Netroot Preload. For example, content list `dist/compute-preload.cray` within config set `p0` has these contents:

```
smw# cat p0/dist/compute-preload.cray
/etc/ld.so.cache
/opt/cray/rca/*/bin/rca-helper
/lib64/libc-*.so
/lib64/ld-*.so
/opt/cray/rca/*/lib*/librca.so.*
[...]
```

Pattern matching is supported.

size_limit The memory-consumption limit (in MB) set for this load, which limits how much can be copied to any node. As the files are copied via Netroot, Netroot Preload checks the sizes and amount of data copied so far. When it reaches the size limit, it stops, and any remaining files are not copied. Setting this to zero (0) indicates no limit.

Cray Provides Default Loads

Cray provides two default loads: the 'compute' load, which targets all compute nodes in the system, and the 'login' load, which targets all internal login nodes in the system. The compute load has a single content list specified: `dist/compute-preload.cray`. This file contains paths that are commonly referenced during the node boot and initialization process. Similarly, the login load specifies this content list as the only entry in its `content_lists` setting: `dist/login-preload.cray`. Note that each of these is a relative path. The full path would be `/var/opt/cray/imps/config/sets/p0/dist/login-preload.cray` for the login content list entry. If a site disables or modifies these default settings, the time it takes to boot and initialize nodes may increase.

Sites can Create Custom Loads to Optimize for Specific Workloads

Sites may define their own loads as well. This enables sites to optimize for specific workloads. For targets, sites can use existing node groups or define their own (see [Update cray_node_groups Worksheet](#) on page 236).

To determine which file paths to add to load content lists, use the DVS request log, which is enabled by default. That log was used to create the Cray default content lists. The `/proc/fs/dvs/request_log` file contains a log of all DVS requests that take more than a certain number of seconds to complete (the default is 15 seconds). Look for file paths that are referenced often; these are good candidates for Netroot Preload.

Use the following commands (as root) to view, disable, enable, and clear the DVS request log.

Table 78. Commands to disable, enable, and clear the DVS request log

view	<code>cat /proc/fs/dvs/request_log</code>
disable	<code>echo 0 > /proc/fs/dvs/request_log</code>
enable	<code>echo 1 > /proc/fs/dvs/request_log</code>
clear	<code>echo 2 > /proc/fs/dvs/request_log</code>

See "DVS Can Log Requests Sent to Servers" in *XC™ Series DVS Administration Guide (S-0005)* for additional information about this request log.

The Netroot Preload Log File and a Note about Symlinks

Netroot Preload creates a log file on affected nodes at `/var/opt/cray/log/netroot_preload.log`. This log file contains details on the files preloaded, which, if any, files were not found in the Netroot, and the size of the files preloaded on the node. Any failures will also be logged to the console file on the SMW.

Note that any symlinks included in a load content list may not be copied from Netroot to the node-local RAM file system (i.e., "promoted" in the log file), which might look confusing. For example, suppose a site content list contains `/etc/alternatives/unzip`, which is a symlink to `/usr/bin/unzip-plain`. While both the link and its target are present in Netroot, neither of them appear in the node-local file system, despite the log saying `Promoted '/new_root/merge/etc/alternatives/unzip'`. This is expected and correct behavior. A site that is concerned about possible confusion for administrators can decide to exclude symlinks from content lists or simply list the target of the symlink in a content list to ensure that it is present in the node-local file system.

6.4.2.28 Update `cray_netroot_preload` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

Netroot Preload is a mechanism for populating a Cray system node's root file system early in the boot process to reduce load on the DVS (Data Virtualization Service) servers providing the data and thereby reduce boot times for Netroot nodes. Netroot Preload also improves post-boot performance—how much improvement depends on the workloads. This service is needed if Netroot is used, and does no harm if Netroot is not used.

This procedure configures some settings in the `cray_netroot_preload` configuration worksheet to add site-specific "load" data. Cray provides two default load settings that define target nodes and files to be preloaded to them. Sites may define custom loads as well (optional). For more information, see [About Configuring Netroot Preload](#) on page 233.

Procedure

1. Edit `cray_netroot_preload_worksheet.yaml`.

```
smw# vi cray_netroot_preload_worksheet.yaml
```

2. Uncomment `cray_netroot_preload.enabled`. Keep it set to `true`, which is the default.

Continue to step 3 to define a custom load (optional).

3. Define a custom load.

In the worksheet, copy the four lines below # ** EXAMPLE 'load' VALUE (with current defaults) ** and paste them below # NOTE: Place additional 'load' setting entries here, if desired.

```
# ** EXAMPLE 'load' VALUE (with current defaults) **
#   cray_netroot_preload.settings.load.data.label.sample_key_a: null  <-- setting a multival key
#   cray_netroot_preload.settings.load.data.sample_key_a.targets: []
#   cray_netroot_preload.settings.load.data.sample_key_a.content_lists: []
#   cray_netroot_preload.settings.load.data.sample_key_a.size_limit: 0
```

Uncomment the lines, replace `sample_key_a` with the label for this load (e.g., `my_load`) in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, add site-specific values. Add each list element on a separate line prefixed by a hyphen and space (`-`).

```
# NOTE: Place additional 'load' setting entries here, if desired.
cray_netroot_preload.settings.load.data.label.my_load: null
cray_netroot_preload.settings.load.data.my_load.targets:
- site-defined_node_group
cray_netroot_preload.settings.load.data.my_load.content_lists:
- relative/path/to/oft-requested/files
cray_netroot_preload.settings.load.data.my_load.size_limit: 0
#***** END Service Setting: load *****
```

6.4.2.29 Update `cray_node_groups` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

Node Groups are a mechanism for defining logical groupings of Cray system nodes to streamline node specifications for use in other Cray configuration services. The node groups defined are non-exclusive, that is, a node may belong to more than one node group. They are referenced in other configuration templates and are used in Ansible plays as well. For more information, see [About Node Groups](#) on page 416.

This procedure configures some basic settings in the Cray Node Groups service configuration worksheet to add site-specific data. This procedure also provides a translation table for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_node_groups_worksheet.yaml`.

```
smw# vi cray_node_groups_worksheet.yaml
```

2. Uncomment `cray_node_groups.enabled` and ensure that it is set to `true`.
3. Customize pre-populated node groups.

These pre-populated (default) node groups are provided by Cray, but sites must customize the `members` setting for most of the node groups. For example, the host ID of the SMW is `1eac199c` in the first node group, "smw_nodes," but this must be replaced by the actual host ID for the SMW at this site. For more information about changing these default settings, including the use of additional platform keywords for finer-grained groupings, see [About Node Groups](#) on page 416. For a migration, see also the translation table that follows these example worksheet lines.

```
# ** 'groups' DATA **

cray_node_groups.settings.groups.data.group_name.compute_nodes: null
cray_node_groups.settings.groups.data.compute_nodes.description: Default node
  group which contains all of the compute nodes for the current partition.
cray_node_groups.settings.groups.data.compute_nodes.members:
- platform:compute

cray_node_groups.settings.groups.data.group_name.service_nodes: null
cray_node_groups.settings.groups.data.service_nodes.description: Default node
  group which contains all of the service nodes for the current partition.
cray_node_groups.settings.groups.data.service_nodes.members:
- platform:service

cray_node_groups.settings.groups.data.group_name.smw_nodes: null
cray_node_groups.settings.groups.data.smw_nodes.description: Default node
  group which contains the primary and failover (if applicable) SMW nodes.
cray_node_groups.settings.groups.data.smw_nodes.members:
- 1eac199c

cray_node_groups.settings.groups.data.group_name.boot_nodes: null
cray_node_groups.settings.groups.data.boot_nodes.description: Default node
  group which contains the primary and failover (if applicable) boot
  nodes associated with the current partition.
cray_node_groups.settings.groups.data.boot_nodes.members:
- c0-0c0s0n1

cray_node_groups.settings.groups.data.group_name.sdb_nodes: null
cray_node_groups.settings.groups.data.sdb_nodes.description: Default node
  group which contains the primary and failover (if applicable) SDB
  nodes associated with the current partition.
cray_node_groups.settings.groups.data.sdb_nodes.members:
- c0-0c0s1n1

cray_node_groups.settings.groups.data.group_name.login_nodes: null
cray_node_groups.settings.groups.data.login_nodes.description: Default node
  group which contains the login nodes for the configured system.
cray_node_groups.settings.groups.data.login_nodes.members:
- c0-0c0s2n2

cray_node_groups.settings.groups.data.group_name.all_nodes: null
cray_node_groups.settings.groups.data.all_nodes.description: Default node
  group which contains all of the nodes applicable to the current system.
  May also contain SMW nodes and external login nodes.
cray_node_groups.settings.groups.data.all_nodes.members:
- platform:compute
- platform:service

cray_node_groups.settings.groups.data.group_name.tier2_nodes: null
cray_node_groups.settings.groups.data.tier2_nodes.description: Default node
  group which contains the tier2 nodes in the system. See the guidance in
  the cray_scalable_services service for a detailed description of tier2
  nodes.
```

```
cray_node_groups.settings.groups.data.tier2_nodes.members:
- c0-0c0s8n0
- c0-0c0s15n0
```

To help with selecting nodes to be tier2 servers, here is a tier2 node FAQ:

- Q. How many tier2 nodes are needed?** **A.** At least one server must be provided, and for resiliency, two nodes placed on different blades. The recommended ratio of tier2 nodes (servers) to tier3 nodes (clients) is 1 to 400.
- Q. Will adding more tier2 nodes help performance?** **A.** Adding more tier2 nodes does not always yield additional performance and is subject to diminishing returns.
- Q. What kind of node can be used as a tier2 node?** **A.** Use these:
- OPTIMAL: dedicated repurposed compute nodes (RCN)
 - dedicated service nodes
 - nodes with uniform light to moderate load
 - nodes with relatively homogeneous single core speeds to reduce resource contention disparity during periods of partial availability
- AVOID these (will result in sub-optimal performance):
- nodes with slower individual CPU cores, such as Intel® Xeon Phi™ "Knights Landing" (KNL) processors
 - direct-attached Lustre (DAL) servers
 - RSIP (realm-specific IP) servers
 - login nodes
- Q. Can a tier2 node have more than one role?** **A.** Small test and development systems (TDS) may use tier2 nodes that have additional roles, but generally, it is better for tier2 nodes to be dedicated.
- Q. Where should tier2 nodes be placed?** **A.** Distribute nodes throughout the system (on different blades) for resiliency in the event of hardware failure.

For a migration, use this translation table for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98. This table shows all of the required node groups for a CLE 6.0 / SMW 8.0 system. All of this data must be completed.

If the CLE 5.2 / SMW 7.2 node_classes file has some nodes in the dsl node class, then those nodes may be good candidates to be added to the tier2 node group.

Table 79. Translation Table: Variables beginning with `cray_node_groups.settings.groups.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
group_name.compute_nodes	null	required	N/A	N/A
compute_nodes.description	Default node group which contains all the compute nodes for the current partition.	basic	N/A	N/A
compute_nodes.members	- platform:compute	required	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
group_name.service_nodes	null	required	N/A	N/A
service_nodes.description	Default node group which contains all the service nodes for the current partition.	basic	N/A	N/A
service_nodes.members	- platform:service	required	N/A	N/A
group_name.smw_nodes	null	required	N/A	N/A
smw_nodes.description	Default node group which contains the primary and failover (if applicable) SMW nodes.	basic	N/A	N/A
smw_nodes.members	[]	required	<i>probe (1)</i> <i>probe (2)</i>	N/A
group_name.boot_nodes	null	required	N/A	N/A
boot_nodes.description	Default node group which contains the primary and failover (if applicable) boot nodes associated with the current partition.	basic	<i>probe (3)</i>	<i>installer: CLEinstall.conf</i> (node_boot_primary node_boot_alternate)
boot_nodes.members	[]	required	N/A	N/A
group_name.sdb_nodes	null	required	N/A	N/A
sdb_nodes.description	Default node group which contains the primary and failover (if applicable) SDB nodes associated with the current partition.	basic	<i>probe (3)</i>	<i>installer: CLEinstall.conf</i> (node_sdb_primary node_sdb_alternate)
sdb_nodes.members	[]	required	N/A	N/A
group_name.login_nodes	null	required	N/A	N/A
login_nodes.description	Default node group which contains the login nodes for the configured system.	basic	N/A	N/A
login_nodes.members	[]	required	N/A	<i>files: bootroot</i> <i>/etc/opt/cray/sdb/</i> <i>node_classes (login)</i>

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/hosts (login) <i>installer</i> : CLEinstall.conf (node_login node_class array with "login")
group_name.all_nodes	null	required	N/A	N/A
all_nodes.description	Default node group which contains all of the nodes applicable to the current system. May also contain SMW nodes and external login nodes.	basic	N/A	N/A
all_nodes.members	- platform:compute - platform:service	required	N/A	N/A
group_name.tier2_nodes	null	required	N/A	N/A
tier2_nodes.description	Default node group which contains the tier 2 nodes in the system. See the guidance in the <code>cray_scalable_services</code> service for a detailed description of tier 2 nodes.	basic	N/A	<i>files</i> : bootroot /etc/opt/cray/sdb/ node_classes (dsl) /etc/hosts (dsl) <i>installer</i> : CLEinstall.conf (DSL_nodes)
tier2_nodes.members	[]	required	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) smw1# hostid (2) smw2# hostid (3) smw# xtcli part_cfg show				

4. Define a custom node group, as needed.

Be sure to check the CLE 5.2 / SMW 7.2 shared root in `/etc/opt/cray/sdb/node_classes`, because this site may have defined some node classes other than the ones similar to the pre-populated node groups in the previous step. Some of these site-defined node classes may need to be re-created as site-defined CLE 6.0 / SMW 8.0 node groups.

Repeat this step for each additional node group.

Copy the three commented lines under `** EXAMPLE 'groups' VALUE` (with current defaults) `**` and paste them under `# NOTE: Place additional 'groups' setting entries here, if desired.`

```
** EXAMPLE 'groups' VALUE (with current defaults) **
#cray_node_groups.settings.groups.data.group_name.sample_key_a: null <--setting a multival key
#cray_node_groups.settings.groups.data.sample_key_a.description: ''
#cray_node_groups.settings.groups.data.sample_key_a.members: []
```

Uncomment the lines, replace `sample_key_a` with the identifier chosen for the node group in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, add values for the `description` (a string) and `members` (a list) fields. For the `members` field, add each list element on a separate line prefixed by a hyphen and space (`-`).

As an example, here is the definition of a node group called `lnet_nodes`, which could be the list of LNet router nodes to an external Lustre file system.

```
# NOTE: Place additional 'groups' setting entries here, if desired.
cray_node_groups.settings.groups.data.group_name.lnet_nodes: null
cray_node_groups.settings.groups.data.lnet_nodes.description: Node group that
contains all the LNet router nodes
cray_node_groups.settings.groups.data.lnet_nodes.members:
- c0-0c2s1n1
- c0-2c2s1n2
#***** END Service Setting: groups *****
```

For a migration, this translation table continues the `lnet_nodes` example and indicates which files (and variables within those files) might contain information from CLE 5.2 / SMW 7.2 that this site might wish to migrate to CLE 6.0 / SMW 8.0.

Table 80. Translation Table: Variables beginning with `cray_node_groups.settings.groups.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>group_name.lnet_nodes</code>	<code>null</code>	required	N/A	N/A
<code>lnet_nodes.description</code>	Node group that contains all the LNet router nodes	basic	N/A	<i>files:</i> bootroot <code>/etc/opt/cray/sdb/node_classes (lnet)</code> <i>installer:</i> CLEinstall.conf (node_class array with "lnet")
<code>lnet_nodes.members</code>	<code>- c0-0c2s1n1</code> <code>- c0-2c2s1n2</code>	required	N/A	N/A

Other custom node groups

Other useful custom node groups might be: `rsip_nodes` (for RSIP server nodes), `mom_nodes` (for MOM nodes with a workload manager), `dvs_nodes` (for a node DVS-projecting an external file system to internal nodes), `datawarp_nodes` (for the DataWarp SSD-endowed nodes), or `postproc_nodes` (for MAMU nodes in the former CLE 5.2 / SMW 7.2 `postproc` node_class).

The following table lists all of the CLE configuration services that require node groups for one or more variables. In some cases, a custom node group may need to be defined.

<code>cray_alps</code>	<code>cray_lnet</code>	<code>cray_persistent_data</code>
<code>cray_auth</code>	<code>cray_local_users</code>	<code>cray_rsip</code>

cray_boot	cray_login	cray_scalable_services
cray_dvs	cray_lustre_client	cray_sdb
cray_dws	cray_lustre_server	cray_simple_shares

A custom node group for use with Simple Sync

Node groups can be used in conjunction with Simple Sync to distribute some files to members of a node group. Here is an example of a custom node group called 'automount' that would have an associated 'automount' directory in the Simple Sync directory structure on the SMW (in `/var/opt/cray/imps/config/sets/p0/files/simple_sync/nodegroups`), which could be used to distribute automount maps to the nodes in that node group (for more information, about the Simple Sync directory structure, see [About Simple Sync](#) on page 419 or see `/var/opt/cray/imps/config/sets/p0/files/simple_sync/README`).

```
# NOTE: Place additional 'group' setting entries here, if desired.
cray_node_groups.settings.groups.data.group_name.automount: null
cray_node_groups.settings.groups.data.automount.description: Node group that
contains all the service nodes which will get automount maps via Simple Sync
cray_node_groups.settings.groups.data.automount.members:
- c0-0c1s4n2
#***** END Service Setting: groups *****
```

For a migration, this translation table continues the automount example.

Table 81. Translation Table: Variables beginning with `cray_node_groups.settings.groups.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
group_name.automount_nodes	null	required	N/A	N/A
automount_nodes.description	Node group that contains all the service nodes which will get automount maps via Simple Sync	basic	N/A	N/A
automount_nodes.members	- c0-0c1s4n2	required	N/A	N/A

6.4.2.30 Update `cray_node_health` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray Node Health service configures the Cray Node Health Checker (NHC). The Cray NHC is automatically invoked by ALPS (Application Level Placement Scheduler) upon the termination of an application. ALPS passes a

list of CNL compute nodes associated with the terminated application to NHC. NHC performs specified tests to determine if compute nodes allocated to the application are healthy enough to support running subsequent applications. If not, it removes any compute nodes incapable of running an application from the resource pool.

This procedure enables the `cray_node_health` service. No other settings need to be changed at this point in the process. Cray recommends that sites install and configure CLE with default plugins first, and then return to the Cray Node Health service after the first system boot to configure custom plugins, if needed, using the `custom_plugins` setting. The MIGRATE CONFIGURATION DATA section of this procedure provides translation tables for migrating site-specific plugin configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0. Use that section when reconfiguring this service later in the process.

For information about NHC configuration, see "Configure Node Health Checker Tests" under "Modify an Installed System" in *XC™ Series System Administration Guide (S-2393)*.

Procedure

1. Edit `cray_node_health_worksheet.yaml`.

```
smw# vi cray_node_health_worksheet.yaml
```

2. Uncomment `cray_node_health.enabled` and set it to `true`.

MIGRATE CONFIGURATION DATA -----

Use the translation tables below for help finding the relevant NHC configuration data in the CLE 5.2 / SMW 7.2 system to be migrated to the CLE 6.0 / SMW 8.0 settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

About NHC plugins.

Individual tests may appear multiple times in the configuration, with different variable values. Every time a test is specified, NHC will run that test. This means if the same line is specified five times, NHC will try to run that same test five times. This functionality is mainly used in the case of the Plugin test (allows the administrator to specify as many additional tests as have been written for the site) and the Filesystem test (allows the administrator to specify as many additional file systems as wanted). However, any test can be specified to run any number of times. Different parameters and test actions can be set for each test. For example, this functionality could be used to set up hard limits and soft limits for the Free Memory Check test. Two Free Memory Check tests could be specified in the configuration file; the first test configured to only warn about small amounts of non-free memory, and the second test configured to `admindown` a node that has large amounts of non-free memory.

The CLE 5.2 / SMW 7.2 values of some of the NHC settings can be found in stanzas in the node health configuration file on the SMW, `/opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf`. For example, here is a stanza from that file (begins with "[Memory]" and ends before "[Filesystem]").

```
[Memory]
Action: Log
WarnTime: 20
[Filesystem]
```

That stanza provides values for these CLE 6.0 / SMW 8.0 variables:

```
cray_node_health.settings.memory_plugins.data.Default Memory.action
```

cray_node_health.settings.memory_plugins.data.Default Memory.warntime

The translation table entries for these two variables look like this:

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
Default Memory.action	Log	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Memory] Action)
Default Memory.warntime	20	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Memory] WarnTime)

Other things to note:

- Plugins are enabled in CLE 5.2 / SMW 7.2 when their stanza is uncommented. This example shows a stanza that is commented in SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf and hence disabled. Each of the plugins in CLE 6.0 / SMW 8.0 has an "enabled" setting that serves the same function and should be set according to whether it was commented (disabled) or uncommented (enabled) in CLE 5.2 / SMW 7.2.

```
#####
# Sigcont plugin
#####
# In some cases it is helpful to send a SIGCONT to processes
# of an APID. This plugin will send SIGCONT to the pids in
# /dev/cpuset/CRAY_NHC_APID/tasks
#[Plugin]
#Command: sigcont.sh
#Action: Log
#WarnTime: 240
#Timeout: 300
#RestartDelay: 10
#Sets: Application
```

- The XEON_PHI_PLUGIN was implemented in CLE 5.2 / SMW 7.2 for the Intel Xeon Phi KNC (Knight's Corner), which is not supported hardware for CLE 6.0 / SMW 8.0. The model of Intel Xeon Phi supported with CLE 6.0 / SMW 8.0 is the KNL (Knight's Landing).
- The Accelerator Test plugin from CLE 5.2 / SMW 7.2 had several options to the `gat.sh` command that were valid only for testing KNC (-o, -M, -c). Of these, only the (-M, -c) options are valid in CLE 6.0 / SMW 8.0 for testing the KNL.
- The Apinit/Slurm Log and Core File Recovery plugin is no longer part of NHC. That functionality is being handled by `xtdumpsys` in CLE 6.0 / SMW 8.0.

3. Configure NHC options, as needed.

Table 82. Translation Table: Variables beginning with `cray_node_health.settings.options.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
debug		advanced	N/A	N/A
nhmdebug		advanced	N/A	N/A
advanced_features	'on'	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] advanced_features)
dumpdon	'on'	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] dumpdon)
anyapid	'off'	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] anyapid)
stack_trace	'2'	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] stack_trace)
suspectenable	y	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] suspectenable)
suspectbegin	180	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] suspectbegin)
suspectend	2100	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] suspectend)
recheckfreq	300	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] recheckfreq)
runtests	errors	advanced	N/A	files: SMW

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] runtests)
connecttime	60	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] connecttime)
maxdumps	1	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] maxdumps)
downaction	Log	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] downaction)
downdumps	1	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] downdumps)
alps_recheck_max	10	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] alps_recheck_max)
alps_sync_timeout	1200	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] alps_sync_timeout)
alps_warn_time	120	advanced	N/A	files: SMW / opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] alps_warn_time)
sdb_recheck_max	10	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Options] sdb_recheck_max)
sdb_warn_time	120	advanced	N/A	files: SMW

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Options] sdb_warn_time)
node_no_contact_warn_time	600	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Options] node_no_contact_warn_time)
node_no_contact_action	admindown	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Options] node_no_contact_action)
unhealthy_state	admindown	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Options] unhealthy_state)
prereboot_state	unavail	advanced	N/A	N/A
cache_hosts	'off'	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Options] cache_hosts)

4. Exclude directories (file systems) from testing by the Filesystem test, as needed.

The "Excluding" keyword may appear multiple times in CLE 5.2 / SMW 7.2

SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf. If so, then multiple directories should be added to the list of directories in the CLE 6.0 / SMW 8.0 `cray_node_health.settings.test_options.data.exclude_dirs` setting. Directories that are automounted can also be added to the `cray_node_health.settings.test_options.data.exclude_dirs` list setting. In this way, an administrator can exclude mount points that should not be tested by the Filesystem test. This allows intentionally excluding specific mount points even though they appear in the `fstab` file. This action prevents NHC from setting nodes to `admindown` because of errors on relatively benign file systems.

Explicitly specified mount points cannot be excluded in this fashion; if they should not be checked, then they should not be specified.

Table 83. Translation Table: Variables beginning with `cray_node_health.settings.test_options.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
exclude_dirs	[]	advanced	N/A	files: SMW

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf (Excluding)

5. Configure NHC file system overrides, as needed.

Table 84. Translation Table: Variables beginning with `cray_node_health.settings.filesys_override.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
name.Lustre	null	advanced	N/A	N/A
Lustre.enabled	false	advanced	N/A	N/A
Lustre.warntime	900	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Lustre] WarnTime)
Lustre.timeout	1800	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Lustre] Timeout)
Lustre.restartdelay	60	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Lustre] RestartDelay)
name.DVS	null	advanced	N/A	N/A
DVS.enabled	false	advanced	N/A	N/A
DVS.warntime	60	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([DVS] WarnTime)
DVS.timeout	120	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([DVS] Timeout)
DVS.restartdelay	30	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([DVS] RestartDelay)

6. Configure file system plugins, as needed.

Several [Filesystem] stanzas could be defined for CLE 5.2 / SMW 7.2. If multiple are defined, then extra stanzas will be needed in the `cray_node_health.settings.filesys_plugins` section of the CLE 6.0 / SMW 8.0

configuration worksheet. Here is an example of multiple file systems in CLE 5.2 / SMW 7.2 for the `/scratch` and `/foo` directories.

```
[Filesystem]
Action: Admindown
WarnTime: 600
Timeout: 4800
RestartDelay: 900
Uid: 0
Gid: 0
Path: /scratch
Sets: Application
[Filesystem]
Action: Log
WarnTime: 300
Timeout: 600
RestartDelay: 900
Uid: 25
Gid: 400
Path: /foo
Sets: Application
```

Table 85. Translation Table: Variables beginning with `cray_node_health.settings.filesys_plugins.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
desc.Default Filesystem	null	advanced	N/A	N/A
Default Filesystem.name	Filesystem	advanced	N/A	N/A
Default Filesystem.enabled	true	advanced	N/A	N/A
Default Filesystem.command		advanced	N/A	N/A
Default Filesystem.action	Admindown	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/ etc/opt/cray/nodehealth/ nodehealth.conf ([Filesystem] Action)
Default Filesystem.warntime	60	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/ etc/opt/cray/nodehealth/ nodehealth.conf ([Filesystem] WarnTime)
Default Filesystem.timeout	120	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/ etc/opt/cray/nodehealth/ nodehealth.conf ([Filesystem] Timeout)
Default Filesystem.restartdelay	30	advanced	N/A	<i>files:</i> SMW

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/opt/xt-images/templates/default/ etc/opt/cray/nodehealth/ nodehealth.conf ([Filesystem] RestartDelay)
Default Filesystem.uid	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/ etc/opt/cray/nodehealth/ nodehealth.conf ([Filesystem] Uid)
Default Filesystem.gid	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/ etc/opt/cray/nodehealth/ nodehealth.conf ([Filesystem] Gid)
Default Filesystem.path		advanced	N/A	files: SMW /opt/xt-images/templates/default/ etc/opt/cray/nodehealth/ nodehealth.conf ([Filesystem] Path)
Default Filesystem.subdir		advanced	N/A	N/A
Default Filesystem.excluding		advanced	N/A	N/A
Default Filesystem.sets	Application	advanced	N/A	files: SMW /opt/xt-images/templates/default/ etc/opt/cray/nodehealth/ nodehealth.conf ([Filesystem] Sets)

7. Configure memory plugins, as needed.

Here is an example of a memory stanza in the CLE 5.2 / SMW 7.2 `nodehealth.conf` file.

```
[Memory]
Action: LogWarn
Time: 20
Timeout: 30
RestartDelay: 30
Threshold: 600
Sets: Reservation
```

Table 86. Translation Table: Variables beginning with `cray_node_health.settings.memory_plugins.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
desc.Default Memory	null	advanced	N/A	N/A
Default Memory.name	Memory	advanced	N/A	N/A
Default Memory.enabled	true	advanced	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
Default Memory.command		advanced	N/A	N/A
Default Memory.action	Log	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Memory] Action)
Default Memory.warntime	20	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Memory] WarnTime)
Default Memory.timeout	30	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Memory] Timeout)
Default Memory.restartdelay	30	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Memory] RestartDelay)
Default Memory.threshold	600	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Memory] Threshold)
Default Memory.uid	0	advanced	N/A	N/A
Default Memory.gid	0	advanced	N/A	N/A
Default Memory.sets	Reservation	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([Memory] Sets)

8. Configure plugins, as needed.

If these pre-populated CLE 6.0 / SMW 8.0 plugins were used in the CLE 5.2 / SMW 7.2 system, use this translation table to find the values to migrate to the CLE 6.0 / SMW 8.0 worksheet.

NOTE: For any pre-populated plugins that were commented out in the CLE 5.2 / SMW 7.2 system, set the "enabled" field to false in the CLE 6.0 / SMW 8.0 worksheet.

Table 87. Translation Table: Variables beginning with `cray_node_health.settings.plugins.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
desc.Default Alps	null	advanced	N/A	N/A
Default Alps.name	Alps	advanced	N/A	N/A
Default Alps.enabled	true	advanced	N/A	N/A
Default Alps.command		advanced	N/A	N/A
Default Alps.action	Admindown	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Alps] Action)
Default Alps.warntime	30	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Alps] WarnTime)
Default Alps.timeout	60	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Alps] Timeout)
Default Alps.restartdelay	30	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Alps] RestartDelay)
Default Alps.uid	0	advanced	N/A	N/A
Default Alps.gid	0	advanced	N/A	N/A
Default Alps.sets	Application	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Alps] Sets)
desc.Plugin DVS Requests	null	advanced	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
Plugin DVS Requests.name	Plugin	advanced	N/A	N/A
Plugin DVS Requests.enabled	true	advanced	N/A	N/A
Plugin DVS Requests.command	dvs_requests.sh	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([DVS Client Request Status/ Plugin] Command)
Plugin DVS Requests.action	Log	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([DVS Client Request Status/ Plugin] Action)
Plugin DVS Requests.warntime	10	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([DVS Client Request Status/ Plugin] WarnTime)
Plugin DVS Requests.timeout	20	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([DVS Client Request Status/ Plugin] Timeout)
Plugin DVS Requests.restartdelay	5	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([DVS Client Request Status/ Plugin] RestartDelay)
Plugin DVS Requests.uid	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
				nodehealth/ nodehealth.conf ([DVS Client Request Status/ Plugin] Uid)
Plugin DVS Requests.gid	0	advanced	N/A	files: SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([DVS Client Request Status/ Plugin] Gid)
Plugin DVS Requests.sets	Application	advanced	N/A	files: SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([DVS Client Request Status/ Plugin] Sets)
desc.Plugin Datawarp	null	advanced	N/A	N/A
Plugin Datawarp.name	Plugin	advanced	N/A	N/A
Plugin Datawarp.enabled	true	advanced	N/A	N/A
Plugin Datawarp.command	datawarp.sh -v	advanced	N/A	files: SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([DataWarp mount point removal check/Plugin] Command)
Plugin Datawarp.action	Admindown	advanced	N/A	files: SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([DataWarp mount point removal check/Plugin] Action)
Plugin Datawarp.warntime	30	advanced	N/A	files: SMW /opt/xt-images/templates/ default/etc/opt/cray/

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
				nodehealth/ nodehealth.conf ([DataWarp mount point removal check/Plugin] WarnTime)
Plugin Datawarp.timeout	360	advanced	N/A	files: SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([DataWarp mount point removal check/Plugin] Timeout)
Plugin Datawarp.restartdelay	65	advanced	N/A	files: SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([DataWarp mount point removal check/Plugin] RestartDelay)
Plugin Datawarp.uid	0	advanced	N/A	files: SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([DataWarp mount point removal check/Plugin] Uid)
Plugin Datawarp.gid	0	advanced	N/A	files: SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([DataWarp mount point removal check/Plugin] Gid)
Plugin Datawarp.sets	Reservation	advanced	N/A	files: SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
				nodehealth.conf ([DataWarp mount point removal check/Plugin] Sets)
desc.Default Application	null	advanced	N/A	N/A
Default Application.name	Application	advanced	N/A	N/A
Default Application.enabled	true	advanced	N/A	N/A
Default Application.command		advanced	N/A	N/A
Default Application.action	Admindown	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Application] Action)
Default Application.warntime	240	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Application] WarnTime)
Default Application.timeout	300	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Application] Timeout)
Default Application.restartdelay	1	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Application] RestartDelay)
Default Application.uid	0	advanced	N/A	N/A
Default Application.gid	0	advanced	N/A	N/A
Default Application.sets	Application	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
				nodehealth/ nodehealth.conf ([Application] Sets)
desc.Default Reservation	null	advanced	N/A	N/A
Default Reservation.name	Reservation	advanced	N/A	N/A
Default Reservation.enabled	true	advanced	N/A	N/A
Default Reservation.command		advanced	N/A	N/A
Default Reservation.action	Admindown	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Reservation] Action)
Default Reservation.warntime	240	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Reservation] WarnTime)
Default Reservation.timeout	300	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Reservation] Timeout)
Default Reservation.restartdelay	1	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Reservation] RestartDelay)
Default Reservation.uid	0	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Reservation] Uid)
Default Reservation.gid	0	advanced	N/A	<i>files:</i> SMW

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
				/opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Reservation] Gid)
Default Reservation.sets	Reservation	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Reservation] Sets)
desc.Plugin Nvidia	null	advanced	N/A	N/A
Plugin Nvidia.name	Plugin	advanced	N/A	N/A
Plugin Nvidia.enabled	true	advanced	N/A	N/A
Plugin Nvidia.command	gat.sh -m 10% -r	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Accelerator Test/Plugin] Command)
Plugin Nvidia.action	Admindown	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Accelerator Test/Plugin] Action)
Plugin Nvidia.warntime	50	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Accelerator Test/Plugin] WarnTime)
Plugin Nvidia.timeout	60	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
				([Accelerator Test/Plugin] Timeout)
Plugin Nvidia.restartdelay	30	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Accelerator Test/Plugin] RestartDelay)
Plugin Nvidia.uid	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Accelerator Test/Plugin] Uid)
Plugin Nvidia.gid	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Accelerator Test/Plugin] Gid)
Plugin Nvidia.sets	Application	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Accelerator Test/Plugin] Sets)
desc.Plugin ugni	null	advanced	N/A	N/A
Plugin ugni.name	Plugin	advanced	N/A	N/A
Plugin ugni.enabled	true	advanced	N/A	N/A
Plugin ugni.command	/opt/cray/ugni/default/nhc_plugins/ugni_nhc_plugins	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([ugni Tests/Plugin] Command)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
Plugin ugni.action	Admindown	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([ugni Tests/Plugin] Action)
Plugin ugni.warntime	260	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([ugni Tests/Plugin] WarnTime)
Plugin ugni.timeout	300	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([ugni Tests/Plugin] Timeout)
Plugin ugni.restartdelay	30	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([ugni Tests/Plugin] RestartDelay)
Plugin ugni.uid	0	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([ugni Tests/Plugin] Uid)
Plugin ugni.gid	0	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([ugni Tests/Plugin] Gid)
Plugin ugni.sets	Application	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
				nodehealth/ nodehealth.conf ([ugni Tests/Plugin] Sets)
desc.Plugin Sigcont	null	advanced	N/A	N/A
Plugin Sigcont.name	Plugin	advanced	N/A	N/A
Plugin Sigcont.enabled	false	advanced	N/A	N/A
Plugin Sigcont.command	sigcont.sh	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Sigcont plugin/Plugin] Command)
Plugin Sigcont.action	Log	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Sigcont plugin/Plugin] Action)
Plugin Sigcont.warntime	240	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Sigcont plugin/Plugin] WarnTime)
Plugin Sigcont.timeout	300	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Sigcont plugin/Plugin] Timeout)
Plugin Sigcont.restartdelay	10	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/ default/etc/opt/cray/ nodehealth/ nodehealth.conf ([Sigcont plugin/Plugin] RestartDelay)
Plugin Sigcont.uid	0	advanced	N/A	N/A
Plugin Sigcont.gid	0	advanced	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
Plugin Sigcont.sets	Application	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Sigcont plugin/Plugin] Sets)
desc.Plugin Hugepage Check	null	advanced	N/A	N/A
Plugin Hugepage Check.name	Plugin	advanced	N/A	N/A
Plugin Hugepage Check.enabled	false	advanced	N/A	N/A
Plugin Hugepage Check.command	/opt/cray/nodehealth/default/bin/hugepages_check	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Hugepages Check/Plugin] Command)
Plugin Hugepage Check.action	Reboot	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Hugepages Check/Plugin] Action)
Plugin Hugepage Check.warntime	240	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Hugepages Check/Plugin] WarnTime)
Plugin Hugepage Check.timeout	300	advanced	N/A	<i>files:</i> SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Hugepages Check/Plugin] Timeout)
Plugin Hugepage Check.restartdelay	100	advanced	N/A	<i>files:</i> SMW

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
				/opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Hugepages Check/Plugin] RestartDelay)
Plugin Hugepage Check.uid	0	advanced	N/A	N/A
Plugin Hugepage Check.gid	0	advanced	N/A	N/A
Plugin Hugepage Check.sets	Reservation	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([Hugepages Check/Plugin] Sets)
desc.Plugin CCM Test	null	advanced	N/A	N/A
Plugin CCM Test.name	Plugin	advanced	N/A	N/A
Plugin CCM Test.enabled	true	advanced	N/A	N/A
Plugin CCM Test.command	nhc_ccm_test.sh	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([CCM plugin/Plugin] Command)
Plugin CCM Test.action	Admindown	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([CCM plugin/Plugin] Action)
Plugin CCM Test.warntime	240	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([CCM plugin/Plugin] WarnTime)
Plugin CCM Test.timeout	360	advanced	N/A	files: SMW

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Pre-populated Value	Level	Probe	Files/Installer
				/opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([CCM plugin/Plugin] Timeout)
Plugin CCM Test.restartdelay	30	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([CCM plugin/Plugin] RestartDelay)
Plugin CCM Test.uid	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([CCM plugin/Plugin] Uid)
Plugin CCM Test.gid	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([CCM plugin/Plugin] Gid)
Plugin CCM Test.sets	Reservation	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/opt/cray/nodehealth/nodehealth.conf ([CCM plugin/Plugin] Sets)

Note that the Apinit/Slurm Log and Core File Recovery plugin shown below is no longer part of NHC. That functionality is being handled by `xtdumpsys` in CLE 6.0 / SMW 8.0, so do not try to use this plugin in the new release.

```
#####
# Apinit/Slurm Log and Core File Recovery
#####
# A plugin script to copy apinit and slurm core dump and log files to a login/
# service node.
# The destination directory should be given as the first argument to this
# script.
# This line should not be uncommented until a destination directory has been
```

```

specified.
# i.e. Replace /lus/<archivedir> with an appropriate path.
# The plugin will check for slurm core files at /var/spool/slurmd
# This path could change depending on the SlurmdSpoolDir setting in
# /etc/opt/slurm/slurm.conf
#
# By default, log files will only be copied if a core file is found. To
unconditionally
# copy logs, use the following options after the destination directory:
#
# Copy alps logs unconditionally:
#   Command: apinitarchive.sh /lus/<archivedir> -a
# Copy slurm logs unconditionally:
#   Command: apinitarchive.sh /lus/<archivedir> -s
# Copy alps and slurm logs unconditionally:
#   Command: apinitarchive.sh /lus/<archivedir> -a -s
#
# [Plugin]
# Command: apinitarchive.sh /lus/<archivedir>
# Action: Log
# WarnTime: 0
# Timeout: 10
# RestartDelay: 5
# Uid: 0
# Gid: 0
# Sets: Application

```

9. Configure custom plugins, as needed.

To configure a custom plugin, copy the 11 commented lines under **** EXAMPLE 'plugins' VALUE** (with current defaults) ****** and paste them under **# NOTE: Place additional 'plugins' setting entries here, if desired.**

```

# ** EXAMPLE 'plugins' VALUE (with current defaults) **
#   cray_node_health.settings.plugins.data.desc.sample_key_a: null  <-- setting a multival key
#   cray_node_health.settings.plugins.data.sample_key_a.name: Plugin
#   cray_node_health.settings.plugins.data.sample_key_a.enabled: false
#   cray_node_health.settings.plugins.data.sample_key_a.command: ''
#   cray_node_health.settings.plugins.data.sample_key_a.action: ''
#   cray_node_health.settings.plugins.data.sample_key_a.warntime: 0
#   cray_node_health.settings.plugins.data.sample_key_a.timeout: 0
#   cray_node_health.settings.plugins.data.sample_key_a.restartdelay: 0
#   cray_node_health.settings.plugins.data.sample_key_a.uid: 0
#   cray_node_health.settings.plugins.data.sample_key_a.gid: 0
#   cray_node_health.settings.plugins.data.sample_key_a.sets: Application

```

Uncomment the lines, replace `sample_key_a` with the identifier chosen for this plugin (e.g., "my_plugin") in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the null value is required; do not remove or change it). Finally, add site-specific values for the rest of the settings.

Repeat this step for each additional custom plugin definition.

This translation table shows an example for a plugin called `my_plugin`, which has values to migrate that are found in a `my_plugin` stanza in the CLE 5.2 / SMW 7.2

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

Table 88. Translation Table: Variables beginning with `cray_node_health.settings.custom_plugins.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
desc.my_plugin	null	advanced	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
my_plugin.name	Plugin	advanced	N/A	N/A
my_plugin.enabled	false	advanced	N/A	N/A
my_plugin.command	none	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([my_plugin] Command)
my_plugin.action		advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([my_plugin] Action)
my_plugin.warntime	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([my_plugin] WarnTime)
my_plugin.timeout	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf b([my_plugin] Timeout)
my_plugin.restartdelay	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([my_plugin] RestartDelay)
my_plugin.uid	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([my_plugin] Uid)
my_plugin.gid	0	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([my_plugin] Gid)
my_plugin.sets	Application	advanced	N/A	files: SMW /opt/xt-images/templates/default/etc/ opt/cray/nodehealth/nodehealth.conf ([my_plugin] Sets)

6.4.2.31 Update `cray_persistent_data` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Cray Persistent Data service provides persistent storage to nodes, which can be configured on a per-node basis. This procedure configures some basic settings in the `cray_persistent_data` configuration worksheet to add site-specific data.

NOTE: `cray_persistent_data` must be enabled when using boot node failover or SDB node failover.

Procedure

1. Edit `cray_persistent_data_worksheet.yaml`.

```
smw# vi cray_persistent_data_worksheet.yaml
```

2. Uncomment `cray_persistent_data.enabled` and set it to `true`.
3. Uncomment `cray_persistent_data.settings.directories.data.persistent_space_mount` and set it to match the `fs_mountpoint` for the CLE storage set (`cledefault`).

The full setting name is

`cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg.volumes.nvolatile.fs_mount_point`, which is a setting in the `cray_bootraid` service in the global config set. Find this value by using `cfgset search`, and then scan the list of matches for this setting.

```
smw# cfgset search --service cray_bootraid --level advanced \
--state all --term nvolatile global
```

4. Ensure that these `client_groups` settings are uncommented.

For each setting, uncomment both the variable and its value (the line that follows it, which is a list containing one node group). They should all be set to a list containing the node group `service_nodes`, except for the NFS mount: `nfs.client_groups` should be set to a list containing `boot_nodes` and `sdb_nodes`.

```
#cray_persistent_data.settings.mounts.data./var/opt/cray/alps.client_groups:
#- service_nodes
```

```
#cray_persistent_data.settings.mounts.data./var/opt/cray/aeld.client_groups:
#- service_nodes
```

```
#cray_persistent_data.settings.mounts.data./var/opt/cray/apptermd.client_groups:
#- service_nodes
```

```
#cray_persistent_data.settings.mounts.data./var/opt/cray/ncmd.client_groups:
#- service_nodes

#cray_persistent_data.settings.mounts.data./var/lib/nfs.client_groups:
#- boot_nodes
#- sdb_nodes
```

5. If the Cray DRC (dynamic RDMA credentials) service will be used with persistent storage, configure space for it by defining a `cray_persistent_data` mount point.

In the worksheet, copy the five lines below `# ** EXAMPLE 'mounts' VALUE` (with current defaults) `**` and paste them below `# NOTE: Place additional 'mounts' setting entries here, if desired.`

```
# ** EXAMPLE 'mounts' VALUE (with current defaults) **
# cray_persistent_data.settings.mounts.data.mount_point.sample_key_a: null <-- setting a multival key
# cray_persistent_data.settings.mounts.data.sample_key_a.alt_storage_path: ''
# cray_persistent_data.settings.mounts.data.sample_key_a.options: ''
# cray_persistent_data.settings.mounts.data.sample_key_a.ancestor_def_perms: '0771'
# cray_persistent_data.settings.mounts.data.sample_key_a.client_groups: []
```

Uncomment the lines, replace `sample_key_a` with `/var/opt/cray/rdma-credentials` in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). For the `client_groups` setting (last line), remove the empty list (`[]`), and add a node group (one that contains the service node that should be running the DRC service) on a separate line prefixed by a hyphen and space (`-`). The `cname` of this node is the same as was set for the `cray_drc.settings.server.data.server_cname` setting in the Cray DRC worksheet (`cray_drc_worksheet.yaml`). To see which node group contains the node with this `cname`, or to create such a node group for this system (**`NODE_GROUP`** in this example), edit `cray_node_groups_worksheet.yaml`.

Leave all other settings at the default values.

```
# NOTE: Place additional 'mounts' setting entries here, if desired.
cray_persistent_data.settings.mounts.data.mount_point./var/opt/cray/rdma-credentials: null
cray_persistent_data.settings.mounts.data./var/opt/cray/rdma-credentials.alt_storage_path: ''
cray_persistent_data.settings.mounts.data./var/opt/cray/rdma-credentials.options: ''
cray_persistent_data.settings.mounts.data./var/opt/cray/rdma-credentials.ancestor_def_perms: '0771'
cray_persistent_data.settings.mounts.data./var/opt/cray/rdma-credentials.client_groups:
- NODE_GROUP

#***** END Service Setting: mounts *****
```

6. If a workload manager (WLM) will be used, configure space for its spool area by defining a `cray_persistent_data` mount point..

Use these spool file paths as mount points for persistent storage, depending on the WLM used at this site. Note that for Moab/TORQUE, two mount points will need to be defined.

- Moab/TORQUE: `/var/spool/moab` and `/var/spool/torque`
- PBS: `/var/spool/PBS`
- Slurm: `/var/spool/slurm`

In the worksheet, copy the five lines below `# ** EXAMPLE 'mounts' VALUE` (with current defaults) `**` and paste them below `# NOTE: Place additional 'mounts' setting entries here, if desired.`

```
# ** EXAMPLE 'mounts' VALUE (with current defaults) **
# cray_persistent_data.settings.mounts.data.mount_point.sample_key_a: null <-- setting a multival key
# cray_persistent_data.settings.mounts.data.sample_key_a.alt_storage_path: ''
```

```
# cray_persistent_data.settings.mounts.data.sample_key_a.options: ''
# cray_persistent_data.settings.mounts.data.sample_key_a.ancestor_def_perms: '0771'
# cray_persistent_data.settings.mounts.data.sample_key_a.client_groups: []
```

Uncomment the lines, replace `sample_key_a` with one the correct spool file path in all lines, and remove the `<--` setting a multival key text at the end of the first line (note that the `null` value is required; do not remove or change it). For the `client_groups` setting (last line), remove the empty list (`[]`), and add a node group (one that contains the WLM server node) on a separate line prefixed by a hyphen and space (`-`). To see which node group contains the node with this `cname`, or to create such a node group for this system (**`NODE_GROUP`** in this example), edit `cray_node_groups_worksheet.yaml`.

Leave all other settings at the default values.

This example shows the Slurm file path as the mount point (`sample_key_a`).

```
# NOTE: Place additional 'mounts' setting entries here, if desired.
cray_persistent_data.settings.mounts.data.mount_point./var/spool/slurm: null
cray_persistent_data.settings.mounts.data./var/spool/slurm.alt_storage_path: ''
cray_persistent_data.settings.mounts.data./var/spool/slurm.options: ''
cray_persistent_data.settings.mounts.data./var/spool/slurm.ancestor_def_perms: '0771'
cray_persistent_data.settings.mounts.data./var/spool/slurm.client_groups:
- NODE_GROUP

#***** END Service Setting: mounts *****
```

6.4.2.32 Update `cray_rsis` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

RSIP (realm-specific IP) helps to maintain packet integrity by allowing an RSIP host to borrow one or more IP addresses from a set of configured RSIP gateways. This procedure configures some simple settings in the Cray RSIP configuration service worksheet to add site-specific data, such as which nodes will be RSIP servers and which will be RSIP clients (the last step provides a translation table for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0).

Systems with service nodes that will provide the RSIP service need to have RSIP configured. For simple RSIP configuration, enable the `cray_rsis` service and provide values for the settings in this worksheet. For more complex RSIP configuration, disable the `cray_rsis` service at this time. The service must be disabled because some of the advanced configuration can be done only after the XC system has booted, and if RSIP is enabled but not fully configured, it will cause boot errors. Therefore, for complex RSIP configurations, this service must be enabled and configured later in the process after the XC system boots successfully.

Procedure

1. Edit `cray_rsis_worksheet.yaml`.

```
smw# vi cray_rsis_worksheet.yaml
```

2. Uncomment `cray_rsis.enabled` and set it as follows.

- Set it to `false` if this system will not use RSIP. Skip the rest of this procedure.
- Set it to `false` if this system requires complex RSIP configuration and the XC system has not yet booted. Skip the rest of this procedure.
- Set it to `true` if this system will use RSIP and the settings in the `cray_rsip` configuration worksheets suffice to configure RSIP. Proceed to the next step.
- Set it to `true` if this system requires complex RSIP configuration and the XC system has booted. Proceed to the next step.

3. Enter the node group (or groups) of the nodes that will be RSIP servers on this system.

To create one or more node groups that contain the RSIP server nodes (by `cname`) for this system (`rsip_nodes` in this example), edit `cray_node_groups_worksheet.yaml`.

Uncomment `cray_rsip.settings.service.data.server_groups`, remove the empty list (`[]`), and add the node group(s) on separate lines prefixed by a hyphen and space (`-`). For a migration, use the translation table in the last step of this procedure for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

```
cray_rsip.settings.service.data.server_groups:
- rsip_nodes
```

4. Enter the node group (or groups) of the service nodes that will be RSIP clients on this system, such as a MOM node.

To create one or more node groups that contain the RSIP client nodes (by `cname`) for this system (`rsip_servicenode_clients` in this example), edit `cray_node_groups_worksheet.yaml`.

Uncomment `cray_rsip.settings.service.data.node_groups_as_client`, remove the empty list (`[]`), and add the node group(s) on separate lines prefixed by a hyphen and space (`-`). For a migration, use the translation table in the last step of this procedure for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

```
cray_rsip.settings.service.data.node_groups_as_client:
- rsip_servicenode_clients
```

5. (For complex RSIP configuration only) If this system requires complex RSIP configuration, and the XC system has booted, generate the advanced configuration files and set the `use_xtrsipcfg` setting.

- Uncomment `cray_rsip.settings.service.data.use_xtrsipcfg` and ensure that it is set to `true`.
- Run `xtrsipcfg_v2` as root.

This command will generate the needed configuration files and place them in `/var/opt/cray/imps/config/sets/p0/files/roles/rsip/`.

NOTICE: `xtrsipcfg_v2` can be run only when the CLE system is booted.

```
smw# /opt/cray/xtrsipcfg/*/bin/xtrsipcfg_v2 -b
```

6. Migrate other configuration settings, as needed.

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Things to note:

- With CLE 5.2 / SMW 7.2, `CLEinstall.conf` had one list of nodes that were the RSIP servers and a second list of Ethernet interfaces on those RSIP servers. This permitted some nodes to use `eth0`, others `eth1`, others `eth2`, and others `eth3`. These were placed uniquely into each RSIP node's `/etc/opt/cray/rsipd/rsipd.conf` file for the `ext_if` variable. However, with CLE 6.0 / SMW 8.0, there is one setting in the config set related to the old `ext_if` setting that applies to all RSIP nodes, so all RSIP server nodes need to use an Ethernet interface with the same name. Only `eth0`, for example, would be used for RSIP on every RSIP server node.
- In CLE 5.2 / SMW 7.2, the `raw_so_rcvbuf` and `raw_so_sndbuf` variables defaulted to `-1`, but in CLE 6.0 / SMW 8.0, `cray_rsip.settings.service.data.raw_so_rcvbuf` and `cray_rsip.settings.service.data.raw_so_sndbuf` default to `0`. In CLE 6.0 / SMW 8.0, the only valid negative value is `"-1"` and should be used only when advised to by Cray.
- Several settings that were `1/0` or `on/off` in CLE 5.2 / SMW 7.2, such as those shown below, are now `true/false` in CLE 6.0 / SMW 8.0.

```
cray_rsip.settings.service.data.arp
cray_rsip.settings.service.data.force_iface
```

- The `cray_rsip.settings.service.data.max_clients` setting value of `0` acts as "auto" to automatically calculate max clients. See the guidance text in `cray_rsip_worksheet.yaml`.

Table 89. Translation Table: Variables beginning with `cray_rsip.settings.service.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>server_groups</code>	<code>[]</code>	required	N/A	<i>installer:</i> <code>CLEinstall.conf</code> (<code>rsip_nodes</code>)
<code>compute_as_client</code>	<code>true</code>	advanced	N/A	<i>installer:</i> <code>CLEinstall.conf</code> (<code>CNL_rsip</code>)
<code>node_groups_as_client</code>	<code>[]</code>	advanced	N/A	<i>installer:</i> <code>CLEinstall.conf</code> (<code>rsip_servicenode_clients</code>)
<code>method_exception_groups</code>	<code>[]</code>	advanced	N/A	N/A
<code>use_xtrsipcfg</code>	<code>false</code>	advanced	N/A	N/A
<code>client_delay</code>	<code>60</code>	advanced	N/A	N/A
<code>use_rsip_local_ports</code>	<code>1</code>	advanced	N/A	N/A
<code>client_method</code>	<code>2</code>	advanced	N/A	N/A
<code>rsaip_method</code>	<code>2</code>	advanced	N/A	<i>files:</i> RSIP server node sharedroot <code>/etc/opt/cray/rsipd/rsipd.conf</code> (<code>rsaip_method</code>)
<code>pool</code>	<code>[]</code>	advanced	N/A	<i>files:</i> RSIP server node sharedroot <code>/etc/opt/cray/rsipd/rsipd.conf</code> (<code>pool</code>)
<code>rsa_reserved</code>	<code>0</code>	advanced	N/A	<i>files:</i> RSIP server node sharedroot <code>/etc/opt/cray/rsipd/rsipd.conf</code> (<code>rsa_reserved</code>)
<code>rsip_port</code>	<code>4555</code>	advanced	N/A	<i>files:</i> RSIP server node sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/opt/cray/rsipd/rsipd.conf (rsip_port)
initial_register_lease_time	90	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (initial_register_lease_time)
maximum_lease_time	3600	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (maximum_lease_time)
arp	false	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (arp)
force_iface	false	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (force_iface)
net_delay	2	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (net_delay)
tunnel_mtu	0	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (tunnel_mtu)
rsip_port_start	8192	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (port_range)
rsip_port_end	60000	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (port_range)
ip_port_start	60001	advanced	N/A	N/A
ip_port_end	65535	advanced	N/A	N/A
listen_start	1	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (listen_range)
listen_end	60000	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (listen_range)
ext_if	eth0	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (ext_if) installer: CLEinstall.conf (rsip_interfaces)
int_if	ipogif0	advanced	N/A	files: RSIP server node sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/opt/cray/rsipd/rsipd.conf (int_if)
max_clients	0	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (max_clients)
initial_lease_time	600	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (initial_lease_time)
raw_so_rcvbuf	'0'	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (raw_so_rcvbuf)
raw_so_sndbuf	'0'	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (raw_so_sndbuf)
log_level	5	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (log_level)
status_level	1	advanced	N/A	files: RSIP server node sharedroot /etc/opt/cray/rsipd/rsipd.conf (status_level)

6.4.2.33 Update `cray_rur` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

RUR (Resource Utilization Reporting) is a scalable framework for collecting utilization data from nodes within a user application. It is also a collection of plugins that report an extensible list of statistics about the hardware and software resources consumed by the application. RUR allows the creation of both data plugins for collecting statistics about the use of additional resources, and output plugins for writing the summarized usage data to additional forms of permanent storage.

This procedure enables the Cray RUR service and shows two Cray ALPS settings that must be set if RUR is used. No other settings need to be changed at this point in the process. Cray recommends that sites install and configure CLE with default plugins first, and then return to the Cray RUR service after the first system boot to configure custom plugins, if needed, using the `data_plugins` or `output_plugins` settings. This procedure

also provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

For information about RUR data collectors and how to enable them, see the procedures in "Resource Utilization Reporting" under "Monitor the System" in *XC™ Series System Administration Guide (S-2393)*.

Procedure

1. Edit `cray_rur_worksheet.yaml`.

```
smw# vi cray_rur_worksheet.yaml
```

2. Uncomment `cray_rur.enabled` and set it to `true`.

3. Ensure that the `prologPath` and `epilogPath` variables in the Cray ALPS service have been set.

The configuration worksheet for the Cray ALPS service has the following two settings that must be configured if RUR is used. See [Update `cray_alps Worksheet`](#) on page 138.

```
cray_alps.settings.apsys.data.prologPath: /opt/cray/rur/default/bin/rur_prologue.py
cray_alps.settings.apsys.data.epilogPath: /opt/cray/rur/default/bin/rur_epilogue.py
```

MIGRATE CONFIGURATION DATA -----

Use the translation tables below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to CLE 6.0 / SMW 8.0 settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

The CLE 5.2 / SMW 7.2 values of some of the RUR settings can be found in stanzas in the RUR configuration file on the default sharedroot, `/etc/opt/cray/rur/rur.conf`. For example, here is a stanza from that file (begins with "[`rur_stage`]" and ends before "[`rur_gather`]").

```
[rur_stage]
stage_timeout: 30
stage_dir: /var/spool/RUR/
[rur_gather]
```

That stanza provides values for these CLE 6.0 / SMW 8.0 variables:

```
cray_rur.settings.rur_stage.data.stage_timeout
cray_rur.settings.rur_stage.data.stage_dir
```

The translation table entries for these two variables look like this:

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>rur_stage.data.stage_timeout</code>	90	advanced	N/A	<i>files:</i> default sharedroot <code>/etc/opt/cray/rur/rur.conf</code> ([<code>rur_stage</code>] <code>stage_timeout</code>)
<code>rur_stage.data.stage_dir</code>	<code>/var/spool/RUR</code>	advanced	N/A	<i>files:</i> default sharedroot <code>/etc/opt/cray/rur/rur.conf</code> ([<code>rur_stage</code>] <code>stage_dir</code>)

Other things to note:

- The kncstats plugin was implemented in CLE 5.2 / SMW 7.2 for the Intel Xeon Phi KNC (Knight's Corner), which is not supported hardware for CLE 6.0 / SMW 8.0. Therefore, the kncstats plugin is not supported in CLE 6.0 / SMW 8.0. The model of Intel Xeon Phi supported with CLE 6.0 / SMW 8.0 is the KNL (Knight's Landing).
- The `cray_rur.settings.file.data.arg` setting may have been a shared file system, such as `/lus/scratch`. With CLE 6.0 / SMW 8.0, this can be a local file on each login/MOM node or can be on a shared file system. The default is to put it in `/tmp` on each login/MOM node, which is not a persistent file system between reboots.

4. Migrate RUR configuration settings, as needed.

Table 90. Translation Table: Variables beginning with `cray_rur.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>base.data.debug_level</code>	ERROR	advanced	N/A	<i>files:</i> default sharedroot <code>/etc/opt/cray/rur/rur.conf</code> ([global] <code>debug_level</code>)
<code>base.data.keep_temp_files</code>	false	advanced	N/A	N/A
<code>base.data.use_json</code>	false	advanced	N/A	N/A
<code>rur_stage.data.stage_timeout</code>	90	advanced	N/A	<i>files:</i> default sharedroot <code>/etc/opt/cray/rur/rur.conf</code> ([<code>rur_stage</code>] <code>stage_timeout</code>)
<code>rur_stage.data.stage_dir</code>	<code>/var/spool/RUR</code>	advanced	N/A	<i>files:</i> default sharedroot <code>/etc/opt/cray/rur/rur.conf</code> ([<code>rur_stage</code>] <code>stage_dir</code>)
<code>rur_gather.data.gather_timeout</code>	90	advanced	N/A	<i>files:</i> default sharedroot <code>/etc/opt/cray/rur/rur.conf</code> ([<code>rur_gather</code>] <code>gather_timeout</code>)
<code>rur_gather.data.gather_dir</code>	<code>/tmp/rur</code>	advanced	N/A	<i>files:</i> default sharedroot <code>/etc/opt/cray/rur/rur.conf</code> ([<code>rur_gather</code>] <code>gather_dir</code>)
<code>rur_post.data.post_timeout</code>	90	advanced	N/A	<i>files:</i> default sharedroot <code>/etc/opt/cray/rur/rur.conf</code> ([<code>rur_post</code>] <code>post_timeout</code>)
<code>rur_post.data.post_dir</code>	<code>/tmp/rur</code>	advanced	N/A	<i>files:</i> default sharedroot <code>/etc/opt/cray/rur/rur.conf</code> ([<code>rur_post</code>] <code>post_dir</code>)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
gpustat.data.enable	false	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([plugins] gpustat)
gpustat.data.stage	/opt/cray/rur/default/bin/ gpustat_stage.py	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([gpustat] stage)
gpustat.data.post	/opt/cray/rur/default/bin/ gpustat_post.py	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([gpustat] post)
taskstats.data.enable	true	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([plugins] taskstats)
taskstats.data.stage	/opt/cray/rur/default/bin/ taskstats_stage.py	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([taskstats] stage)
taskstats.data.post	/opt/cray/rur/default/bin/ taskstats_post.py	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([taskstats] post)
taskstats.data.arg	json-dict	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([taskstats] arg)
energy.data.enable	true	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([plugins] energy)
energy.data.stage	/opt/cray/rur/default/bin/ energy_stage.py	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([energy] stage)
energy.data.post	/opt/cray/rur/default/bin/ energy_post.py	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([energy] post)
energy.data.arg	json-dict	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([energy] arg)
timestamp.data.enable	true	advanced	N/A	<i>files:</i> default sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/opt/cray/rur/rur.conf ([plugins] timestamp)
timestamp.data.stage	/opt/cray/rur/default/bin/ timestamp_stage.py	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([timestamp] stage)
timestamp.data.post	/opt/cray/rur/default/bin/ timestamp_post.py	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([timestamp] post)
memory.data.enable	false	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([plugins] memory)
memory.data.stage	/opt/cray/rur/default/bin/ memory_stage.py	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([memory] stage)
memory.data.post	/opt/cray/rur/default/bin/ memory_post.py	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([memory] post)
memory.data.arg	none	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([memory] arg)
nodeuse.data.enable	false	advanced	N/A	N/A
nodeuse.data.stage	/opt/cray/rur/default/bin/ nodeuse_stage.py	advanced	N/A	N/A
nodeuse.data.post	/opt/cray/rur/default/bin/ nodeuse_post.py	advanced	N/A	N/A
dws.data.enable	false	advanced	N/A	N/A
dws.data.stage	/opt/cray/rur/default/bin/ dws_stage.py	advanced	N/A	N/A
dws.data.post	/opt/cray/rur/default/bin/ dws_post.py	advanced	N/A	N/A
llm.data.enable	true	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([outputplugins] llm)
llm.data.output	/opt/cray/rur/default/bin/ llm_output.py	advanced	N/A	files: default sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/opt/cray/rur/rur.conf ([llm] output)
file.data.enable	false	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([outputplugins] file)
file.data.output	/opt/cray/rur/default/bin/ file_output.py	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([file] output)
file.data.arg	/tmp/rur.output	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([file] arg)
user.data.enable	true	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([outputplugins] user)
user.data.output	/opt/cray/rur/default/bin/ user_output.py	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([user] output)
user.data.arg	single, opt-in	advanced	N/A	files: default sharedroot /etc/opt/cray/rur/rur.conf ([user] arg)

5. Migrate RUR data plugin configuration settings, as needed.

If this site used RUR plugins in CLE 5.2 / SMW 7.2, that configuration data can be migrated in this step. This example shows a site data plugin called "siteplug." Look for the name of site data plugin(s) in the [plugins] stanza from CLE 5.2 / SMW 7.2 to use when adding this stanza to the CLE 6.0 / SMW 8.0 `cray_rur` configuration worksheet.

Note: The custom data for data plugins may be in the default sharedroot `/opt/cray/rur/default/bin`, but it could be in any file system that is readable by compute nodes, owned by root, and not writeable by non-root users. Ensure that these scripts are saved in the archive so they can be transferred to the CLE 6.0 / SMW 8.0 system. It may be necessary to use Simple Sync to distribute these site plugins to the appropriate login nodes and MOM nodes (see [Configure Simple Sync for Custom RUR Plugins](#)).

Configure the following group of settings if this system has a site data plugin. Repeat this step for each site data plugin.

In the worksheet, copy the lines below `# ** EXAMPLE 'data_plugins' VALUE` (with current defaults) `**` and paste them below `# NOTE: Place additional 'data_plugins' setting entries here, if desired.`

```
# ** EXAMPLE 'data_plugins' VALUE (with current defaults) **
#   cray_rur.settings.data_plugins.data.plugin_name.sample_key_a: null <--
```

```

setting a multival key
#   cray_rur.settings.data_plugins.data.sample_key_a.stage: none
#   cray_rur.settings.data_plugins.data.sample_key_a.post: none
#   cray_rur.settings.data_plugins.data.sample_key_a.arg: none
#   cray_rur.settings.data_plugins.data.sample_key_a.enable: true

```

Uncomment the lines, replace `sample_key_a` with a string that identifies the plugin (`siteplug` in the example below), then remove the `<-- setting a multival key` text at the end of the first line in each set (note that the null value is required; do not remove or change it). Finally, modify the values as needed for this site. For a migration, use the translation table provided below.

```

cray_rur.settings.data_plugins.data.plugin_name.siteplug: null
cray_rur.settings.data_plugins.data.siteplug.stage: none
cray_rur.settings.data_plugins.data.siteplug.post: none
cray_rur.settings.data_plugins.data.siteplug.arg: none
cray_rur.settings.data_plugins.data.siteplug.enable: true

```

Table 91. Translation Table: Variables beginning with `cray_rur.settings.data_plugins.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>plugin_name.siteplug</code>	null	advanced	N/A	N/A
<code>siteplug.stage</code>	none	advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/rur/rur.conf ([siteplug] stage)
<code>siteplug.post</code>	none	advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/rur/rur.conf ([siteplug] post)
<code>siteplug.arg</code>	none	advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/rur/rur.conf ([siteplug] arg)
<code>siteplug.enable</code>	true	advanced	N/A	<i>files</i> : default sharedroot /etc/opt/cray/rur/rur.conf ([plugins] siteplug)

6. Migrate RUR output plugin configuration settings, as needed.

If this site used RUR output plugins in CLE 5.2 / SMW 7.2, that configuration data can be migrated in this step. This example shows a site data plugin called "siteout." Look for the name of site data plugin(s) in the `[plugins]` stanza from CLE 5.2 / SMW 7.2 to use when adding this stanza to the CLE 6.0 / SMW 8.0 `cray_rur` configuration worksheet.

Note: The custom data for output plugins may be in the shared root `/opt/cray/rur/default/bin`, but it could be in any file system that is readable by compute nodes, owned by root, and not writeable by non-root users. Ensure that these scripts are saved in the archive so they can be transferred to the CLE 6.0 / SMW 8.0 system. It may be necessary to use Simple Sync to distribute these site plugins to the appropriate login nodes and MOM nodes (see [Configure Simple Sync for Custom RUR Plugins](#)).

Configure the following group of settings if this system has a site output plugin. Repeat this step for each site output plugin.

In the worksheet, copy the lines below `# ** EXAMPLE 'output_plugins' VALUE` (with current defaults) `**` and paste them below `# NOTE: Place additional 'output_plugins' setting entries here, if desired.`

```
# ** EXAMPLE 'output_plugins' VALUE (with current defaults) **
#   cray_rur.settings.output_plugins.data.plugin_name.sample_key_a: null  <--
#   setting a multival key
#   cray_rur.settings.output_plugins.data.sample_key_a.output: none
#   cray_rur.settings.output_plugins.data.sample_key_a.arg: none
#   cray_rur.settings.output_plugins.data.sample_key_a.enable: true
```

Uncomment the lines, replace `sample_key_a` with a string that identifies the plugin (`siteout` in the example below), then remove the `<-- setting a multival key` text at the end of the first line in each set (note that the null value is required; do not remove or change it). Finally, modify the values as needed for this site. For a migration, use the translation table provided below.

```
cray_rur.settings.output_plugins.data.plugin_name.siteout: null
cray_rur.settings.output_plugins.data.siteout.output: none
cray_rur.settings.output_plugins.data.siteout.arg: none
cray_rur.settings.output_plugins.data.siteout.enable: true
```

Table 92. Translation Table: Variables beginning with `cray_rur.settings.output_plugins.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>plugin_name.siteout</code>	null	advanced	N/A	N/A
<code>siteout.output</code>	none	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([siteout] output)
<code>siteout.arg</code>	none	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([siteout] arg)
<code>siteout.enable</code>	true	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/rur/rur.conf ([outputplugins] siteout)

6.4.2.34 Update `cray_scalable_services` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

Cray Scalable Services defines a tree of servers (nodes), starting with the server of authority (SoA), that are used in the scaling of the system. Configuration of Scalable Services is required for a functioning system. For more

information, see [About Cray Scalable Services](#) on page 410. This procedure configures some basic settings in the `cray_scalable_services` configuration worksheet to add site-specific data.

Procedure

1. Edit `cray_scalable_services_worksheet.yaml`.

```
smw# vi cray_scalable_services_worksheet.yaml
```

2. Uncomment `cray_scalable_services.enabled` and ensure that it is set to `true`.
3. Uncomment `cray_scalable_services.settings.scalable_service.data.server_of_authority` and ensure that it is set to `smw`.
4. Enter the node group (or node groups) of the nodes that will be tier1 servers on this system.

Ensure that these node groups include the cname of the boot node and any other nodes that have an Ethernet connection to the SMW. The SDB node should also have a connection to the SMW, so it can be a tier1 server.

IMPORTANT: If enabling boot node failover or SDB node failover, ensure that all boot nodes and all SDB nodes are in a tier1 node group and none of them are in a tier2 node group.

Uncomment `cray_scalable_services.settings.scalable_service.data.tier1_groups`, remove the empty list (`[]`), and add these predefined node groups on separate lines prefixed by a hyphen and space (`-`).

```
cray_scalable_services.settings.scalable_service.data.tier1_groups:
- boot_nodes
- sdb_nodes
- OTHER_TIER1_NODE_GROUP
```

To verify that these node groups contain the tier1 server nodes (by cname) for this system, to add the correct tier1 nodes to them, or to add a new node group for tier1 servers, (`OTHER_TIER1_NODE_GROUP` in this example), edit `cray_node_groups_worksheet.yaml` (see [Update cray_node_groups Worksheet](#) on page 236).

5. Enter the node group (or node groups) of the nodes that will be tier2 servers on this system.

Uncomment `cray_scalable_services.settings.scalable_service.data.tier2_groups` and the line below it, which is a list of one predefined node group.

```
cray_scalable_services.settings.scalable_service.data.tier2_groups:
- tier2_nodes
```

To verify that the predefined tier2 node group contains the correct tier2 server nodes (by cname) for this system or to add the correct tier2 nodes to them, edit `cray_node_groups_worksheet.yaml`.

Q. How many tier2 nodes are needed?

A. At least one server must be provided, and for resiliency, two nodes placed on different blades. The recommended ratio of tier2 nodes (servers) to tier3 nodes (clients) is 1 to 400.

Q. Will adding more tier2 nodes help performance?

A. Adding more tier2 nodes does not always yield additional performance and is subject to diminishing returns.

Q. What kind of node can be used as a tier2 node?

A. Use these:

- OPTIMAL: dedicated repurposed compute nodes (RCN)
- dedicated service nodes
- nodes with uniform light to moderate load
- nodes with relatively homogeneous single core speeds to reduce resource contention disparity during periods of partial availability

AVOID these (will result in sub-optimal performance):

- nodes with slower individual CPU cores, such as Intel® Xeon Phi™ "Knights Landing" (KNL) processors
- direct-attached Lustre (DAL) servers
- RSIP (realm-specific IP) servers
- login nodes

Q. Can a tier2 node have more than one role?

A. Small test and development systems (TDS) may use tier2 nodes that have additional roles, but generally, it is better for tier2 nodes to be dedicated.

Q. Where should tier2 nodes be placed?

A. Distribute nodes throughout the system (on different blades) for resiliency in the event of hardware failure.

Check the guidance for tier2 nodes in this configuration worksheet for additional requirements or limitations.

6.4.2.35 Update `cray_sdb` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray Software Database (SDB) service configures the services and settings for the SDB node. This procedure configures some basic settings in the `cray_sdb` service configuration worksheet. The last step of this procedure provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_sdb_worksheet.yaml`.

```
smw# vi cray_sdb_worksheet.yaml
```

2. Uncomment `cray_sdb.enabled` and ensure that it is set to `true`.
3. Configure the SDB node groups setting.

- a. Uncomment the SDB node groups setting.

Be sure to uncomment both lines.

```
#cray_sdb.settings.node_groups.data.sdb_groups:
#- sdb_nodes
```

- b. Verify that the `sdb_nodes` node group has been accurately defined for this site.

To verify, edit `cray_node_groups_worksheet.yaml` and search for `sdb_nodes`. For a migration, use the translation table in the last step of this procedure for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

4. Configure the admin and root database passwords.

Uncomment the following two password settings and replace the default values with site-specific values. For a migration, use the translation table in the last step of this procedure for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

These passwords will be stored in clear text in the config set. Note that the values of these passwords are excluded when the config set is distributed to eLogin nodes.

```
#cray_sdb.settings.database.data.db_admin_password: sys_mgt
#cray_sdb.settings.database.data.db_current_root_password: ''
```

5. (Optional) Set the host for the daemon that syncs the HSS database.

Uncomment this setting to configure it. Cray recommends keeping the default value of 'sdb'; however, if this site wishes `xtdbsyncd` to run on the boot node instead, change the value to 'boot.' For a migration, use the translation table in the last step of this procedure for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to this setting.

```
#cray_sdb.settings.database.data.synchost: sdb
```

6. Migrate other database configuration settings, as needed.

It is rare for these to be changed to values other than the defaults.

Use the translation tables below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to CLE 6.0 / SMW 8.0 settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

The CLE 5.2 / SMW 7.2 values of some of the SDB settings can be found in stanzas in the SDB configuration file in shared root `/root/.my.cnf`. For example, here are two variables in the `[client]` stanza from that file.

```
[client]
user=value
password=value
```

It provides values for these CLE 6.0 / SMW 8.0 variables:

```
cray_sdb.settings.database.data.db_admin_user
cray_sdb.settings.database.data.db_admin_password
```

The translation table entry for the first variable looks like this. Note how the last column shows the context, file path, stanza, and variable.

Table 93. Translation Table: Variables beginning with `cray_sdb.settings.database.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
db_admin_user	sys_mgmt	advanced	N/A	files: sharedroot /root/.my.cnf ([client] user) installer: CLEintall.conf (sdb_accounting_user_name)

Here are the translation tables for the `cray_sdb` service.

Table 94. Translation Table: Variables beginning with `cray_sdb.settings.node_groups.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
sdb_groups	sdb_nodes	advanced	N/A	N/A

Table 95. Translation Table: Variables beginning with `cray_sdb.settings.database.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
directory	/var/lib/mysql	advanced	N/A	files: default sharedroot /etc/.my.cnf ([mysqld] datadir)
owner_user	mysql	advanced	N/A	files: default sharedroot /etc/.my.cnf ([mysqld] user)
owner_group	mysql	advanced	N/A	N/A
db_admin_user	sys_mgmt	advanced	N/A	files: sharedroot /root/.my.cnf ([client] user) installer: CLEintall.conf (sdb_accounting_user_name)
db_admin_password	sys_mgmt	basic	N/A	files: sharedroot /root/.my.cnf ([client] password) installer: CLEintall.conf (sdb_accounting_password)
db_current_root_password		basic	N/A	N/A
max_connections	2500	advanced	N/A	N/A
max_connect_errors	10000	advanced	N/A	N/A
port	3306	advanced	N/A	files: default sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/.my.cnf ([mysqld] port)
socket	/var/run/mysql/ mysql.sock	advanced	N/A	files: default sharedroot /etc/.my.cnf ([mysqld] socket)
synchost	sdb	advanced	N/A	N/A

6.4.2.36 Update `cray_service_node` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray Service Node configuration service configures the services and settings for service nodes. This procedure enables the `cray_service_node` service, which is sufficient for a fresh install. The last step of this procedure provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_service_node_worksheet.yaml`.

```
smw# vi cray_service_node_worksheet.yaml
```

2. Uncomment `cray_service_node.enabled` and set it to `true`.
3. Migrate values for advanced settings, as needed.

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to CLE 6.0 / SMW 8.0 settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Table 96. Translation Table: Variables beginning with `cray_service_node.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
kernel.data.vm_min_free_kbytes	102400	advanced	<i>probe (1)</i>	files: bootroot /etc/sysctl.conf installer: CLEinstall.conf (sysctl_conf_vm_min_free_kbytes)

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
nodehealth.data.pcmd_suid	false	advanced	<i>probe (2)</i>	installer: CLEinstall.conf (NHC_pcmd_suid)
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) <code>boot# sysctl -a grep vm.min_free_kbytes</code> (2) In sharedroot: <code>default# ls -l /opt/cray/nodehealth/default/bin/pcmd</code>				

6.4.2.37 Update `cray_shifter` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

Shifter is an HPC-focused implementation of Linux containers that was created at the Berkeley Labs NERSC supercomputing facility. It enables a large-scale HPC system to efficiently and safely allow end-users to run a docker image. The `cray_shifter` configuration service configures Shifter for Cray XC systems.

Shifter includes the following:

- A utility that typically runs on the compute node that creates the run-time environment for the application.
- An image gateway service that pulls images from a registry and repacks it in a format suitable for the HPC system.
- Scripts and plugins to integrate Shifter with various batch scheduler systems.

This procedure enables or disables `cray_shifter`, depending on whether it is needed for this site. If enabled, Cray recommends configuring the rest of the Shifter settings later after the system has been booted. To install and configure Shifter at that time, see *XC™ Series Shifter Installation Guide (S-2572)*. To migrate site-specific Shifter configuration data from CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0, use the translation tables provided in the last step, in conjunction with the Shifter installation guide, to configure Shifter after system boot.

Procedure

1. Edit `cray_shifter_worksheet.yaml`.
2. Uncomment `cray_shifter.enabled` and do one of the following:
 - Set it to `false` for systems that will NOT use Shifter. Skip the rest of the procedure.
 - Set it to `true` for systems that will use Shifter. Proceed to the next step.
3. Migrate `cray_shifter` settings, as needed.

For a migration, use these translation tables for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Table 97. Translation Table: Variables beginning with `cray_shifter.settings.options.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
udiMount	/var/udiMount	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/cnrte/roots.conf (UDI)
loopMount	/var/udiLoopMount	advanced	N/A	N/A
imagePath	/lus/scratch/UDI	basic	N/A	<i>files:</i> default sharedroot /etc/opt/cray/shifter/ shifter.conf (imagePath)
CacheDirectory	/lus/scratch/cache	basic	N/A	N/A
perNodeCachePath	/lus/scratch/cache	advanced	N/A	N/A
ExpandDirectory	dev/shm	basic	N/A	N/A
udiRootPath	/opt/cray/shifter/ default/	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/shifter/ shifter.conf (udiRootPath)
sitePreMountHook	advanced	N/A	N/A	N/A
sitePostMountHook	advanced	N/A	N/A	N/A
optUdiImage	/opt/cray/shifter/ default/lib/shifter/opt/ udiImage	advanced	N/A	N/A
etcPath	/etc/shifter/ shifter_etc_files	basic	N/A	<i>files:</i> default sharedroot /etc/opt/cray/shifter/ shifter.conf (etcPath)
allowLibcPwDCalls	true	advanced	N/A	N/A
allowLocalChroot	false	advanced	N/A	N/A
autoLoadKernelModule	true	advanced	N/A	N/A
mountUdiRootWritable	true	advanced	N/A	N/A
maxGroupCount	31	advanced	N/A	N/A
rootfsType	ramfs	advanced	N/A	N/A
gatewayTimeout	10	advanced	N/A	N/A
siteFs	/home:/home	advanced	N/A	N/A

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
siteEnv	SHIFTER_RUNTIME=1	advanced	N/A	N/A
siteEnvAppend	PATH=/opt/udilimage/bin	advanced	N/A	N/A
siteEnvPrepend	advanced	N/A	N/A	N/A
imageGatewayHostname	eloin	basic	N/A	files: default sharedroot /etc/opt/cray/shifter/ shifter.conf (imageGateway)
imageGateway	http://127.0.0.1:5000	basic	N/A	N/A
system	system_name	basic	N/A	N/A
defaultImageType	docker	advanced	N/A	N/A
allowedImageTypes	docker custom	advanced	N/A	N/A
defaultImageLocation	registry-1.docker.io	advanced	N/A	N/A

Table 98. Translation Table: Variables beginning with `cray_shifter.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
imageLocations.data.imageURL.registry-1.docker.io		advanced	N/A	N/A
imageLocations.data.registry-1.docker.io.remoteType	dockerv2	advanced	N/A	N/A
imageLocations.data.registry-1.docker.io.authentication	http	advanced	N/A	N/A
imagegwadmins.data.admin.root		advanced	N/A	N/A

6.4.2.38 Update `cray_simple_shares` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Cray Simple File-system Sharing service quickly shares files between compute nodes that are connected to the high speed network (HSN). This procedure configures some basic settings in the `cray_simple_shares` configuration worksheet.

Procedure

1. Edit `cray_simple_shares_worksheet.yaml`.

```
smw# vi cray_simple_shares_worksheet.yaml
```

2. Uncomment `cray_simple_shares.enabled` and ensure that it is set to `true`.

3. Update the NFS mount settings.

- a. Ensure that the node groups settings are configured.

Search in the file for 'NFS' DATA, and below that line, find these `server_groups` and `client_groups` settings for several pre-populated NFS client mounts. If they are commented, uncomment them.

```
# ** 'NFS' DATA **

#cray_simple_shares.settings.NFS.data./alps_shared.server_groups:
#- sdb_nodes
#cray_simple_shares.settings.NFS.data./alps_shared.client_groups:
#- service_nodes
#cray_simple_shares.settings.NFS.data./alps_shared.client_exclude_groups:
#- boot_nodes
...
#cray_simple_shares.settings.NFS.data./cray_home.server_groups:
#- boot_nodes
#cray_simple_shares.settings.NFS.data./cray_home.client_groups:
#- service_nodes
...
#cray_simple_shares.settings.NFS.data./var/opt/cray/imps.server_groups:
#- boot_nodes
#cray_simple_shares.settings.NFS.data./var/opt/cray/imps.client_groups:
#- tier2_nodes
...
#cray_simple_shares.settings.NFS.data./non_volatile.server_groups:
#- boot_nodes
#cray_simple_shares.settings.NFS.data./non_volatile.client_groups:
#- service_nodes
```

- b. If the home directory was changed in other configuration worksheets (e.g., `cray_local_users_worksheet.yaml`), change it here also.

Under 'NFS' DATA, look for settings with `cray_home` or `home` as the 'path' key. Ensure that they reflect the same home directory as used in `cray_local_users_worksheet.yaml`.

```

cray_simple_shares.settings.NFS.data./cray_home.server_groups:
- boot_nodes
cray_simple_shares.settings.NFS.data./cray_home.fs_root: /cray_home
cray_simple_shares.settings.NFS.data./cray_home.fs_export_opt:
  'secure,rw,no_subtree_check,no_root_squash,no_acl'
cray_simple_shares.settings.NFS.data.path./cray_home: null
cray_simple_shares.settings.NFS.data./cray_home.client_groups:
- service_nodes
cray_simple_shares.settings.NFS.data./cray_home.unconditional_mount: false

```

4. Update the DVS mount settings.

Search in the file for 'DVS' DATA, and below that line, find these settings for a pre-populated DVS client mount. If they are commented, uncomment them.

```

# ** 'DVS' DATA **
...
#cray_simple_shares.settings.DVS.data./var/opt/cray/imps.spath: /var/opt/cray/imps
#cray_simple_shares.settings.DVS.data./var/opt/cray/imps.client_groups:
#- all_nodes

```

Disambiguation. Notice that the path '/var/opt/cray/imps' appears twice in the first setting. The first instance is the path where clients will mount the file system. It is the 'key' (*mount_point*) for this client mount, so it appears in all of the settings for this client mount. The second instance is the path to the file system on the server node that is to be projected. It is the default value provided for this pre-populated DVS client mount. That first setting is simply specifying that the file system will be projected from the same path on the server as it is mounted from the client.

5. Verify that the node groups referenced in steps 3 and 4 have been accurately defined for this site.

To verify, edit `cray_node_groups_worksheet.yaml` and search for these node groups:

```

all_nodes
boot_nodes
sdb_nodes
service_nodes
tier2_nodes

```

```

smw# vi cray_node_groups_worksheet.yaml

```

6.4.2.39 Update `cray_simple_sync` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```

smw# cd /var/adm/cray/release/p0_worksheet_workarea

```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

Simple Sync is a mechanism for automatically distributing files to targeted locations on the Cray system. This procedure enables the `cray_simple_sync` service.

Procedure

1. Edit `cray_simple_sync_worksheet.yaml`.

```
smw# vi cray_simple_sync_worksheet.yaml
```

2. Uncomment `cray_simple_sync.enabled` and set it to `true`.
No other settings need to be changed.

6.4.2.40 Update `cray_ssh` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The SSH service allows the system to be accessed through a secure shell. This procedure enables the Cray SSH configuration service. The last step of this procedure provides a translation table for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_ssh_worksheet.yaml`.

```
smw# vi cray_ssh_worksheet.yaml
```

2. Uncomment `cray_ssh.enabled` and set it to `true`.
3. Migrate values for advanced settings, as needed.

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Note that in CLE 5.2 / SMW 7.2 with SLES 11 SP3, the default was to permit all SSH host key formats: `rsa_key`, `dsa_key`, and `ecdsa_key`. With CLE 6.0 / SMW 8.0 and SLES 12, a fourth SSH host key format is available: `ed25519`. If all of the CLE 5.2 / SMW 7.2 SSH host key formats are permitted at this site, then leave `cray_ssh.settings.sshd.data.hostkeys_v2` as the empty list. This will mean that the `sshd_config` file will have no entries and that `sshd` will permit all key types.

Table 99. Translation Table: Variables beginning with `cray_ssh.settings.sshd.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>permitrootlogin</code>	<code>true</code>	advanced	<i>probe (1)</i>	<i>files:</i> default sharedroot <code>/etc/ssh/sshd_config</code> (PermitRootLogin)
<code>passwordauthentication</code>	<code>true</code>	advanced	<i>probe (2)</i>	<i>files:</i> default sharedroot <code>/etc/ssh/sshd_config</code> (PasswordAuthentication)
<code>hostkeys_v2</code>	<code>[]</code>	advanced	<i>probe (3)</i>	<i>files:</i> default sharedroot <code>/etc/ssh/sshd_config</code> (HostKey)
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) <code>sdb# sshd -T grep permitrootlogin</code> (2) <code>sdb# sshd -T grep passwordauthentication</code> (3) <code>sdb# sshd -T grep hostkey</code>				

6.4.2.41 Update `cray_storage` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Storage service defines which storage set the current partition or system may use for persistent storage. Storage sets are defined in the global config set. This procedure configures some basic settings in the `cray_storage` configuration worksheet.

Procedure

1. Edit `cray_storage_worksheet.yaml`.

```
smw# vi cray_storage_worksheet.yaml
```

2. Uncomment `cray_storage.enabled` and set it to `true`.

3. Uncomment `cray_storage.settings.storage.data.active_storage_set` and set it to be the name of the CLE storage set in the `cray_bootraid` service, which is in the global config set.

Use this command to show all storage sets defined in the global config set.

```
smw# cfmset search -s cray_bootraid global |awk -F'.' '{print $5}' | sort -u
```

4. (For reinstall only) Uncomment `cray_storage.settings.storage.data.zero_volumes_on_create` and set it to true if this system is reinstalling to a CLE storage set that had been in use previously.

6.4.2.42 Update `cray_sysconfig` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray System Configuration service controls configuration of files in `/etc/sysconfig`. The `sysconfig` service can be used to specify particular configuration file settings and values.

This procedure enables the `cray_sysconfig` service and provides an example of changing a configuration file in `/etc/sysconfig`. For a migration, this procedure also provides translation tables (in step 4 on page 294) for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_sysconfig_worksheet.yaml`.

```
smw# vi cray_sysconfig_worksheet.yaml
```

2. Uncomment `cray_sysconfig.enabled` and set it to true.
3. Change configuration settings in a file in `/etc/sysconfig`, as needed.

Repeat this step for each file with settings to be changed. For a migration, check the advanced settings in the `cray_sysconfig` translation table (in step 4 on page 294) for what might have been changed on the CLE 5.2 / SMW 7.2 system.

In the worksheet, copy the six lines below `# ** EXAMPLE 'sysconfig_files' VALUE` (with current defaults) `**` and paste one set for each external Lustre server below the line `# NOTE: Place additional 'sysconfig_files' setting entries here, if desired.`

```
# ** EXAMPLE 'sysconfig_files' VALUE (with current defaults) **
#  cray_sysconfig.settings.sysconfig_files.data.name.sample_key_a: null  <-- setting a multival key
#  cray_sysconfig.settings.sysconfig_files.data.sample_key_a.file: ''
#  cray_sysconfig.settings.sysconfig_files.data.sample_key_a.scope:
#  - service
#  - compute
#  cray_sysconfig.settings.sysconfig_files.data.sample_key_a.key_values: []
```

Uncomment the lines, replace `sample_key_a` in all lines with an identifier for the file to be changed (sitekey in the example below), and remove the `<-- setting a multival key` text at the end of the first line

(note that the null value is required; do not remove or change it). Finally, modify the values as needed for this site.

There are two list settings in the `sysconfig_files` setting: `scope` and `key_values`. To enter a list, add each list item on a separate line prefixed by a hyphen and space (-). If the list was initially set to `[]`, an empty list, remove the brackets before adding list items.

- For the `scope` list setting, enter a list of target node types (service, compute) and/or cnames (NOT node groups).
- For the `key_values` list setting, enter a list of key=value pairs.

The following example uses `sitekey` to identify the `/etc/sysconfig/filename` file and change the value of its `MYVAR` variable to `newsetting` for all service and compute nodes.

```
# NOTE: Place additional 'sysconfig_files' setting entries here, if desired.
cray_sysconfig.settings.sysconfig_files.data.name.sitekey: null
cray_sysconfig.settings.sysconfig_files.data.sitekey.file: filename
cray_sysconfig.settings.sysconfig_files.data.sitekey.scope:
- service
- compute
cray_sysconfig.settings.sysconfig_files.data.sitekey.key_values:
- MYVAR="newsetting"
```

4. Migrate other configuration settings, as needed.

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

NOTE: Use the CLE 6.0 / SMW 8.0 default values for `cray_sysconfig.settings.sysconfig_files.data.nfs_service` instead of the CLE 5.2 / SMW 7.2 values from `/etc/sysconfig/nfs` if the CLE 6.0 / SMW 8.0 default values are larger than the CLE 5.2 / SMW 7.2 values.

Table 100. Translation Table: Variables beginning with `cray_sysconfig.settings.sysconfig_files.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>name.nfs_service</code>	null	advanced	N/A	N/A
<code>nfs_service.file</code>	nfs	advanced	N/A	N/A
<code>nfs_service.scope</code>	- service	advanced	N/A	N/A
<code>nfs_service.key_values</code>	- USE_KERNEL_NFSD_NUMBER="512" - MOUNTD_OPTIONS="--num-threads=256"	advanced	N/A	files: default sharedroot /etc/sysconfig/nfs
<code>name.cron</code>	null	advanced	N/A	N/A
<code>cron.file</code>	cron	advanced	N/A	N/A
<code>cron.scope</code>	- service - compute	advanced	N/A	N/A
<code>cron.key_values</code>	- DAILY_TIME="00:00"	advanced	N/A	files: default sharedroot

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
				/etc/sysconfig/cron (DAILY_TIME)

Table 101. Translation Table: Variables beginning with `cray_sysconfig.settings.ping.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
setuid	true	advanced	<i>probe (1)</i>	N/A
scope	- service - compute	advanced	N/A	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system: (1) <code>login# ls -l /bin/ping</code>				

6.4.2.43 Update `cray_sysenv` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The Cray System Environment service specifies values to be defined within the system environment. This procedure configures some basic settings in the `cray_sysenv` configuration worksheet.

Procedure

1. Edit `cray_sysenv_worksheet.yaml`.

```
smw# vi cray_sysenv_worksheet.yaml
```

2. Uncomment `cray_sysenv.enabled` and set it to `true`.

6.4.2.44 Update `cray_time` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray Time service configures the time zone and several advanced features, such as the minimum poll interval for NTP messages. This procedure configures the inheritance setting in the Cray Time service configuration worksheet. This service has a global template as well as a CLE template, and therefore it can be configured to inherit settings from the global config set.

Procedure

1. Edit `cray_time_worksheet.yaml`.

```
smw# vi cray_time_worksheet.yaml
```

2. Uncomment `cray_time.inherit` and set it to `true`.

This means that time settings in the global config set will be used instead of time settings in the CLE config set. See [Update `cray_time` Worksheet in Global Config Set](#) on page 135. No other settings need to be changed.

6.4.2.45 Update `cray_user_settings` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray User Settings service sets the environment modules that should be loaded automatically when a user logs in to the SMW, login node, or service nodes. The SMW modules can be extended by adding to `/etc/opt/cray/modules/Base-opts.local`.

This procedure enables the `cray_user_settings` service. The last step of this procedure provides a translation table for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_user_settings_worksheet.yaml`.

```
smw# vi cray_user_settings_worksheet.yaml
```

2. Uncomment `cray_user_settings.enabled` and set it to `true`.
3. Migrate configuration settings, as needed.

Use the translation table below for help finding the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Things to note:

- In CLE 5.2 / SMW 7.2, default module information was found on the SMW, boot node, and login node in `/etc/bash.bashrc.local` and `/etc/csh.cshrc.local`. The default set of Cray modules to be loaded is done differently in CLE 6.0 / SMW 8.0. Do not copy all of the old module names from CLE 5.2 / SMW 7.2 files to CLE 6.0 / SMW 8.0.
- After a workload manager (WLM) has been installed, add the module file for that WLM to the list of modules for the login nodes.
- If cluster compatibility mode (CCM) is used in CLE 5.2 / SMW 7.2, then add `ccm` to the list of modules for the login nodes. However, CCM requires a workload manager (WLM) to have been installed.

Table 102. Translation Table: Variables beginning with `cray_user_settings.settings.default_modules.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
login	- nodestat - sdb - alps - udreg - ugni - gni-headers - dmapp - xpmem - llm - nodehealth - system-config	advanced	N/A	files: login node sharedroot <code>/etc/bash.bashrc.local</code> <code>/etc/csh.cshrc.local</code>
service	- sysadm - nodehealth - nodestat - sdb - alps - llm - system-config	advanced	N/A	files: default sharedroot <code>/etc/bash.bashrc.local</code> <code>/etc/csh.cshrc.local</code>

As other software is installed later, it might be necessary to change the set of module files loaded by default on login and service nodes.

6.4.2.46 Update `cray_wlm_detect` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray WLM (workload manager) Detect service is a C library and command used to identify the native WLM on the system. If this service is not configured, the default ALPS will be used.

This procedure enables the `cray_wlm_detect` configuration service. The last step of this procedure provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_wlm_detect_worksheet.yaml`.

```
smw# vi cray_wlm_detect_worksheet.yaml
```

2. Uncomment `cray_wlm_detect.enabled` and set it to `true`.

3. Set the active WLM (workload manager).

This setting identifies the native WLM running on the system. For WLMs using BASIL, or to indicate no WLM, set the value to ALPS. For a native WLM, enter its name in uppercase (for example, enter SLURM for Slurm). Currently only ALPS and Slurm are supported.

```
cray_wlm_detect.settings.common.data.active_wlm: ALPS
```

For a migration, use the translation table below to find the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

Table 103. Translation Table: Variables beginning with `cray_wlm_detect.settings.common.data`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>active_wlm</code>	ALPS	advanced	<i>probe (1)</i>	N/A
Commands for probing the CLE 5.2 / SMW 7.2 system:				
(1) <code>login# cat /proc/cmdline grep active_wlm</code>				

6.4.2.47 Update `cray_wlm_trans` Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

The Cray WLM (workload manager) Trans service is a library that provides WLM-agnostic functions for common tasks such as setting node state and getting a list of jobs being run by a user. It is used primarily by node health checker.

This procedure enables the `cray_wlm_trans` configuration service. The last step of this procedure provides translation tables for migrating site-specific configuration data from CLE 5.2 / SMW 7.2 to the CLE config set for CLE 6.0 / SMW 8.0.

Procedure

1. Edit `cray_wlm_trans_worksheet.yaml`.

```
smw# vi cray_wlm_trans_worksheet.yaml
```

2. Uncomment `cray_wlm_trans.enabled` and ensure that it is set to `true`.
3. Migrate configuration settings, as needed.

For a migration, use the translation table below to find the relevant CLE 5.2 / SMW 7.2 data to be migrated to these settings. For an explanation of variable names and translation table column headings, see [Transfer Configuration Data to Configuration Worksheets](#) on page 98.

The CLE 5.2 / SMW 7.2 values of some of the general settings can be found in stanzas in the WLM Trans initialization file on the default shared root, `/etc/opt/cray/wlm_trans/wlm_trans.ini`. For example, the value to migrate to the CLE 6.0 / SMW 8.0 variables `cray_wlm_trans.settings.alps.data.state_retries` and `cray_wlm_trans.settings.alps.data.state_interval` can be found in this `wlm_trans.ini` stanza for ALPS that begins with `"[alps]"` and ends before the next item in brackets `"[slurm]"`:

```
[alps]
state_retries=20
state_interval=60
[slurm]
```

The table entries for these are the first two rows of the following translation table.

Table 104. Translation Table: Variables beginning with `cray_wlm_trans.settings`.

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
<code>alps.data.state_retries</code>	20	advanced	N/A	<code>files: default sharedroot</code> <code>/etc/opt/cray/wlm_trans/wlm_trans.ini</code> <code>([alps] state_retries)</code>

CLE 6.0 / SMW 8.0			CLE 5.2 / SMW 7.2	
Setting/Field Name	Default	Level	Probe	Files/Installer
alps.data.state_interval	60	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/wlm_trans/wlm_trans.ini ([alps] state_interval)
slurm.data.state_retries	10	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/wlm_trans/wlm_trans.ini ([slurm] state_retries)
slurm.data.state_interval	30	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/wlm_trans/wlm_trans.ini ([slurm] state_interval)
slurm.data.bindir	/opt/slurm/ default/bin	advanced	N/A	<i>files:</i> default sharedroot /etc/opt/cray/wlm_trans/wlm_trans.ini ([slurm] bindir)

6.4.2.48 Update cray_zonesort Worksheet

Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

About this task

(FOR SITES MIGRATING FROM CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0) This configuration service does not have translation tables for migrating configuration data because the settings (variables) in this service have no analog in the CLE 5.2 / SMW 7.2 software.

The zonesort_module kernel module sorts free memory on the node to improve the predictability of the MCDRAM (multi-channel dynamic random-access memory) cache performance. The Cray zone sort configuration service configures the loading of the zonesort_module kernel module on compute nodes. This procedure enables that service.

Procedure

1. Edit `cray_zonesort_worksheet.yaml`.

```
smw# vi cray_zonesort_worksheet.yaml
```

2. Uncomment `cray_zonesort.enabled` and ensure that it is set to `true`.

6.5 Load and Validate Configuration Data on the Migration SMW

After updating worksheets for the global config set and the CLE config set, this part of the migration process uses the `cfgset` command to import the configuration worksheet information into those config sets and update them. All configuration data items are then validated.

The `cfgset validate` command checks variable types and regular expressions, correlates related data items to ensure consistency, and applies validation rules. Cray provides over 30 validation rules that, if applied, ensure that certain services are enabled when services that depend on them have been enabled, that every system has configured at least one boot node and at least one SDB node, that hosts defined in `cray_net` must be on a network defined in either `cray_net` or `cray_global_net`, and many other common-sense checks. System administrators can specify whether to apply rules, and which rules to include or exclude.

Use [Migration Checklist 2.4: Load and Validate Configuration Data on the Migration SMW](#) on page 438 to track progress while performing the following procedures.

1. [Disable Pre- and Post-configuration Scripts](#) on page 301
2. [Update Global Config Set from Worksheets](#) on page 302
3. [Create New CLE Config Set from Worksheets](#) on page 302
4. [Update CLE Config Set](#) on page 303
5. [Update /etc/motd for Nodes](#) on page 304
6. [Copy Files for External Lustre Fine-grained Routing](#) on page 305
7. [Configure Files for Cray Simple Sync Service](#) on page 305
 - [Configure Simple Sync for DVS Server Nodes](#) on page 307
 - [Configure Simple Sync for TCP Wrappers](#) on page 309
 - (if using Slurm) [Configure Simple Sync for Slurm](#) on page 310
 - (as needed) [Create Node Groups and Their Simple Sync Directories](#) on page 311
8. [Validate Config Sets](#) on page 312
9. [Ensure Time Zone Setting Accessible by Cabinet and Blade Controllers](#) on page 313
10. [Continue Initial DataWarp Configuration](#) on page 314

6.5.1 Disable Pre- and Post-configuration Scripts

About this task

Because the migration SMW is not connected to the CLE hardware, the configurator must not call pre- and post-configuration scripts, some of which require HSS daemons and other CLE services to be running. Rather than add the `--no-scripts` option to each `cfgset` command in the following procedures, this step disables those scripts by setting these two environment variables, which will last for the duration of the login session.

Procedure

Disable pre- and post-configuration scripts.

```
smw# export IMPS_SKIP_PRECONFIG_SCRIPTS=1
smw# echo $IMPS_SKIP_PRECONFIG_SCRIPTS

smw# export IMPS_SKIP_POSTCONFIG_SCRIPTS=1
smw# echo $IMPS_SKIP_POSTCONFIG_SCRIPTS
```

6.5.2 Update Global Config Set from Worksheets

Prerequisites

This procedure assumes that worksheets have been obtained, copied to a work area outside of `/var/opt/cray/imps/config/sets/global/worksheets`, and modified to include site-specific configuration data.

About this task

This procedure updates the global config set from existing global configuration worksheets. Note that the worksheet path provided must be enclosed in single quotes because of the file glob used.

Procedure

Update the global config set by uploading worksheets.

```
smw# cfgset update --worksheet-path \
'/var/adm/cray/release/global_worksheet_workarea/*_worksheet.yaml' global
```

6.5.3 Create New CLE Config Set from Worksheets

Prerequisites

This procedure assumes that worksheets have been obtained, copied to a work area outside of `/var/opt/cray/imps/config/sets/CONFIG_SET_NAME/worksheets`, and modified to include site-specific configuration data.

About this task

This procedure creates a new CLE config set from existing CLE configuration worksheets. Use one or more of the following commands, depending on whether this system is partitioned or not. Note that the worksheet path provided must be enclosed in single quotes because of the file glob used. There is no need to specify the config set type because the default is type CLE.

full system Create a config set for a full (unpartitioned) system p0:

```
smw# cfgset create --worksheet-path \
'/var/adm/cray/release/p0_worksheet_workarea/*_worksheet.yaml' p0
```

partitioned Create a config set for each partition. For partition p1:

```
smw# cfgset create --worksheet-path \  
'/var/adm/cray/release/p1_worksheet_workarea/*_worksheet.yaml' p1
```

For partition p2:

```
smw# cfgset create --worksheet-path \  
'/var/adm/cray/release/p2_worksheet_workarea/*_worksheet.yaml' p2
```

Additional partitions follow the same pattern.

6.5.4 Update CLE Config Set

Prerequisites

This procedure assumes that one or more CLE config sets have been created.

About this task

This procedure uses the configurator in auto mode to check for any required or basic settings that were not configured earlier in the process. The `crayadm` and `root` passwords from the `cray_local_users` service were not configured earlier using worksheets because they must be encrypted, and it is difficult to enter encrypted values in a worksheet. Therefore, the configurator will prompt for those values now. In addition, the configurator may prompt for the value of the `flat_routes` setting or the `fgr_routes` setting or both (from the `cray_lnet` service), depending on which one is not being used for external Lustre servers or whether direct-attached Lustre (DAL) is used.

Procedure

1. Invoke `cfgset` to update the config set.

full system Update the config set for a full (unpartitioned) system p0 (in this example, the config set is named p0):

```
smw# cfgset update p0
```

partitioned Update the config set for each partition. For partition p1 (in this example, the config set is named p1):

```
smw# cfgset update p1
```

For partition p2 (in this example, the config set is named p2):

```
smw# cfgset update p2
```

Additional partitions follow the same pattern.

2. Set root and `crayadm` passwords when prompted by the configurator.

These two settings from the `cray_local_users` service are for CLE/Linux accounts. They are of type "protected," which means that they must be entered twice (the second time for confirmation) and are not displayed while being entered. The configurator will encrypt them before storing them in the config set. To

enter or change the value of a protected setting, enter +, then enter and re-enter the value (in its not-yet-encrypted form) at the prompts.

- a. Set the crayadm password.

```
cray_local_users.settings.users.data.crayadm.crypt
[+=modify, ?=help, @=less] $ +
Modify crypt (Ctrl-d to cancel, <cr> to set) $
Re-enter value for crypt (Ctrl-d to cancel, <cr> to set) $
```

- b. Set the root password.

```
cray_local_users.settings.users.data.root.crypt
[+=modify, ?=help, @=less] $ +
Modify crypt (Ctrl-d to cancel, <cr> to set) $
Re-enter value for crypt (Ctrl-d to cancel, <cr> to set) $
```

- c. Set the "users" entries.

```
cray_local_users.settings.users
[<cr>=set N entries, ?=help, @=less] $ <cr>
```

Not prompted for all of these? If the configurator did not prompt for one or more of these settings, wait until `cfgset` finishes, then run `cfgset` in interactive mode (example shows command for config set `p0`), and select and set these settings from the `cray_local_users` service.

```
smw# cfgset update -m interactive -s cray_local_users p0
```

For more information about using the configurator, see *XC™ Series Configurator User Guide (S-2560)*.

3. Enter values for any other settings presented by the configurator.

If no more settings are presented, it means that all required and basic settings have been set.

When the configurator is done, it displays a message indicating the file name of the changelog file for this configuration session. The changelog is written to a file in the `/var/opt/cray/imps/config/sets/p0/changelog` directory (for a CLE config set named `p0`).

6.5.5 Update `/etc/motd` for Nodes

About this task

The standard `/etc/motd` on CLE nodes has this information.

```
Identity of node
Compute or service node
Boot image
Size of boot image
CLE release and build
Core and memory info
```

To append a custom message to the standard message of the day for all nodes, edit the `/etc/motd` file as shown in the example, which uses the config set common role to distribute the `/etc/motd` file to all nodes.

Procedure

1. Create the `files/roles/common/etc` path below the config set directory.

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

2. Edit the message of the day to append the custom message.

```
smw# vi files/roles/common/etc/motd
```

6.5.6 Copy Files for External Lustre Fine-grained Routing

Prerequisites

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

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

About this task

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

Procedure

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

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

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

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

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

```
cray_lnet.settings.fgr_routes.data.sonexion.ip2nets_file
cray_lnet.settings.fgr_routes.data.sonexion.routes_file
```

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

```
smw# cd directory_containing_ip2nets.conf_and_routes.conf
```

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

6.5.7 Configure Files for Cray Simple Sync Service

About this task

Cray Simple Sync provides a generic mechanism to automatically distribute files to targeted locations on the system. This mechanism can be used to override or change default system behavior through the contents of the distributed files. When enabled, the Simple Sync service is executed on all CLE nodes at boot time and whenever the administrator executes `/etc/init.d/cray-ansible start` on a CLE node. When Simple Sync is executed, files placed in the following directory structure are copied to the root file system (/) on the target nodes.

Action Needed for Migration

When the config set was created on the migration SMW, no pre- or post-configuration callback scripts were called. As a consequence, no node group directories have been created in the Simple Sync directory structure. Use this command to create the node groups directories in the Simple Sync directory structure now.

```
smw# /opt/cray/imps_config/system-config/default/configurator/callbacks/post/\
simple_sync_create_dirs.sh /var/opt/cray/imps/config/sets/p0 cle
```

If more node groups need to be added later with the intent of using those node groups with Simple Sync, use the procedure in [Create Node Groups and Their Simple Sync Directories](#) on page 311.

About the Simple Sync Directory Structure

The Simple Sync directory structure has this root:

```
smw:/var/opt/cray/imps/config/sets/<config_set>/files/simple_sync/
```

Below that root are the directories listed on the left:

Files placed here	are copied to
<code>./common/files/</code>	all nodes
<code>./platform/[compute, service]/files/</code>	all compute or service nodes
<code>./hardwareid/<hardwareid>/files/</code>	nodes with matching hardware ID, which is the cname of a CLE node or the output of the <code>hostid</code> command (e.g., <code>1eac0b0c</code>) on other nodes
<code>./hostname/<hostname>/files/</code>	nodes with matching host name (use this for eLogin nodes ONLY)
<code>./nodegroups/<node_group_name>/files/</code>	nodes in the matching node group

NOTE: The directory structure for a particular hardware ID or host name (everything below `./hardwareid/` and `./hostname/`) must be created manually as needed. This is unnecessary for node groups because their associated directories are created automatically by post-configuration callback scripts when the config set is created or updated using `cfgset`.

Anything (directory structure and files) placed below `./files/` in the Simple Sync directory structure on the SMW is replicated on the target node starting at root (/). For example, this path on the SMW

```
/var/opt/cray/imps/config/sets/p0/files/simple_sync/common/files/etc/myapplication.conf
```

will place the `myapplication.conf` file on all nodes in this directory:

```
/etc/myapplication.conf
```

Note that the ownership and permissions of files in the config set are preserved in the copies made to nodes. For more information and use cases, see [About Simple Sync](#) on page 419.

6.5.7.1 Configure Simple Sync for DVS Server Nodes

About this task

The DVS servers for this system might mount the remote file system via NFS or GPFS directly in `/etc/fstab`, or they might use the automounter to mount it upon request. This procedure shows how to place the requisite files from either method in the correct Simple Sync directory structure on the migration SMW so that they will be pushed to targeted nodes when Simple Sync is run (after the migration SMW is connected to XC system hardware later in the migration process). Simple Sync is one of the Ansible plays run when `cray-ansible` is invoked at boot time for the nodes or when invoked after a boot by the system administrator on a node.

This procedure uses the example client mount on CSS that was introduced in [Update cray_dvs Worksheet](#) on page 165.

Procedure

1. (If `/etc/fstab` used) Configure Simple Sync for DVS server nodes with `/etc/fstab`.

If the CLE 5.2 / SMW 7.2 system used `/etc/fstab` for mounting files, then the `/etc/fstab` files must be put into the Simple Sync directory structure in a CLE config set so that it will be distributed to the nodes in the node group. This example shows an NFS mount, but we know some customers do a GPFS mount. Mounting either on the login node and the "DVS" node which will project this external non-Lustre filesystem to the compute nodes is similar in use of Simple Sync to send the `/etc/fstab` files.

- a. Check the `fstab` entry on the DVS server node.

This example shows how to check on one of the DVS server nodes (node `c0-0c0s4n1` or `nid00017`) for `/cray/css`. That mount is an NFS readonly (`ro`) mount from a hostname called `csse12`. (Recall the example setting in [Update cray_dvs Worksheet](#) on page 165, which set `cray_dvs.settings.client_mount.data.CSS.readonly` to `true`.)

```
nid00017# grep /cray/css /etc/fstab
csse12:/css          /cray/css   nfs        tcp,ro    0 0
```

- b. Create a node group for the DVS servers.

To define the `css_dvs_servers` node group in the `cray_node_groups` configuration service, use the procedure in [Create Node Groups and Their Simple Sync Directories](#) on page 311. That procedure also creates the Simple Sync directory for that node group.

- c. Add `/etc/fstab` to the Simple Sync directory structure on the migration SMW.

Later in the process, Simple Sync will push the `/etc/fstab` file out to nodes targeted by the `css_dvs_servers` node group. This example places the `/etc/fstab` file (that mounts `/cray/css` from `csse12:/css`) in the `css_dvs_servers` node group directory in the 'p0' CLE config set Simple Sync directory structure.

```
smw# cd /var/opt/cray/imps/config/sets/p0/files/simple_sync
smw# mkdir -p nodegroups/css_dvs_servers/files/etc/
```

```
smw# echo "csse12:/css          /cray/css  nfs      tcp,ro  0 0" \
> nodegroups/css_dvs_servers/files/etc/fstab

smw# cat nodegroups/css_dvs_servers/files/etc/fstab
csse12:/css          /cray/css  nfs      tcp,ro  0 0
```

The login nodes may also be mounting this same file system, so check the `/etc/fstab` file of the login node, and if it is mounting the same file system, then add this to Simple Sync for the `login_nodes` node group.

```
smw# cd /var/opt/cray/imps/config/sets/p0/files/simple_sync
smw# mkdir -p nodegroups/login_nodes/files/etc/

smw# echo "csse12:/css          /cray/css  nfs      tcp,ro  0 0" \
> nodegroups/login_nodes/files/etc/fstab

smw# cat nodegroups/login_nodes/files/etc/fstab
csse12:/css          /cray/css  nfs      tcp,ro  0 0
```

2. (If automounter used) Configure Simple Sync for DVS server nodes with automounter.

If the CLE 5.2 / SMW 7.2 system used automounter for mounting files, then the files for automounting must be put into the Simple Sync directory structure in a CLE config set so that it will be distributed to the target nodes.

a. Check the automount map on the DVS server node.

There are many different ways to configure the automounter, but look for files in these locations on CLE 5.2 / SMW 7.2 DVS server nodes. The same automount maps may also be in use on the login nodes.

```
/etc/auto.master
/etc/auto.*
```

The master auto map in `/etc/auto.master` on SLES 11 SP3 shows any mount points, but may also include other files that contain maps.

b. Prepare the `/etc/auto.master.d/auto.css` file for CLE 6.0 / SMW 8.0.

In CLE 5.2 / SMW 7.2 with SLES 11 SP3, there may be an automount entry in `/etc/auto.master` like this.

```
/cray/css /etc/auto.css --timeout 600
```

In CLE 6.0 / SMW 8.0 with SLES 12, move this single line entry to a file in `/etc/auto.master.d`.

```
node# cat /etc/auto.master.d/css.autofs
/cray/css /etc/auto.css --timeout 600
```

Then copy the `/etc/auto.css` file from the CLE 5.2 / SMW 7.2 system to the CLE 6.0 / SMW 8.0 migration SMW without any changes.

c. Add automount files to Simple Sync directory structure.

Instead of using the `login_nodes` node group and the `css_dvs_servers` node group, it may make sense to create a new node group called `automount_nodes`, assign the login nodes and the CSS DVS servers to it, and use that new node group as the target to which Simple Sync will distribute the automount files.

If defining an `automount_nodes` node group, that must be done before continuing with this step. Perform the procedure in [Create Node Groups and Their Simple Sync Directories](#) on page 311 and then return here.

```
smw# cd /var/opt/cray/imps/config/sets/p0/files/simple_sync
smw# mkdir -p nodegroups/automount_nodes/files/etc/
smw# mkdir -p nodegroups/automount_nodes/files/etc/auto.master.d
```

Copy the `/etc/auto.master.d/auto.css` file and the `/etc/auto.css` from the CLE 5.2 / SMW 7.2 system to the Simple Sync directory just created on the CLE 6.0 / SMW 8.0 migration SMW.

```
smw# cd /path/to/automount/files

smw# cp -p auto.css /var/opt/cray/imps/config/sets/p0\
/files/simple_sync/nodegroups/automount_nodes/files/etc

smw# cp -p auto.master.d/css.autofs /var/opt/cray/imps/config/sets/p0\
/files/simple_sync/nodegroups/automount_nodes/files/etc/auto.master.d
```

- d. Confirm that the automount files have been put into the proper location for the "automount_nodes" node group.

```
smw# ls -lR /var/opt/cray/imps/config/sets/p0/files/\
simple_sync/nodegroups/automount_nodes/files
/var/opt/cray/imps/config/sets/p0/files/simple_sync/nodegroups/
automount_nodes/files:
total 0
drwxr-xr-x 1 root root 42 Dec 14 11:34 etc

/var/opt/cray/imps/config/sets/p0/files/simple_sync/nodegroups/
automount_nodes/files/etc:
total 4
-rw-r--r-- 1 root root 1630 Sep 27 12:46 auto.css
drwxr-xr-x 1 root root 16 Dec 14 11:34 auto.master.d

/var/opt/cray/imps/config/sets/p0/files/simple_sync/nodegroups/
automount_nodes/files/etc/auto.master.d:
total 4
-rw-r--r-- 1 root root 38 Dec 14 11:34 css.autofs
```

6.5.7.2 Configure Simple Sync for TCP Wrappers

About this task

A TCP wrapper is a simple security tool that monitors and controls network traffic through the use of an `/etc/hosts.allow` file and an `/etc/hosts.deny` file. If TCP wrappers (`tcpd`) are configured in the CLE 5.2 / SMW 7.2 system, then the `/etc/hosts.allow` and `/etc/hosts.deny` files should be prepared on the CLE 6.0 / SMW 8.0 migration SMW for distribution to CLE nodes using Simple Sync.

SMW files from an SMW running CLE 5.2 / SMW 7.2 can be placed in the same location on an SMW running CLE 6.0 / SMW 8.0. However, for CLE nodes on the CLE 6.0 / SMW system, Simple Sync can be used to distribute these files to node groups, which is similar to using classes from `/etc/node_classes` in CLE 5.2 / SMW 7.2 to control where files are distributed.

Procedure

1. Copy the following CLE 5.2 / SMW 7.2 files to the equivalent location on the migration SMW.

```
/etc/hosts.allow
/etc/hosts.deny
```

The format of these files is unchanged in CLE 6.0 / SMW 8.0. See the man pages for `tcpd(8)` and `hosts_access(5)`. The default contents are identical between SLES 11 SP3 and SLES 12. The files used on the CLE 5.2 / SMW 7.2 system for SMW, bootroot, and sharedroot can be copied to the equivalent location on the CLE 6.0 / SMW 8.0 migration SMW.

2. Add `/etc/hosts.allow` and `/etc/hosts.deny` to the Simple Sync directory structure.

If this site uses a DVS class in the CLE 5.2 / SMW 7.2 system, and if this site has a CLE 6.0 / SMW 8.0 node group called "css_dvs_servers" on the migration SMW (as described in [Configure Simple Sync for DVS Server Nodes](#) on page 307), then the same node group can be used by Simple Sync to distribute the `/etc/hosts.allow` and `/etc/hosts.deny` files to CLE nodes after the migration SMW is connected to XC system hardware (later in the migration process).

This example shows the CLE 5.2 / SMW 7.2 `/etc/hosts.allow` and `/etc/hosts.deny` files from the DVS class being in `/var/adm/cray/migration_data/dvs_etc_hosts.*`. This `hosts.deny` file prevented SSH access from external hosts to the DVS node with this line in `/etc/hosts.deny`, while still allowing ssh access from hosts on the high-speed network (HSN).

```
sshd : ALL EXCEPT LOCAL
```

```
smw# cd /var/opt/cray/imps/config/sets/p0/files/simple_sync/nodegroups
smw# mkdir -p css_dvs_servers/files/etc

smw# cp -p /var/adm/cray/migration_data/dvs_etc_hosts.allow css_dvs_servers/
files/etc
smw# cp -p /var/adm/cray/migration_data/dvs_etc_hosts.deny css_dvs_servers/
files/etc
```

6.5.7.3 Configure Simple Sync for Slurm

About this task

If this system uses Slurm as the workload manager (WLM), this procedure saves `/etc/opt/slurm` files from persistent storage on the Slurm server or MOM nodes (most but not all sites use the SDB node as the WLM server) and places them in the Simple Sync directory structure on the migration SMW. They will be pushed to targeted nodes when Simple Sync is run after the migration SMW is connected to XC system hardware (later in the migration process).

Procedure

1. Create the necessary Simple Sync directory on the migration SMW.

```
smw# cd /var/opt/cray/imps/config/sets/p0/files/simple_sync/common/files
smw# mkdir -p etc/opt/slurm
```

2. Copy the `etc/opt/slurm` files from the Slurm server or MOM nodes on the CLE 5.2 / SMW 7.2 system to the Simple Sync directory structure on the migration SMW.

6.5.7.4 Create Node Groups and Their Simple Sync Directories

Prerequisites

This procedure assumes a situation in which configurator pre- and post-configuration callback scripts must not be run (for example, when the SMW is not connected to XC hardware).

About this task

Typically, when a config set is created or updated, pre- and post-configuration callback scripts are run automatically. One of those scripts creates a node group directory in the Simple Sync directory structure for each node group defined in the config set. In some phases of migration and in certain other circumstances, running those scripts must be suppressed using the `--no-scripts` option or environment variables. In such cases, when a node group is created, its associated Simple Sync directory is not created automatically. That is a problem if the node group is intended for use with Simple Sync.

This procedure describes how to create a new node group interactively and run the specific callback script that creates the associated Simple Sync directory.

Procedure

1. Update the `cray_node_groups` configuration service.

```
smw# cfgset update --no-scripts -m interactive -s cray_node_groups p0
```

2. Select the groups setting.

The groups setting is the first and only setting in this configuration service, so enter `1` at the configurator prompt, then enter `c` to configure that setting.

```
Cray Node Groups Configuration Service Menu [default: save & exit - Q] $ 1
...
Cray Node Groups Configuration Service Menu [default: configure - C] $ c
```

3. Add a new node group entry.

Repeat this step for each new node group.

```
cray_node_groups.settings.groups
[<cr>=set 8 entries, +=add an entry, ?=help, @=less] $ +
```

```
cray_node_groups.settings.groups.data.group_name
[<cr>=set '', <new value>, ?=help, @=less] $ automount_nodes
```

```
cray_node_groups.settings.groups.data.automount_nodes.description
[<cr>=set '', <new value>, ?=help, @=less] $ Simple Sync will distribute
automount files to these nodes.
```

```
cray_node_groups.settings.groups.data.automount_nodes.members
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add members [Ctrl-d to exit] $ c0-0c0s1n1
Add members [Ctrl-d to exit] $ c0-0c0s2n2
```

```
cray_node_groups.settings.groups.data.automount_nodes.members
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <cr>
```

4. When finished adding node groups, set the node groups entries, then save and exit the configurator.

```
cray_node_groups.settings.groups
[<cr>=set 9 entries, +=add an entry, ?=help, @=less] $ <cr>
```

```
Cray Node Groups Configuration Service Menu [default: save & exit - Q] $ Q
```

5. Run the callback script that creates node groups directories in the Simple Sync directory structure.

```
smw# /opt/cray/imps_config/system-config/default/configurator/callbacks/post/\
simple_sync_create_dirs.sh /var/opt/cray/imps/config/sets/p0 cle
```

6.5.8 Validate Config Sets

About this task

It is important to validate any config set that has been modified, because there is currently no mechanism to prevent the system from trying to use an invalid config set. Validation is useful for determining if the config set is minimally viable for use with the system it is intended to configure.

IMPORTANT: Validation ensures that a config set passes all rules stored on the system. A validated config set does not necessarily equate to a config set with configuration data that will result in a properly configured system.

When validating a config set, the configurator checks the following:

- Config set has the proper directory structure and permissions.
- All configuration templates have correct YAML syntax.
- All configuration templates adhere to the configurator schema.
- All fields of type `lookup` reference values and settings that exist in the available configuration services.
- All level `required` fields in enabled services are configured (i.e., their state is `set`).
- Pre-configuration and post-configuration callback scripts ran successfully during the latest config set update.
- `cfgset validate` has run all validation rules installed on the system.

For more information on how `lookup` fields work, see the "Advanced: Lookup" section in "Configurator Data Types and How to Set Them," which is in *XC™ Series Configurator User Guide (S-2560)*. For more information about validation rules, see "Validate a Config Set and List Validation Rules," also in that publication.

Procedure

Validate the CLE and global config sets.

For a migration, the configurator will produce the following WARNING messages because no pre- or post-configuration scripts have been run in this phase of the migration process. Ignore these two messages. Do NOT update the config set without the `--no-scripts` option, as instructed in the warning message, because the migration SMW is not connected to XC system hardware. The config set will be updated and re-validated later in the migration process.

```
WARNING - The ConfigSet pre-/post-configuration scripts were skipped or failed
in the previous update/create operation.
WARNING - Update the ConfigSet without the '--no-scripts' option to run the
scripts and then re-validate.
Error: 1 of 1 config sets failed to validate.
```

These example commands use CLE config sets named p0, p1, and p2. Substitute the correct config set names for this site.

full system Validate the CLE and global config sets for a full, unpartitioned system:

```
smw# cfigset validate global
smw# cfigset validate p0
```

partitioned Validate the CLE and global config sets for a partitioned system, with partitions p1 and p2:

```
smw# cfigset validate global
smw# cfigset validate p1
smw# cfigset validate p2
```

6.5.9 Ensure Time Zone Setting Accessible by Cabinet and Blade Controllers

Prerequisites

This procedure assumes that the global config set on the physical migration SMW has been updated and validated.

About this task

This procedure runs `cray-ansible` to apply global config set changes on the physical migration SMW, then places time configuration data in a place where the cabinet and blade controllers can access it. This ensures that any changes to the time zone are seen by the cabinet and blade controllers when they are rebooted.

Procedure

1. Run Ansible plays on the SMW.

After the global config set has been updated, reapply any Ansible plays that consume global config set data.

NOTE: (SMW HA only) Both SMWs require this command. The procedure to install and configure the second SMW includes this command.

Logs from running Ansible plays, such as `cray-ansible`, are stored on the SMW in `/var/opt/cray/log/ansible`.

```
smw# /etc/init.d/cray-ansible start
```

CHECK TIME SETTINGS

2. Check for external NTP servers.

Check that external NTP servers have been set as desired in the global config set.

NOTE: (SMW HA only) Both SMWs require this command. The procedure to install and configure the second SMW includes this command.

```
smw# grep server /etc/ntp.conf
server ntpserver1 minpoll 4 iburst
server ntpserver2 minpoll 4 iburst
```

3. Put the SMW time zone setting where the cabinet and blade controllers can access it.

This SMW time zone setting will be applied to the cabinet and blade controllers when they are rebooted later in the process.

NOTE: (SMW HA only) Both SMWs require this command. The procedure to install and configure the second SMW includes this command.

```
smw# cp -p /etc/localtime /opt/tftpboot/localtime
```

6.5.10 Continue Initial DataWarp Configuration

Prerequisites

This procedure assumes the following:

- This site uses DataWarp.
- This site has a Cray XC series system running CLE 6.0.UP03 with one or more service nodes with SSD hardware.
- This is a migration from CLE 5.2 to CLE 6.0.
- Nodes with SSD hardware have been identified by `cname`.

Additional requirement: a parallel file system (PFS) must be mounted in the same location on all compute nodes as well as all service nodes included in `managed_nodes_groups` within this procedure. In other words, the mount points must look the same on compute and SSD-endowed service nodes. More than one PFS is allowed. This requirement can be implemented before or after the installation of DataWarp.

About this task

If this site does not use DataWarp in the CLE 5.2 / SMW 7.2 system but plans to enable it in the CLE 6.0 / SMW 8.0 system, complete the migration without DataWarp enabled (skip this procedure and proceed to [Update Non-](#)

[config-set Configuration Files on the Migration SMW](#) on page 319), then follow the instructions in *XC™ Series DataWarp™ Installation and Administration Guide (S-2564)* to install and configure it later.

This procedure verifies that the node groups needed for DataWarp were created earlier in the process. It and the three procedures that follow complete the initial configuration of DataWarp begun earlier in [Update cray_dws Worksheet](#) on page 171. The rest of DataWarp installation and configuration must wait until after the first boot of the XC system with CLE 6.0 / SMW 8.0, and it is addressed later in the migration process.

Procedure

1. Determine the node group names used for the DataWarp-managed nodes and API gateway nodes.

```
smw# cfgset search -s cray_dws -t groups p0
INFO - Checking services for valid YAML syntax
INFO - Checking services for schema compliance
# 2 matches for 'groups' from cray_dws_config.yaml
#-----
-
cray_dws.settings.service.data.managed_nodes_groups: datawarp_nodes
cray_dws.settings.service.data.api_gateway_nodes_groups: login_nodes
```

2. Verify that the datawarp_nodes node group, which is assigned to the managed_nodes_groups variable (from output of step 1) has the correct cnames of SSD-endowed service nodes.

```
smw# cfgset search -s cray_node_groups -t datawarp_nodes p0
INFO - Checking services for valid YAML syntax
INFO - Checking services for schema compliance
# 2 matches for 'datawarp_nodes' from cray_node_groups_config.yaml
#-----
-
cray_node_groups.settings.groups.data.datawarp_nodes.description: Node group
that contains all DataWarp managed nodes with SSDs
cray_node_groups.settings.groups.data.datawarp_nodes.members: c0-0c0s5n1,
c0-0c1s6n1, c0-1c2s4n0
```

3. Verify that the login_nodes node group, which is assigned to the api_gateway_nodes_groups variable (from output of step 1) has the correct cnames of DataWarp API gateway nodes.

```
smw# cfgset search -s cray_node_groups -t login_nodes p0
INFO - Checking services for valid YAML syntax
INFO - Checking services for schema compliance
# 2 matches for 'login_nodes' from cray_node_groups_config.yaml
#-----
-
cray_node_groups.settings.groups.data.login_nodes.description: Default node
group which contains the login nodes for the configured system.
cray_node_groups.settings.groups.data.login_nodes.members: c0-0c0s3n2,
c0-0c0s1n1
```

4. If either of the node groups is not defined as expected, return to [Update cray_dws Worksheet](#) on page 171 for instructions on defining those node groups.

To complete initial DataWarp configuration perform the next three procedures.

6.5.10.1 Ensure that `cray_ipforward`, `cray_lnet`, `cray_munge`, and `cray_dw_wlm` are Enabled

Prerequisites

This procedure assumes DataWarp is enabled.

About this task

The following configuration services must be enabled for DataWarp to function in a CLE 6.0 / SMW 8.0 system. This procedure checks to make sure they are.

- `cray_ipforward`
- `cray_lnet`
- `cray_munge`
- `cray_dw_wlm`

Procedure

1. Ensure that the `cray_lnet`, `cray_munge`, and `cray_dw_wlm` services are enabled.

```
smw# cfgset search --service-status p0 | grep lnet.enabled
cray_lnet.enabled: True

smw# cfgset search --service-status p0 | grep munge.enabled
cray_munge.enabled: True

smw# cfgset search --service-status p0 | grep dw_wlm.enabled
cray_dw_wlm.enabled: True
```

If any of these services is disabled (set to false), edit the corresponding configuration worksheet and change that setting to true.

2. Ensure that the `cray_ipforward` service is enabled.

Because the `cray_ipforward` service has both a global and CLE template, the CLE template can be configured to inherit settings from the global template. Therefore, to check the status of `cray_ipforward`, it is first necessary to determine which template governs the configuration of that service: global or CLE.

- a. Check to see whether `cray_ipforward` inherits from global config set.

Search the `cray_ipforward` worksheet in the CLE config set (p0 in example) to find the `inherit` setting.

In this example, that setting is false, so `cray_ipforward` does not inherit from global, and the CLE config set worksheet is the correct one to check to find the status of the service.

```
smw# grep inherit: /var/opt/cray/imps/config/sets/p0/worksheets/\
cray_ipforward_worksheet.yaml
cray_ipforward.inherit: False
```

If the setting had been `cray_ipforward.inherit: True`, it would mean that `cray_ipforward` inherits from global, and the global config set worksheet is the correct one to check.

- b. (If `cray_ipforward.inherit` set to false) Verify that `cray_ipforward` is enabled in the CLE config set (p0 in example).

```
smw# cfgset search --service-status p0 | grep ipforward.enabled
cray_ipforward.enabled: true
```

In the above example, `cray_ipforward` is enabled. If it is not enabled, edit `cray_ipforward_worksheet.yaml` in the CLE config set and change that setting to true.

- c. (If `cray_ipforward.inherit` set to true) Verify that `cray_ipforward` is enabled in the global config set.

```
smw# cfgset search --service-status global | grep ipforward.enabled
cray_ipforward.enabled: true
```

In the above example, `cray_ipforward` is enabled. If it is not enabled, edit `cray_ipforward_worksheet.yaml` in the global config set and change that setting to true.

6.5.10.2 Set Up DataWarp Persistent Storage

Procedure

1. Edit `cray_persistent_data_worksheet.yaml`.

```
smw# vi cray_persistent_data_worksheet.yaml
```

2. Ensure that `cray_persistent_data` is enabled.

If `cray_persistent_data.enabled` is commented out, uncomment it and ensure that it is set to true.

3. Configure persistent storage for DataWarp.

In the worksheet, copy the five lines below `# ** EXAMPLE 'mounts' VALUE` (with current defaults) `**` and paste them below `# NOTE: Place additional 'mounts' setting entries here, if desired.`

```
# ** EXAMPLE 'mounts' VALUE (with current defaults) **
# cray_persistent_data.settings.mounts.data.mount_point.sample_key_a: null <-- setting a multival key
# cray_persistent_data.settings.mounts.data.sample_key_a.alt_storage_path: ''
# cray_persistent_data.settings.mounts.data.sample_key_a.options: ''
# cray_persistent_data.settings.mounts.data.sample_key_a.ancestor_def_perms: '0771'
# cray_persistent_data.settings.mounts.data.sample_key_a.client_groups: []
```

Uncomment the lines, replace `sample_key_a` with `/var/opt/cray/dws` in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Change setting values as indicated in the example below.

```
# NOTE: Place additional 'mounts' setting entries here, if desired.
cray_persistent_data.settings.mounts.data.mount_point./var/opt/cray/dws: null
cray_persistent_data.settings.mounts.data./var/opt/cray/dws.alt_storage_path: ''
cray_persistent_data.settings.mounts.data./var/opt/cray/dws.options: rw
cray_persistent_data.settings.mounts.data./var/opt/cray/dws.ancestor_def_perms: '0755'
cray_persistent_data.settings.mounts.data./var/opt/cray/dws.client_groups:
- service_nodes

#***** END Service Setting: mounts *****
```

6.5.10.3 Update and Validate Global and CLE Config Sets

Procedure

1. Update the CLE and global config sets.

Correct any issues before proceeding.

full system Update the CLE and global config sets for a full, unpartitioned system:

```
smw# cfgset update p0
smw# cfgset update global
```

partitioned Update the CLE and global config sets for a partitioned system, with partitions p1 and p2:

```
smw# cfgset update p1
smw# cfgset update p2

smw# cfgset update global
```

2. Validate the CLE and global config sets.

Correct any discrepancies before proceeding, with this exception: the configurator will produce the following WARNING messages because no pre- or post-configuration scripts have been run in this phase of the migration process. Ignore these two messages. Do NOT update the config set without the `--no-scripts` option, as instructed in the warning message, because the migration SMW is not connected to XC system hardware. The config set will be updated and re-validated later in the migration process.

```
WARNING - The ConfigSet pre-/post-configuration scripts were skipped or failed
in the previous update/create operation.
WARNING - Update the ConfigSet without the '--no-scripts' option to run the
scripts and then re-validate.
Error: 1 of 1 config sets failed to validate.
```

These example commands use CLE config sets named p0, p1, and p2. Substitute the correct config set names for this site.

full system Validate the CLE and global config sets for a full, unpartitioned system:

```
smw# cfgset validate p0
smw# cfgset validate global
```

partitioned Validate the CLE and global config sets for a partitioned system, with partitions p1 and p2:

```
smw# cfgset validate p1
smw# cfgset validate p2

smw# cfgset validate global
```

Initial DataWarp configuration is complete. Proceed to [Update Non-config-set Configuration Files on the Migration SMW](#) on page 319.

6.6 Update Non-config-set Configuration Files on the Migration SMW

There are a few configuration files used in CLE 6.0 / SMW 8.0 that are not controlled by data in a config set. Use [Migration Checklist 2.5-P: Update Non-config-set Configuration Files on the Migration SMW](#) on page 439 to track progress while performing the following steps/procedures.

1. Check for files on the CLE 5.2 / SMW 7.2 system that are the same as non-config-set files in CLE 6.0 / SMW 8.0. Compare them to the CLE 6.0 / SMW 8.0 version of the files, and move settings to the CLE 6.0 / SMW 8.0 files on the migration SMW.
2. Configure the HSS database when XC hardware is not connected.
This part of the migration process creates the hardware test configuration in the HSS Data Store, marks service nodes, and indicates which components are disabled.
 - a. [Create Hardware Test Configuration with xtdiscover](#) on page 319
 - b. [Assign Service Nodes Manually and Disable Components](#) on page 320
3. [Check for Site Modifications in SMW xtrim.conf](#) on page 321
4. [Check for Site Modifications to SEDC Files](#) on page 322
5. [Check for Site Modifications to SMW Firewall and IP Tables](#) on page 322

6.6.1 Create Hardware Test Configuration with xtdiscover

Prerequisites

This procedure assumes that the following information has been gathered. Enter this information in response to system prompts when performing this procedure. The probe output from running `xtcli part_cfg show` on the CLE 5.2 / SMW 7.2 SMW has the needed information.

Information needed	Default value
maximum X cabinet size (columns)	There is no default value. Find the X and Y cabinet sizes and the network topology class from Site-dependent Configuration Values in Configuration Values on page 24.
maximum Y cabinet size (rows)	No default value. See above.
network topology class	0 or 2 for Cray XC Series liquid-cooled systems, 0 for Cray XC Series air-cooled systems (XC30-AC, XC40-AC)
boot node name	c0-0c0s0n1
sdb node name	c0-0c0s1n1

About this task

Because the migration SMW is not connected to XC system hardware, the normal process of running `xtdiscover --bootstrap` and then `xtdiscover` cannot be done. For the `cnode` command to assign parameters to nodes in a NIMS map, the HSS database must be initialized with data.

This procedure uses the `xtdiscover --testconfig` command to collect some basic information that will be used to change the hardware discovery process so that it skips probing the XC cabinet controllers. If boot node failover or SDB node failover will be enabled, then when `xtdiscover` asks for the boot node or the SDB node, instead of entering a single node, enter a pair of nodes with a comma between them, for example "c0-0c0s0n1,c0-2c0s0n1." For more detailed information, see the `xtdiscover(8)` man page.

Procedure

1. Run `xtdiscover` in `testconfig` mode.

```
smw# xtdiscover --testconfig
```

The system prompts the user to enter the information gathered as a prerequisite to this procedure. Here are the prompts from `xtdiscover`. Set the X and Y cabinet sizes to match the CLE 5.2 / SMW 7.2 system. Set the boot node(s) and SDB node(s) to the correct ones for this XC system. The probe output from running `xtcli part_cfg show` on the CLE 5.2 / SMW 7.2 SMW has the needed information.

```
Please enter 'c' to continue, or 'a' or 'q' to abort [c]:
```

```
Enter maximum X cabinet size (columns) [1-64, last: 1], q=quit:
Enter maximum Y cabinet size (rows) [1-32, last: 1], q=quit:
```

```
Enter your system's network topology class [last: 2]:
```

```
Enter the boot node name [c0-0c0s0n1]:
Enter the SDB node name [c0-0c0s1n1]:
```

(Virtual migration SMW only) If the migration SMW is a virtual SMW and it reports any errors about `eth1` not being found, such as shown here, ignore them.

```
Adding hosts and routes for 12 cabinets...Cannot find device "eth1"
RTNETLINK answers: Network is unreachable
```

2. Confirm that the expected settings have been added to the HSS data store.

```
smw# xtcli part_cfg show
smw# xtcli status s0
```

6.6.2 Assign Service Nodes Manually and Disable Components

Prerequisites

This procedure assumes that the hardware test configuration has been created on the migration SMW using `xtdiscover` in `testconfig` mode.

About this task

This procedure manually sets the service nodes for the migration SMW to match what is currently on the CLE 5.2 / SMW 7.2 system. Although this might seem tedious to do when the XC system is not connected to the migration SMW, it will save time later when completing the switch to the CLE 6.0 / SMW 8.0 system later in the migration process. The final step disables any components that are disabled on the CLE 5.2 / SMW 7.2 SMW.

Procedure

1. Use `xtcli mark_node service` to set the service nodes on the migration SMW to match the CLE 5.2 / SMW 7.2 SMW.

This enables the `cnode` command to assign boot images to nodes of the proper type for the NIMS map.

The list of components should be a comma separated list of nodes, blades, chassis, cabinets, or partitions. Substitute the correct cnames for this site.

```
smw# xtcli mark_node service c0-0c0s0n1,c0-0s0s1n1
```

Set all of the nodes that are actually service node hardware to be service nodes. Also set any compute nodes that will be repurposed to become service nodes. These are often the tier2 nodes, but there may be others.

2. Confirm nodes were marked as service.

```
smw# xtshow_service
```

3. Disable any nodes, blades, or other components on the migration SMW to match the SMW running CLE 5.2 / SMW 7.2.

If there are any blades or other components to be disabled, disable them there. The list of components should be a comma separated list of nodes, blades, chassis, cabinets, or partitions. Substitute the correct cnames for this site.

```
smw# xtcli disable c1-1c0s0,c1-1c0s1
```

6.6.3 Check for Site Modifications in SMW `xttrim.conf`

About this task

The `xttrim` command provides a simple and configurable method to automate the compression and deletion of old log files. It is enabled to run via cron on the SMW for both CLE 5.2 / SMW 7.2 and CLE 6.0 / SMW 8.0. This procedure compares the current (CLE 5.2 / SMW 7.2) version of `xttrim.conf` and the CLE 6.0 / SMW 8.0 version of it on the migration SMW to identify site-specific differences. The site must decide whether to move any site-specific content forward to CLE 6.0 / SMW 8.0.

Procedure

1. Compare the `xttrim.conf` on the migration SMW to the version of `xttrim.conf` on the SMW running CLE 5.2 / SMW 7.2.

This example command compares the CLE 6.0 / SMW 8.0 version to the version of `xttrim.conf` extracted from the CLE 5.2 / SMW 7.2 system earlier in the migration process and placed

in `/migration_data/migration/smw` on the migration SMW. The example output shows that the CLE 6.0 / SMW 8.0 version of the file has three lines instead of one for `/var/opt/cray/log`, because this release supports separate settings for subdirectories, providing finer-grained control over log subdirectories, if desired.

```
smw# diff /etc/opt/cray/llm/xttrim.conf \
/migration_data/migration/smw/etc/opt/cray/llm/xttrim.conf
2c2
< # Copyright (c) 2012 Cray Inc. All Rights Reserved.      #
---
> # Copyright (c) 2012, 2016 Cray Inc. All Rights Reserved.  #
30c30,32
< /var/opt/cray/log 30 0
---
> /var/opt/cray/log 30 0 xtdiag controller
> /var/opt/cray/log/xtdiag 0 90
> /var/opt/cray/log/controller 2 30
```

2. If any settings were changed from the default for CLE 5.2 / SMW 7.2, decide whether to move them forward to CLE 6.0 / SMW 8.0.

There are some new settings in the CLE 6.0 / SMW 8.0 version of the `xttrim.conf` file, so do not copy the CLE 5.2 / SMW 7.2 version of that file directly to the migration SMW.

6.6.4 Check for Site Modifications to SEDC Files

In CLE 5.2 / SMW 7.2, the SEDC sensor scanning configuration may have been modified to be different than the default settings. This data was in two files that were loaded by individual blades when the `sedc_enable_default` script was called for the first time. Look in these two files for any customized settings that should be brought forward into the CLE 6.0 / SMW 8.0 versions:

- `/opt/cray/hss/default/etc/cab_json.sedc`
- `/opt/cray/hss/default/etc/blade_json.sedc`

With the CLE 6.0 / SMW 8.0 release, these files have newer hardware types available, such as the Intel® Xeon Phi™ processors (formerly code named Knights Landing or KNL). Because the CLE 6.0 / SMW 8.0 versions of those files have better defaults, it is likely they will not require customization. Seek advice from Cray field support.

6.6.5 Check for Site Modifications to SMW Firewall and IP Tables

About this task

Because this site may have customized firewall and iptables configuration on the CLE 5.2 / SMW 7.2 system, that configuration information from the SMW was saved earlier in the migration process. The firewall configuration files will be saved again just before the "Shutdown and Switch" phase to capture any changes that may have occurred since they were saved initially. Cray recommends returning to this procedure at that time if there are changes.

Compare that saved information to what is on the migration SMW running CLE 6.0 / SMW 8.0. Check for any site modifications to the firewall and iptables configuration of the SMW on the CLE 5.2 / SMW 7.2 system.

There may also be differences between the default firewall and iptables configurations provided by Cray in CLE 5.2 / SMW 7.2 for SLES11SP3 and the default configurations provided by Cray in CLE 6.0 / SMW 8.0 for

SLES12. For this reason, Cray recommends analyzing these archived files and settings for differences rather than simply restoring them from the old CLE 5.2 / SMW 7.2 system to the new CLE 6.0 / SMW 8.0 system.

Procedure

1. Compare SuSEfirewall2 configuration of the SMW.

The archive of SMW files on the SMW running CLE 5.2 / SMW 7.2 included `/etc/sysconfig/SuSEfirewall2` and `/etc/sysconfig/SuSEfirewall2.d/*`. Compare these files in the archive to the similar files on the SMW running CLE 6.0 / SMW 8.0. For any differences, investigate the source of the difference (site modification, change in default, and so forth) and decide whether a change needs to be made to the CLE 6.0 / SMW 8.0 files.

If any changes are required, make them now on the migration SMW.

2. Compare iptables configuration of the SMW.

Save the output from this command to compare to what was saved earlier on the SMW running CLE 5.2 / SMW 7.2. It will be produced in a format that could be used with `iptables-restore`; however, do not restore the old SMW CLE 5.2 / SMW 7.2 version of this file without comparing it to the new CLE 6.0 / SMW 8.0 version of the file and understanding any differences.

```
smw# iptables-save
```

If any changes are required, make them now on the migration SMW.

6.7 Choose Image Recipes to Build

This step of the "Preparation of Configuration Data and Software Images" phase of the migration process prepares for building images in the next step.

Do the following tasks in the order shown:

1. Identify which recipes will be needed at this site.

In addition to the basic set of images that are needed to boot CLE—admin, service, login, and compute—additional images may be needed for these features:

- DAL (direct-attached Lustre)
- DataWarp with Fusion I/O SSDs
- Netroot (Read [Where to Place the Root File System—tmpfs versus Netroot](#) on page 324 to decide whether to use a Netroot image for compute and/or login nodes.)

2. Identify which recipes might need to be enhanced with site content.

Image recipes are extended for a variety of reasons:

- (common) To add workload manager content and site-specific RPMs, whether the additional RPMs are from Cray-provided software repositories or from site-built RPMs.
- To add non-RPM content.
- To run certain commands in a chrooted context as the recipe is built into an image root.

For information about creating a custom image recipe or extending an existing recipe, see [Install Third-Party Software with a Custom Image Recipe](#) on page 424.

3. Gain familiarity with the `recipe`, `repo`, and `pkgcoll` commands.
4. [Create a NIMS Map](#) on page 325
5. Read these to prepare for building images in the next step of the "Preparation of Configuration Data and Software Images" phase of the migration process.
 - a. [About Image Groups and How to Customize Them](#) on page 326
 - b. [About the Admin Image](#) on page 328

6.7.1 Where to Place the Root File System—tmpfs versus Netroot

The Cray XC™ Series root file system for nodes can either reside in RAM (tmpfs) or be mounted from a network source (Netroot), depending on the type of node. The boot and SDB nodes, all other service nodes (except login nodes), and all DAL (direct-attached Lustre) nodes must use tmpfs. Compute nodes and login nodes may use either tmpfs or Netroot. Use the information provided here to decide whether to use Netroot for some or all compute and login nodes at this site.

About Netroot and Dynamic Shared Objects and Libraries (DSL)

In releases prior to CLE 6.0 / SMW 8.0, the dynamic shared objects and libraries (DSL) feature was optional. It was necessary for many sites because it enabled both dynamic shared libraries and large network-based images, which were needed for systems with NVIDIA GPUs and for most production workloads.

In the current release, DSL is supported by default. Note, however, that the DSL feature no longer includes provision for large network-based images. That capability is now provided by Netroot.

- Sites that require large network-based images and additional storage should use Netroot.
- Sites using NVIDIA GPUs must use Netroot.

Comparison of tmpfs and Netroot

tmpfs The default location of the root file system on Cray XC™ Series systems is tmpfs, a type of memory-resident file system or RAM disk.

tmpfs has these characteristics and limitations:

- always used for service nodes (except login nodes) and DAL (direct-attached Lustre) nodes
- efficient and fast root file system access
- large memory footprint
- file system content needs to be restricted to reduce memory footprint
- typically used when minimal commands and libraries required
- works well for compute nodes with well defined workloads and for service nodes that are used primarily for internal services

Netroot Netroot is an alternative approach that mounts the root file system from a network source. It is used only for compute and login nodes. It uses overlays to layer tmpfs on top of a read-only network file system.

Due to the reliance on overlays, the decision to use Netroot should include consideration of the characteristics and limitations of overlays in addition to those of Netroot listed here.

Netroot has these characteristics and limitations:

- used only for compute and login nodes, never for service nodes (except login nodes)
- slower root file system access
- increased node boot time
- minimized memory footprint (mounted from network, so requires less disk space)
- no restriction on file system content
- typically used when a robust set of commands and libraries required (Netroot enables large network-based images, formerly enabled through the DSL feature)
- works well for compute nodes with diverse workloads and for compute nodes with a high memory footprint
- always used for GPUs

This comparison of tmpfs and Netroot memory footprints is based on a fresh install with nothing extra added. These numbers could be larger or smaller for a site depending on whether the Cray image recipes for tmpfs and Netroot have been extended (by adding necessary RPMs) or reduced (by removing unnecessary RPMs).

Table 105. Comparison of tmpfs and Netroot Memory Footprints

Image Type	Memory Consumption	Number of RPMs
Admin image root - tmpfs	1400 MB	600
Service image root – tmpfs	1700 MB	670
Login image root – tmpfs	3600 MB	1100
Compute image root – tmpfs	1500 MB	745
Login image root – Netroot	125 MB	2500
Compute image root – Netroot	150 MB	2380

6.7.2 Create a NIMS Map

Prerequisites

This procedure assumes that `xtdiscover` has been run in `testconfig` mode earlier in the migration process.

About this task

For a migration, a new NIMS (Node Image Mapping Service) map needs to be created on the migration SMW. This procedure creates a NIMS map and designates it as the active map.

Procedure

Create a NIMS map and set it as active.

These example commands use CLE config sets named `p0`, `p1`, and `p2`. Substitute the correct config set names for this site.

full system For a full, unpartitioned system:

```
smw# cmap create p0
smw# cmap setactive p0
```

partitioned For a partitioned system, with partitions p1 and p2:

```
smw# cmap create p1 --partition p1
smw# cmap setactive p1 -p p1
```

```
smw# cmap create p2 -p p2
smw# cmap setactive p2 -p p2
```

6.7.3 About Image Groups and How to Customize Them

Image group configuration information is used by the `imgbuilder` command to build boot images. Image groups are defined in the global config set in the `cray_image_groups` configuration file (`/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml`). Here is an example of the contents of that file:

```
cray_image_groups:
  default:
    - recipe: "admin_cle_6.0up03_sles_12_x86-64_ari"
      dest: "admin{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
      nims_group: "admin"
    - recipe: "compute_cle_6.0up03_sles_12_x86-64_ari"
      dest: "compute{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
      nims_group: "compute"
    - recipe: "login_cle_6.0up03_sles_12_x86-64_ari"
      dest: "login{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
      nims_group: "login"
    - recipe: "service_cle_6.0up03_sles_12_x86-64_ari"
      dest: "service{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
      nims_group: "service"
    - recipe: "diag-all_cle_6.0up03_sles_12_x86-64_ari"
      dest: "diag-all_cle_60up03_sles_12_x86-64_ari"
    ...
  testing:
    - recipe: "compute_cle_6.0up03_sles_12_x86-64_ari"
      dest: "{my_custom_prefix}_compute-TEST-{my_other_value}_{date}_{time}.cpio"
      nims_group: "compute-test"
```

The only way to modify this information to customize it for a site is to edit this YAML file directly.

```
smw# vi /var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml
```

The following sections describe important things to know to successfully customize the `cray_image_groups` configuration file.

What image groups contain

- The `cray_image_groups` configuration file can contain multiple *image groups* (this example shows two: default and testing). When invoked, `imgbuilder` builds one of these. It builds "default" if no image group name is passed as a parameter.
- Each image group contains a list of *image specifications* that will be built: by default, the standard admin, compute, service, and login images.
- Each image specification is a stanza containing these three fields:

recipe	An IMPS (Image Management and Provisioning System) image recipe name. This can be customized to specify which image recipe is used to build a specific boot image.
dest	The destination file name used for the IMPS image root (which may or may not be a bootable cpio file). This can be customized as described below.
nims_group	The NIMS group to which this image is mapped. The <code>nims_group</code> field is specified only for images that are intended as boot images, so not all specifications have this field (for example, the <code>diag-all</code> image in the default image group).

How to customize an image root file name using placeholders

Placeholders like `{date}` can be used to customize an image root name. The `dest` values in the above example contain several such placeholders. At build time, relevant values are substituted for these placeholders. Currently, `imgbuilder` supports the following built-in placeholders for use in the `cray_image_groups` configuration file:

{date}	the current system date (e.g., 20140314)
{time}	the current system time (e.g., 134514)
{host}	the current system host name
{user}	the current username
{cle_release}	the currently active CLE release
{cle_build}	the currently active CLE build
{patch}	the currently active patch

IMPORTANT: When adding one or more placeholders to `dest`, ensure that the whole expression is enclosed by double quotes. For example,

```
dest: "login_cle_{cle_release}-build{cle_build}_sles_12-created{date}.cpio"
```

User-defined placeholders (optional) are also supported for further customization of image names. An example of a user-defined placeholder is `{note}`, which Cray has added to the image root name in several of the standard image specifications. `{note}` does not need to be defined in order for the image specifications to work; however, if a site wishes to add something more to the image root file names that contain `{note}`, a value for `{note}` can be specified on the command line when running `imgbuilder`, and substitution occurs at runtime. For example, if a site wanted to add the string "favorite" to those image root names, the following command could be used.

```
(EXAMPLE ONLY - DO NOT USE) smw# imgbuilder --map -- note=favorite
```

Other custom placeholders can be defined as well. As with `{note}`, the key/value pair defining the placeholder would be added to the `imgbuilder` command on the command line. The syntax is two hyphens and a space (`--`) followed by any number of placeholder definitions as `key=value` pairs separated by spaces.

For example, this command would tell `imgbuilder` to build the images in the "testing" image group, map them to the NIMS groups specified in that group, and "foo" everywhere for "my_custom_prefix" and "bar" everywhere "my_other_value" appears.

```
(EXAMPLE ONLY - DO NOT USE) smw# imgbuilder --map --image-group testing \  
-- my_custom_prefix=foo my_other_value=bar
```

6.7.4 About the Admin Image

About the admin image. The admin image can be used on boot and SDB nodes ("admin" nodes) instead of the general service node image. The admin recipe produces an image root that is smaller than that produced by the general service recipe, resulting in a boot image small enough for a PXE boot. Using the admin boot image on the boot and SDB nodes may enable them to PXE boot at the same time. And because the general service image is no longer used for nodes that are intended to PXE boot, content can be added to the general service image without regard for the PXE boot size limitation.

Should this site use the admin recipe for both boot nodes and SDB nodes?

- | | |
|---------------------|--|
| boot node(s) | Yes. This will enable a PXE boot of the boot node(s). |
| SDB node(s) | It depends. <ul style="list-style-type: none">• Yes, if nothing needs to be added to the recipe for the SDB node. This will enable a PXE boot of the SDB node(s).• Maybe, if the site needs to create a custom recipe for the SDB node (e.g., to add content for a workload manager), and the admin recipe can be used as a base. Create a custom recipe for the SDB node and add the admin recipe as a sub-recipe. A PXE boot of the SDB node(s) may be possible if the resulting boot image size does not exceed the PXE boot size limit.• No, if the admin recipe is missing content that is needed for the custom SDB recipe. Use the service recipe as the base, instead. Create a custom recipe for the SDB node and add the service recipe as a sub-recipe. A PXE boot of the SDB node(s) may be possible if the resulting boot image size does not exceed the PXE boot size limit. |

For an example of creating and extending a recipe, see [Install Third-Party Software with a Custom Image Recipe](#) on page 424.

6.8 Build Image Roots and Boot Images from Recipes

This part of the migration process uses the recipes identified in the last step to build image roots and boot images. It also uses knowledge about image groups gained in the last step (see [About Image Groups and How to Customize Them](#) on page 326).

About `imgbuilder`. The `imgbuilder` command uses information in the `cray_image_groups` configuration file (`/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml`) to know which images to build, how to build them, what to call the built images, and which NIMS (Node Image Mapping Service) groups to map those images to. When invoked, the `imgbuilder` command builds all of the image specifications from one of the image groups defined in the `cray_image_groups` configuration file, beginning with the first image specification and working down the list of specifications within that group.

Several of the procedures listed below use the `imgbuilder` command with one of these options:

--bootstrap-nims	To successfully map an image to a node, <code>imgbuilder</code> also needs to know which NIMS group that node belongs to, which means the node must have its NIMS group (i.e., its "group" field) populated. But for an initial fresh install, that field may not be populated yet. To ensure that the required node information gets populated prior to building boot images, use the <code>--bootstrap-nims</code> option. With this option, <code>imgbuilder</code> looks at the "group" field of each node, and if it is empty, <code>imgbuilder</code> adds "compute" or "service" depending on the type of that node, as reported by HSS (Hardware Supervisory System).
--image-group	To specify which image group to build, use the <code>--image-group</code> option. If that option is not used, <code>imgbuilder</code> will build the group called "default."
--map	When <code>imgbuilder</code> is invoked with the <code>--map</code> option, it maps the image in each image specification to the associated NIMS group (the <code>nims_group</code> field).
--dry-run	To see what IMPS and NIMS commands <code>imgbuilder</code> would run, without actually running them, use the <code>--dry-run</code> option.

`imgbuilder` logs are found at `var/adm/cray/logs/imgbuilder`. For more information, see the `imgbuilder` man page or type `imgbuilder -h`. Cray recommends gaining familiarity with both the `imgbuilder` command to build a set of images and the `image` command to build a single image.

Procedures to follow. Use [Migration Checklist 2.6: Build Image Roots and Boot Images from Recipes](#) on page 439 to track progress while performing the following procedures.

1. [Bootstrap NIMS with `imgbuilder`](#) on page 329
2. [Install SMW/CLE Patches on the Migration SMW](#) on page 330
3. [Prepare Cray Image Groups and Custom Recipes](#) on page 332
4. [Assign Nodes to New NIMS Groups](#) on page 335
5. [Build Images and Map Them to NIMS Groups](#) on page 339

After the images have been built, sites must be careful to verify that the desired content is in the image roots, especially if the recipes have been extended.

6.8.1 Bootstrap NIMS with `imgbuilder`

Prerequisites

This procedure assumes the following tasks have been completed in the migration process:

- `xtdiscover` has been run in testconfig mode
- service nodes have been assigned manually
- image recipes have been selected

About this task

Using the `imgbuilder` command with the `--bootstrap-nims` option ensures that the required node information gets populated prior to building boot images. With this option, `imgbuilder` looks at the "group" field of each node, and if it is empty, adds "compute" or "service" depending on the type of that node.

Procedure

Bootstrap NIMS (Node Image Mapping Service) using `imgbuilder` with the `bootstrap` option.

full system For a full, unpartitioned system:

```
smw# imgbuilder --bootstrap-nims
```

partitioned For a partitioned system, with partitions `p1` and `p2`:

```
smw# imgbuilder --bootstrap-nims -p p1
```

```
smw# imgbuilder --bootstrap-nims -p p2
```

All nodes have now been assigned to the correct NIMS service or compute group (i.e., have their "group" field set to either "service" or "compute"). The boot and SDB nodes will be assigned to the "admin" group later in the migration process.

6.8.2 Install SMW/CLE Patches on the Migration SMW

Prerequisites

This procedure assumes that config sets and non-config-set configuration files have been updated on the migration SMW.

About this task

This procedure installs SMW and CLE patches on the migration SMW to ensure that the latest release software is on the machine in preparation for building images. It includes steps to suppress building NIMS maps and running pre- and post-configuration scripts, activities that are not needed or should not be done at this point in the migration process.

Procedure

1. Check CrayPort for any available CLE and SMW patches for the CLE 6.0.UP03 / SMW 8.0.UP03 release.
2. Make a directory on the SMW (if it does not already exist) to hold any patches that may be available on CrayPort.

```
smw# mkdir -p /var/adm/cray/release/patchsets
```

3. Download SMW and CLE patches to the patchset directory on the SMW, as described in the release notes.
4. Confirm which patches are now in the patchset directory.

```
smw# ls -l /var/adm/cray/release/patchsets
```

5. Set environment variables to suppress some patch actions.
 - a. Temporarily suppress building and mapping images.

This prevents the patch scripts from creating images and mapping them to NIMS. Image creation and NIMS mapping are done later in the migration process.

```
smw# export PATCHSET_BUILD_IMAGES=false
smw# echo $PATCHSET_BUILD_IMAGES
```

```
smw# export PATCHSET_NIMS_TIMING=deferred
smw# echo $PATCHSET_NIMS_TIMING
```

- b. Disable running of configurator pre-configuration and post-configuration scripts.

Because the migration SMW is not connected to the CLE hardware, the configurator must not call pre- and post-configuration scripts, some of which require HSS daemons and other CLE services to be running. Rather than add the `--no-scripts` option to each `cfgset` command in the following procedures, this step disables those scripts by setting these two environment variables, which will last for the duration of the login session.

```
smw# export IMPS_SKIP_PRECONFIG_SCRIPTS=1
smw# echo $IMPS_SKIP_PRECONFIG_SCRIPTS
```

```
smw# export IMPS_SKIP_POSTCONFIG_SCRIPTS=1
smw# echo $IMPS_SKIP_POSTCONFIG_SCRIPTS
```

6. Follow the instructions in the SMW patch README files.

- a. Review what the patch is about from the README file.
- b. Run the LOAD script.
- c. Apply the SMW patch.

To apply the patch, review and run the patch set specific instructions in INSTALL file for each SMW patch. The last step will be to record the installation of this patch.

IMPORTANT: Do NOT run any of these commands on the SMW when directed to do so by the INSTALL script. Because the migration SMW is not connected to XC hardware, these commands would fail.

```
xtalive
xtbounce
xtccreboot
xtzap
```

7. Follow the instructions in the CLE patch README files.

- a. Review what the patch is about from the README file.
- b. Run the LOAD script.
- c. Do this workaround for an invalid CLE config set.

IMPORTANT:

IMPORTANT: This step is intended only for the **migration SMW**. This is not an action to take on an SMW that is connected to XC system hardware.

Because the configurator has been running with `IMPS_SKIP_PRECONFIG_SCRIPTS=1` and `IMPS_SKIP_POSTCONFIG_SCRIPTS=1` on the migration SMW, the config set will always be invalid. Only after all of the configurator pre-configuration and post-configuration callback scripts have been run

will the config set be marked valid. That will be done on the SMW that will be connected to XC system hardware later in the migration process.

The config set is invalid (health=false) until changed by this `sed` command. This disables the check for a valid CLE config set (p0 in the example).

```
smw# grep health /var/opt/cray/imps/config/sets/p0/.imps_ConfigSet_metadata
health: false

smw# sed -i 's/health: false/health: true/' /var/opt/cray/imps/config/\
sets/p0/.imps_ConfigSet_metadata

smw# grep health /var/opt/cray/imps/config/sets/p0/.imps_ConfigSet_metadata
health: true
```

- d. Run the INSTALL script for the CLE patch set.

The INSTALL script will run the patch-set-specific instructions. The last step will be to record the installation of this patch.

IMPORTANT: Do NOT run any of these commands on the SMW when directed to do so by the INSTALL script. Because the migration SMW is not connected to XC hardware, these commands would fail.

```
cfgset push
image create
image export
```

IMPORTANT: Do NOT run any of these commands on the CMC when directed to do so by the INSTALL script.

```
add_configset
deploy_*.sh
```

8. Return CLE config set to "invalid" state.

IMPORTANT:

IMPORTANT: This step is intended only for the **migration SMW**. This is not an action to take on an SMW that is currently connected to XC system hardware.

Because the configurator has been running with `IMPS_SKIP_PRECONFIG_SCRIPTS=1` and `IMPS_SKIP_POSTCONFIG_SCRIPTS=1` on the migration SMW, the config set will always be invalid. Only after all of the configurator pre-configuration and post-configuration callback scripts have been run will the config set be marked valid. That will be done on the SMW that is or will be connected to XC system hardware later in the migration process.

The config set was temporarily set to valid (health=true). This `sed` command changes it back to its real state of invalid (health=false) for a CLE config set named p0.

```
smw# grep health /var/opt/cray/imps/config/sets/p0/.imps_ConfigSet_metadata
health: true

smw# sed -i 's/health: true/health: false/' /var/opt/cray/imps/config/\
sets/p0/.imps_ConfigSet_metadata

smw# grep health /var/opt/cray/imps/config/sets/p0/.imps_ConfigSet_metadata
health: false
```

6.8.3 Prepare Cray Image Groups and Custom Recipes

Prerequisites

This procedure assumes that all SMW and CLE patch sets have been installed on the migration SMW.

About this task

Customize the `cray_image_groups` configuration file, as needed, by editing `/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml`. Move recipe stanzas into the default group for anything to be built by default, and modify or create other image groups as appropriate for this site.

Procedure

1. Edit `cray_image_groups.yaml`.

```
smw# vi /var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml
```

2. Prepare direct-attached Lustre (DAL) image stanza, if applicable.

For systems using DAL, ensure that this DAL stanza (image specification) is in the default image group, or customize and use the `tmpfs-w-dal` image group, which already has it.

```
cray_image_groups:
  default:
    ...
    - recipe: "dal_cle_6.0up02_centos_6.5_x86-64_ari"
      dest: "dal{note}_cle_{cle_release}-build{cle_build}{patch}_centos_6.5-created{date}.cpio"
      nims_group: "dal"
    ...
```

3. Prepare Netroot compute and login image stanza, if applicable.

- a. Add these two Netroot image specifications to the default image group, if they are not already there.

The safest way to do this is to find these two image specifications elsewhere in the file, then copy and paste them from there to the default image group.

NOTE: If Netroot will be used on only a subset of compute and login nodes instead of all of them, then create a separate NIMS group for the subset of compute nodes ("compute_netroot") and one for the subset of login nodes ("login_netroot") in the procedure that follows this one ([Assign Nodes to New NIMS Groups](#) on page 335). Substitute those names for "compute" and "login" as the values assigned to `nims_group` in these two image specifications (keep the double quotes).

```
cray_image_groups:
  default:
    ...
    - recipe: "initrd-compute-large_cle_6.0.up02_sles_12_x86-64_ari"
      dest: "initrd-compute-large{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-created{date}.cpio"
      nims_group: "compute"
    - recipe: "initrd-login-large_cle_6.0.up02_sles_12_x86-64_ari"
      dest: "login-large{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-created{date}.cpio"
      nims_group: "login"
    ...
```

Each of these Netroot image recipes builds two image roots and only one boot image (the `.cpio` file). For example, the first builds an `initrd-compute-large` image root, a `compute-large` image root, and an `initrd-compute-large` boot image.

NOTE: The value for 'dest' in the login Netroot image specification begins with "login-large" but it should begin with "initrd-login-large" to be similar to the value of 'dest' in the compute Netroot stanza. The omission of 'initrd-' does not affect the behavior of `imgbuilder`: the correct image roots and boot image are created.

- b. Comment out image specifications with redundant NIMS group assignments.

If Netroot-specific NIMS groups are being used, skip this substep.

If this site is using Netroot for all compute and login nodes, then comment out any other image specifications in the "default" image group that have these NIMS group assignments: `nims_group: "compute"` or `nims_group: "login"`. This will avoid building unnecessary image roots.

4. Prepare DataWarp Fusion IO image stanza, if applicable.

Sites with Fusion IO (Sandisk) SSD cards must integrate the driver software into the service node image. No special boot image is required for DataWarp using Intel SSDs. For more information about installation of third-party software with a custom image, see [Install Third-Party Software with a Custom Image Recipe](#) on page 424.

Fusion ioScale2 SSD cards, though supported in CLE 5.2, are not supported in CLE 6.0. Perform this step if DataWarp will be used for nodes that are endowed with Fusion IO SSDs. The information for this step is taken from "Create a New Service Node Image for Fusion IO SSDs" in the *XC™ Series DataWarp™ Installation and Administration Guide (S-2564)*.

- a. Add two DataWarp stanzas to the Cray image groups file.

Add this stanza to the end of the default image group so that this image will be built whenever the default image group is built.

```
cray_image_groups:
  default:
    ...
    - recipe: "fio-service_cle_6.0up02_sles_12_x86-64_ari"
      dest: "fio-service{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
      nims_group: "fio-service"
    ...
```

Add this stanza elsewhere in the Cray image groups file to create a new image group that can be used for building the fio-service recipe without rebuilding all of the recipes in the default image group.

```
cray_image_groups:
  default:
    ...
  fio-service:
    - recipe: "fio-service_cle_6.0up02_sles_12_x86-64_ari"
      dest: "fio-service{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
      nims_group: "fio-service"
    ...
```

- b. Create a new recipe for fio-service.

Sites with Fusion IO (Sandisk) SSD cards must integrate the driver software into the service node image.

```
smw# recipe create fio-service_cle_6.0up02_sles_12_x86-64_ari

smw# recipe update fio-service_cle_6.0up02_sles_12_x86-64_ari --add-recipe \
service_cle_6.0up02_sles_12_x86-64_ari

smw# recipe update fio-service_cle_6.0up02_sles_12_x86-64_ari --add-coll \
datawarp-xtra_cle_6.0up02_sles_12
```

```
smw# recipe update fio-service_cle_6.0up02_sles_12_x86-64_ari --add-repo \
passthrough-common_cle_6.0up02_sles_12_x86-64

smw# recipe update fio-service_cle_6.0up02_sles_12_x86-64_ari --add-repo \
passthrough-common_cle_6.0up02_sles_12_x86-64_updates

smw# recipe update fio-service_cle_6.0up02_sles_12_x86-64_ari --add-repo \
common_cle_6.0up02_sles_12_x86-64_ari

smw# recipe update fio-service_cle_6.0up02_sles_12_x86-64_ari --add-repo \
common_cle_6.0up02_sles_12_x86-64_ari_updates
```

For more information, see [Install Third-Party Software with a Custom Image Recipe](#) on page 424.

c. Verify content in the fio-service recipe.

```
smw# recipe show fio-service_cle_6.0up02_sles_12_x86-64_ari
fio-service_cle_6.0up02_sles_12_x86-64_ari:
  name: fio-service_cle_6.0up02_sles_12_x86-64_ari
  created: 2016-12-14T18:18:10
  repositories:
    common_cle_6.0up02_sles_12_x86-64_ari_updates
    common_cle_6.0up02_sles_12_x86-64_ari
    passthrough-common_cle_6.0up02_sles_12_x86-64_updates
    passthrough-common_cle_6.0up02_sles_12_x86-64
  recipes:
    service_cle_6.0up02_sles_12_x86-64_ari
  package_collections:
    datawarp-xtra_cle_6.0up02_sles_12
  path: /etc/opt/cray/imps/image_recipes.d/image_recipes.local.json
  history:
    2016-12-14T18:18:41: Extended with recipe
  service_cle_6.0up02_sles_12_x86-64_ari
    2016-12-14T18:19:54: Extended package_collections attribute with 1
    Package Collection.
    2016-12-14T18:21:24: Extended repositories attribute with 1 Repository.
    2016-12-14T18:22:16: Extended repositories attribute with 1 Repository.
    2016-12-14T18:22:51: Extended repositories attribute with 1 Repository.
    2016-12-14T18:22:59: Extended repositories attribute with 1 Repository.
```

5. Edit `cray_image_groups.yaml` again and add any site custom stanzas, as needed.

Ensure that any site custom recipes are added to the default image group or a site-specific stanza so that they will get built.

```
smw# vi /var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml
```

Regardless of which workload manager (WLM) is being used, there is content that must be added to the image root.

PBS: Following the PBS install instructions, recipes will be created for `pbs-login`, `pbs-admin` (or `pbs-service`) and extended with PBS content. There will also be content added to `/var/opt/pbs` on the SMW that will be used for `post_build_files` in the recipes to be copied into the PBS image roots. Add a stanza for each of these recipes to `cray_image_groups.yaml` so that `imgbulder` can build them.

See [Install and Configure a Workload Manager \(WLM\)](#) on page 403.

6.8.4 Assign Nodes to New NIMS Groups

Prerequisites

This procedure assumes that all nodes have been assigned to either the NIMS compute group or NIMS service group.

About this task

Any nodes that need a different boot image than the standard service or compute boot image must be assigned to different NIMS groups.

Procedure

ADMIN NODE ASSIGNMENT -----

1. Assign the boot and SDB nodes to the NIMS admin group.

Assign the boot and SDB nodes to the NIMS admin group so that they will be assigned the admin boot image for booting. This example uses `c0-0c0s0n1` and `c0-0c0s1n1` as the admin (boot and SDB) nodes. Substitute the correct cnames for this site when using these commands.

full system For a full, unpartitioned system, remove from the NIMS service group and add to the NIMS admin group:

```
smw# cnode update -G service -g admin c0-0c0s0n1 c0-0c0s1n1
```

partitioned For a partitioned system, with partitions p1 and p2, remove from the NIMS service group and add to the NIMS login group:

```
smw# cnode update -G service -g admin c0-0c0s0n1 c0-0c0s1n1 -p p1
```

```
smw# cnode update -G service -g admin c0-0c0s0n1 c0-0c0s1n1 -p p2
```

NOTE: If a custom recipe will be created and used for the SDB node(s) instead of the admin recipe (for example, to add content for a workload manager), assign the SDB node(s) to a different NIMS group, where the name of the NIMS group may have the same name as the custom recipe.

2. Confirm that the intended nodes were added to the NIMS admin group.

```
smw# cnode list --filter group=admin
```

LOGIN NODE ASSIGNMENT -----

3. Assign login nodes to the NIMS login group.

Assign login nodes to the NIMS login group so that they will be assigned the login boot image for booting. To assign more than one node, use a space-separated list of nodes. This example uses `c0-0c0s1n1` and `c0-0c0s3n2` as the login nodes. Substitute the correct cnames for this site when using these commands.

full system For a full, unpartitioned system, remove from the NIMS service group and add to the NIMS login group:

```
smw# cnode update -G service -g login c0-0c0s1n1 c0-0c0s3n2
```

partitioned For a partitioned system, with partitions p1 and p2, remove from the NIMS service group and add to the NIMS login group:

```
smw# cnode update -G service -g login c0-0c0s1n1 -p p1
```

```
smw# cnode update -G service -g login c0-0c0s3n2 -p p2
```

4. Confirm that the intended nodes were added to the NIMS login group.

```
smw# cnode list --filter group=login
```

```
DAL NODE ASSIGNMENT -----
```

5. For systems using direct-attached Lustre (DAL), assign DAL nodes to the NIMS dal group.

Assign DAL service nodes to the NIMS dal group so that they are assigned the DAL boot image for booting. To assign more than one node, use a space-separated list of nodes. This example uses c0-0c0s2n1 and c0-0c0s2n2 as the DAL nodes. Substitute the correct cnames for this site when using these commands.

full system For a full, unpartitioned system, remove from the NIMS service group and add to the NIMS dal group:

```
smw# cnode update -G service -g dal c0-0c0s2n1 c0-0c0s2n2
```

partitioned For a partitioned system, with partitions p1 and p2, remove from the NIMS service group and add to the NIMS dal group:

```
smw# cnode update -G service -g dal c0-0c0s2n1 c0-0c0s2n2 -p p1
```

```
smw# cnode update -G service -g dal c0-1c0s2n1 c0-1c0s2n2 -p p2
```

6. Confirm that the intended nodes were added to the NIMS dal group.

```
smw# cnode list --filter group=dal
```

```
NETROOT NODE ASSIGNMENT -----
```

If this site plans to use Netroot for ALL compute and login nodes, then the next two steps can be skipped because the NIMS login group will be used for all login nodes and the NIMS compute group will be used for all compute nodes. If Netroot will be used on only a subset of compute and login nodes instead of all of them, then create a separate NIMS group for the subset of compute nodes and one for the subset of login nodes.

7. (Skip if Netroot used for all compute/login nodes) Create and assign Netroot-specific NIMS groups.

To use Netroot for only a subset of compute and login nodes, create and assign Netroot-specific NIMS groups for those compute/login subsets. In the example, the new NIMS groups are called *login_netroot* and *compute_netroot*, and each subset of nodes (*SUBSET_LOGIN_NODES* and *SUBSET_COMPUTE_NODES*) is a space-separated list of nodes.

full system For a full, unpartitioned system, remove the subset of login nodes from the NIMS login group, remove the subset of compute nodes from the NIMS compute group, and add to the NIMS *login_netroot* or *compute_netroot* group, respectively:

```
smw# cnode update -G login -g login_netroot SUBSET_LOGIN_NODES
smw# cnode update -G compute -g compute_netroot SUBSET_COMPUTE_NODES
```

partitioned For a partitioned system, with partitions p1 and p2, remove from the NIMS service group and add to the NIMS dal group:

```
smw# cnode update -G login -g login_netroot SUBSET_LOGIN_NODES -p p1
smw# cnode update -G compute -g compute_netroot \
SUBSET_COMPUTE_NODES -p p1
```

```
smw# cnode update -G login -g login_netroot SUBSET_LOGIN_NODES -p p2
smw# cnode update -G compute -g compute_netroot \
SUBSET_COMPUTE_NODES -p p2
```

8. (Skip if Netroot used for all compute/login nodes) Confirm that the intended nodes were added to the NIMS Netroot groups.

```
smw# cnode list --filter group=login_netroot
```

```
smw# cnode list --filter group=compute_netroot
```

DATAWARP FUSION IO NODE ASSIGNMENT -----

9. Assign DataWarp Fusion IO nodes to NIMS fio-service group.

For systems using DataWarp Fusion IO nodes, assign those nodes to the NIMS fio-service group. To assign more than one node, use a space-separated list of nodes. Substitute the correct cnames for this site when using these commands.

full system For a full, unpartitioned system, remove from the NIMS service group and add to the NIMS fio-service group:

```
smw# cnode update -G service -g fio-service cname1 cname2
```

partitioned For a partitioned system, with partitions p1 and p2, remove from the NIMS service group and add to the NIMS fio-service group:

```
smw# cnode update -G service -g fio-service cname1 -p p1
```

```
smw# cnode update -G service -g fio-service cname1 -p p2
```

See "Create a New Service Node Image for Fusion IO SSDs" in the *XC™ Series DataWarp™ Installation and Administration Guide* (S-2564).

10. Confirm that the intended nodes were added to the NIMS fio-service group.

```
smw# cnode list --filter group=fio-service
```

WORKLOAD MANAGER (WLM) NODE ASSIGNMENT -----

11. If a site has added special recipes for WLMs that are assigned to nodes not in one of the other NIMS groups, create a new NIMS group and assign them now.

For example, create a NIMS group for MOM nodes (wlm-service) while the rest of the service nodes use the NIMS service group.

full system For a full, unpartitioned system, remove from the NIMS service group and add to the NIMS wlm-service group:

```
smw# cnode update -G service -g wlm-service cname_mom1 cname_mom2
```

partitioned For a partitioned system, with partitions p1 and p2, remove from the NIMS service group and add to the NIMS wlm-service group:

```
smw# cnode update -G service -g wlm-service cname_mom1 -p p1
```

```
smw# cnode update -G service -g wlm-service cname_mom2 -p p2
```

12. Confirm that the intended nodes were added to the NIMS wlm-service group.

```
smw# cnode list --filter group=wlm-service
```

6.8.5 Build Images and Map Them to NIMS Groups

Prerequisites

This procedure assumes the following:

- all nodes have been correctly assigned to NIMS groups
- image groups and custom recipes have been prepared

About this task

This procedure provides two methods (step 1 and step 2) to build the images defined in `/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml` and map them to NIMS groups. Step 1 uses the `imgbuilder --map` command to both build and map images, and step 2 uses `imgbuilder` to build images and then manually maps them to NIMS groups. Perform step 1 or step 2, but not both steps.

NOTICE: Building images takes approximately 5 minutes for each type of recipe in the image group. Building Netroot images takes slightly longer because there are more RPMs to be installed. If four different recipes are in the default image group, it will take about 20 minutes to build the images. If `fio-service`, `Netroot for compute (initrd-compute-large)`, and `Netroot for login (initrd-login-large)` are added to the set it may take about 40 minutes.

Procedure

1. Build images and map them to NIMS groups.

Create a set of images as defined

in `/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml` and map them to the specified NIMS groups using the `--map` option.

full system To build the images in the "default" image group and map them to the NIMS groups specified in "default" for a full, unpartitioned system:

```
smw# imgbuilder --map
```

partitioned To build the images in the "default" image group and map them to the NIMS groups specified in "default" for a partitioned system, with partitions p1 and p2:

```
smw# imgbuilder --map -p p1 -p p2
```

2. (Alternative to step 1 on page 339) Build images and then manually map them to NIMS groups.

As an alternative to mapping the images using the `--map` option, that mapping can be done manually.

- a. Build the images.

```
smw# imgbuilder
```

- b. Map the images to specified NIMS groups.

Replace the cpio file names in these examples with the file names from the `imgbuilder` output in step a.

If any other boot images have been created for special nodes, assign them with similar `cnode update` commands filtered for the NIMS groups to which those special nodes have been assigned.

full system To map the images to specified NIMS groups for a full, unpartitioned system:

```
smw# cd /var/opt/cray/imps/boot_images
smw# ls -ltr

smw# cnode update -i compute_img.cpio --filter group=compute
smw# cnode update -i service_img.cpio --filter group=service
smw# cnode update -i login_img.cpio --filter group=login
smw# cnode update -i dal_img.cpio --filter group=dal
```

partitioned To map the images to specified NIMS groups for a partitioned system, with partitions p1 and p2:

```
smw# cd /var/opt/cray/imps/boot_images
smw# ls -ltr

smw# cnode update -i compute_img.cpio --filter group=compute -p p1
smw# cnode update -i service_img.cpio --filter group=service -p p1
smw# cnode update -i login_img.cpio --filter group=login -p p1
smw# cnode update -i dal_img.cpio --filter group=dal -p p1

smw# cnode update -i compute_img.cpio --filter group=compute -p p2
smw# cnode update -i service_img.cpio --filter group=service -p p2
smw# cnode update -i login_img.cpio --filter group=login -p p2
smw# cnode update -i dal_img.cpio --filter group=dal -p p2
```

6.9 Assign Kernel Parameters to Nodes

The `imgbuilder` command can be used to map the boot images it builds to nodes, but use the `cnode` command for further manipulation of boot image assignment and for changing kernel parameters or config sets for nodes and groups of nodes.

Gain familiarity with commands to create NIMS maps (`cmap`) and manipulate information for nodes (`cnode`) to create NIMS groups and to assign NIMS groups, boot images, and kernel parameters to nodes.

The next procedure, [Set the Turbo Boost Limit](#) on page 341, shows how to use the `cnode` command to view and change a kernel parameter.

6.9.1 Set the Turbo Boost Limit

Turbo boost limiting is NOT supported on Intel® Xeon Phi™ "Knights Landing" (KNL) or on Intel® Xeon® "Sandy Bridge" processors.

Because Intel 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. Turbo boost can be limited by setting the `turbo_boost_limit` kernel parameter to one of these valid values:

Value	Result
0	Disable turbo boost.
100	Limits turbo boost to 100 MHz.
200	Limits turbo boost to 200 MHz.
300	Limits turbo boost to 300 MHz.
400	Limits turbo boost to 400 MHz.
999 (default)	No limit is applied.

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. For example, on a 12-core processor, the limit is in effect when 12, 11, 10, or 9 cores are active.

Set or Change the Turbo Boost Limit Parameter

To make a persistent change, use `cnode` (as `crayadm` or `root`) to change the parameter. This change will take effect later when the nodes are rebooted. Note that the following commands target all nodes or all compute nodes. To specify individual nodes, add their `cnames` at the end of the command line.

- To list the current kernel parameters for a full or partitioned system:

full system For a full, unpartitioned system:

```
smw# cnode list
```

partitioned For a partitioned system:

```
smw# cnode list --partition p1
```

2. To change the `turbo_boost_limit` kernel parameter for all compute nodes in a full or partitioned system, substitute one of the values listed above for `value` in one of these commands:

full system For a full, unpartitioned system:

```
smw# cnode update --filter group=compute \  
--add-parameter turbo_boost_limit=value
```

partitioned For a partitioned system:

```
smw# cnode update --filter group=compute --partition p1 \  
--add-parameter turbo_boost_limit=value
```

3. To remove the change, if needed, use one of these commands:

full system For a full, unpartitioned system:

```
smw# cnode update --filter group=compute \  
--remove-parameter turbo_boost_limit
```

partitioned For a partitioned system:

```
smw# cnode update --filter group=compute --partition p1 \  
--remove-parameter turbo_boost_limit
```

6.10 Check NIMS Information

About this task

This procedure lists NIMS (Node Image Mapping Service) information: which maps are active on the SMW and what NIMS information is stored for each node.

Procedure

1. Check active NIMS maps.

full system For a full, unpartitioned system:

```
smw# cmap list
```

partitioned For a partitioned system, with partitions p1 and p2:

```
smw# cmap list --partition p1
```

```
smw# cmap list -p p2
```

2. Check NIMS information for each node.
-

full system For a full, unpartitioned system:

```
smw# cnode list
```

partitioned For a partitioned system, with partitions p1 and p2:

```
smw# cnode list -p p1
```

```
smw# cnode list -p p2
```

3. Check NIMS information for each NIMS group.

```
smw# cnode list --filter group=admin
smw# cnode list --filter group=service
smw# cnode list --filter group=login
smw# cnode list --filter group=compute
```

Check any additional NIMS groups that may have been created for Netroot compute and login nodes (typically only when Netroot is used on only a subset of compute and login nodes instead of all of them, so the NIMS compute and login groups cannot be used for that subset).

```
smw# cnode list --filter group=compute_netroot
smw# cnode list --filter group=login_netroot
```

Check any additional NIMS groups that may have been created for DataWarp with Fusion IO SSDs.

```
smw# cnode list --filter group=fio-service
```

Check any additional NIMS groups that may have been created with WLM (workload manager) or other site names.

```
smw# cnode list --filter group=wlm-admin
smw# cnode list --filter group=wlm-service
smw# cnode list --filter group=wlm-login
```

6.11 Identify and Port Site-local Scripts

About this task

Identify and begin porting any site-local scripts that need to be ported from SLES 11 SP3 / CLE 5.2 / SMW 7.2 to SLES 12 / CLE 6.0 / SMW 8.0 on the migration SMW.

Procedure

1. Identify and port boot scripts.

Some site-local scripts are called from `/etc/init.d` when nodes boot, such as `boot.local` or `boot.last`. These scripts may need to be modified to work with SLES 12.

2. Identify and port additions to the boot automation file.

Other scripts may be called from the boot automation script. What these scripts do for configuration in CLE 5.2 / SMW 7.2 may be handled a different way with the CLE 6.0 / SMW 8.0 release, so they may no longer be needed or may need to be modified to work with the new software.

There have been changes to the default order of actions in the boot automation file `auto.generic` with the CLE 6.0 / SMW 8.0 release. In particular, the boot and SDB nodes can now be booted at the same time via PXE boot, and all other service nodes can be booted at once before the compute nodes are booted.

Here are some possible entries in the CLE 5.2 / SMW 7.2 boot automation file that will still be needed in CLE 6.0 / SMW 8.0. If using boot node failover, the blade that contains the primary boot node should have `stonith=true` set. If using SDB node failover, the blade that contains the primary SDB node should have `stonith=true` set. For guidance about where to place boot nodes and SDB nodes in the system when enabling these failover features, see "Discover Hardware and HSN Routing, Prepare STONITH" in *XC™ Series Software Installation and Configuration Guide (S-2559)*.

```
# set stonith for boot & sdb node failover
lappend actions { crms_exec "xtdaemonconfig c0-0c0s0 stonith=true" }
lappend actions { crms_exec "xtdaemonconfig c0-0c0s1 stonith=true" }
```

3. Identify and port Node Health Checker (NHC) plugin scripts.

If there are site-local changes to Node Health plugin scripts, these may need to be modified for the CLE 6.0 / SMW 8.0 release. The site-local NHC plugins could live anywhere a compute node can access. The only way to preserve site-local plugins would be to look at `/etc/opt/cray/nodehealth/nodehealth.conf` and check for any lines like this.

```
Command: </path/to/plugin>
```

4. Identify and port RUR (Resource Utilization and Reporting) plugin scripts.

If there are site-local changes to RUR scripts, these may need to be modified for the CLE 6.0 / SMW 8.0 release. Look in default sharedroot `/etc/opt/cray/rur/rur.conf` on the CLE 5.2 / SMW 7.2 system.

5. Identify and port account creation scripts.

If there are site-local changes to the skeleton files used to create local Linux accounts for the SMW or for CLE nodes, these may need to be modified for the CLE 6.0 / SMW 8.0 release.

6.12 Install Cray Programming Environment (PE) Software

About this task

The Cray Developers Toolkit (CDT) for Cray XC Series systems is a package that consists of the basic libraries and components needed to develop and compile code on Cray systems, including the GNU Fortran, C, and C++ compilers. The CDT also includes the Cray Compiling Environment (CCE), but a valid license key is required before CCE can be installed. All other compilers are sold, installed, and licensed separately.

This procedure installs and configures the Cray Programming Environment (PE) software to make its content available on Cray XC Series compute nodes. A typical PE installation takes about 30 minutes.

For a migration, this procedure can be done as part of the preparation work if the migration SMW is a physical SMW, but cannot be done that early if the migration SMW is virtual because the PE image root is too large (about 80 GB) for the virtual SMW's 150 GB disk on a laptop.

Procedure

1. Create the PE image root.

Use a PE image for several of the monthly releases of PE software, and use a fresh image with each new CLE release.

a. Determine the name of the PE image root used in the PE profile in `cray_image_binding`.

Because the name of the PE image root must match the image name configured in the Image Binding Service (`cray_image_binding`), this command can be used to find out what it is.

```
smw# cfgset search -t image -s cray_image_binding p0 | grep PE
cray_image_binding.settings.profiles.data.PE.image:
pe_compute_cle_6.0up03_sles_12
```

b. Set an environment variable for the PE image name.

```
smw# export PEIMAGE=pe_compute_cle_6.0up03_sles_12
smw# echo $PEIMAGE
```

If this site wishes to use a different name for the PE image when setting the `$PEIMAGE` environment variable, update the name in the PE profile of the `cray_image_binding` service for the CLE configuration set to match.

Note that although the PE image name has 'compute' in it, the same image is also used for login nodes.

c. Create PE image on the SMW.

1. Get the name of the PE image recipe on the system.

```
smw# recipe list | grep ^pe
pe_image_cle_6.0up03_sles_12
```

2. Create the PE image (`$PEIMAGE`) using the recipe name discovered by the command in the previous step.

```
smw# image create -r pe_image_cle_6.0up03_sles_12 $PEIMAGE
```

2. Install the compiler license RPMs.

The Cray Compiling Environment (CCE), Intel, and PGI compilers all require licenses. These licenses must be installed at this point before installing any of the PE software. For instructions, see

CCE *Cray Compiling Environment Release Overview and Installation Guide*, available at <http://pubs.cray.com>

Intel compilers <http://software.intel.com/en-us/articles/intel-software-technical-documentation>

PGI compilers <http://www.pgroup.com>

3. Copy the most recent PE ISO to the SMW and mount the ISO.

```
smw# mkdir -p /var/adm/cray/release/pe
smw# cd /var/adm/cray/release/pe
smw# mkdir -p /var/adm/cray/release/pe/mount_iso
```

```
smw# mount -o loop,ro <downloaded PE ISO> /var/adm/cray/release/pe/mount_iso
```

4. Install the `craype-installer` rpm from the PE ISO on the SMW.

```
smw# rpm -ivh /var/adm/cray/release/pe/mount_iso/installer/\
craype-installer-*.x86_64.rpm
```

5. Configure the installer configuration file.

- a. Copy the install configuration file from the `craype-installer` installation directory.

```
smw# cp -p /opt/cray/craype-installer/default/conf/install-cdt.yaml .
```

- b. Create logs directory that will be used by the installer.

```
smw# mkdir ./logs
```

- c. Update the configuration file, `install-cdt.yaml`.

When `install-cdt.yaml` is opened, there are comment blocks before every keyword listed below describing the valid values for each.

1. For `IMAGE_DIRECTORIES` specify the directory (or directories) for the installer to install into. This example uses an `image_root` of `pe_compute_cle_6.0up03_sles_12`. This parameter must have data on the next line. The data must have four space characters and then a dash character and then a space character and the path to the directory.
2. Specify **YES** in each of the `INSTALL_*_LIBRARIES` for the compiler specific PE libraries to be installed. The Pathscale compiler is no longer supported by PE.
3. If the system includes an `ACCELERATOR`, change **NONE** to a comma separated list of one or more of the supported accelerators - **FERMI** or **KEPLER**. See the comments in `install-cdt.yaml` for examples and more information.
4. If the system has more than one type of processor installed, then specify the lowest common denominator for the processor for `CRAY_CPU_TARGET`.

Because this file supports older releases as well, some of the items are not applicable for this release. Those that are applicable for this release are shown in bold. For a migration, note that the items between `CRAY_CPU_TARGET` and `IMAGE_DIRECTORIES` were used for CLE 5.2 / SMW 7.2 systems. They should be set to the values shown.

```
smw# vi install-cdt.yaml
```

```
---
HAS_MAMU_NODES : NO
ACCELERATORS : NONE
NETWORK_TYPE : NONE
CRAY_CPU_TARGET : sandybridge
BOOTNODE_HOSTNAME : NONE
BOOTNODE_ROOT_DIRS :
- /rr/current
ESMS_HOSTNAME : NONE
ESMS_IMAGE_DIRS :
- /cm/images/<your image name>
UNMANAGED_ESLOGINS : NONE
```

```

IMAGE_DIRECTORIES :
  - /var/opt/cray/imps/image_roots/pe_compute_cle_6.0up03_sles_12
LOGS_DIR           : ./logs
ISO_MOUNT_DIR     : ./mount_iso
INSTALL_CCE_LIBRARIES : YES
INSTALL_GNU_LIBRARIES : YES
INSTALL_INTEL_LIBRARIES : YES
INSTALL_PATHSCALE_LIBRARIES : NO
INSTALL_PGI_LIBRARIES : YES

```

6. Install PE software from the most recent PE installation media and installer.

- a. Link `/opt/cray/pe/bin` to `/opt/cray`.

```

smw# chroot /var/opt/cray/imps/image_roots/$PEIMAGE ln \
-s /opt/cray/pe/bin /opt/cray/bin

```

- b. Run the PE installer.

This step can take about 30 minutes.

```

smw# module load craype-installer
smw# craype-installer.pl --install --install-yaml-path ./install-cdt.yaml

```

When the installation completes, output such as the following will be displayed, summarizing the installed packages.

```

1) atp-1.7.5-0_3605.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
2) cray-ccdb-1.0.3-0_3575.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
3) cray-dwarf-14.2.0-0.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
<snip>
71) perftools-clients-6.2.2-1.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)

```

- c. Set the default versions for PE (if the install succeeds) by running `set_default` scripts.

```

smw# craype-installer.pl --set-default --install-yaml-path ./install-cdt.yaml

```

Note that at the monthly PE update, this step would not be done until after the image is pushed to the boot node and tested.

- d. Unmount the ISO.

```

smw# umount ./mount_iso

```

- e. Clean up the PE ISO and PE RPMs.

These can be removed since they are large and use up disk space.

```

smw# rm *.iso *.rpm *.tar.gz

```

6.12.1 Install Additional Cray Programming Environment (PE) Software Releases to Image Root

Prerequisites

This procedure assumes that a single Cray Programming Environment (PE) software release has been installed, which created `/var/adm/cray/release/pe/install-cdt.yaml` to control what software the `craype-installer` will install and to where.

About this task

Additional PE software releases can be installed into the same PE image root used in [Install Cray Programming Environment \(PE\) Software](#) and defined in `var/adm/cray/release/pe/install-cdt.yaml`. However, ensure that the `setdefault` script has been run for whichever version is intended to be the default. The desired default may not be the most recently installed version.

Note that any version installed on the CLE 5.2 / SMW 7.2 system prior to CDT 15.09, released in September 2015, cannot be installed on CLE 6.0 / SMW 8.0. However all of the newer releases can be.

Procedure

1. Copy the desired monthly release of the PE ISO to the SMW and mount the ISO.

```
smw# mkdir -p /var/adm/cray/release/pe
smw# cd /var/adm/cray/release/pe
smw# mkdir -p /var/adm/cray/release/pe/mount_iso
smw# mount -o loop,ro <downloaded PE ISO> /var/adm/cray/release/pe/mount_iso
```

2. Install the `craype-installer` RPM from the PE ISO on the SMW.

```
smw# rpm -ivh /var/adm/cray/release/pe/mount_iso/installer/\
craype-installer-*.x86_64.rpm
```

3. Install PE software from the most recent PE installation media and installer.

- a. Run the PE installer.

This step can take about 30 minutes.

```
smw# module load craype-installer
smw# craype-installer.pl --install --install-yaml-path ./install-cdt.yaml
```

When the installation completes, the following will be displayed, summarizing the installed packages.

```
1) atp-1.7.5-0_3605.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up02_sles_12_x86-64_ari)
2) cray-ccdb-1.0.3-0_3575.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up02_sles_12_x86-64_ari)
3) cray-dwarf-14.2.0-0.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up02_sles_12_x86-64_ari)
<snip>
71) perftools-clients-6.2.2-1.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up02_sles_12_x86-64_ari)
```

- b. Set the default versions for PE (if the install succeeds) by running `set_default` scripts.

```
smw# craype-installer.pl --set-default --install-yaml-path ./install-cdt.yaml
```

- c. Unmount the ISO.

```
smw# umount ./mount_iso
```

- d. Clean up the PE ISO and PE RPMs.

```
smw# rm *.iso *.rpm *.tar.gz
```

7 Preservation of Other Data Prior to Final Shutdown

These steps should be performed just before shutting down the CLE 5.2 / SMW 7.2 system for the last time, so that the most current site accounting, operational, and user data is preserved. Cray recommends saving this data separately from the configuration and image data captured and archived in the previous stage of this migration process.

1. Run final accounting reports.

If process accounting, SAR (system activity reporter), or RUR (resource utilization reporting) data has been generated on the CLE 5.2 / SMW 7.2 system that needs to be used for accounting or billing based on application job usage, run those reports just before the XC system is shut down.

2. Save operational data.

Save CLE 5.2 / SMW 7.2 operational data, such as boot automation files, tuning settings in HSS configuration files, crontabs, error logs, and other hardware status information.

1. **Site-local plugins.** If there are any site-local plugins for Node Health, archive them now. Look for references to site-local plugins in `/etc/opt/cray/nodehealth/nodehealth.conf`.
2. **Old dumps.** If there are any dumps on the SMW running CLE 5.2 / SMW 7.2 that have not been transferred elsewhere for analysis, archive them now.

```
smw# ls -l /var/opt/cray/dump
```

3. **Crontab entries from SMW.** Check for any crontab entries on the SMW for accounts such as root and crayadm and any site-added accounts.

```
smw# crontab -l
crayadm@smw> crontab -l
```

Save the information. Some of the crontab entries will point towards scripts that need to be saved.

Some of the crontab entries will need to be re-created on the SMW running CLE 6.0 / SMW 8.0. For example, the `/opt/cray/sec/default/bin/check_xt_wrapper.ex` script, which is part of SEC.

4. **Mail configuration files on SMW.** Check `/etc/aliases` on the SMW running CLE 5.2 / SMW 7.2 to confirm that all site email aliases are present on the SMW running CLE 6.0 / SMW 8.0. There may be Postfix or other mail transfer agent (MTA) configuration files to be copied.
5. **SMW files.** Archive all of these SMW files and directories. Decide how much log data to save.
 - Archive SEC logs from running `check_xt` in `/var/log/check_xt`.
 - Archive SEDC files in `/tmp/SEDC_FILES` (if not in postgresql database).
 - Archive SuSEfirewall2 firewall settings and defined services in `/etc/sysconfig/SuSEfirewall2` and `/etc/sysconfig/SuSEfirewall2.d/services/`. These files were saved earlier in the process to enable comparison with the CLE 6.0 / SMW 8.0 versions of those files and analysis of whether any differences should be migrated. They are saved again here to capture any changes that may have occurred since those files were saved earlier. To determine if those changes, if any, need to be migrated, revisit [Check for Site Modifications to SMW Firewall and IP Tables](#) on page 322.

- Archive files from the current boot session in `/var/opt/cray/log/p0-current` (for partition `p0`). Files from earlier boots in `/var/opt/cray/log/*` can be archived also. Sites must decide how many days of logs they wish to retain. These boot session logs include:
 - `xthwerrlogd` files, which have information about the past health of the system that might be needed for fault analysis. The binary files are on the SMW for each boot session in `hwerrlog.p0-SESSIONID*` files, but the CLE 5.2 / SMW 7.2 files can be read by the CLE 6.0 / SMW 8.0 `xthwerrlogd`.
 - `netwatch` files, which are text files in `netwatch.p0-SESSIONID`.
 - network link resiliency files in `nldr-YYYYMMDD`.
 - `pcimon` files in `pcimon-YYYYMMDD*`.
 - RUR data will be contained in the `messages-YYYYMMDD` file.
- Archive `smwmessages-YYYYMMDD` in `/var/opt/cray/log`, which have SMW messages that may include information about the SMW hardware and environmental history.
- Archive user home directories on SMW file system.

Note that the `mysql` database (HSS database) does not need to be archived. A full `xtdiscover` will be run to regenerate the HSS database for the CLE 6.0 / SMW 8.0 system, so it does not need to be dumped and restored.

6. CLE files. Archive all of these CLE files and directories. Decide how much log data to save.

- Dump the SDB databases for reference. This data will be regenerated, but the dump may include site-specific system labels data.

```
sdb# mysqldump --databases XTAcct --log-error=ERRORS \
--result-file=XTAcct_dump.YYYYMMDD.out
```

```
sdb# mysqldump --databases XTAdmin --log-error=ERRORS \
--result-file=XTAdmin_dump.YYYYMMDD.out
```

- (If DataWarp enabled) Save the DataWarp service configuration, which is stored in the `dwsd.db` database on the SDB node.

1. Archive `/var/opt/cray/dws/dwsd.db` on the SDB node.
2. Archive output of these probe commands:

```
sdb# module load dws
sdb# dwstat -b nodes pools > 52dwstat.output
```

- Archive any site custom plugins for Node Health. These would be referenced from `/etc/opt/cray/nodehealth/nodehealth.conf`.
- Archive RUR data contained in the boot session `messages-YYYYMMDD` file.
- Archive ALPS data in `/ufs/alps_shared`, specifically the `apschedNextId` and `apschedPDomain` files to preserve the next `apid` and `alps` protection domains.
- Archive from persistent storage any workload manager (WLM) files that need to be preserved from the WLM server or MOM nodes (most but not all sites use the SDB node as the WLM server).
 - Moab/TORQUE: `/var/spool/moab` and `/var/spool/torque`
 - PBS: `/var/spool/PBS`

See section 7.7, "Migration Upgrade Under Linux" in the PBS Pro *Big Book*, which is available on the PBS Professional documentation site: <http://www.pbsworks.com/PBSProductGT.aspx?n=PBS-Professional&c=Overview-and-Capabilities&d=PBS-Professional,-Documentation>. It lists everything that can be backed up and restored.

- Slurm: `/var/spool/slurm` and configuration files in `/etc/opt/slurm`
- Run this command on the WLM server or MOM nodes to preserve server configuration:

```
wlm# opt/torque/default/bin/qmgr -c 'p s' > /var/spool/torque/  
server_settings
```

3. Save site user data.

Save any CLE 5.2 / SMW 7.2 site user data, such as the following:

- SMW `/home` directories for SMW accounts with local home directories.
- CLE `/ufs/home` directories for CLE accounts with local home directories.
- CLE boot node home directories under `/home`. There may be accounts other than `crayadm`, which is in `/home/crayadm`.
- DataWarp files on the SSDs—the DataWarp SSDs will have to be reformatted after the transition to CLE 6.0 / SMW 8.0.
- Persistent `/var` from boot node `/snv` may have important files, but this depends on what the site has stored from certain nodes in that node's `/var`.

Data on direct-attached Lustre (DAL) file systems should not need to be archived. The Lustre file system provided by DAL nodes does not need to be reformatted during the migration from CLE 5.2 / SMW 7.2 to CLE 6.0 / SMW 8.0.

4. Drain WLM queues before shutting down.

For example, for Slurm, use this command for each partition:

```
smw# scontrol update PartitionName=p1 State=drain
```

For other WLMs, use the equivalent command for that WLM.

8 Shutdown and Switch using a Physical Migration SMW

The final phase of the migration process is to shut down the currently running system and make the switch to the new release. Because this process uses a physical migration SMW and boot RAID, the original SMW and boot RAID must be disconnected from the XC hardware and the migration SMW and boot RAID must be connected to the XC hardware. Use the following procedures in the order listed for this phase of the process.

1. [Shut Down the CLE System](#) on page 353
2. (SMW HA only) [Put the SMW HA Cluster into Maintenance Mode during a Migration](#) on page 354
3. [Switch Cabling to Migration SMW and Boot RAID](#) on page 354
4. [Discover XC System Hardware](#) on page 356
5. [Complete CLE Configuration](#) on page 365
6. [Boot the CLE System during a Migration](#) on page 370
7. Configure other CLE 6.0 / SMW 8.0 features and services and install additional software.
 - a. [Complete Post-boot Configuration of Config Services](#) on page 384
 - b. [Install and Configure Additional Software](#) on page 400
 - c. [Back Up the Newly Installed and Configured SMW/CLE Software](#) on page 404
 - d. [Back Up Site Data](#) on page 404
8. [Restore Operational Data during a Migration](#) on page 406
9. (If needed) [Roll Back Changes during a Migration](#) on page 407

8.1 Shut Down the CLE System

About this task

To shut down the CLE system, if it is booted, use the shutdown automation file to shut it down gracefully.

Procedure

1. Check whether the boot node is up.

full system

For a full, unpartitioned system:

```
smw# ping -c3 boot
```

partitioned For a partitioned system, with partitions p1 and p2:

```
smw# ping -c3 boot-p1
smw# ping -c3 boot-p2
```

2. If the boot node is up, then shut down the CLE system.

full system For a full, unpartitioned system:

```
smw# su - crayadm
crayadm@smw> xtbootsys -s last -a auto.xtshutdown
crayadm@smw> exit
smw#
```

partitioned For a partitioned system, with partitions p1 and p2:

```
smw# su - crayadm
crayadm@smw> xtbootsys --partition p1 -s last -a auto.xtshutdown
crayadm@smw> xtbootsys --partition p2 -s last -a auto.xtshutdown
crayadm@smw> exit
smw#
```

8.2 Put the SMW HA Cluster into Maintenance Mode during a Migration

About this task

NOTICE: This procedure is for the migration of SMW HA systems only.

After CLE has been shut down, put the cluster into maintenance mode before disconnecting any cables. This will ensure that SMW HA takes no action if it detects loss of connectivity. This is a cluster-wide action and needs to be done on only one SMW.

Procedure

1. (For SMW HA systems running SLEHA11SP3 only) Put the SMW HA cluster into maintenance mode.

```
smw# crm configure property maintenance-mode=true 2> /dev/null
```

2. (For SMW HA systems running SLEHA12SP0 only) Put the SMW HA cluster into maintenance mode.

```
smw# maintenance_mode_configure enable
```

8.3 Switch Cabling to Migration SMW and Boot RAID

Prerequisites

This procedure assumes the following:

- Configuration data and software images have been prepared.
- Current site accounting, operational, and user data has been preserved.

About this task

This procedure describes how to disconnect the currently running SMW and boot RAID ("old" SMW and "old" boot RAID) from the XC hardware and connect the migration SMW and boot RAID ("new" SMW and "new" boot RAID) to the XC hardware.

Note that the old SMW can continue to be connected to the old boot RAID. This may be advantageous to access files from the file systems on the old boot RAID using the old SMW to facilitate transfer of user data. However, if the old SMW is connected to the old boot RAID via a switch and the switch is disconnected, the old boot RAID may need to be directly connected to the old SMW.

Procedure

1. Disconnect old SMW Ethernet.

TIP: Label all cables before disconnecting them.

Label and disconnect the old SMW Ethernet cables on eth1 and eth3. These cables will be used to connect the new SMW to the Ethernet switch that has the HSS network and admin network (boot and SDB nodes).

IMPORTANT: Do not have two SMWs connected to the same network at the same time. If two SMWs have eth1 connected to the HSS network at the same time using the same IP address, then the HSS controllers will have failures from DHCP, TFTP, and booting. If two SMWs have eth3 connected to the admin network at the same time using the same IP address, then the boot and SDB nodes would have failures from DHCP, TFTP, and booting.

2. Disconnect from the old boot RAID.

- a. Disconnect the cable between the old SMW and the SAS or fibre channel (FC) switch at the SMW side.
- b. Disconnect the cables between the old boot RAID controllers (A and B) and the SAS/FC switch at the boot RAID side.

This approach clears the way for simply connecting the new boot RAID and new SMW to the SAS/FC switch.

Alternative: Keep the old SMW and boot RAID connected to the SAS/FC switch. However, before the new SMW and new boot RAID are connected to that SAS/FC switch in step 4, the switch would need to be zoned for the new storage.

3. Connect new SMW Ethernet.

Connect the Ethernet cables for eth1 and eth3 to the new SMW.

IMPORTANT: Do not mix up which cable belongs on eth1 and on eth3 to avoid booting problems for the HSS controllers (eth1) or the boot and SDB nodes (eth3).

4. Connect new boot RAID.
-

- a. Connect the cable from the SAS/FC switch to the new SMW.
- b. Connect the cables from the SAS/FC switch to the new boot RAID controllers (A and B).

Alternative: Zone the SAS/FC switch for new storage, then connect the new SMW and new boot RAID to the SAS/FC switch in addition to keeping the old SMW and boot RAID connected to the SAS/FC switch.

8.4 Discover XC System Hardware

Use these procedures in the order shown to discover XC system hardware.

1. [Start a Typescript File](#)
2. [Bootstrap Hardware Discovery](#) on page 357
3. [Update Firmware](#) on page 359
4. [Discover Hardware and HSN Routing, Prepare STONITH](#) on page 360
5. [\(Optional\) Configure Partitions](#) on page 362
6. [Repurpose Compute Nodes](#) on page 363
7. [Finish Configuring the SMW for the CLE System Hardware](#) on page 363
8. [Enable System Environmental Data Collections \(SEDC\)](#) on page 364

8.4.1 Start a Typescript File

About this task

Sites can make as few or as many typescripts as they deem useful. Cray recommends starting a typescript file at these milestones:

- just before extracting and archiving current configuration information during a software migration
- just before installing a new software release
- just before configuring the newly installed software

Procedure

1. Log in as root to the SMW.
2. (First time only) Create a release directory for the typescript file.

```
smw# mkdir -p /var/adm/cray/release
```

3. Change to the release directory.

```
smw# cd /var/adm/cray/release
```

4. Set a variable equal to today's date.

```
smw# export TODAY=`date +%Y%m%d`
smw# echo $TODAY
```

5. Start a typescript file.

```
smw# script -af ${TODAY}.suffix
```

For *suffix*, substitute a unique string to distinguish among typescript files, such as `install.1` or `update.2`.

6. Change prompt to include a timestamp.

```
smw# PS1="\u@\h:\w \t# "
```

8.4.2 Bootstrap Hardware Discovery

Prerequisites

This procedure assumes that the following information has been gathered. Enter this information in response to system prompts when performing this procedure.

Information needed	Default value
maximum X cabinet size (columns)	There is no default value. Find the X and Y cabinet sizes and the network topology class from Site-dependent Configuration Values in Configuration Values on page 24.
maximum Y cabinet size (rows)	No default value. See above.
network topology class	0 or 2 for Cray XC Series liquid-cooled systems, 0 for Cray XC Series air-cooled systems (XC30-AC, XC40-AC)
boot node name	c0-0c0s0n1
sdb node name	c0-0c0s1n1

About this task

This procedure uses the `xtdiscover --bootstrap` command to collect some basic information that will be used to bootstrap the hardware discovery process. If boot node failover or SDB node failover will be enabled, then when `xtdiscover` asks for the boot node or the SDB node, instead of entering a single node, enter a pair of nodes with a comma between them, for example "c0-0c0s0n1,c0-2c0s0n1." For more detailed information, see the `xtdiscover(8)` man page.

NOTE: (SMW HA only) Hardware discovery is done only on the first SMW. Do not repeat hardware discovery on the second SMW.

Trouble?

- If the `xtdiscover --bootstrap` command is unable to power up the cabinets, try running `xtdiscover --testconfig` and then run `xtdiscover --bootstrap` again.

- If a step in this procedure fails because of a hardware issue, such as a cabinet failing to power up, resolve that issue and then go back to the last successful step in the procedure and continue from there. Do not skip steps or continue out of order.

Procedure

1. Power down the system.

```
smw# xtcli power down s0
Turning off power to cabinet and waiting for confirmation...
```

Component	Flags:	Result
c0-0	noflags :	Success
c0-0c0s0	noflags :	Success
c0-0c0s1	noflags :	Success
c0-0c0s2	noflags :	Success
c0-0c0s3	noflags :	Success

2. Reboot the cabinet controllers (CC), then verify that all CCs are up.

a. Reboot the cabinet controllers.

```
smw# xtccreboot -c all
xtccreboot: reboot sent to specified CCs
smw# sleep 180
```

b. Are all cabinet controllers up now? Repeat this command until all of the cabinet controllers report in.

```
smw# xtalive -a llsysd -l ll s0
The expected responses were received.
```

3. Run `xtdiscover` in bootstrap mode.

```
smw# xtdiscover --bootstrap
```

The system prompts the user to enter the information gathered as a prerequisite to this procedure. Prior to powering on the cabinets, the system prompts the user to disable any blades that should not be powered on.

```
xtdiscover is about to power on the cabinets.
*** IF YOU NEED TO DISABLE COMPONENTS TO AVOID THEM
*** BEING POWERED ON, PLEASE DO SO NOW USING 'xtcli disable'

Please enter 'c' to continue, or 'a' or 'q' to abort [c]:
```

4. Disable any blades that should not be powered on.

If there are any blades or other components to be disabled, open a separate window and disable them (as `crayadm`) there. In this command, replace `cname` with the `cname` of the component to be disabled.

IMPORTANT: For a migration, if any nodes were disabled on CLE 5.2 / SMW 7.2, then disable them now before continuing `xtdiscover --bootstrap`. That information was saved in the output from the `xtshow_disabled` command on the SMW running CLE 5.2 / SMW 7.2.

```
crayadm@smw> xtcli disable cname
```

- Return to the `xtdiscover --bootstrap` window and enter **c** to continue the hardware discovery bootstrap.

```
Please enter 'c' to continue, or 'a' or 'q' to abort [c]: c
```

The `xtdiscover` command proceeds without further prompts.

Trouble? If the `xtdiscover` command fails with the message, The following cabinets were not detected by heartbeat, power cycle the cabinet controller and retry the `xtdiscover --bootstrap` command.

The bootstrap process is now complete. The next task is to discover the Cray system hardware.

8.4.3 Update Firmware

Prerequisites

This procedure assumes that Cray hardware discovery has been completed successfully.

About this task

This procedure first checks whether the firmware of these components (controllers) needs to be updated, then updates the firmware only if there are Revision Mismatches.

all cabinet-level components

cc_mc (CC Microcontroller)
cc_bios (CC Tolapai BIOS)
cc_fpga (CC FPGA)
chia_fpga (CHIA FPGA)

all blade-level components

cbb_mc (CBB BC Microcontroller)
ibb_mc (IBB BC Microcontroller)
anc_mc (ANC BC Microcontroller)
bc_bios (BC Tolapai BIOS)
lod_fpga (LOD FPGA)
node_bios (Node BIOS)
loc_fpga (LOC FPGA)
qloc_fpga (QLOC FPGA)

Procedure

- Update firmware, if any components are not current.



CAUTION: The `xtzap` command is normally intended for use by Cray Service personnel only. Improper use of this restricted command can cause serious damage to the computer system.

Run `xtzap -a` to update all components.

```
crayadm@smw> xtzap -a s0
```

Note that it is possible to update firmware in cabinets or blades only rather than the entire system. For more information, see *XC™ Series System Administration Guide (S-2393)*.

2. Run `xtbounce --linktune` if any components were not current.

Force `xtbounce` to do a `linktune` on the full system before checking firmware again.

full system For a full, unpartitioned system or a fresh install:

```
crayadm@smw> xtbounce --linktune=all s0
```

partitioned For a partitioned system (software update process only, because partition instructions come later in the fresh install process):

```
crayadm@smw> xtbounce --linktune=all p1
crayadm@smw> xtbounce --linktune=all p2
crayadm@smw> xtbounce --linktune=all p3
```

3. Confirm that all components with out-of-date firmware have been updated.

Check firmware again after updating and linktuning those components.

```
crayadm@smw> xtzap -r -v s0
```

8.4.4 Discover Hardware and HSN Routing, Prepare STONITH

Prerequisites

This procedure assumes that the `xtdiscover --bootstrap` command has been run successfully.

About this task

About Hardware Discovery. This procedure uses `xtdiscover` to detect the Cray system hardware components on the system. The `xtdiscover` command confirms some basic information (entered earlier with `xtdiscover --bootstrap`) for the hardware discovery process, warns that changes will be made, and then confirms whether to abort or continue. Finally, this command creates entries in the system database to describe the hardware. To display the configuration, use the `xtcli` command after running `xtdiscover`. For more detailed information, see the `xtdiscover(8)` man page.

About STONITH. This procedure prepares STONITH (shoot the other node in the head), a Linux service that automatically powers down a node that is not working correctly. If either boot node failover or SDB node failover will be used, then STONITH needs to be set on the primary blade.

IMPORTANT: The primary boot node and primary SDB node should not be on the same blade. Likewise the secondary boot node and secondary SDB node should not be on the same blade. Four different blades should be used if there are two boot nodes and two SDB nodes.

Trouble? If a step in this procedure fails because of a hardware issue, such as a cabinet failing to power up, resolve that issue and then go back to the last successful step in the procedure and continue from there. Do not skip steps or continue out of order.

Procedure

DISCOVER CRAY SYSTEM HARDWARE

1. Log on to the SMW as `root`, if not already logged in.
2. Run the `xtdiscover` command.

`xtdiscover` will continue until it pauses with instructions to bounce the system in a separate window.

```
smw# xtdiscover
***** xtdiscover started *****
...
...
```

In a separate window, please bounce the system now to continue discovery.

3. If prompted, bounce the system (as `crayadm`) in a separate window.

```
crayadm@smw> /opt/cray/hss/default/etc/xtdiscover-bounce-cmd
```

4. After the `xtbounce` command from the previous step has finished, return to the `xtdiscover` window and enter "c" to continue the hardware discovery.

```
After bounce completes, enter 'c' to complete discovery
or 'q' or 'a' to abort [c]: c
```

5. Commit the results of `xtbounce` to the database.

When asked whether to commit the `xtdiscover` results to the database, enter **y**.

(optional) PREPARE STONITH FOR BOOT NODE AND SDB NODE FAILOVER

6. For sites using boot node failover, set STONITH for the primary boot node's blade.

Skip this step if there will be no boot node failover at this site.

In the example, the primary boot node is `c0-0c0s0n1`, so its blade is `c0-0c0s0`.

```
smw# xtdaemonconfig c0-0c0s0 stonith=true
```

7. For sites using SDB failover, set STONITH for primary SDB node's blade.

Skip this step if there will be no SDB node failover at this site.

In the example, the primary SDB node is `c0-0c2s0n1`, so its blade is `c0-0c2s0`.

```
smw# xtdaemonconfig c0-0c2s0 stonith=true
```

DISCOVER HSN ROUTING CONFIGURATION

- Discover the routing configuration of the high-speed network (HSN).

After `xtdiscover` finishes, run the `rtr` command as `crayadm` to determine the exact configuration of the HSN.

```
smw# su - crayadm
crayadm@smw> PS1="\u@\h:\w \t> "
crayadm@smw> rtr --discover
```

The `rtr` command may produce the following message and prompt. Answer "y" to allow `rtr` to bounce the system in diagnostic mode.

```
rtr:WARNING: No HSN discover info found, Using defaults (100% bandwidth
assumed)
System was not bounced in diagnostic mode, should I re-bounce? y
```

8.4.5 (Optional) Configure Partitions

About this task

This procedure describes how to divide the CLE system into "logical machines" or partitions. By definition, `p0` is the entire system, and `p1` through `p31` are smaller partitions. Each partition must have its own set of boot, `sdb`, and other service nodes and compute nodes to boot the partition. See the `xtcli_part(8)` man page for more details.

NOTE: (SMW HA only) For a partitioned SMW HA system, only the first SMW requires this procedure, because the hardware configuration is stored in a shared MariaDB (formerly MySQL) database.

To add a partition, specify the boot node, `SDB` node, and the components that will be members of the partition. As an example, the following steps show how to add these two partitions to an unpartitioned system (`p0`).

```
partition: p1
boot node: c0-0c0s0n1
sdb node: c0-0c0s1n1
members:
c0-0c0s0,c0-0c0s1,c0-0c0s4,c0-0c0s5,c0-0c0s6,c0-0c0s7,c0-0c0s8,c0-0c0s9,c0-0c0s10
,c0-0c0s11,c0-0c0s12,c0-0c0s15

partition: p2
boot node: c0-0c0s3n1
sdb node: c0-0c0s3n1
members: c0-0c0s2,c0-0c0s3,c0-0c0s13,c0-0c0s14
```

Procedure

- Deactivate `p0`.

```
smw# xtcli part_cfg deactivate p0
```

2. Add a partition.

Note that `-b` identifies the boot node, `-d` identifies the SDB node, and `-m` identifies all members of the partition.

```
smw# xtcli part_cfg add p1 -i /raw0 -b c0-0c0s0n1 -d c0-0c0s1n1 \
-m c0-0c0s0,c0-0c0s1,c0-0c0s4,c0-0c0s5,c0-0c0s6,c0-0c0s7,\
c0-0c0s8,c0-0c0s9,c0-0c0s10,c0-0c0s11,c0-0c0s12,c0-0c0s15
```

3. Activate the new partition.

```
smw# xtcli part_cfg activate p1
```

4. Add and activate a second partition.

```
smw# xtcli part_cfg add p2 -i /raw0 -b c0-0c0s3n1 -d c0-0c0s3n1 \
-m c0-0c0s2,c0-0c0s3,c0-0c0s13,c0-0c0s14
```

```
smw# xtcli part_cfg activate p2
```

8.4.6 Repurpose Compute Nodes

When a compute node is configured for a non-compute role, that node is a *repurposed compute node*. Compute nodes can be repurposed to become service nodes for use as tier2 servers (recommended) or in other capacities. Compute nodes should not be repurposed as service nodes for services that require external connectivity.

Use the `xtcli mark_node` command to repurpose a node in a compute blade. In this example, two compute nodes are being repurposed as service nodes and marked accordingly in the HSS database.

```
crayadm@smw> xtcli mark_node service c0-0c0s2n0,c0-0c0s2n1
```

Note that service nodes can be repurposed as compute nodes as well. In that case, the command would be `xtcli mark_node compute`.

8.4.7 Finish Configuring the SMW for the CLE System Hardware

Prerequisites

This procedure assumes that Cray hardware has been discovered and component firmware has been updated (if needed).

About this task

This procedure contains the final steps of configuring the SMW for the CLE system hardware. Note that a full system is referred to as "s0" here. The term "p0" could have been used, because in this context, the two terms are interchangeable. In contrast, commands that operate on config sets use only the term "p0" when referring to a full system. In the config set context, the terms are not interchangeable.

Procedure

1. Check status on all components.

full system For a full, unpartitioned system:

```
crayadm@smw> xtcli status s0
```

partitioned For a partitioned system, with partitions p1 and p2:

```
crayadm@smw> xtcli status p1
```

```
crayadm@smw> xtcli status p2
```

2. Check routing configuration of the system.

full system For a full, unpartitioned system:

```
crayadm@smw> rtr -R s0
```

partitioned For a partitioned system, with partitions p1 and p2:

```
crayadm@smw> rtr -R p1
```

```
crayadm@smw> rtr -R p2
```

Note that the `rtr -R` command produces no output unless there is a routing problem.

3. Examine the hardware inventory and verify that all nodes are visible to the SMW.

full system For a full, unpartitioned system:

```
crayadm@smw> xthwinv s0 > xthwinv.out
```

```
crayadm@smw> xthwinv -x s0 > xthwinv.xml
```

partitioned For a partitioned system, with partitions p1 and p2:

```
crayadm@smw> xthwinv p1 > xthwinv.p1.out
```

```
crayadm@smw> xthwinv -x p1 > xthwinv.p1.xml
```

```
crayadm@smw> xthwinv p2 > xthwinv.p2.out
```

```
crayadm@smw> xthwinv -x p2 > xthwinv.p2.xml
```

4. Check microcontroller information.

Execute the `xtmcinfo -u` command to retrieve microcontroller information from cabinet control processors and blade control processors. Ensure that all blade controllers have output and show similar uptime values.

```
crayadm@smw> xtmcinfo -u s0
```

5. Exit from crayadm back to root account.

```
crayadm@smw> exit  
smw#
```

8.4.8 Enable System Environmental Data Collections (SEDC)

SEDC is a tool that collects and reports in real time the environmental data on all Cray systems. Data includes information from sensors located on significant hardware components at the cabinet and blade level, such as power supplies, processors, memory and fans. SEDC refers to these sensors as *scan IDs*. Examples of collected data include cabinet and blade/node temperatures, voltage, current, power, cooling system air pressure, humidity, and statuses. At the node level, data is collected only from the nodes that are powered on.

Use this command to enable SEDC data collection after a fresh install or after XC hardware is connected during a migration:

```
crayadm@smw> sedc_enable_default
```

For information about how SEDC data is stored in the power management database (PMDB) and querying the PMDB for that data, see *System Environmental Data Collections Guide (S-2491)* for release SMW 8.0.UP03.

8.5 Complete CLE Configuration

Now that the migration SMW is connected to XC hardware, use the following procedures in the order listed to complete CLE configuration.

1. [Update and Validate Global Config Set after Migration Switch](#) on page 365
2. [Update and Validate CLE Config Sets after Migration Switch](#) on page 366
3. [Check CLE Hostnames in /etc/hosts File](#) on page 366
4. [Display and Capture all Config Set Information](#) on page 367
5. [Make a Post-config Snapshot using snaputil](#) on page 368
6. [Make a Post-config Backup of Current Global and CLE Config Sets](#) on page 369
7. [Check NIMS Information](#) on page 342

8.5.1 Update and Validate Global Config Set after Migration Switch

Prerequisites

This procedure assumes that the SMW with CLE 6.0 / SMW 8.0 software is connected to XC hardware.

About this task

This procedure updates the global config set without the `--no-scripts` option so that the pre- and post-configuration scripts can execute and process data from this SMW, which is connected to XC system hardware, and the global config set can then be validated.

Procedure

1. Update the global config set.

```
smw# cfgset update global
```

2. Validate the global config set.

```
smw# cfgset validate global
```

8.5.2 Update and Validate CLE Config Sets after Migration Switch

Prerequisites

This procedure assumes that the SMW with CLE 6.0 / SMW 8.0 software is connected to XC hardware.

About this task

This procedure updates the CLE config sets without the `--no-scripts` option so that the pre- and post-configuration scripts can execute and process data from this SMW, which is connected to XC system hardware, and the CLE config sets can then be validated.

Procedure

1. Update the CLE config set(s).

full system Update the config set for a full (unpartitioned) system p0 (in this example, the config set is named p0):

```
smw# cfgset update p0
```

partitioned Update the config set for each partition. For partition p1 (in this example, the config set is named p1):

```
smw# cfgset update p1
```

For partition p2 (in this example, the config set is named p2):

```
smw# cfgset update p2
```

Additional partitions follow the same pattern.

2. Validate the CLE config set(s).

full system Validate the config set for a full (unpartitioned) system p0:

```
smw# cfgset validate p0
```

partitioned Validate the config set for each partition.

```
smw# cfgset validate p1  
smw# cfgset validate p2
```

Additional partitions follow the same pattern.

8.5.3 Check CLE Hostnames in /etc/hosts File

Prerequisites

This procedure assumes that the CLE config set has been created and updated.

About this task

This procedure confirms that the post-configuration callback scripts, which were run when the CLE config set was updated, added the correct host name entries to the `/etc/hosts` file.

Procedure

1. Confirm that host name entries exist in the CLE `/etc/hosts` file for `boot`, `sdb`, `login`, `lnet`, `rsip`, `dvs`, and any other names defined on this system.

full system For a full, unpartitioned system:

```
smw# egrep "boot|sdb|login|lnet|rsip|dvs" /var/opt/cray/\
imps/config/sets/p0/files/roles/common/etc/hosts
```

partitioned For a partitioned system, with partitions `p1` and `p2`:

```
smw# egrep "boot|sdb|login|lnet|rsip|dvs" /var/opt/cray/\
imps/config/sets/p1/files/roles/common/etc/hosts
```

```
smw# egrep "boot|sdb|login|lnet|rsip|dvs" /var/opt/cray/\
imps/config/sets/p2/files/roles/common/etc/hosts
```

Trouble? If any expected host names are missing, proceed to step 2.

2. If any expected host names are missing, do one of the following:

Option	Description
Option 1: Update the config set (preferred)	Use <code>cfgset</code> to update the <code>cray_net</code> configuration service in config set <code>p0</code> and add any missing hostname, hostname alias, or network interface information.
	<pre>smw# cfgset update -m interactive -s cray_net p0</pre>
Option 2: Edit the <code>/etc/hosts</code> file	Add external host names and IP addresses directly to the following file on the SMW. The additional entries and any comments will be retained every time the config set is updated. Do not add them to the <code>/etc/hosts</code> file on a CLE node.
	<pre>smw# vi /var/opt/cray/imps/config/sets/p0/files/roles/\ common/etc/hosts</pre>

8.5.4 Display and Capture all Config Set Information

About this task

This procedure displays all of the configuration settings in a config set and captures them in a typescript file of this software update. It is not required, but it may aid in troubleshooting. Note that the `cfgset search` command

does not search guidance text in the configuration templates and worksheets, so that information will not be included in the output.

Procedure

Display all configuration settings in the CLE and global config sets, and capture them in a typescript file.

full system Display/capture full information (more verbose) for a full, unpartitioned system:

```
smw# cfgset search -l advanced --format full p0 | tee /var/adm/\
cray/release/p0.${TODAY}.fresh_install.advanced.conf.full

smw# cfgset search -l advanced --format full global | tee /var/adm/\
cray/release/global.${TODAY}.fresh_install.advanced.conf.full
```

Display/capture just the settings and values for a full, unpartitioned system:

```
smw# cfgset search -l advanced p0 | tee /var/adm/cray/release/\
p0.${TODAY}.fresh_install.advanced.conf.full

smw# cfgset search -l advanced global | tee /var/adm/cray/release/\
global.${TODAY}.fresh_install.advanced.conf.full
```

partitioned Display/capture full information (more verbose) for a partitioned system, with partitions p1 and p2:

```
smw# cfgset search -l advanced --format full p1 | tee /var/adm/\
cray/release/p1.${TODAY}.fresh_install.advanced.conf

smw# cfgset search -l advanced --format full global | tee /var/adm/\
cray/release/global.${TODAY}.fresh_install.advanced.conf

smw# cfgset search -l advanced --format full p2 | tee /var/adm/\
cray/release/p2.${TODAY}.fresh_install.advanced.conf

smw# cfgset search -l advanced --format full global | tee /var/adm/\
cray/release/global.${TODAY}.fresh_install.advanced.conf
```

8.5.5 Make a Post-config Snapshot using snaputil

About this task

This procedure uses `snaputil` to make an archival snapshot of the system after configuring CLE and before booting the CLE system.

Best Practice. Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

For more information, see [About Snapshots and Config Set Backups during a Migration](#) on page 424.

Procedure

1. List the available snapshots on the system.

```
smw# snaputil list
```

2. Set the `SNAPSHOT` environment variable using the currently booted snapshot name.

```
smw# export SNAPSHOT=$(snaputil list |grep ^cur| awk '{print $2}')
smw# echo $SNAPSHOT
```

Setting a variable for the snapshot name enables better command substitution in later commands dealing with snapshots.

(SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

3. Create a new snapshot.

```
smw# snaputil create ${SNAPSHOT}.postconfig
```

8.5.6 Make a Post-config Backup of Current Global and CLE Config Sets

About this task

This procedure uses the `cfgset` command to create a post-install backup of the global and CLE config sets after configuring CLE and before booting the CLE system.

Procedure

1. Back up the current global config set.

```
smw# cfgset create --clone global global-postconfig-`${TODAY}
```

2. Back up the current CLE config set.

This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

```
smw# cfgset create --clone p0 p0-postconfig-`${TODAY}
```

8.5.7 Check NIMS Information

About this task

This procedure lists NIMS (Node Image Mapping Service) information: which maps are active on the SMW and what NIMS information is stored for each node.

Procedure

1. Check active NIMS maps.

full system For a full, unpartitioned system:

```
smw# cmap list
```

partitioned For a partitioned system, with partitions p1 and p2:

```
smw# cmap list --partition p1
```

```
smw# cmap list -p p2
```

2. Check NIMS information for each node.

full system For a full, unpartitioned system:

```
smw# cnode list
```

partitioned For a partitioned system, with partitions p1 and p2:

```
smw# cnode list -p p1
```

```
smw# cnode list -p p2
```

3. Check NIMS information for each NIMS group.

```
smw# cnode list --filter group=admin
smw# cnode list --filter group=service
smw# cnode list --filter group=login
smw# cnode list --filter group=compute
```

Check any additional NIMS groups that may have been created for Netroot compute and login nodes (typically only when Netroot is used on only a subset of compute and login nodes instead of all of them, so the NIMS compute and login groups cannot be used for that subset).

```
smw# cnode list --filter group=compute_netroot
smw# cnode list --filter group=login_netroot
```

Check any additional NIMS groups that may have been created for DataWarp with Fusion IO SSDs.

```
smw# cnode list --filter group=fio-service
```

Check any additional NIMS groups that may have been created with WLM (workload manager) or other site names.

```
smw# cnode list --filter group=wlm-admin
smw# cnode list --filter group=wlm-service
smw# cnode list --filter group=wlm-login
```

8.6 Boot the CLE System during a Migration

It is now time to complete the first boot of the new CLE software. If there are any problems booting CLE, see the *XC™ Series Boot Troubleshooting Guide (S-2565)* for techniques on how to determine what might be the problem.

To boot CLE and perform post-boot activities, use the following procedures in the order shown.

1. [Boot the Boot and SDB Nodes](#) on page 371
2. [Restore ALPS Files to /alps_shared](#) on page 371
3. [Push Diag Image to Boot Node](#) on page 372
4. [Push Netroot Images to Boot Node](#) on page 373
5. [Push PE Image Root to Boot Node](#) on page 374
6. [Boot the Rest of the System using a Boot Automation File](#) on page 376
7. [Run Tests after Boot is Complete](#) on page 378
8. [Prepare Site and Software Revision Information Reporting using xtgetrev and xtshowrev](#) on page 379
9. [Test xtdumpsys and cdump](#) on page 380
10. [Make a Post-boot Snapshot using snaputil](#) on page 382
11. [Make a Post-boot Backup of Current Global and CLE Config Sets](#) on page 383

8.6.1 Boot the Boot and SDB Nodes

About this task

Boot the boot and SDB nodes first. The boot session will be continued later, after image roots have been pushed to the boot node and any ALPS files (apschedNextId and apschedPDomain) have been restored. .

NOTE: Some large sites run `xtbounce` first to ensure that a clean bounce happens, and then they suppress having `xtbootsys` call `xtbounce`.

Procedure

Boot the boot and SDB nodes.

Use the `auto.bootnode+sdb` boot automation file, which will end after the boot and SDB nodes have been booted.

```
smw# su - crayadm
crayadm@smw> xtbootsys -a auto.bootnode+sdb
```

8.6.2 Restore ALPS Files to /alps_shared

Prerequisites

This procedure assumes that the ALPS files mentioned have been archived from the CLE 5.2 / SMW 7.2 system.

About this task

If the following files were saved from the CLE 5.2 / SMW 7.2 system, then restore them to the `/alps_shared` file system on the SDB node of the CLE 6.0 / SMW 8.0 system now.

Procedure

1. To have ALPS apids continue to increase, restore this file.

From the CLE 5.2 / SMW 7.2 system:

```
/ufs/alps_shared/apschedNextId
```

To the CLE 6.0 / SMW 8.0 system:

```
/alps_shared/apschedNextId
```

2. To preserve the ALPS protection domains, restore this file.

From the CLE 5.2 / SMW 7.2 system:

```
/ufs/alps_shared/apschedPDomain
```

To the CLE 6.0 / SMW 8.0 system:

```
/alps_shared/apschedPDomain
```

8.6.3 Push Diag Image to Boot Node

Prerequisites

This procedure assumes that the diag image root has been built and the boot node has been booted.

About this task

The online diagnostics image provides some useful tools that are made available on CLE nodes through the Cray Image Binding service using the profile for the diag image root. This procedure describes how to push the diag image root to the boot node.

Procedure

1. Determine the name of the image root used in the diag profile in `cray_image_binding`.

In this example, `p0` is the name of the CLE config set.

```
smw# cfsset search -t image -s cray_image_binding p0 | grep diag
cray_image_binding.settings.profiles.data.diags.image: diags_cle_6.0up03_sles_12_x86-64
```

2. Check for an existing diag image root.

```
smw# image list | grep diag
diag-all_cle_6.0up03_sles_12_x86-64
```

3. Push the diag image root to the boot node.

In this example, the diag image root is `diags_cle_6.0up03_sles_12_x86-64`.

```
smw# image push -d boot diags_cle_6.0up03_sles_12_x86-64
```

Trouble? If passwordless `ssh` has not been prepared between `root@smw` and `root@boot`, then the system will prompt for the password for `root@boot` twice.

8.6.4 Push Netroot Images to Boot Node

Prerequisites

This procedure assumes the following:

- The boot node is booted.
- Netroot images have been built using `imgbuilder`, and the output of that command provided the specific image name that needs to be pushed to the boot node.

Procedure

1. Check for existing Netroot image roots for both `compute-large` and `login-large`.

```
smw# image list | grep "^compute-large"
compute-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222

smw# image list | grep "^login-large"
login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

2. (Optional) If both image names have a common string, set an environment variable for it.

Using the output generated by one of the image list commands in step 1, set an environment variable to represent the common string appearing in both image names. This example assumes that the common string is everything that follows `-large`. If this is not the case (for example, if the date-time stamp is different), creating an environment variable may not be worthwhile.

```
smw# export BASEIMAGE=cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

3. Push the Netroot images to the boot node.

Note that these commands may take 10 minutes or more to complete.

full system For a full, unpartitioned system:

```
smw# image push -d boot \
compute-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222

smw# image push -d boot \
login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

If an environment variable was defined that applies to both image names, use these commands instead:

```
smw# image push -d boot compute-large_$BASEIMAGE
smw# image push -d boot login-large_$BASEIMAGE
```

partitioned For a partitioned system, with partitions p1 and p2:

```
smw# image push -d boot-p1 \  
compute-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

```
smw# image push -d boot-p1 \  
login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

```
smw# image push -d boot-p2 \  
compute-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

```
smw# image push -d boot-p2 \  
login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

If an environment variable was defined that applies to both image names, use these commands instead:

```
smw# image push -d boot-p1 compute-large_$BASEIMAGE  
smw# image push -d boot-p1 login-large_$BASEIMAGE
```

```
smw# image push -d boot-p2 compute-large_$BASEIMAGE  
smw# image push -d boot-p2 login-large_$BASEIMAGE
```

Trouble? If passwordless `ssh` has not been prepared between `root@smw` and `root@boot`, then the system will prompt for the password for `root@boot` twice.

4. Push custom Netroot image roots to boot node.

If any custom image roots were created with Netroot content or something that will be used by a profile in `cray_image_binding`, push that image root to the boot node. Installed workload managers (WLM) will have such custom images if login Netroot is included in the WLM recipe for the login nodes. For example, if a WLM ('wlm') is installed, there will be `wlm-login` and `wlm-admin` or `wlm-service` image roots created. However, only if `wlm-login-large` was created as a Netroot image root will it need to be pushed to the boot node.

a. Check for existing custom Netroot image roots.

This example shows checking for 'wlm' image roots. Substitute the name for the particular WLM used in this system.

```
smw# image list | grep wlm  
wlm-login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

b. Push custom Netroot image roots to the boot node, if any were found.

This example uses the output of the previous substep as the image name. Substitute the image name(s) displayed in the output of the `image list` command for this system. If this image name contains the same base string as the images from step 1, and an environment variable was defined, that can be substituted for everything after 'wlm-login-large' in this push command.

```
smw# image push -d boot \  
wlm-login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

8.6.5 Push PE Image Root to Boot Node

Prerequisites

This procedure assumes that the PE image root has been built and the boot node has been booted.

About this task

This procedure identifies the PE image root and pushes it to the boot node, which can take about 10 minutes.

Note that although the PE image name has 'compute' in it, the same image is also used for login nodes.

Procedure

1. Determine the name of the image root used in the PE profile in `cray_image_binding`.

In this example, `p0` is the name of the CLE config set.

```
smw# cfgset search -t image -s cray_image_binding p0 | grep PE

INFO - Checking services for valid YAML syntax
INFO - Checking services for schema compliance
# 1 matches for 'image' from cray_image_binding_config.yaml
#-----
-
cray_image_binding.settings.profiles.data.PE.image:
pe_compute_cle_6.0up02_sles_12
```

2. Set the `PEIMAGE` variable to the PE image root name found in the previous step.

The `pe_compute_cle_6.0up02_sles_12` name is preconfigured in the Image Binding Service. If setting the `$PEIMAGE` environment variable to a different name for the PE image, update the name in the PE profile of the `cray_image_binding` service for the CLE configuration set so that they match.

```
smw# export PEIMAGE=pe_compute_cle_6.0up02_sles_12
smw# echo $PEIMAGE
```

3. Check for an existing PE image.

```
smw# image list | egrep "^[ ]*$PEIMAGE"
```

4. Push the PE image root to the boot node.

This step can take about 10 minutes.

For `p0`:

```
smw# image push -d boot $PEIMAGE
INFO - Remotely cloning Image '<name of image>' to 'boot'...
INFO - Checking remote destination...
INFO - Transferring Image '<name of image>' to 'root@boot:/var/opt/cray/imps/
image_roots/<name of image>'...
INFO - Cloned Image '<name of image>' to remote host 'root@boot:/var/opt/cray/
imps/image_roots/<name of image>'.
```

For partitioned systems, push to the boot node for that partition, `boot-p1`:

```
smw# image push -d boot-p1 $PEIMAGE
```

8.6.6 Boot the Rest of the System using a Boot Automation File

Prerequisites

This procedure assumes that configuration and image preparation are complete, the boot and SDB nodes have booted, and the rest of the system is now ready to boot.

About this task

This procedure describes how to customize a boot automation file and use it to boot the XC system with `xtbootsys`. For more information about boot automation files, see [About Boot Automation Files](#) on page 423.

For a migration, if a virtual migration SMW was used and a boot automation file was prepared on it, that boot automation file should be copied into place (`/opt/cray/hss/default/etc`).

Procedure

1. Create a site boot automation file.

Copy the Cray generic boot automation file and rename it. Add site customizations, as needed. For sites booting tmpfs images (instead of Netroot) with no SDB node failover, no changes may be necessary.

Replace `hostname` with the host name of the system that will use this automation file.

```
smw# cp -p /opt/cray/hss/default/etc/auto.generic \  
/opt/cray/hss/default/etc/auto.hostname.start
```

2. Create a site automation file for shutting down the system.

Copy the Cray shutdown automation file and rename it. Add site customizations, as needed. For example, customization may be needed to cleanly shut down queues for the workload manager (WLM) on MOM or SDB nodes. The specific commands will vary based on the WLM.

Replace `hostname` with the host name of the system that will use this automation file.

```
smw# cp -p /opt/cray/hss/default/etc/auto.xtshutdown \  
/opt/cray/hss/default/etc/auto.hostname.stop
```

3. If the SDB boot image is too large for a PXE boot (often the case if a WLM is installed in that image), change `auto.hostname.start` to enable booting the SDB node(s) via HSN rather than PXE. See [About Boot Automation Files](#) on page 423 for more information.
4. If boot or SDB node failover is used, add boot node or SDB node failover to `auto.hostname.start`.

If either boot node failover or SDB node failover will be used, then the boot automation file should have a setting to ensure that STONITH has been enabled on the blade that has the primary boot node and the

primary SDB node. The STONITH setting does not survive a power cycle. To maintain the STONITH setting, add these lines to the boot automation file.

Use the blade that contains the primary boot node. For example, if the primary boot node is c0-0c0s0n1, then the blade to use is c0-0c0s0. Add these lines **before** the line for booting the boot node.

```
# Set STONITH for primary boot node
lappend actions {crms_exec "xtdaemonconfig c0-0c0s0 stonith=true"}
```

Use the blade that contains the primary SDB node. For example, if the primary SDB node is c0-0c1s0n1, then the blade to use is c0-0c1s0. Add these lines **before** the line for booting the SDB node.

```
# Set STONITH for primary SDB node
lappend actions {crms_exec "xtdaemonconfig c0-0c1s0 stonith=true"}
```

5. If `cray_login.settings.login_nodes.data.login_prohibited_after_boot` is set to true, then remove the `/etc/nologin` file using one of the following methods:

- Remove it manually. This can be done only after all of the CLE nodes have been booted and the system is ready for users to log in. To choose this option, wait until step 7 on page 378.
- Remove it by means of an entry in the boot automation file. Place an entry like the following after lines that boot all of the compute nodes and after any other special commands in the boot automation file that prepare the system for user access.

```
# Remove /etc/nologin from all service nodes as the last step in the system boot.
lappend actions { crms_exec_on_bootnode "sdb" "pcmd -r -n ALL_SERVICE 'rm /etc/nologin'" }
```

The following boot automation file entry is the equivalent for a CLE 5.2 / SMW 7.2 system:

```
lappend actions { crms_exec_on_bootnode "root" "xtunspec -r /rr/current -d /etc/nologin" }
```

6. Run `xtbootstsys` with `auto.hostname.start`.

Using the `-s last` option for `xtbootstsys` will attempt to continue the last boot session. Because the boot node is already booted, `xtbootstsys` will ask a few questions to confirm that the system should be booted. Here is an example of the additional questions.

full system For a full, unpartitioned system (this example shows the questions that will be asked):

```
smw# su - crayadm
crayadm@smw> xtbootstsys -s last -a auto.hostname.start
12
'grep -e ^enabled=.*yes.* /etc/opt/cray/llm/llm.conf' completed
with status 0
LLM is enabled
'/opt/cray/llm/default/bin/xtlog -t 23 6 xtbootstsys "xtbootstsys llm
check"' completed with status 0
Logging via rsyslog
'xtcli part_cfg show' completed with status 0
INFO: partition defaulting to 'p0'
INFO: id-list defaulting to 'p0'

WARNING: Partition p0 is currently booted.
Bootnode c0-0c0s0n1 c0-0c0s4n1 state is "ready standby"
Continuing will result in a new boot session. Stopping the
previous boot session uncleanly may result in a bad system
```

```

state.
Do you want to Continue ? [yN] y

WARNING: Your idlist does not include a full partition!
Cold booting a partial partition may result in a bad system
state.
Do you want to Continue ? [yN] y
INFO: last boot session is p0-20161222t102014
INFO: debug dir set to /var/opt/cray/debug/p0-20161222t102014
INFO: dump dir set to /var/opt/cray/dump/nodedumps/p0-20161222t102014

```

partitioned For a partitioned system, with partitions p1 and p2:

```
smw# su - crayadm
```

```
crayadm@smw> xtbootsys -s last -p p1 -a auto.hostname.start.p1
```

```
crayadm@smw> xtbootsys -s last -p p2 -a auto.hostname.start.p2
```

The boot automation files have different names in this example to indicate that partitions may use different automation files, if needed.

Trouble? If there are any problems booting CLE, see the *XC™ Series Boot Troubleshooting Guide (S-2565)* for techniques to determine what might be causing the problem.

7. Remove the `/etc/nologin` file manually after the system boots, as needed.

If `cray_login.settings.login_nodes.data.login_prohibited_after_boot` is set to `true`, and the `/etc/nologin` file was NOT removed by means of an entry in the boot automation file in step 5 on page 377, remove it manually after all of the CLE nodes have been booted and the system is ready for users to log in.

```
sdb# pcmd -r -n ALL_SERVICE "rm /etc/nologin"
```

8.6.7 Run Tests after Boot is Complete

Prerequisites

This procedure assumes the following:

- The system has completed booting.
- The compute nodes are "interactive," not under workload manager (WLM) control.
- ALPS is available.

If ALPS is not available and Slurm is used as the WLM, then the compute nodes can be either "interactive" or "batch," and `srun` (the equivalent Slurm command) should be used instead of the `aprun` commands in the steps that follow.

About this task

Log in to the login node as `crayadm`. This can be done from the SMW to the boot node to the login node or directly from another computer to the login node without passing through the SMW and boot node. Then perform these rudimentary functionality checks.

Procedure

1. Run `apstat` to get the number of nodes to use for the following commands.

```
crayadm@login> NUMNODES=$((apstat -v | grep XT | awk "{print \$3}"))
crayadm@login> echo NUMNODES is $NUMNODES
```

2. Verify that all nodes run (from `/tmp`).

```
crayadm@login> cd /tmp
crayadm@login> aprun -b -n $NUMNODES -N 1 /bin/cat /proc/sys/kernel/hostname
```

3. Verify that the home directory is working by running a job.

```
crayadm@login> cd ~
crayadm@login> aprun -b -n $NUMNODES -N 1 /bin/cat /proc/sys/kernel/hostname
```

4. Verify that the Lustre directory is working by running a job.

```
crayadm@login> cd /lustre_file_system
crayadm@login> aprun -b -n $NUMNODES -N 1 /bin/cat /proc/sys/kernel/hostname
```

8.6.8 Prepare Site and Software Revision Information Reporting using `xtgetrev` and `xtshowrev`

Prerequisites

To run `xtgetrev`, the boot node must be booted and accessible.

About this task

System administrators use the `xtgetrev` and `xtshowrev` commands to gather and display machine, software revision, Field Notice (FN), and patch set information. The `xtgetrev` command collects information from the administrator and from the SMW and boot node. The `xtshowrev` command displays that information, even when CLE is not running. These tools are useful for gathering information to send to Cray after installing a software upgrade, FN, or patch set and for help with troubleshooting.

This procedure describes how to use these two tools on a Cray XC Series system. These steps (except for running `xtshowrev`) must be executed as root.

ATTENTION: Any information that is submitted to `site_install_data@cray.com` will be used only within Cray, Inc. and will not be made public. The `xtshowrev` command does not submit any information to Cray automatically.

Procedure

1. Load the module to enable use of the tools.

```
smw# module load xtshowrev
```

2. Run `xtgetrev` to create and populate the initial files.

Only root can run this command. The first time `xtgetrev` is executed, when there are no files populated, the tool will prompt for site information. If the boot node does not have passwordless ssh, then the tool will prompt for the password.

This example uses `CRAY/INTERNAL` as the site name and `9999` as the serial number of the machine. Substitute the actual values for this site.

```
smw# xtgetrev
xtgetrev: No site information has been defined.
```

```
Site name: CRAY/INTERNAL
Serial Number: 9999
System Name [panda1]:
System Type [XC40]:
```

<snip>

Trouble? If `xtgetrev` does not allow entry of those values, it may be because the initial configuration files have been created already. In that case, manually edit `/etc/opt/cray/release/pkginfo/site_config` and modify 'site name:' and 'serial number:' values.

```
smw# vi /etc/opt/cray/release/pkginfo/site_config
```

3. Run `xtshowrev` to see the formatted information.

Note the prompt, which indicates that any user can run this command.

```
user@smw> xtshowrev
Site:                CRAY/INTERNAL
S/N:                 9999
System Type:         XC40
Install Date:        2016-06-01
```

```
<snip>
user@smw>
```

8.6.9 Test `xtdumpsys` and `cdump`

Prerequisites

This procedure assumes that the system has been booted.

About this task

This procedure tests the `xtdumpsys` and `cdump` tools. The example output is for illustrative purposes only. Actual output may differ for the current release.

Procedure

1. Start an `xtdumpsys` typescript.

Start a new window. Start a typescript session for `xtdumpsys` in that new window.

```
smw# su - crayadm
crayadm@smw> export TODAY=`date +%Y%m%d`
crayadm@smw> . /etc/opt/cray/release/cle-release
crayadm@smw> mkdir -p /home/crayadm/dump/${TODAY}_${BUILD}
crayadm@smw> cd /home/crayadm/dump/${TODAY}_${BUILD}/
crayadm@smw> script -af hss.xtdumpsys
```

2. Start `xtdumpsys`.

Start the dump, but do not press **Ctrl-d** until step 5 on page 382. When `xtdumpsys` asks for a dump reason, it will have created the dump directory.

For a full system:

```
crayadm@smw> xtdumpsys
INFO: Beginning dump
INFO: Gathering system partition information
INFO: Gathering system hardware information
INFO: No session specified, defaulting to current.
INFO: Moving temporary log files to the dump directory.
INFO:
#####
INFO: # Your dump is available in /var/opt/cray/dump/p0-YYYYMMDDtHHMMSS-
NNNNNNNNNN #
INFO:
#####
Enter reason for dump:
(an EOF terminates input, usually CTRL-D)
```

For a partitioned system:

```
crayadm@smw> xtdumpsys -p p1
crayadm@smw> xtdumpsys -p p2
```

3. Start a `cdump` typescript in a different window.

Start another window. Start a typescript session for `cdump` in that window.

```
smw# su - crayadm
cdump crayadm@smw> export TODAY=`date +%Y%m%d`
cdump crayadm@smw> . /etc/opt/cray/release/cle-release
cdump crayadm@smw> cd /home/crayadm/dump/${TODAY}_${BUILD}/
cdump crayadm@smw> script -af hss.cdump
```

4. Dump a node with `cdump`.

Change to the directory created in the `xtdumpsys` window (after `INFO: # Your dump is available in`), then use `cdump` to dump a compute node that successfully booted.

```
cdump crayadm@smw> cd /var/opt/cray/dump/p0-YYYYMMDDtHHMMSS-NNNNNNNNNN
cdump crayadm@smw> mkdir cdumps; cd cdumps
```

For a full system (example uses the `c0-0c0s3n0` node):

```
cdump crayadm@smw> cdump -AmD -r xt-hsn@boot c0-0c0s3n0
Wed Mar 1 09:08:08 CDT 2017 start cdump
...
makedumpfile Completed.
- done
Wed Mar 1 09:08:08 CDT 2017 cdump: # of nodes 1
  success 1
  failed 0
  skipped 0
cdump crayadm@smw> exit
```

For a partitioned system, use the host name to specify which boot node. This example uses `boot-p1` to `cdump` the `c0-0c0s4n0` node in the `p1` partition.

```
cdump crayadm@smw> cdump -AmD -r xt-hsn@boot-p1 c0-0c0s4n0
cdump crayadm@smw> exit
```

5. Continue `xtdumpsys`: enter a reason.

After `cdump` completes, return to the `xtdumpsys` window and enter a reason.

```
xtdumpsys window> testdump
```

Then enter an end-of-file (**Ctrl-d**) to end the dump reason.

```
xtdumpsys window> <Ctrl-d>
testdump
INFO: Dump reason:
...
INFO:
#####
INFO: # Your dump is available in /var/opt/cray/dump/
p0-20170301t081927-1304240904 #
INFO:
#####
INFO: No post-processing plugin found at '/etc/opt/cray/dumpsys/
postprocessing.py'
INFO: Example plugins can be found at '/opt/cray/dumpsys/
1.2.5-1.0000.35873.20.1/bin/plugins/examples/postprocessing.py.*'
INFO: Cleaning up
```

```
xtdumpsys crayadm@smw> exit
```

6. Remove dump directory, if desired.

If there are no errors, it is probably safe to delete the dump directory.

```
xtdumpsys crayadm@smw> rm -rf /var/opt/cray/dump/pX-YYYYMMDDtHHMMSS-NNNNNNNNNN
crayadm@smw> exit
```

8.6.10 Make a Post-boot Snapshot using snaputil

About this task

This procedure uses `snaputil` to make an archival snapshot of the system after booting the CLE system.

Best Practice. Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

For more information, see [About Snapshots and Config Set Backups during a Migration](#) on page 424.

Procedure

1. List the available snapshots on the system.

```
smw# snaputil list
```

2. Set the `SNAPSHOT` environment variable using the currently booted snapshot name.

```
smw# export SNAPSHOT=$(snaputil list |grep ^cur| awk '{print $2}')
smw# echo $SNAPSHOT
```

Setting a variable for the snapshot name enables better command substitution in later commands dealing with snapshots.

(SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

3. Create a new snapshot.

```
smw# snaputil create ${SNAPSHOT}.postboot
```

8.6.11 Make a Post-boot Backup of Current Global and CLE Config Sets

About this task

This procedure uses the `cfgset` command to create a post-boot backup of the global and CLE config sets.

Procedure

1. Back up the current global config set.

```
smw# cfgset create --clone global global-postboot-$(TODAY)
```

2. Back up the current CLE config set.

This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

```
smw# cfgset create --clone p0 p0-postboot-$(TODAY)
```

8.7 Complete Post-boot Configuration of Config Services

Prerequisites

This procedure assumes the following:

- the prepared configuration data and software images have been migrated
- the XC system has been booted

About this task

This procedure updates several configuration services that Cray recommends configuring after the XC system has booted.

- `cray_node_health`: if custom node health checker (NHC) plugins are used
- `cray_rur`: if custom Resource Utilization Reporting (RUR) `data_plugins` or `output_plugins` used
- `cray_rsip`: if complex RSIP configuration needed
- `cray_shifter`: if Shifter used

It also restores any workload manager (WLM) spool directories prior to starting a WLM and provides a procedure to apply any site-specific firewall and iptables configuration to CLE nodes. It also restores any workload manager (WLM) spool directories prior to starting a WLM, provides a procedure to apply any site-specific firewall and iptables configuration to CLE nodes, and configures direct-attached Lustre (DAL).

Note that any change to the config set—whether by worksheet, by using the configurator interactively, or by adding/changing files within the config set—must be followed by updating and validating the config set. A procedure is provided to do that in case any configuration services are updated here.

Procedure

1. Prepare fresh worksheets for editing.
 - a. Generate fresh configuration worksheets for the CLE config set (p0 in the example) using prepare mode and the no-scripts option.

```
smw# cfgset update -m prepare --no-scripts p0
```

- b. Copy the new worksheets to the work area that was set up earlier in the process.

This will overwrite the worksheets placed there earlier, but a copy of the originals was saved at that time in case the originals are needed.

```
smw# cp -a /var/opt/cray/imps/config/sets/p0/worksheets/* \  
/var/adm/cray/release/p0_worksheet_workarea
```

2. If this system uses custom node health checker (NHC) plugins, configure them in the `cray_node_health` worksheet now.
 - a. Perform the procedure in [Update cray_node_health Worksheet](#) on page 242.

- b. Import the revised worksheet to the config set.

This command uses the `--no-scripts` option to save time. The config set will be updated and validated after all revised worksheets are imported.

```
smw# cfgset update --no-scripts --worksheet-path \  
/var/adm/cray/release/p0_worksheet_workarea/cray_node_health_worksheet.yaml  
p0
```

3. If this system uses custom Resource Utilization Reporting (RUR) `data_plugins` or `output_plugins`, configure them in the `cray_rur` worksheet now.

- a. Perform the procedure in [Update cray_rur Worksheet](#) on page 273.

- b. Import the revised worksheet to the config set.

This command uses the `--no-scripts` option to save time. The config set will be updated and validated after all revised worksheets are imported.

```
smw# cfgset update --no-scripts --worksheet-path \  
/var/adm/cray/release/p0_worksheet_workarea/cray_rur_worksheet.yaml p0
```

4. If this system requires a complex RSIP configuration, configure it now.

- a. Perform the procedure in [Update cray_rsip Worksheet](#) on page 269.

- b. Import the revised worksheet to the config set.

This command uses the `--no-scripts` option to save time. The config set will be updated and validated after all revised worksheets are imported.

```
smw# cfgset update --no-scripts --worksheet-path \  
/var/adm/cray/release/p0_worksheet_workarea/cray_rsip_worksheet.yaml p0
```

5. If `cray_shifter` was enabled earlier but not fully configured, complete Shifter configuration now.

- a. Perform the procedure in [Update cray_shifter Worksheet](#) on page 286 in conjunction with *XC™ Series Shifter Installation Guide (S-2572)*.

- b. Import the revised worksheet to the config set.

This command uses the `--no-scripts` option to save time. The config set will be updated and validated after all revised worksheets are imported.

```
smw# cfgset update --no-scripts --worksheet-path \  
/var/adm/cray/release/p0_worksheet_workarea/cray_shifter_worksheet.yaml p0
```

6. Before a WLM is started, restore the contents of its spool directory to the appropriate nodes (not the SMW).

To restore the contents of the WLM spool directory that was saved from the CLE 5.2 / SMW 7.2 system to the migration SMW, copy the files in that directory to the boot

node `/nonvolatile/<node cname>/<wlm_spool_directory_path>`, where `<node cname>` is the cname of a WLM server in this system, and `<wlm_spool_directory_path>` is one of the following paths:

- Moab/TORQUE: `/var/spool/moab` and `/var/spool/torque`
- PBS: `/var/spool/PBS`
- Slurm: `/var/spool/slurm`

The WLM servers are listed in the `client_groups` setting of the mount point defined in [Update `cray_persistent_data` Worksheet](#) on page 266 for the WLM spool directory (or directories, in the case of Moab/TORQUE).

7. (Optional) [Apply Site Firewall and IP Tables Configuration via Config Set and Ansible Play](#) on page 386.
8. If any configuration worksheets were changed or files were added to Simple Sync in the previous steps, use this procedure now: [Update and Validate CLE Config Set for Post-boot Changes](#) on page 387.
9. If this system uses direct-attached Lustre (DAL), configure it now.
 - a. [Configure Direct-attached Lustre \(DAL\)](#) on page 388
 - b. (Optional if using DAL) [LMT Configuration for DAL](#) on page 395 (Lustre Monitoring Tool for direct-attached Lustre)

8.7.1 Apply Site Firewall and IP Tables Configuration via Config Set and Ansible Play

Prerequisites

This procedure assumes that the XC system has been booted.

About this task

If any adjustments to the firewall or iptables configuration on CLE nodes are needed, follow these suggestions to make those changes. Adjustments, if they were needed, to the firewall and iptables configuration on the SMW were made earlier in the migration process.

Procedure

1. Make SUSE firewall2 changes, as needed.

To make a change for the SUSE firewall, `SuSEfirewall2`, determine what changes are needed and then make the changes in the config set directory structure in the Simple Sync area of the config set.

- a. Log in to the node with a network interface that needs firewall adjustments.
- b. Compare the files in `/etc/sysconfig/SuSEfirewall2` and `/etc/sysconfig/SuSEfirewall2.d/*` between the archived information from CLE 5.2 / SMW 7.2 to what is on the booted node.

Some differences will be related to the change between SLES11SP2 to SLES12 and some will be related to Cray changes. If it is determined that changes are needed, copy the file to be changed from the node to the SMW and then merge the new information into it.

- c. If there are changes, use Simple Sync to distribute the new file from the previous substep.

Depending on which nodes and how many need the same adjustment, distribute this file to a single node, to an existing node group, or to a new node group. See [Configure Files for Cray Simple Sync Service](#) on page 305 for instructions on where to place the file.
- d. Warm boot the node to test the configuration change and ensure that the desired settings are in place on the node.

If the desired change is not present, check the Ansible logs in `/var/opt/cray/log/ansible/file-changelog-init` and `/var/opt/cray/log/ansible/file-changelog-booted` to see what other Ansible play might have modified the file.

2. Make iptables changes, as needed.

Unlike SuSEfirewall2, changes for iptables must be applied to the running kernel on a node, so this type of change cannot be delivered via Simple Sync with a change to the files in `/etc/sysconfig`. Determine what changes are needed, and then develop an Ansible play to make the changes to the IPV4 packet filter rules in the Linux kernel. For help with writing an Ansible play, see *XC™ Series Ansible Play Writing Guide (S-2582)*.

- a. Login to the node with a network interface that needs iptables adjustments.
- b. Get the current iptables configuration.

Save the output of this command.

```
node# iptables-save
```

- c. Compare the output of running `iptables-save` on this node to the output from running `iptables-save` on the same node when it was running SMW 7.2/CLE 5.2.

This output should have been collected while archiving configuration information from the old software. Some differences will be related to the change from SLES11SP2 to SLES12 and some will be related to Cray software changes.

- d. If it is determined that changes are needed, choose one of these two options to get the changes to the node:
 - Option 1:
 1. Create a file in the same format as the output from `iptables-save` and place it in the Simple Sync directory structure. Depending on which nodes and how many need the same adjustment, distribute this file to a single node, to an existing node group, or to a new node group. See [Configure Files for Cray Simple Sync Service](#) on page 305 for instructions on where to place the file. Different nodes might have different iptables configuration distributed to them via Simple Sync.
 2. Develop a custom Ansible play to run on the desired node(s). This play should run the `iptables-restore` command using the file that was distributed in the first step of this option. A single Ansible play could run on all of the nodes that have special iptables settings, but each node might have a different configuration file passed to `iptables-restore`.
 - Option 2: Develop a custom Ansible play to run on the desired node(s). This play could run the `iptables` command once or several times to apply simple settings to the node. Logic could be added to run different iptables commands on different nodes.
- e. Warm boot the node to test the configuration change and ensure that the desired settings are in place on the node.

If the desired change is not present, check the Ansible logs in `/var/opt/cray/log/ansible/sitelog-init` and `/var/opt/cray/log/ansible/sitelog-booted` to see whether the Ansible play had errors.

8.7.2 Update and Validate CLE Config Set for Post-boot Changes

Prerequisites

This procedure assumes that changes have been made to the CLE config set since the XC system was booted.

About this task

When a config set has new or changed configuration information, it must be updated and validated. This procedure updates the CLE config set and validates it after post-boot changes have been made.

Procedure

1. Update the CLE config set (p0 in the example).

```
smw# cfgset update p0
```

2. Validate the CLE config set (p0 in the example).

```
smw# cfgset validate p0
```

8.7.3 Configure Direct-attached Lustre (DAL)

Prerequisites



CAUTION: As stated in the "Migration Caveats" section of the introduction, this migration process does not include a tested procedure for preserving a DAL file system during migration. If this site wishes to preserve an existing DAL file system, do not use this procedure, because it will reformat the existing DAL file system, and the existing data will be lost.

This procedure assumes the following:

- Service nodes to support direct-attached Lustre® (DAL) have been identified with `xtdiscover` as management server (MGS), metadata server (MDS), or object storage server (OSS) nodes.
 - Configuration worksheets for Cray Linux environment (CLE) have been created and updated for DAL:
 - The `cray_lnet` worksheet is updated and the `cray_lnet.enabled` setting is uncommented and set to `true`. See [Update cray_lnet Worksheet](#) on page 180.
 - The `cray_lustre_client` worksheet is updated and the `cray_lustre_client.enabled` setting is uncommented and set to `true`. See [Update cray_lustre_client Worksheet](#) on page 194.
 - `cray_lustre_server` worksheet is updated and the `cray_lustre_server.enabled` setting is uncommented and set to `true`. See [Update cray_lustre_server Worksheet](#) on page 199
- NOTE:** There are additional settings which tune the Lustre kernel modules.
- If using the Lustre Monitoring Tool (LMT), a MySQL database, storage space for that database, Cerebro, and the LMT GUI must be configured on the MGS node. The `cray_lmt` worksheet includes settings for configuring LMT.

- All DAL service nodes are assigned to the DAL group so that they are assigned the DAL boot image for booting.
- The `imgbuilder` configuration for DAL has the DAL stanza added to the to the default image group.

About this task

This procedure configures direct-attached Lustre (DAL) nodes that provide a Lustre file system.

Procedure

Identify Logical Unit Numbers (LUNs) for DAL

1. Identify the LUNs used for DAL.

Log in to the DAL service nodes to identify the persistent storage device names to be used for the Lustre file system. Identify all disk device names that will be used for the metadata target (MDT) / management target (MGT) and object storage target (OST) devices.

```
smw# ssh boot
boot# ssh dal-mds
```

2. If the LUN number is known, then use the `lsscsi` command to map the LUN to the short disk name.

This example shows that LUN 17 is `/dev/sdr`.

```
dal-mds# lsscsi | grep 17
[0:0:0:17] disk LSI INF-01-00 0786 /dev/sdr
```

3. Use the short disk name from the previous step to determine the long persistent disk device name.

This example shows that `sdr` has two different persistent device names that could be used.



CAUTION: Use persistent device names in the Lustre file system definition. Non-persistent device names (for example, `/dev/sdc`) can change when the system reboots. If non-persistent names are specified in the `fs_name.fs_defs` file, then Lustre may try to mount the wrong devices and fail to start when the system reboots.

For more information about Lustre control utilities, see the `lustre_control(8)` and `lustre.fs_defs(5)` man pages.

```
dal-mds# ls -l /dev/disk/by-id | grep sdr
lrwxrwxrwx 1 root root 9 Aug 4 13:23 scsi-360080e500036ae3e000002e6524a8369 -
> ../../sdr
lrwxrwxrwx 1 root root 9 Aug 4 13:23 wwn-0x60080e500036ae3e000002e6524a8369 -
> ../../sdr
```

Create and Install the Lustre fs_defs File

4. Prepare the Lustre `fs_defs` file on the system management workstation (SMW).

This file is used by `lustre_control` to format, reformat, start, and stop the file system. When creating the Lustre `fs_defs` file in this example, use `/dev/disk/by-id/`

scsi-0x60080e500036ae3e000002e6524a8369 for LUN 17. Refer to the *XC™ Series Lustre® Administration Guide (S-2648)* for detailed information about how to create an `fs_defs` file for a Lustre file system.

5. Create a variable called `FS_NAME` to be the name of the file system using 8 characters or less ("dal" in this example). The file name of the `fs_defs` file should be similar to the file system it defines.

```
smw# export FS_NAME=dal
smw# echo $FS_NAME
dal
```

6. Copy the `example.fs_defs` file to the one named after the DAL file system.

```
smw# cp -p /opt/cray-xt-lustre-utils/default/etc/example.fs_defs \
/home/crayadm/$FS_NAME.fs_defs
```

7. Edit the `$FS_NAME.fs_defs` file. This is a simple example for the p0 partition, which calls the file system "dal" and has the MGT on `nid00027`, MDT on `nid00027` and `nid00029`, first OST on `nid00028`, and second OST on `nid00031`. Substitute site-specific values in this site's `fs_defs` file.

```
smw# vi /home/crayadm/$FS_NAME.fs_defs
```

8. Locate `fs_name: example` and change `example` to the name defined by `$FS_NAME` ("dal" in this example).

```
fs_name: dal
```

9. Set the Lustre server hosts to LNet NIDs mapping.

```
# Lustre server hosts to LNET NIDs mapping.
# Multiple lines are additive.
# Use multiple lines with the same nodes if you have more than one nid for each
# node.
# Nodes and nids can be specified using range expressions. See the
# lustre.fs_defs man page for more information on range expressions.
# 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
```

10. Update the `fs_defs` file with these settings (substituting appropriate site-specific values). Identify which nodes and devices are being used for MGT, MDT, and OSTs. There are other settings in the `fs_defs` file that can be changed, but they are probably acceptable for most sites.

```
## 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-360001fff020021101061ad79111170300
    index=0
ost: node=nid00031
    dev=/dev/disk/by-id/scsi-360001fff020021101061ad7a811170400
    index=1
```

11. Install the `fs_defs` file into the appropriate CLE config_set (p0 in the example).

```
smw# lustre_control install -c p0 /home/crayadm/$FS_NAME.fs_defs
```

The `lustre_control install` command copies the `fs_defs` file into a directory in the config set, makes a `lustre_control` readable version of it with the suffix `.config.data`, and updates the list of installed file systems.

12. Verify that the `fs_defs` file is installed in the config set by listing the files in the `lustre/.lctrl/` directory of the config set.

```
smw# ls /var/opt/cray/imps/config/sets/p0/lustre/.lctrl/
dal.config.data      dal.filesys.data      dal.service.data
dal.failover.data    dal.fs_defs.20160421.1461256838  installed_filesystems
```

Modify the Config Set to Load the `lustre-utils` Module

13. Modify `cray_user_settings.settings.default_modules.data.service` to add `lustre-utils`.

- a. Update the `cray_user_settings` service in config set p0.

```
smw# cfgset update -s cray_user_settings -m interactive -l advanced p0
```

- b. Select the default modules `service` setting (a list of autoloaded modules for non-login service nodes) to configure it.

Enter `2` and press **Enter** to select `service`, then enter `c` and press **Enter** to configure it.

```
Cray User Settings Menu [default: save & exit - Q] $ 2
...
Cray User Settings Menu [default: configure - C] $ c
```

- c. Add the `lustre-utils` module to the list.

Enter `+` to add an entry, then enter "lustre-utils" and press **Enter**. Press **Ctrl-d** to finish adding entries, then press **Enter** to set the entries for this setting.

```
cray_user_settings.settings.default_modules.data.service
[<cr>=set 7 entries, +=add an entry, ?=help, @=less] $ +
Add service (Ctrl-d to exit) $ lustre-utils
Add service (Ctrl-d to exit) $ <Ctrl-d>
...
cray_user_settings.settings.default_modules.data.service
[<cr>=set 8 entries, +=add an entry, ?=help, @=less] $ <cr>
```

- d. Save the changes and exit the configurator.

```
Cray User Settings Menu [default: save & exit - Q] $ Q
```

14. Validate the config set.

- Entire system:

```
smw# cfgset validate p0
```

- Partitioned system:

```
smw# cfgset validate p1  
smw# cfgset validate p2
```

Boot the System and Reformat the DAL File System

The DAL file system must be formatted using `lustre_control` from the boot node after initial set up, and before automating the start up and mounting of the DAL file system.

15. Boot the system.

- If CLE is not booted, proceed to step [16](#) on page 392.
- If CLE is booted, proceed to step [19](#) on page 392.

16. If CLE is not booted:

```
crayadm@smw> xtbootsys -a auto.hostname.start
```

17. Reformat the DAL file system after a full system boot.

```
smw# ssh boot  
boot# export FS_NAME=dal  
boot# lustre_control reformat -f $FS_NAME
```

18. Proceed to step [21](#) on page 393

19. If CLE is booted, run `cray-ansible`, then reboot only the DAL nodes.

Restarting `/etc/init.d/cray-ansible` refreshes the config set cache on the boot node. This example specifies a comma-separated list of `cnames` (for example `c0-0c0s0n0`) for all DAL nodes (MGS, MDS, and OSS) to create a `$DALNODES` variable.

```
boot# /etc/init.d/cray-ansible start
```

Note that the following commands are run as `crayadm`, not `root`.

```
crayadm@smw> export DALNODES=mgsnode,mdsnode,ossnode1,ossnode2  
crayadm@smw> xtbounce -s $DALNODES  
crayadm@smw> xtcli boot DEFAULT $DALNODES
```

20. Reformat the DAL file system after a reboot of only the DAL nodes.

```
smw# ssh boot  
  
boot# module load lustre-utils  
boot# export FS_NAME=dal  
boot# lustre_control reformat -f $FS_NAME  
Continue? (y|n|q) y
```

Start and Mount the DAL File System

21. Start the DAL file system using `lustre_control` on the boot node.

```
boot# lustre_control start -p -f $FS_NAME
```

22. Verify that the Lustre targets are mounted on each DAL node.

```
boot# lustre_control status -f $FS_NAME
```

23. Test mount the DAL file system on a login node.

```
boot# ssh login
login# export FS_NAME=dal
login# mkdir -p /lus/$FS_NAME
```

```
login# mount -t lustre 27@gni:/$FS_NAME /lus/$FS_NAME
```

In the above mount command, substitute the site-specific value for `27@gni`, which is a combination of the nid of the MGT node and the LNet name by which the external Lustre server is accessed (will be something like `gni` or `gni1`). The MGT node nid was defined in the `fs_defs` file in step 10, and the LNet name can be found by searching for "gni" the CLE config set (p0 in this example) on the SMW.

```
smw# cfgset search -t gni -l advanced -s cray_lnet p0
# 2 matches for 'gni' from cray_lnet_config.yaml
#-----
-
cray_lnet.settings.local_lnet.data.lnet_name: gni4
cray_lnet.settings.flat_routes.data.o2ib.src_lnet: gni4
```

Add DAL file system to `cray_lustre_client` Configuration

24. Add the DAL file system to `cray_lustre_client` configuration so that Lustre clients can mount the file system from the Lustre server.

Note that the `cray_lustre_client` service must be enabled in addition to setting information like the settings below (substitute appropriate site-specific values).

```
smw# cfgset update -s cray_lustre_client -l advanced -m interactive p0
```

In the `client_mounts` setting, add two new entries for the DAL file system. One will be for the compute nodes, which can mount the file system at boot time. The other will be for the login node(s). These cannot currently mount the file system at boot time since they are booted before the DAL file system is started. Follow the guidance for the `client_mounts` settings. Set the `mgs_lnet_nids` to the NID number of the MGS (and failover MGS if applicable) followed by `@gni`. Set `mount_at_boot` to `false` for the login node entry and set it to `true` for the compute node entry.

```
cray_lustre_client.settings.client_mounts.data.fs_name.dal_login: null
cray_lustre_client.settings.client_mounts.data.dal_login.lustre_fs_name: dal
cray_lustre_client.settings.client_mounts.data.dal_login.mount_point: /lus/dal
cray_lustre_client.settings.client_mounts.data.dal_login.mgs_lnet_nids:
- 27@gni
- 29@gni
cray_lustre_client.settings.client_mounts.data.dal_login.mount_options:
rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.dal_login.mount_at_boot: false
cray_lustre_client.settings.client_mounts.data.dal_login.client_groups:
```

- login_nodes

```
cray_lustre_client.settings.client_mounts.data.fs_name.dal_compute: null
cray_lustre_client.settings.client_mounts.data.dal_compute.lustre_fs_name: dal
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_point: /lus/dal
cray_lustre_client.settings.client_mounts.data.dal_compute.mgs_lnet_nids:
- 27@gni
- 29@gni
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_options:
rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_at_boot: true
cray_lustre_client.settings.client_mounts.data.dal_compute.client_groups:
- compute_nodes
```

Add DAL File System to cray_lustre_server Configuration

25. Add the DAL file system node groups to the cray_lustre_server service.

```
smw# cfgset update -s cray_lustre_server -l advanced -m interactive p0
```

```
cray_lustre_server.settings.lustre_servers.data.mgs_group: MGS_NODE_GROUP
cray_lustre_server.settings.lustre_servers.data.mds_groups:
- MDS_NODE_GROUP_1
- MDS_NODE_GROUP_2
cray_lustre_server.settings.lustre_servers.data.oss_groups:
- OSS_NODE_GROUP_1
- OSS_NODE_GROUP_2
```

Configure LMT to Monitor DAL

26. (Optional) If using LMT to enable monitoring of DAL, see [LMT_Configuration_for_DAL.ditamap#C320691](#).

Enable Realm-Specific Internet Protocol (RSIP) on DAL Nodes

27. Enable RSIP on DAL nodes so they can communicate with an external LDAP or NIS server.

DAL nodes do not have external network connections, but require access to LDAP or NIS servers external to the system for uid/gid information associated with the Lustre file system.

- a. Add DAL node groups to the list of


```
cray_rsip.settings.service.data.node_groups_as_client.
```

```
smw# cfgset update -s cray_rsip -l advanced -m interactive p0
```

Add the DAL MDS node group(s).

```
cray_rsip.settings.service.data.node_groups_as_client:
- MDS_NODE_GROUP_1
- MDS_NODE_GROUP_2
```

28. Validate the config set.

- Entire system:

```
smw# cfgset validate p0
```

- Partitioned system:

```
smw# cfgset validate p1
smw# cfgset validate p2
```

Update the Boot Automation File for DAL

29. Edit the site boot automation file (in `/opt/cray/hss/default/etc/`) so that the DAL file system is started during the CLE boot.

Because the config set modifications made in an earlier step set it up so that login and elogin nodes do not attempt to mount DAL at boot time, but the compute nodes do, add these DAL lines to the site boot automation file **after** the boot of the service nodes but **before** the boot of the compute nodes.

```
#Boot all the service nodes
lappend actions {crms_boot_all_serv}

# start Lustre server on DAL nodes & mount Lustre filesystem on login nodes
lappend actions { crms_exec_on_bootnode "root" "lustre_control start -f dal" }
lappend actions { crms_exec_on_bootnode "root" "lustre_control mount_clients -
f dal -w login[1-2]" }

#Boot specific compute nodes
#lappend actions [list crms_boot_loadfile DEFAULT compute "c0-0c0s7n0
c0-0c0s7n1" linux]

#Boot compute nodes
lappend actions {crms_boot_all_comp}
```

This uses a `pdsh` style list of nodes as an argument for the `mount_clients` command. For example, `lustre_control` will interpret `login[1-8]` as nodes `login1` through `login8`. Replace `dal` in the command with the name of the DAL file system for this site.

With `client_mounts.data.dal_compute.mount_at_boot` set to `true` in the `cray_lustre_clients` service, the compute nodes automatically mount the DAL file system when they boot. This also ensures that they mount the DAL file system even when rebooted individually, outside the control of the auto boot file.

8.7.4 LMT Configuration for DAL



CAUTION: As stated in the "Migration Caveats" section of the introduction, this migration process does not include a tested procedure for preserving a DAL LMT database during migration. If this site wishes to preserve an existing LMT database, do not use this procedure, because it will result in loss of existing data.

The Lustre® monitoring tool (LMT) for direct-attached Lustre (DAL) on Cray Linux environment (CLE 6.0) requires some manual configuration during the software installation process.

Configure Storage for the LMT Database	At least 40GB of storage space must be made available to the MGS node. See LMT Disk Usage on page 399.
Configure the LMT MySQL Database	The IMPS configuration does not set up this database, so this must be configured manually for CLE 6.0 UP01 and later releases. See Configure LMT MySQL Database for DAL on page 396.
Configure the LMT GUI (Optional)	See Configure the LMT GUI on page 398.

Use the configurator to configure the LMT for DAL on CLE 6.0. Guidance is provided for each LMT configuration setting in the `cfgset` utility.

The `cray_lmt` configurator template configures LMT settings for specific nodes when they are booted. The default system configuration value for the LMT service is disabled (`false`). Log in to the SMW as `root` and use the `cfgset` command to modify the `cray_lmt` configuration settings to configure LMT.

```
smw# cfgset update -s cray_lmt -m interactive CONFIG_SET
```

8.7.4.1 Configure LMT MySQL Database for DAL

Prerequisites

A MySQL server instance must be configured on the management server (MGS) node. All commands described below should be executed on the MGS for the direct-attached Lustre (DAL) file system.

About this task

A MySQL server instance on the management server (MGS) node stores real-time and historical Lustre monitoring tool (LMT) data. The configurator does not handle the initial setup of the LMT MySQL users and database. It must, therefore, be done manually. All commands described below should be executed on the MGS for the DAL file system.

Procedure

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

(Where `nidMGS` is the node ID (NID) of the MGS node.)

```
boot# ssh nidMGS
```

2. Start the MySQL server daemon (if not already running).

```
mgs# /sbin/service mysqld start
```

3. Run the `mysql_secure_installation` script to improve MySQL server instance security.

This sets the password for the `root` MySQL user, disallows remote `root` access to the database, removes anonymous users, removes the test database, and reloads privileges. If this is the first time configuring LMT, create a symlink before running `mysql_secure_installation` to ensure that MySQL uses the correct socket.

- a. Create a symbolic link.

```
mgs# ln -s /var/run/mysql/mysql.sock /var/lib/mysql/mysql.sock
```

- b. Run `mysql_secure_installation` utility.

```
mgs# mysql_secure_installation
```

- c. Respond to script prompts.

Prompts and recommended responses generated by the script.

```
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`.

```
mgs# chmod 600 /usr/share/lmt/mkusers.sql
```

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

This file is not used at run time by LMT or MySQL processes. This script creates the MySQL users on the persistent storage configured for the MySQL databases. After it is run through MySQL, it is no longer needed.

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`.

Edit the GRANT Statement Edit the GRANT statements to grant privileges on only `filesystem_`*fname*.* where *fname* is the name of the file system. This will only grant permissions on the database for the file system being monitored.

Edit the Password Edit the password for `lwatchadmin` by changing `mypass` to the desired password. Also add a password for the `lwatchclient` user.

```

CREATE USER 'lwatchclient'@'localhost' IDENTIFIED BY 'foo';
GRANT SELECT ON filesystem_scratch.* TO 'lwatchclient'@'localhost';

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

FLUSH PRIVILEGES;

```

6. Save the changes and execute the following command. (This prompts for the MySQL `root` user password, which was set when `mysql_secure_installation` was executed.)

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

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

(Where *fname* is the name of the DAL file system.)

```
mgs# lmtinit -a fname
```

LMT data will be inserted into the LMT MySQL database the next time the Cerebro service is restarted on the MGS.

8. Restart Cerebro.

```
mgs# service cerebrod restart
```

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

- a. Initiate the LMT shell.

```
mgs# lmtsh -f fname
```

- b. List tables.

```
fsname> t
```

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

8.7.4.2 Configure the LMT GUI

About this task

The Lustre monitoring tool (LMT) graphical user interface (GUI) package is installed on login nodes. It contains a GUI called `lwatch` and a command-line tool for viewing live data called `lstat`. The configuration file `~/ .lmtrc` must be set up prior to using either tool.

Procedure

1. Login to the MGS node as `root`.
2. 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

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

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

Here is an example for configuring access to the LMT database for the file system named `scratch_1`, which was set up so that the user `lwatchclient` has no password. In this example, access is being configured on the LMT server node, so the database is local. Thus, the `db_host` is `localhost`.

```
filesystem.1.name=scratch_1
filesystem.1.mountname=/lus/scratch_1
filesystem.1.dbhost=localhost
filesystem.1.dbport=3306
filesystem.1.dbuser=lwatchclient
filesystem.1.dbauth=
filesystem.1.dbname=filesystem_scratch_1
```

After setting up `~/ .lmtrc`, `lwatch` and `lstat` can be run on this node. To run the GUI from a remote node, the MySQL database must be configured to allow remote access for the read-only user, `lwatchclient`. See [Configure LMT MySQL for Remote Access](#) on page 398.

8.7.4.3 Configure LMT MySQL for Remote Access

In order to run the Lustre monitoring tool (LMT) graphical user interface (GUI) on a separate node from the LMT server, the MySQL server instance (running on the LMT server) must be configured to enable remote access for the LMT read-only user, `lwatchclient`. These MySQL statements can be added to `/usr/share/lmt/mkusers.sql` prior to executing the statements in that file. They can also be executed directly. In these examples, `FSNAME` is the name of the file system being monitored.

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

To execute these statements directly, log on to the DAL MGS node, open a mysql shell as the root MySQL user, and run the statements as follows.

1. Connect to the database as root.

```
mgs# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
...
```

2. Create lwatchclient user.

```
mysql> CREATE USER 'lwatchclient'@'%';
Query OK, 0 rows affected (0.00 sec)
...
```

3. Grant privileges to lwatchclient user.

```
mysql> GRANT SELECT ON filesystem_FSNAME.* TO 'lwatchclient'@'%';
Query OK, 0 rows affected (0.00 sec)
```

This enables the user named lwatchclient to connect from any hostname.

To allow connections from a certain IP address, replace the '%' with an IP address in single quotes.

```
CREATE USER 'lwatchclient'@'10.11.255.252' IDENTIFIED BY 'foo';
GRANT SELECT ON filesystem_FSNAME.* TO 'lwatchclient'@'10.11.255.252';
```

8.7.4.4 LMT Disk Usage

LMT requires at least 40GB persistent storage attached to the LMT server (i.e., the management server (MGS)) to store historical data. If the storage becomes full, data can be deleted from the database using MySQL delete statements.

MySQL Tables

Five tables store general file system statistics. These tables are populated by `lmt_agg.cron` script.

Table 106. General File System Tables

Table Name	On-Disk Growth Rate
FILESYSTEM_AGGREGATE_HOUR	0.8 KB/hour
FILESYSTEM_AGGREGATE_DAY	0.8 KB/day
FILESYSTEM_AGGREGATE_WEEK	0.8 KB/week
FILESYSTEM_AGGREGATE_MONTH	0.8 KB/month
FILESYSTEM_AGGREGATE_YEAR	0.8 KB/year

Table 107. MDS Aggregate Tables and Growth Rates

Table Name	Approximate On-Disk Growth Rate
MDS_AGGREGATE_HOUR	0.5 KB/hour/MDS

Table Name	Approximate On-Disk Growth Rate
MDS_AGGREGATE_DAY	0.5 KB/day/MDS
MDS_AGGREGATE_WEEK	0.5 KB/week/MDS
MDS_AGGREGATE_MONTH	0.5 KB/month/MDS
MDS_AGGREGATE_YEAR	0.5 KB/year/MDS

Table 108. OST Aggregate Tables and Growth Rates

Table Name	On-Disk Growth Rate
OST_AGGREGATE_HOUR	0.7 KB/hour/OST
OST_AGGREGATE_DAY	0.7 KB/day/OST
OST_AGGREGATE_WEEK	0.7 KB/week/OST
OST_AGGREGATE_MONTH	0.7 KB/month/OST
OST_AGGREGATE_YEAR	0.7 KB/year/OST

Calculate Expected Disk Usage for a File System

Use this formula to calculate the approximate rate of disk space usage for a file system. Disregard the AGGREGATE tables as they grow so much more slowly than the raw data tables.

$$(56 \text{ KB/hour/filesystem}) * (\# \text{ of filesystems}) + (1000 \text{ KB/hour/MDS}) * (\# \text{ of MDSs}) \\ + (44 \text{ KB/hour/OSS}) * (\# \text{ of OSSs}) + (70 \text{ KB/hour/OST}) * (\# \text{ of OSTs}) = \text{Total KB/hour}$$

Calculate the Disk Usage for a File System for 1 Year

In this example, LMT is monitoring one file system with one MDS, four object storage servers (OSS), and eight object storage targets (OST). The amount of disk space used by the LMT database to is expected to grow at this hourly rate.

$$56 \text{ KB/hour/filesystem} * 1 \text{ filesystem} + 1000 \text{ KB/hour/MDS} * 1 \text{ MDS} \\ + 44 \text{ KB/hour/OSS} * 4 \text{ OSSs} + 70 \text{ KB/hour/OST} * 8 \text{ OSTs} = 1792 \text{ KB/hour}$$

Which translates to this yearly rate.

$$1792 \text{ KB/hour} * 24 \text{ hours/day} * 365 \text{ days/year} * 1 \text{ MB}/1024\text{KB} \\ * 1 \text{ GB}/1024\text{MB} = 15 \text{ GB / year}$$

8.8 Install and Configure Additional Software

This is the final installation/configuration of the migration process. These procedures complete the configuration of DataWarp, install a workload manager, and continue the migration of an SMW HA system. The features are optional, but these procedures are required for any feature in use at this site.

1. [Complete DataWarp Configuration](#) on page 401
2. [Install and Configure a Workload Manager \(WLM\)](#) on page 403

3. (SMW HA only) The migration of the first SMW in an SMW HA system is complete. To complete migration of the full SMW HA system, proceed with the standard SMW HA fresh install process, beginning with section 2.3 "Prepare to Install SMW HA Software" in *XC™ Series SMW HA Installation Guide (SLEHA12.SP0.UP03) S-0044*.
 - a. Install SMW HA software on the first SMW.
 - b. Install SLE, SMW, CLE, and SMW HA software on the second SMW.
 - c. Configure the SMW HA cluster.
 - d. Configure the power management database (PMDB) on a DRBD device.

8.8.1 Complete DataWarp Configuration

Prerequisites

This procedure assumes the following (procedures for these tasks were provided in earlier phases of the migration process):

- The necessary configuration services have been modified to enable DataWarp in the config set.
- The DataWarp service configuration database file, `dwstat.db`, from CLE 5.2.UP04 was archived and is available to copy to the CLE 6.0 system.
- Output from the command `dwstat` was saved to `52dwstat.output` from the CLE 5.2.UP04 system and is available to copy to the CLE 6.0 system.
- For sites using DataWarp with Fusion IO SSDs, those `fio`-service images have been built and assigned to the nodes with Fusion IO SSDs.
- For sites using DataWarp with Intel SSDs, the regular service image includes support for the Intel SSDs and has been assigned to service nodes with Intel SSDs.
- The XC system has completed its first boot.

About this task

After the XC system boots, sites that use DataWarp can use the following procedures to complete DataWarp installation and configuration.

Procedure

1. Ensure that all Intel P3608 cards have been over-provisioned.

If this site reformatted/over-provisioned Intel P3608 SSD cards as directed in FN6121a *Datawarp - Performance Issues* under CLE 5.2, verify that the Intel P3608 SSD service node has the correct value set. (Note that `0xba4d3a1f` is equivalent to `3125623327`, which is the correct value.)

NOTE: Sites with a large number of SSD server nodes, use the `pdsh` command to group the following commands and issue to all Intel P3608 SSD-enabled nodes.

```
nid00350# module load linux-nvme-ctl
nid00350# nvme get-feature /dev/nvme0 -n 1 -f 0XC1 --sel=0
getfeature:193 (Unknown), value:0xba4d3a1f
```

If this site has NOT reformatted/over-provisioned Intel P3608 SSD cards, then do that now using procedure 7.1 "Over-provision an Intel P3608 SSD" in *XC™ Series DataWarp™ Installation and Administration Guide (CLE 6.0.UP03) S-2564*.

2. Update the firmware for all Fusion IO cards.

If this system has Fusion IO SSD hardware, then update the firmware of all Fusion IO cards now.



CAUTION: Once updated, the firmware revision cannot be reverted to the previous version, so the SSDs will NOT be usable in a CLE 5.2 / SMW 7.2 system.

Use procedure 7.2 "Update Fusion ioMemory Firmware" in *XC™ Series DataWarp™ Installation and Administration Guide (CLE 6.0.UP03) S-2564*.

IMPORTANT: Repeat steps 1 and 2 for each SSD-endowed node that the Cray DataWarp Service (DWS) manages, that is, those added to the node group "managed_nodes_groups," which was defined in [Update cray_dws Worksheet](#) on page 171.

3. Restore DataWarp service configuration for nodes and pools.

The `dwsd.db` file and the output of the `dwstat` command from the SDB node were archived in [Preservation of Other Data Prior to Final Shutdown](#) on page 350. Locate those items from the CLE 5.2 archive area.

- a. Copy the file archived CLE 5.2 `dwsd.db` file to the SDB node and rename it `/var/opt/cray/dws/52dwsd.db`.

After that file has been copied, verify that it is there.

```
sdb# cd /var/opt/cray/dws
sdb# ls
boot_session_cache  dwsd.db  dwsd.pid  log  52dwsd.db
```

- b. Convert the `52dwsd.db` file to JSON format.

```
sdb# module load dws
sdb# dwbackup --state-file=52dwsd.db > 52dwsd.json
```

- c. Stop the `dwsd` daemon that runs on the SDB node.

```
sdb# systemctl stop dwsd
```

- d. Back up and read in the CLE 5.2 DataWarp configuration.

This step is a fail-safe measure to ensure compatibility in case the format of the database file has changed since the earlier release.

```
sdb# dwbackup >OLD60dwsd.json
sdb# systemctl start dwsd
sdb# dwcli config restore <52dwsd.json
note: creating pool 'wlm_pool' with granularity=200GiB and units=bytes
pool add progress [=====] 1/1 100% done
```

- e. Back up the new DataWarp configuration.

```
sdb# dwbackup > /persistent/storage/dwsd_backup_<DATE>.json
```

- f. Verify DataWarp configuration and compare with the archived CLE 5.2 `dwstat` output (52dwstat.output).

In this example, the CLE 6.0 output of `dwstat` is saved to `60dwstat.output` to aid in comparison.

```
sdb# dwstat -b nodes pools > dwstat60.output
sdb# cat dwstat60.output
      pool units quantity      free      gran
wlm_pool bytes  5.66TiB 5.66TiB 200GiB

      node      pool online drain  gran capacity insts activs
nid00021 wlm_pool online  fill 16MiB  5.82TiB     0     0

did not find any sessions, instances, scratch configurations, cache
configurations, swap configurations, registrations, activations, fragments,
namespaces
```

Check visually for differences between `dwstat60.output` and `dwstat50.output`. The `dwstat` output for CLE 5.2 is different than for CLE 6.0, but ensure that the pool names, size, and granularity are consistent. Also check that the nodes are in the same pool, capacity, and granularity, that online is shown as "online" (was "true" in CLE 5.2), and that drain is set to "fill" (was "true" in CLE 5.2).

If there are differences that need to be adjusted, then reference these procedures in *XC™ Series DataWarp™ Installation and Administration Guide (CLE 6.0.UP03) S-2564* to make the adjustments:

- 7.4 "Create a Storage Pool"
- 7.5 "Assign a Node to a Storage Pool "

8.8.2 Install and Configure a Workload Manager (WLM)

Cray XC Series systems support the use of workload manager (WLM) software products. The SMW 8.0.UP03 release supports these three WLM products: PBS, Moab/TORQUE, and Slurm. Each product requires installation and configuration prior to use. For a migration, check with the WLM vendor to find out what version of their product supports SLES 12 or both SLES 11 SP3 and SLES 12.

PBS Professional™

PBS Professional is a commercial product licensed by Altair Engineering, Inc.

- For general product information: <http://www.altair.com>
- For PBS Professional documentation: <http://www.pbsworks.com/PBSProductGT.aspx?n=PBS-Professional&c=Overview-and-Capabilities&d=PBS-Professional,-Documentation>
- Note that PBS Professional uses a license manager, which requires a network connection between the license server and the SDB node on a Cray system.

Moab™ and TORQUE

Moab and TORQUE are commercial products licensed by Adaptive Computing.

- For product information: <http://www.adaptivecomputing.com>
- For a CLE 5.2 to CLE 6.0 migration,

Slurm

Slurm (Simple Linux Utility for Resource Management) is an open source application that is commercially supported by SchedMD, among others.

- For more product information: <http://www.schedmd.com/>
- For Cray-specific installation/configuration instructions: *XC™ Series Slurm Installation Guide (S-2538)*

For the most up-to-date information regarding workload manager software compatibility with CLE releases, look on the CrayPort website at <http://crayport.cray.com>.

8.9 Back Up the Newly Installed and Configured SMW/CLE Software

After installing and configuring the new SMW/CLE software, create a backup of it, if needed.

Dell R815 SMW with software RAID If an R815 SMW was configured using the recommended software RAID1 configuration for the boot disk, there is no need to make a backup of the SMW boot disk because the boot disks are mirrored automatically by the RAID software.

Dell R630 SMW with software RAID If an R630 SMW was configured using the recommended hardware RAID5 virtual disk, there is no need to make a backup of the SMW boot disk. Any one of the four disks in the RAID5 configuration can fail without losing any data.

8.10 Back Up Site Data

This procedure helps sites identify and back up important data from the SMW, boot, and SDB nodes. Back up site data before and after installing new software, depending on circumstances and site policy.

Before installation When a fresh install is performed on a system, disks are wiped clean. Before beginning any installation procedures, back up configuration files, log files, or other files that need to be preserved. The migration process already includes procedures to do this for a CLE 5.2 / SMW 7.2 system.

After installation Sites may also want to archive important SMW and CLE information even if there are no immediate plans to install or reinstall a software release. Saving such information elsewhere will make a later reinstall easier, whether it is planned or part of disaster recovery.

What data should be saved at a particular site depends on several things, such as what is currently installed and where data is stored. A site might have CLE 5.x / SMW 7.x installed, or it might already have CLE 6.0.x / SMW 8.0.x installed and is now planning to do a fresh install and wants to reuse configuration data files. The information to save would be different in each case. And there could be site data in home directories or other parts of the file system unknown to Cray and therefore not listed here. The following suggestions about what data to preserve assume a reinstallation of a CLE 6.x / SMW 8.x release that wipes out an earlier installation of that release.

SMW Data to Save from a CLE 6.x / SMW 8.x release

SMW Configuration Data

`/var/opt/cray/imps`

Save the entire directory, which has global config sets (`/var/opt/cray/imps/global`) and CLE config sets (`/var/opt/cray/config/sets/p0`). Saving only the worksheet YAML would miss any site files added for distribution by simple sync or any site Ansible plays. Of particular importance in the global config set is `cray_bootstrap_config.yaml` (or

<code>/etc/</code>	cray_bootraid_worksheel.yaml) which describes how the storage on the Boot RAID is being used.
<code>/opt/cray/hss/default/etc</code>	Save the entire directory. Information related to image recipes is stored in <code>/etc/opt/cray/imps/image_recipes.d</code> (especially any site changes to <code>image_recipes.local.json</code>) and <code>/etc/opt/cray/imps/package_collections.d</code> .
<code>/var/opt/cray/repos</code>	Save the boot automation files (<code>/opt/cray/hss/default/etc/auto.*</code>) and any other files with custom settings.
<code>/home/crayadm/*fs_defs</code>	Save any site repos which have been created in this directory.
<code>/var/adm/cray/release/pe/install-cdt.yaml</code>	Save this file if direct-attached Lustre (DAL) was configured.
Command output	Save the PE installer YAML configuration file.
	Save output from these commands:
	<ul style="list-style-type: none"> • Are any nodes disabled?
	<pre>smw# xtcli status s0</pre>
	<ul style="list-style-type: none"> • What are the boot and SDB nodes and are any CLE partitions present?
	<pre>smw# xtcli part_cfg show</pre>

SMW Operational Data

<code>/home</code>	Save any user data in this directory, especially in <code>/home/crayadm</code> .
<code>/var/opt/cray/disk/1</code>	Save all files in this directory, which has logs, dumps, and debugging information.
<code>/var/opt/cray/imps/image_roots</code> and <code>/var/opt/cray/imps/boot_images</code>	No need to save data in these two directories as long as the image recipes are saved, because these files can be rebuilt from the image recipes. And when they are rebuilt, they can be pushed to the boot node or CMC (for eLogin).
<code>/var/lib/mysql</code>	Perform a <code>mysql dump</code> of <code>/var/lib/mysql</code> . This data will be regenerated by rerunning <code>xtdiscover</code> .

CLE Data to Save from a CLE 6.x / SMW 8.x release

CLE Boot Node Data

<code>/var/opt/cray/imps</code>	No need to save the files in this directory. They are all copies of files on the SMW.
---------------------------------	---

`/non-volatile`
and `/cray_home` Save the data in these two directories for possible restoration after the fresh install.

CLE SDB Node Data

`/alps_shared`
and `/var/lib/mysql` No need to save the data in these two file systems. It will be regenerated at the first boot with the newly installed software. The only side effect is that all ALPS apids will start over at apid 100.

8.11 Restore Operational Data during a Migration

Prerequisites

This procedure assumes that operational and user data were archived earlier in the migration process.

About this task

Some of the operational data is in log files that can be moved into position before starting RSMS daemons or before booting CLE for the first time so that they will be available for log analysis as needed by the system administrator.

Procedure

1. Load simple log files.

- a. Move simple files from archive to new location on the CLE 6.0 / SMW 8.0 system.
- b. Restore SEC logs.

If this site wishes to retain any SEC historical reference logs from running `check-xt` to gather system status data and statistics, then restore the contents of the SMW `/var/log/check_xt` directory from the CLE 5.2 / SMW 7.2 archive to the same location on the SMW running CLE 6.0 / SMW 8.0.

c. Restore other archived logs, as needed.

- Files from the current boot session in `smw:/var/opt/cray/log/p0-current` (for partition p0).
- Files from earlier boots in `smw:/var/opt/cray/log/*`.
 - `xthwerrlogd` files
 - `netwatch` files (`netwatch.p0-SESSIONID`)
 - `network link resiliency` files (`nldr-YYYYMMDD`)
 - `pcimon` files (`pcimon-YYYYMMDD*`)
 - `RUR` data (contained in the `messages-YYYYMMDD` file)
- Files from `/var/opt/cray/log`.
 - `smwmessages-YYYYMMDD` (information about the SMW hardware and environmental history)

2. Restore any site user data.

Restore any site user data from the CLE 5.2 / SMW 7.2 system that was archived earlier in the migration process, such as home directories for Linux accounts or workload manager logs.

- SMW `/home` directories for SMW accounts with local home directories.
- CLE directories for CLE accounts with local home directories. These were in `/ufs/home` for CLE 5.2 / SMW 7.2, but are in `/cray_home` for CLE 6.0 / SMW 8.0, unless the location was changed from the default setting.

3. Restore any administrative account files (root, crayadm).

For the root and crayadm accounts, sites may wish to restore SSH keys and `known_hosts` files.

```
/root/.ssh/*  
/home/crayadm/.ssh/*
```

For the crayadm account, sites may wish to restore any `xtdumpsys` plugins in `/home/crayadm/.xtdumpsys-plugin`.

4. Continue porting any site local scripts.

8.12 Roll Back Changes during a Migration

Prerequisites

This procedure assumes the following:

- CLE 6.0.UP03 is booted and running on the Cray XC system.
- A physical migration SMW is being used to perform a migration from CLE 5.2.UP04 / SMW 7.2.UP04 to CLE 6.0.UP03 / SMW 8.0.UP03.

About this task

This procedure provides steps to roll back to the CLE 5.2.UP04 system if problems are encountered that prevent full operational status of the CLE 6.0.UP03 system. On a small system, this procedure takes about two hours to complete.

Nomenclature:

- The SMW and boot RAID from the CLE 5.2.UP04 / SMW 7.2.UP04 system are the "old" SMW and "old" boot RAID, with command prompt "smw-old."
- The physical migration SMW and boot RAID for the CLE 6.0.UP03 / SMW 8.0.UP03 system are the "new" SMW and "new" boot RAID, with command prompt "smw-new."

IMPORTANT: If this site updated DataWarp Fusion ioMemory3/SX300 cards for the migration to CLE 6.0 / SMW 8.0, the driver firmware cannot be reverted to the previous version, so the SSDs will NOT be usable in a CLE 5.2 / SMW 7.2 system.

Note that some commands in this procedure must be executed as `crayadm`, and some commands must be executed as `root`.

Procedure

1. Shut down CLE using a boot automation script.

This example uses `auto.xtshutdown`; substitute the name of the automation file used at this site, if different (for example, `auto.hostname.stop`).

```
smw-new# su - crayadm
crayadm@smw-new> xtbootsys -s last -a auto.xtshutdown
```

2. When shutdown is complete, power off the XC system.

```
crayadm@smw-new> xtcli power down s0
```

3. (SMW HA only) Put the new SMW HA cluster into maintenance mode.

After CLE has been shut down, put the cluster into maintenance mode before disconnecting any cables. This will ensure that SMW HA takes no action if it detects loss of connectivity. This is a cluster-wide action and needs to be done on only one SMW.

```
crayadm@smw-new> exit
smw-new# maintenance_mode_configure enable
```

4. When power-down is complete, recable the SMWs.

Recable the SMWs to disconnect the new SMW (or SMWs, in the case of an SMW HA system) from the XC system and reconnect the old SMW(s) to it.

- a. Remove the eth1 and eth3 connections of the new SMW from the Ethernet switch and connect eth1 and eth3 of the old SMW to the Ethernet switch.
- b. Recable the old and new boot RAID(s) to the original state by reversing the actions taken in [Switch Cabling to Migration SMW and Boot RAID](#) on page 354.

5. (SMW HA only) When recabling is complete, take the old SMW HA cluster out of maintenance mode.

For SMW HA systems running SLEHA11SP3, the command is as follows:

```
smw-old# crm configure property maintenance-mode=false 2> /dev/null
```

For SMW HA systems running SLEHA12SP0, the command is this:

```
smw-old# maintenance_mode_configure disable
```

After taking the cluster out of maintenance mode, switch to the `crayadm` account and continue to the next step.

```
smw-old# su - crayadm
```

6. When recabling is complete, check the running image in a cabinet controller (CC).

```
crayadm@smw-old> xtlogin c0-0
(CC) c0-0# cat /image.manifest
Clone Source: /opt/cray/hss-images/master/8.0.3000-73.3
Clone Dest: /opt/cray/hss-images/image-8.0.3000-73.3
Template: /opt/cray/hss-images/templates/8.0.3000-73.3
```

The values above indicate that the CC is still running the SMW 8.0.UP03 controller image.

```
(CC) c0-0# exit  
crayadm@smw-old>
```

7. Reboot the cabinet controllers to load the SMW 7.2.UP04 cabinet controller image, then verify that all CCs are up.

- a. Reboot the CCs (as root).

```
crayadm@smw-old> exit  
smw-old# xtccreboot -c all  
smw# sleep 180
```

- b. Wait until all CCs have rebooted (about three minutes) and they respond to this command:

```
smw-old# xtalive -l cc  
The expected response was received.
```

8. When all CCs are up, check the running image in a cabinet controller (CC) again (as crayadm).

```
smw-old# su - crayadm  
crayadm@smw-old> xtlogin c0-0  
(CC) c0-0# cat /image.manifest  
Clone Source: /opt/cray/hss-images/72UP04PS16  
Clone Dest: /opt/cray/hss-images/72UP04PS17  
Template: /opt/cray/hss-images/templates/7.2.0-1.0702.37275.648
```

The values above indicate that the CC is now running the SMW 7.2.UP04 controller image.

```
(CC) c0-0# exit  
crayadm@smw-old>
```

9. Power up the XC system.

```
crayadm@smw-old> xtcli power up s0
```

Wait for the power-up to complete.

10. When the power-up is complete, update the SMW 7.2.UP04 firmware to the XC system.

```
crayadm@smw-old> xtzap -a -f s0
```

11. When `xtzap` is complete, bounce and linktune the XC system.

```
crayadm@smw-old> xtbounce --linktune=all s0
```

12. Boot the XC system using site-local procedures.

```
crayadm@smw-old> xtbootsys -a auto.hostname.start -c xtbounce=0
```

There is no need to bounce the system again, because this command bounces it cleanly with `-c xtbounce=0`.

9 Supplemental Information

This collection of topics is provided as background information about some aspects of the new Cray management system software referenced in this migration guide.

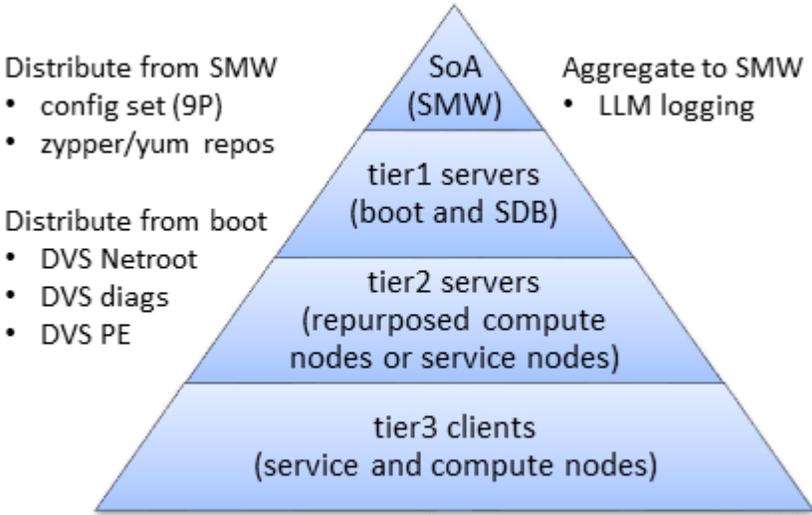
- [About Cray Scalable Services](#) on page 410
- [Cray XC System Configuration](#) on page 412
- [About Config Sets](#) on page 414
- [About Config Set Caching](#) on page 415
- [About Variable Names in the Configurator and Configuration Worksheets](#) on page 416
- [About Node Groups](#) on page 416
- [About Simple Sync](#) on page 419
- [About Boot Automation Files](#) on page 423
- [About Snapshots and Config Set Backups during a Migration](#) on page 424
- [Install Third-Party Software with a Custom Image Recipe](#) on page 424
- [Prefixes for Binary and Decimal Multiples](#) on page 430

9.1 About Cray Scalable Services

Cray Scalable Services is an essential part of the Cray Management System that is used to both distribute and aggregate information. Within Cray Scalable Services, nodes are designated as SoA (server of authority), tier1, tier2, or tier3. A node can be a member of only one of these groups. Tier1 nodes are clients of the SoA and servers for tier2 nodes. Tier2 nodes are clients of tier1 nodes and servers for tier3 nodes. Tier3 nodes are clients of tier2 nodes. Configuration of nodes as SoA, tier1, and tier2 is defined in the `cray_scalable_services` configuration service, which must be configured properly for the system to function.

As indicated in this figure, the SMW is the designated SoA in Cray XC systems. The boot and SDB nodes are designated tier1 nodes, and they must have direct network connectivity to the SMW via Ethernet. Typically, tier2 nodes are service nodes or repurposed compute nodes that have no other duties beyond being part of the Scalable Services. All other nodes are tier3 nodes.

Figure 17. Cray Scalable Services



This table shows what gets distributed or aggregated using Cray Scalable Services.

from SMW to rest of system	<ul style="list-style-type: none"> • config set data is shared using a 9P file system and DIOD (distributed I/O daemon) • zypper software repositories can be used from any node with the Live Update feature (http forwarding from the SMW through the tiers)
from boot node to rest of system	<ul style="list-style-type: none"> • PE (Programming Environment) image root • diag (online diagnostics) image root • Netroot image roots¹
from rest of system to SMW	<ul style="list-style-type: none"> • Lightweight Logging Manager (LLM) logging

Here is an example of how Scalable Services works with Live Updates to distribute software out to nodes. Any tier3 node can run zypper to access the repositories on the SMW because it has an entry in `/etc/zypp/repos.d/liveupdates.repo` that points to the tier2 nodes by means of a baseurl, which uses http protocol listing all of the tier2 nodes. The tier2 nodes, in turn, have an entry in `/etc/zypp/repos.d/liveupdates.repo` that lists at least one tier1 node. All tier1 nodes have an entry in `/etc/zypp/repos.d/liveupdates.repo` that lists the SMW.

Services that Depend on Cray Scalable Services

It is important to configure Cray Scalable Services correctly. The following features and services use data from the `cray_scalable_services` configuration service, and may they not be functional if `cray_scalable_services` is configured incorrectly.

- Node Image Mapping Service (NIMS) plugin** Uses `cray_scalable_services` data to determine tier1 servers and adds the tier1 kernel command line parameter to each tier1 server.

¹ Netroot is a mechanism that enables nodes booted with a minimal, local in-memory file system to execute within the context of a larger, full-featured root file system which available to the node via a network mount.

IMPS Distribution Service (IDS)	Uses <code>cray_scalable_services</code> data to set the <code>ids</code> kernel command line parameter to the node's parent, from whom it will receive config set data.
DVS Ansible configuration	Uses <code>cray_scalable_services</code> data to determine which nodes should serve DVS file systems. This will also impact Netroot functionality, which uses DVS.
CLE liveupdates functionality	Configured using <code>cray_scalable_services</code> data to determine the parent each node should contact en route to the package repos stored on the SMW.
LLM Ansible configuration	Uses <code>cray_scalable_services</code> data to determine the next server to which a node should send its log data, which depends on the node's tier.
NFS Ansible configuration	Uses <code>cray_scalable_services</code> data to determine which nodes should act as clients and servers.
IP forwarding Ansible configuration	Uses <code>cray_scalable_services</code> data to enable IP forwarding and configure servers' routes depending on their tier.

9.2 Cray XC System Configuration

To configure Cray XC systems and manage configuration content, system administrators use the Cray configuration management framework (CMF). The CMF comprises configuration data, the tools to manage and distribute that data, and software to apply the configuration data to the running image at boot time. Its major components include configuration service packages, config sets, the IMPS distribution service (IDS), the configurator, `cray-ansible`, and Ansible.

Configuration Starts with Configuration Service Packages

Configuration content (data and software) is installed as configuration service packages on the management node of Cray XC systems (in `/opt/cray/imps_config/<service package>/default/configurator` by default). Each service package delivers configuration content for one or more system services. The contents of each service package reside in the following subdirectories:

- ansible** Drop zone for Cray-provided Ansible play content.
- callbacks** Pre- and post-configuration scripts.
- dist** Drop zone for other Cray-provided content, such as static files required for the configuration of a service.
- template** Configuration templates that define the configuration settings to be set and provide some default values. These templates are never modified by administrators or other users.

Configuration service packages are installed for system upgrades and updates as well as for initial installation.

Configuration Information is Stored in Config Sets

Administrators use the `cfgset` command to manage configuration information. It takes configuration content delivered in service packages and invokes the `configurator` tool to combine that content with site-specific configuration content gathered from administrators either interactively or through bulk import. The results are used by `cfgset` to create a configuration set or *config set*. A config set is a central repository that stores all configuration information necessary to operate the system. Config sets reside on the management node (e.g., the

SMW) in `/var/opt/cray/imps/config/sets` by default. The contents of each config set reside in the following subdirectories:

- ansible** Drop zone for local site-provided Ansible play content to be distributed with the config set. When the config set is created, `cfgset` copies Ansible content from service packages to this location. Whenever the config set is updated, `cfgset` copies Ansible content from service packages again, overwriting the previous service-package Ansible content and leaving the site-provided content unchanged.
- changelog** YAML change logs from previous sessions with the configurator.
- config** Configuration templates containing configuration information. When the config set is created, the configurator copies service package templates to this location. Administrators can modify the content of these templates using `cfgset` and the configurator. Whenever the config set is updated, the configurator merges service package templates with the templates in this location.
- dist** Drop zone for other site-provided content, such as static files required for the configuration of a service. When the config set is created, `cfgset` copies dist content from service packages to this location. Whenever the config set is updated, `cfgset` copies dist content from service packages again, overwriting the previous service-package dist content and leaving the site-provided content unchanged.
- files** Files necessary for system configuration that are generated by configuration callback scripts or manually and distributed with the config set (e.g., `/etc/hosts`).
- worksheets** Configuration worksheets generated by the configurator using data stored in the configuration templates in the `config` subdirectory of the config set. Administrators copy these worksheets to a location outside the config set, edit them with site-specific configuration data, and then import them to create a new config set or update an existing one.

An administrator may create multiple config sets to support partitions or alternate configurations. Typically a config set of type `cle` is created for each partition to store partition- and CLE-specific content, and another config set of type `global` is created to store management node and global configuration data.

IDS Distributes Config Sets to Nodes

IDS, a read-only network share of content from the management node to the rest of the system, distributes config sets to every node in the system. All config sets are shared throughout the system, but only one `cle` config set is active on a given node at a time (in addition to an active `global` config set, which is applied to the entire system). Currently, IDS leverages the 9P network file system and the Linux automounter facility as its distribution mechanism; however, the content and use of the config sets is independent of the distribution mechanism.

Ansible Plays Apply Configuration during System Boot

Prior to booting the system, each node will have an image, the `global` config set, and the `cle` config set. When the system boots, each node boots an unconfigured software image. Then Ansible plays, which can be located in both the image and the config set (config set is the preferred location for site-supplied Ansible plays), apply configuration to that image, bringing up the services pertinent to each node.

Administrators Configure/Reconfigure the System on an Ongoing Basis

Configuration happens at times other than initial installation. New configuration service packages can be installed during system upgrades and updates, sites can decide to enable a new service or change the configuration of an existing service, and so forth. In all of these scenarios, an administrator uses the `cfgset` command to manage

config sets and the `cray-ansible` script to apply any configuration changes. The `cfgset` command and its associated subcommands and options enable administrators to perform a variety of operations on config sets in addition to create and update, such as search, diff, list, show, validate, push, and remove. See the `cfgset` man page for a description of its subcommands and options and some examples of each.

9.3 About Config Sets

Users invoke the `cfgset` command to take configuration content delivered in service packages and combine it with site-specific configuration content gathered either interactively or through bulk import. The results are used by `cfgset` to create a config set, which is a central repository that stores all configuration information necessary to operate the system. Config sets reside on the management node (e.g., the SMW) in `/var/opt/cray/imps/config/sets` by default. The contents of each config set reside in the following subdirectories:

- ansible** Local site-provided Ansible play content can be placed here for distribution with the config set. When the config set is created, `cfgset` copies Ansible content from service packages to this location. Whenever the config set is updated, `cfgset` copies Ansible content from service packages again, overwriting the previous service-package Ansible content and leaving the site-provided content unchanged.
- changelog** YAML change logs from previous sessions with the configurator.
- config** Configuration templates containing configuration information. When the config set is created, the configurator copies service package templates to this location. Users can modify the content of these templates using `cfgset` to invoke the configurator. Whenever the config set is updated, the configurator merges service package templates with the templates in this location.
- dist** Other site-provided content, such as static files required for the configuration of a service, can be placed here for distribution with the config set. When the config set is created, `cfgset` copies dist content from service packages to this location. Whenever the config set is updated, `cfgset` copies dist content from service packages again, overwriting the previous service-package dist content and leaving the site-provided content unchanged.
- files** Files necessary for system configuration that are distributed with the config set. They can be placed here by:
 - the `cfgset` command, which runs configuration callback scripts to generate some configuration files (e.g., `/etc/hosts`)
 - the Simple Sync service
 - local site administrators
- worksheets** Configuration worksheets generated by the configurator using data stored in the configuration templates in the `config` subdirectory of the config set. Administrators copy these worksheets to a location outside the config set, edit them with site-specific configuration data, and then import them to create a new config set or update an existing one.

Config Set Types

All config sets have a *type* associated with them that is specified upon creation. XC systems require both a `global` config set type and a `cle` config set type. After a config set of a given type is created, its type cannot be changed. A user may create multiple config sets to support partitioned systems or alternate configurations.

Typically a config set of type `cle` is created for each partition to store partition- and CLE-specific content, and another config set of type `global` is created to store configuration data that pertains to the management node domain as well as configuration data that can be easily shared among `cle` config sets. Config sets can be portable between partitions or to other systems if their partition-specific information is modified accordingly.

Configuration Service Inheritance

When a config set is created or updated, only service package templates that match the type of the config set can be included in the config set. Cray provides several service package templates that match both types and can be included in both `cle` and `global` config sets. In such cases, the user can choose which template will be used to configure the service in question. When a `cle` config set is created, and a service that has a template of both types is ready for configuration, the configurator will inject an initial question for the user to choose between configuring the service (i.e., using the `cle` version of the template) or letting the service inherit configuration values from the `global` config set (i.e., inheriting values from the `global` version of the template). Configuration worksheets for such services also provide that choice by including an `inherit` field, which can be set to `true` or `false`. If the user sets it to `true`, the configuration data from the `global` config set version of the service will be used. When the Cray-provided `cray-ansible` service (part of the Cray Configuration Management Framework) is run at boot time or at the system administrator's discretion, it uses the value of the `inherit` field to determine which configuration template data (`global` or `cle`) to use.

Inheritance is useful for systems with multiple partitions where a subset of partitions need custom configuration of a service, but another subset of partitions can all share the same global configuration.

9.4 About Config Set Caching

Config sets are defined and reside on the Server of Authority, which on XC systems is the SMW. Config set content is made available to all nodes in the system by means of Cray Scalable Services.

To make the sharing of config set content both quick and reliable, the `cray-cfgset-cache` service was created. It caches config sets locally on nodes (compressed for a smaller footprint). On the SMW, it does the following:

- notices changes to config sets on the SMW
- refreshes the local caches dynamically
- detects failures and retries automatically

The `cray-cfgset-cache` service ensures that config set content gets refreshed on all nodes whenever config sets are created or updated on the SMW. It is triggered when `cray-ansible` is run on a node with the `start`, `restart`, or `link` commands.

ATTENTION: If the `cray-cfgset-cache` service is stopped, config set content in node-local memory will not get refreshed when `cray-ansible` is run. If that happens, nodes will continue to use the most recent compressed copy of the config set data created before the service was stopped.

What Gets Cached

The `cray-cfgset-cache` service does not copy an entire config set to node-local memory. Instead, it uses the config set on the SMW to create these two files in the root of the config set:

- a compressed copy of the config set using SquashFS tools, (typically < 3 MB)
- a checksum of the compressed copy of the config set

The compressed copy is made available (effectively copied) to node-local RAM, and the checksum is used to know when the config set in node-local memory no longer matches the config set on the SMW. Even though Scalable Services makes the entire config set directory structure on the SMW available to the rest of the system, only the compressed copy and its associated checksum are used by nodes. They are the key to the performance, scalability, and reliability improvements provided by config set caching.

When `cray-ansible` is run on a node, the node will do the following:

1. Check to see if the cached node-local version of the compressed config set is out of date.
2. If it is stale, replace it with a newer version available on the SMW and start using that newer version.

9.5 About Variable Names in the Configurator and Configuration Worksheets

In the configurator and configuration worksheets, variable names can be quite long because they are composed of a data structure hierarchy. Each variable name begins with the name of the service to which it belongs. The next part of each name is always 'settings' to indicate that what follows is a *service setting*, one of the available settings for that service. After 'settings' comes the name of the setting, which could be a simple data type (string, boolean, integer, etc.) or a more complex data type (list, multival, etc.). The next part after the name of the setting is always 'data' to indicate that what follows is one of the fields of that setting. For a full description of data types, see *XC™ Series Configurator User Guide (S-2560)*.

For example, here is the variable for the IP address of the high-speed network (HSN), one of several networks.

```
cray_net.settings.networks.data.hsn.ipv4_network
```

This variable belongs to the `cray_net` service and the `networks` setting of that service. The `networks` setting is of type multival, which means it can have multiple entries, and each entry can have multiple fields to set. This variable targets the `ipv4_network` field of the `hsn` network entry.

This example shows the variable for the IP address of the HSN SDB node alias interface (one of several interfaces) of the SDB node (one of several hosts).

```
cray_net.settings.hosts.data.sdbnode.interfaces.hsn_sdb_alias.ipv4_address
```

This variable belongs to the `cray_net` service and the `hosts` setting of that service. The `hosts` setting is of type multival, and this variable belongs to the `sdbnode` host entry. The `sdb_node` host has a field `interfaces`, which is also of type multival. This variable targets the `ipv4_address` field of the `hsn_sdb_alias` interface entry.

9.6 About Node Groups

The Cray Node Groups service (`cray_node_groups`) enables administrators to define and manage logical groupings of system nodes. Nodes can be grouped arbitrarily, though typically they are grouped by software functionality or hardware characteristics, such as login, compute, service, DVS servers, and RSIP servers.

Node groups that have been defined in a config set can be referenced by name within all CLE services in that config set, thereby eliminating the need to specify groups of nodes (often the same ones) for each service individually and greatly streamlining service configuration. Node groups are used in many Cray-provided Ansible

configuration playbooks and roles and can be also used in site-local Ansible plays. Node groups are similar to but more powerful than the class specialization feature of releases prior to CLE 6.0. For example, a node can be a member of more than one node group but could belong to only one class.

Sites are encouraged to define their own node groups and specify their members. Administrators can define and manage node groups using any of these methods:

- Edit and upload the node groups configuration worksheet (`cray_node_groups_worksheet.yaml`).
- Use the `cfgset` command to view and modify node groups interactively with the configurator.
- Edit the node groups configuration template (`cray_node_groups_config.yaml`) directly. Use `cfgset` to update the config set afterwards so that pre- and post-configuration scripts are run (unless performing a migration, in which case follow the instructions provided to do config set update and validation at the appropriate time in the process).

After using any of these methods, remember to validate the config set.

Characteristics of Node Groups

- Node group membership is not exclusive, that is, a node may be a member of more than one node group.
- Node group membership is specified as a list of cnames. However, if the SMW is part of a node group, it is specified with the output of the `hostid` command. Also, host names can be used for eLogin nodes that are to be included in node groups.
- All compute nodes and/or all service nodes can be added as node group members by including the keywords “platform:compute” and/or “platform:service” in a node group.
- Any CLE configuration service is able to reference any defined node group by name.
- The Configuration Management Framework (CMF) exposes node group membership of the current node through the local system “facts” provided by the Ansible runtime environment. This means that each node knows what node groups it belongs to, and that knowledge can be used in Cray and site-local Ansible playbooks.

Default Node Groups

Default node groups are groups of nodes that

- are likely to be customized and used by many sites
- support useful default values for many of the migrated services

Several of the default node groups require customization by a site to provide the appropriate node membership information. This table lists the Cray default groups and indicates which ones require site customization.

Table 109. cray_node_groups

Default Node Group	Requires Customization?	Notes
compute_nodes	No	Defines all compute nodes for the given partition. The list of nodes is determined at runtime.
service_nodes	No	Defines all service nodes for the given partition. The list of nodes is determined at runtime.

Default Node Group	Requires Customization?	Notes
smw_nodes	Yes	Add the output of the <code>hostid</code> command for the SMW. For an SMW HA system, add the host ID of the second SMW also.
boot_nodes	Yes	Add the <code>cname</code> of the boot node. If there is a failover boot node, add its <code>cname</code> also.
sdb_nodes	Yes	Add the <code>cname</code> of the SDB node. If there is a failover SDB node, add its <code>cname</code> also.
login_nodes	Yes	Add the names of internal login nodes on the system.
all_nodes	Maybe	Defines all compute nodes and service nodes on the system. Add external nodes (e.g., eLogin nodes), as needed.
tier2_nodes	Yes	Add the <code>cnames</code> of nodes that will be used as tier2 servers in the <code>cray_scalable_services</code> configuration.

Why is there no "tier1_nodes" default node group? Cray provides a default `tier2_nodes` node group to support defaults in the `cray_simple_shares` service. Cray does not provide a `tier1_nodes` node group because no default data in any service requires it. Because it is likely that tier1 nodes will consist of only the boot node and the SDB node, for which node groups already exist, Cray recommends using those groups to populate the `cray_scalable_services tier1_groups` setting rather than defining a `tier1_nodes` group.

About eLogin nodes. To add eLogin nodes to node groups, use their 'hostname' values instead of `cnames`, because unlike CLE nodes, eLogin nodes do not have `cname` identifiers. If eLogin nodes are intended to receive configuration settings associated with the `all_nodes` group, add them to that group, or create a new group for eLogin nodes only (`elogin_nodes`), and then change the appropriate settings in other configuration services to include both `all_nodes` and `elogin_nodes`.

Additional Platform Keywords

Cray uses these two platform keywords to create default node groups that contain all compute or all service nodes.

```
platform:compute
platform:service
```

Sites that need finer-grained groupings can use these additional platform keywords to create custom node groups that contain all compute or service nodes with a particular core type.

```
platform:compute-XXNN
platform:service-XXNN
```

For `XXNN`, substitute a four-character processor/core designation, such as `KL64` or `KL68`, which designate two Intel® Xeon Phi™ "Knights Landing" (KNL) processors with different core counts.

Table 110. Cray Supported Intel Processor/Core (XXNN) Designations

Processor (XX)	Core (NN)	Intel Code Name
BW	12, 14, 16, 18, 20, 22, 24, 28, 32, 36, 40, 44	"Broadwell"

Processor (XX)	Core (NN)	Intel Code Name
HW	04, 06, 08, 10, 12, 14, 16, 18, 20, 24, 28, 32, 36	"Haswell"
IV	02, 04, 06, 08, 10, 12, 16, 20, 24	"Ivy Bridge"
KL	60, 64, 66, 68, 72	"Knights Landing"
SB	04, 06, 08, 12, 16	"Sandy Bridge"

9.7 About Simple Sync

The Cray Simple Sync service (`cray_simple_sync`) provides a simple, easy-to-use, generic mechanism for administrators to make configuration changes to their system without resorting to writing a custom Ansible play. When enabled, the service automatically copies files found in source directories in the config set on the SMW to one or more target nodes. Simple Sync is a simple tool and not intended as the sole solution for making configuration changes to the system. Writing custom Ansible plays might provide better maintainability, flexibility and scalability in the long term.

The Simple Sync service is enabled by default and has no additional configuration options. It can be enabled or disabled during the initial installation using worksheets or with the `cfgset` command at any time.

```
smw# cfgset update --service cray_simple_sync --mode interactive <config_set_name>
```

For more information, see `man cfgset(8)`.

How Simple Sync Works

When enabled, Simple Sync is executed on all CLE nodes at boot time and whenever the site administrator executes `/etc/init.d/cray-ansible start` on a CLE node. When Simple Sync is executed, files placed in the following directory structure are copied onto nodes that match these criteria:

```
smw:/var/opt/cray/imps/config/sets/<config_set>/files/simple_sync/
```

- `./common/files/` Matches all nodes.
- `./hardwareid/<hardwareid>/files/` Matches a specific node with that hardware ID, which is the `cname` of a CLE node or the output of the `hostid` command (e.g., `1eac0b0c`) on other nodes. An admin must create both the `<hardwareid>` directory and the `files` directory.
- `./hostname/<hostname>/files/` Matches a node with the specified host name. An admin must create both the `<hostname>` directory and the `files` directory. Use for eLogin nodes ONLY.
- `./nodegroups/<node_group_name>/files/` Matches all nodes in the specified node group. The directories for this `nodegroups` directory are automatically stubbed out when the config set is updated after node groups are defined and configured in the `cray_node_groups` service.

<code>./platform/[compute, service]/files/</code>	Matches all compute nodes or all service nodes, depending on whether they are placed in <code>platform/compute/files</code> or <code>platform/service/files</code> . Each time the config set is updated, the HSS data store is queried to update which nodes are service and which are compute.
<code>./README</code>	Provides brief guidance on using Simple Sync and a list of existing node groups in the order in which files will be copied. This ordering enables an administrator to predict behavior in cases where a file may be duplicated within the Simple Sync directory structure.

Simple Sync copies content into place prior to the standard Linux startup (`systemd`) and before `cray-ansible` runs any other services. As a result, Cray services that make small changes to files will operate on the administrator-provided file. Afterwards, the file will contain both non-conflicting administrator-provided content as well as the changes made by the Cray service. Because these changes happen prior to Linux startup, the changes will be in place when the services start up.

Note that there are some config files that are entirely managed by Cray services. Where possible, such files have a comment at the top indicating that the file is completely under the management of the Cray service. Files that have been changed by Cray services can be identified by checking the change logs on the running node in `/var/opt/cray/log/ansible`. Simple Sync does not provide a mechanism to override changes made by Cray services. To override changes made by Cray services, refer to the documentation for the specific service.

The ownership and permissions of copied directories and files are preserved when they are copied to root (`/`) on the matching target nodes. An administrator can run `cray_ansible` multiple times, as needed, and only the files that have changed will be copied to the target nodes.

Because of the way it works, Simple Sync can be used to configure services that have configuration parameters not currently supported by configuration templates and worksheets. An administrator can create a configuration file with the necessary settings and values, place it in the Simple Sync directory structure, and it will be distributed and applied to the specified node(s).

Characteristics of Simple Sync

Simple Sync is:	Simple Sync is NOT:
for simple and straightforward use cases	a comprehensive system management solution
for copying a moderate number of moderately sized files*	intended to transfer large objects or a large volume of files
	an interface to configure Cray "turnkey" services such as ALPS, Node Health or Lightweight Log Manager (LLM)

* Bear in mind that anything in the Simple Sync directory structure is part of a config set, and a SquashFS copy of the current config set is distributed to all nodes in the system. Even though it is a reduced-size config set that is distributed, it is good practice to not add very large files to a config set, hence the use of "moderate" here.

Introduced with the CLE 6.0.UP00 / SMW 8.0.UP00 release, Simple Sync has been enhanced to:

- run as early in the Ansible execution sequence as possible (it runs BEFORE other `cray-ansible` plays, so it can be used to make changes to files that Cray updates, like `sshd_config`)

- run during the Netroot setup sequence, so it can be used to change LNet and DVS settings, if needed
- support Node Groups for targeting which system nodes to copy files to (see [About Node Groups](#) on page 416)

Simple Sync does not support:

- removing files
- appending to files
- changing file ownership and permissions (the permissions of the file in the config set are mirrored on-node)
- backing up files
- overriding Cray-set values (it cannot be used to change files that Cray completely overwrites, such as `alps.conf`, or change values in files that Cray modifies such as `PermitRootLogin` in `/etc/ssh/sshd_config`)

Cautions about the Use of Simple Sync

- Simple Sync copies files from the config set, which in the case of nodes without a persistent root file-system is cached in a compressed form, locally, in memory. As a result, each file stored in the config set uses some memory on the node. Therefore, using Simple Sync to copy binary files or large numbers of files is inadvisable.
- Be aware of differences in node environments when using Simple Sync. For example, systems configured with direct-attached Lustre (DAL) have nodes running CentOS instead of SLES. Administrators would have to be very careful to avoid putting an inappropriate configuration file into place when using the Simple Sync platform/service target in such a situation.
- Storage and distribution of verbatim config files through Simple Sync creates the potential for unintentional impact to the system when config files evolve due to software changes. Making minimal necessary changes through a site-local Ansible playbook provides more flexibility and minimizes the potential for unintended consequences.

Use Cases

Copy a non-conflicting file to all nodes

1. Place `etc/myfile` under `./common/files/` in the Simple Sync directory structure.
2. Simple Sync copies it to `/etc/myfile` on all nodes.

Copy a non-conflicting file to a service node

1. Place `etc/servicefile` under `./platform/service/files/` in the Simple Sync directory structure.
2. Simple Sync copies it to `/etc/servicefile` on all service nodes.

Copy a non-conflicting file to a compute node

1. Place `etc/computefile` under `./platform/compute/files/` in the Simple Sync directory structure.

2. Simple Sync copies it to `/etc/computefile` on all compute nodes.

Copy a non-conflicting file to a specific node

1. Place `etc/mynode` under `./hardwareid/c0-0c0s0n0/files/` in the Simple Sync directory structure.
2. Simple Sync copies it to `/etc/mynode` on `c0-0c0s0n0`.

Copy a non-conflicting file to a user-defined collection of nodes

1. Create a node group called "my_nodes" containing a list of nodes.
2. Update the config set.

```
smw# cfgset update p0
```

3. Place `etc/mynodes` under `./nodegroups/my_nodes/files/` in the Simple Sync directory structure.
4. Simple Sync copies it to `/etc/mynodes` on all nodes listed in node group `my_nodes`.

Copy to a node a file that is exclusively maintained by Cray

Files exclusively maintained by Cray such as `alps.conf` cannot be updated using Simple Sync. Please refer to the owning service (such as ALPS) for information on how to update the contents.

Copy to a node a file that resides on a file system that will be mounted during Linux boot

No special operational changes are necessary. However, Simple Sync will put the file in place early in the boot sequence, and then it will be over-mounted by the file system. Because Simple Sync runs again later, it will copy the file into the mounted file system. Due to the ordering of operations, the file will not be present between the time the file system was mounted until the late execution of Ansible.

On Netroot login nodes, modify an LNet modprobe parameter

1. Generate a file `zz_lnet.conf` containing options `lnet router_ping_timeout=100`.
2. Place `zz_lnet.conf` under `./nodegroups/login/files/etc/modprobe.d/` in the Simple Sync directory structure.
3. The `lnet router_ping_timeout` value will be 100.

Note that normally Simple Sync does not allow the user to override Cray values, but this procedure takes advantage of the standard Linux mechanism to override Kernel module options.

Copy a file with an incompatible content to a node file that has Cray-maintained content

While Simple Sync allows an administrator to make changes to the same configuration files as modified by Cray, be very careful to avoid introducing syntax errors or incompatible values that may cause the system to fail to operate correctly.

9.8 About Boot Automation Files

New for the CLE 6.0.UP03 release. With this release, the default boot behavior for Cray systems without direct-attached Lustre (DAL) nodes is to boot all service nodes (other than the boot and SDB nodes) and all compute nodes can boot at the same time, thereby decreasing overall boot time.

- Default for systems without DAL:
 1. Boot + SDB (if SDB image small enough to PXE boot)
 2. SDB (if SDB image too large to PXE boot)
 3. Service + Compute
- Default for systems with DAL:
 1. Boot + SDB (if SDB image small enough to PXE boot)
 2. SDB (if SDB image too large to PXE boot)
 3. Service
 4. Compute

Cray provides the following boot automation files with this release.

auto.generic	Used to boot the entire XC system.
auto.xtshutdown	Used to shut down the entire XC system.
auto.bootnode	Used to boot only the boot node(s).
auto.bootnode+sdb	Used to boot only the boot node(s) and SDB node(s).

During a fresh install, sites typically copy `auto.generic`, rename it with the host name of the system for which it will be used (`auto.hostname.start`), and customize it for that site and system. Likewise, sites typically copy `auto.xtshutdown`, rename it with the host name of the system for which it will be used (`auto.hostname.stop`), and customize it, as needed. The host name is included because different systems may have different software installed, resulting in different boot or shutdown requirements. For example, on a system with a workload manager (WLM) installed, extra commands may be needed in the `auto.hostname.stop` file to cleanly stop the WLM queues on SDB or MOM nodes before shutting down the nodes.

When is customization of an automation file needed?

- For systems booting tmpfs images (instead of Netroot) with no SDB node failover, no changes may be necessary.
- For systems booting Netroot images, instructions for making Netroot-related changes after the first boot with tmpfs are provided at the appropriate place in the fresh install process.
- For systems booting direct-attached Lustre (DAL) images, instructions for making DAL-related changes are provided at the appropriate place in the fresh install process.

- For systems with added content in the recipe used for SDB nodes, if the resulting custom recipe produces a boot image too large for a PXE boot, changes to the boot automation file are necessary. If based on `auto.generic`, the system boot automation file will have an option (commented out by default) to boot the boot node via PXE boot and then boot the SDB node via the HSN.
- For systems with a workload manager (WLM) installed, WLM-related changes may be needed. Specific commands to add will vary based on the WLM.

9.9 About Snapshots and Config Set Backups during a Migration

Sites can make as few or as many snapshots and config set backups as they deem useful, but Cray recommends that sites make a snapshot and back up config sets at certain milestones during the installation and configuration process. Most of these will be for archival purposes, but snapshots and config set backups can be used to stage updates/upgrades and roll back to or switch between SMW and CLE releases as well.

Snapshots are created and managed using `snaputil`, a Python utility delivered with the `cray-install-support` RPM that is installed by default on the SMW. However, the fresh install procedure makes the first snapshot manually, because at that point in the process, `snaputil` has not yet been installed. Config sets are created and managed using `cfgset`. Procedures for how to create snapshots and config set backups are included at each point in the process where they are needed.

What does a snapshot contain? Snapshots capture content in these three file systems on the SMW: root (`/`), `/var/lib/mysql`, and `/var/opt/cray/repos`. Used in conjunction with backups of config sets, they provide enough information to be able to re-create the state of the system at the time of the snapshot and config set backup.

What does a config set contain? See [About Config Sets](#) on page 414 for details about the contents of a config set.

Best Practice. Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

Table 111. Suffixes and Corresponding Milestones for Snapshots and Config Set Backups during a Migration

Suffix	Description	Snapshot	Config Set
postinstall	after installing the new software release and before configuring CLE	yes	no
postconfig	after configuring CLE and before booting the CLE system	yes	yes
postboot	after booting the CLE system	yes	yes

9.10 Install Third-Party Software with a Custom Image Recipe

About this task

Any software that is created independent from Cray *and* that is not delivered with a Cray system is third-party software that administrators install as add-ons to the Cray system. (The information in this section does not

pertain to software installed on an external file system that is connected to a Cray system.) There are several ways to install third-party software:

- Add a third-party software package to a custom image recipe (recommended).
- Use the `chroot` command to install the software to an existing image.
- Use the `zypper` command to install software on a node.

IMPORTANT: Do not directly modify a Cray-provided recipe.

Installing software with a custom image recipe is the best method to use because the update to the image is persisted in the recipe. Each time the recipe is built into an image root, packaged into a boot image, and then the node is booted from the boot image, the third-party software is available. This approach follows the ideal model of a *prescriptive recipe*. Using `chroot` or `zypper` to install software is usually less desirable because the installations are not persisted. Software installed using `zypper` is lost the next time the node is booted; software installed with `chroot` is lost when a node image is rebuilt from a recipe. However, using the `chroot` or `zypper` method can be useful when persistence is not important, such as when testing third-party software.

Cray recommends customizing a Cray-provided recipe by creating a new recipe, then adding the Cray-provided recipe as a subrecipe. It is possible to clone a Cray-provided recipe and modify the clone, but cloned recipes do not receive updates from patches. For more information, see [About the Image Management and Provisioning System \(IMPS\)](#).

The following procedure shows how to create a new image recipe, add a Cray-provided image recipe as a subrecipe, then add the third-party RPM and non-RPM content for the image. This procedure starts with adding the RPMs to a package collection and repository. After the image recipe is defined, the procedure describes how to use the IMPS `image` command build an image root, export the image root into a boot image, and test it on a single node, then assign it to all applicable nodes. (After a recipe has been defined and tested, the `imgbuilder` command can be used to rebuild boot images.)

Procedure

1. Create a new repository and add the third-party packages (RPMs).
 - a. Use the `repo create` command to create the new repository (for example, `my_sles12_repo`). This command requires the architecture (such as `x86-64`) and operating system type (either `SLES12` or `CentOS`).

```
smw# repo create --arch x86-64 --type SLES12 my_sles12_repo
```

- b. Verify that the new repository was created.

```
smw# repo list my*
my_sles12_repo
```

- c. Add the third-party RPMs to the repository. This example takes all RPMs starting with `myrpm` in the example repository path `/path/to/repos/` and copies them to the example repo `my_sles12_repo`.

```
smw# repo update -a "/path/to/repos/myrpm*.rpm" my_sles12_repo
smw# ls -l /var/opt/cray/repos/my_sles12_repo
-rw-r--r-- 1 crayadm crayadm 485137 Nov 23 08:56 myrpm-11.13.1.1-4.x86_64.rpm
```

- d. Validate the repository.

```
smw# repo validate my_sles12_repo
```

2. Create a package collection and add the RPM package names.

A package collection represents a logical grouping of RPMs. Cray recommends using a package collection because the RPMs can be used in multiple image types (such as compute and service node images). Package collections are stored on the SMW in `/etc/opt/cray/imps/package_collections.d/`.

Cray provides the following package collections for workload manager (WLM) software:

- `service-pbs_cle_6.0up03_sles_12` (packages needed to build and run PBS)
- `service-torque_cle_6.0up03_sles_12` (packages needed to build and run Moab/TORQUE)
- `slurm-build_cle_6.0up03_sles_12` (packages needed to build Slurm)
- `compute-slurm_cle_6.0up03_sles_12` (packages needed to run Slurm on compute nodes)
- `login-slurm_cle_6.0up03_sles_12` (packages needed to run Slurm on login nodes)
- `service-slurm_cle_6.0up03_sles_12` (packages needed to run Slurm on service nodes)

a. Create an empty package collection (for example, `my_collection`).

```
smw# pkgcoll create my_collection
```

b. Verify that the package collection was created.

```
smw# pkgcoll list my*
my_collection
```

c. Add the RPM package name or names (for example, `myrpm`) to the package collection.

```
smw# pkgcoll update -p myrpm my_collection
```

d. Display information about the package collection.

```
smw# pkgcoll show my_collection
my_collection:
name: my_collection
package_collections:
myrpm
```

3. Create a new recipe and customize it by adding a sub-recipe (the Cray-provided image) and the content for the third-party software.

a. List the existing recipes to determine which image recipe to include.

```
smw# recipe list
compute-large_cle_6.0up01_sles_12_x86-64_ari
compute-large_cle_6.0up02_sles_12_x86-64_ari
compute-large_cle_6.0up03_sles_12_x86-64_ari
compute_cle_6.0up01_sles_12_x86-64_ari
compute_cle_6.0up02_sles_12_x86-64_ari
compute_cle_6.0up03_sles_12_x86-64_ari
dal_cle_6.0up01_centos_6.5_x86-64_ari
dal_cle_6.0up02_centos_6.5_x86-64_ari
dal_cle_6.0up03_centos_6.5_x86-64_ari
elogin_cle_6.0up01_sles_12_x86-64_ari
...
```

b. Create a new image recipe. This example uses the recipe name `site_compute`.

```
smw# recipe create site_compute
```

- c. Add the existing image recipe as a sub-recipe. This example uses the Cray-provided recipe `compute_cle_6.0up03_sles_12_x86-64_ari`.

```
smw# recipe update -i compute_cle_6.0up03_sles_12_x86-64_ari site_compute
```

- d. Add the package collection that contains the third-party RPMs (in this example, `my_collection`).

```
smw# recipe update -c my_collection site_compute
```

- e. Add the repository that contains the third-party RPMs (for example, `my_sles12_repo`).

```
smw# recipe update -r my_sles12_repo site_compute
```

- f. Add the objects mentioned in the sub-recipe that are also needed for the parent recipe.

IMPORTANT: The objects mentioned in a sub-recipe are used to build that sub-recipe but are not available to the parent recipe. If a package (RPM) or package collection is specified in the parent recipe, the custom recipe must explicitly contain the set of repositories where the packages can be found.

1. Determine which repository contains the necessary RPM or RPMs. This example `find` command identifies the Cray repository that contains the RPM `otherrpm`.

```
smw# find /var/opt/cray/repos -name otherrpm\* -ls
```

2. Select the correct repository:

- Choose the repository for the image's operating system type — use a SLES repository for a SLES image recipe; use a CentOS repository for a CentOS recipe.
- Most operating system and Cray repositories come in pairs (base and updates), such as `sles_12_x86-64` and `sles_12_x86-64_updates`. Be sure to select both the `base` and `base_updates` repositories if they exist.

3. Add the required repository or repositories (in this example, `otherrepo`).

```
smw# recipe update -r otherrepo site_compute
```

Repeat the `-r` option to add multiple repositories, such as a `base` and `base_updates` repository pair.

```
smw# recipe update -r sles_12_x86-64 -r sles_12_x86-64_updates \
site_compute
```

- g. (Optional.) Use post-build actions to add non-RPM content to the recipe. For example, post-build actions could include copying a tar file into the image then using `chroot` to run the commands to untar it and run an install script.

To add post-build actions, manually edit the image recipe: Open the local image recipe `/etc/opt/cray/imps/image_recipes.d/image_recipes.local.json`. Locate the image recipe definition for the custom image (for example, `site_compute`). In the `postbuild_copy` section, add the files to copy into the image. In the `postbuild_chroot` section, add the commands to run in a `chroot` environment for this image root.

```
smw# vi /etc/opt/cray/imps/image_recipes.d/image_recipes.local.json
"site_compute": {
    ...
    "package_collections": { ... },
```

```

    "packages": { ... },
    "postbuild_chroot": [
        "chroot_command1",
        ...
        "chroot_commandN"
    ],
    "postbuild_copy": [
        "/file/1",
        ...
        "/dir/2/content"
    ],
    "recipes": { ... },
    "repositories": { ... }
},

```

TIP: Post-build scripts can use the following environmental variables:

- IMPS_IMAGE_NAME
- IMPS_VERSION
- IMPS_IMAGE_RECIPE_NAME
- IMPS_POSTBUILD_FILES

h. Validate the image recipe.

```

smw# recipe validate site_compute
INFO - Repository 'my_sles12_repo' validates.
INFO - Recipe 'site_compute' is valid.

```

This command checks that the JSON syntax of the image recipe is correct. It also validates all repositories and package collections referenced by the image recipe and ensures that it can access any files in the `postbuild_copy` section.

4. Build the image recipe to create the image root. For the image root name, Cray recommends using the image recipe name plus the current date/time. This example shows the image root name `site_compute_timestamp`.

The `image create` command builds the image recipe starting with the package manager installation and then proceeds to step through the `postbuild copy` and `chroot` commands (in that order).

```

smw# image create -r site_compute site_compute_timestamp
INFO - Repository 'my_sles12_repo' validates.
INFO - Recipe 'site_compute' is valid for building.
INFO - Calling Package manager to build new image root; this will take a few
minutes.
INFO - Rebuilding RPM database for Image 'site_compute_timestamp'.
INFO - RPM database does not need to be rebuilt.
INFO - Running post-build scripts for Image 'site_compute_timestamp'.
INFO - Copying postbuild files to /tmp/tmpmAYzG1 in Image
'site_compute_timestamp'
INFO - * Executing post-build chroot script: 'chroot_command1'
INFO - post-build chroot script output will be located in /tmp/
site_compute_postbuild_out_20150713-15:55:11g4WA6p
INFO - Build of Recipe 'site_compute' has completed successfully.

```

5. (Optional.) Display the build history of the image root.

```

smw# image show site_compute_timestamp
site_compute_timestamp:

```

```

name: site_compute_timestamp
created: 2016-07-13T15:54:06
history:
  2016-07-13T15:55:16:      Successful build of Recipe
                          'site_compute into Image 'site_compute_timestamp'.
  2016-07-13T15:55:17:      Successful rebuild of RPM database.
path: /var/opt/cray/imps/image_roots/site_compute_timestamp

```

6. Package the image root into a boot image.

```

smw# image export site_compute_timestamp

INFO - Copying kernel /var/opt/cray/imps/image_roots/site_compute_timestamp/boot/
bzImage-3.12.28-4.6_1.0000.8685-cray_ari_c into /tmp/temp_tempfs_50LJ93/DEFAULT
INFO - Copying parameters file /var/opt/cray/imps/image_roots/site_compute_timestamp/
boot/parameters-ari_c into /tmp/temp_tempfs_50LJ93/DEFAULT
.
.
.
INFO - Image 'site_compute_timestamp' has been packaged into /var/opt/cray/imps/
boot_images/site_compute_timestamp.cpio.

```

The `image export` command displays the boot image file name at the end of the output. This `cpio` file is used in the next step.

7. Test the new boot image on a single node.

- a. Assign the boot image to a node with the NIMS `cnode` command. This example assigns the boot image file `site_compute_timestamp.cpio` (in the directory `/var/opt/cray/imps/boot_images/`) to the compute node with the `cname` `c0-0c0s15n3`.

```

smw# cnode update -i \
/var/opt/cray/imps/boot_images/site_compute_timestamp.cpio c0-0c0s15n3

```

- b. Warm-boot the node to test the boot image.

```

smw# xtcli shutdown c0-0c0s15n3
.
.
.
crayadm@smw> xtbootsys --reboot \
-r "testing new boot image site_compute_timestamp" c0-0c0s15n3

```

8. Assign the new boot image to all applicable nodes.

```

smw# cnode update --group compute \
-i /var/opt/cray/imps/boot_images/site_compute_timestamp.cpio

```

9. Choose when the compute nodes should switch to the new image.

- To immediately use the new image, warm-boot all applicable nodes with the new image. This example specifies the compute nodes as a comma-separated list of `cnames`; see the `xtcli(8)` man page for other ways of specifying multiple nodes.

```

smw# xtcli shutdown cname, cname, ... cname
.
.
.

```

```
smw# xtbootsys --reboot -r "Booting custom image on all compute nodes" \
cname, cname, ... cname
```

- To have the workload manager (WLM) reboot the node once the current user's job finishes, see [Apply Rolling Patches to Compute Nodes with cnat](#).
- Otherwise, wait until the next full system reboot. The nodes will boot with the new image.

9.11 Prefixes for Binary and Decimal Multiples

The International System of Units (SI) prefixes and symbols (e.g., kilo-, Mega-, Giga-) are often used interchangeably (and incorrectly) for decimal and binary values. This misuse not only causes confusion and errors, but the errors compound as the numbers increase. In terms of storage, this can cause significant problems. For example, consider that a kilobyte (10^3) of data is only 24 bytes less than 2^{10} bytes of data. Although this difference may be of little consequence, the table below demonstrates how the differences increase and become significant.

To alleviate the confusion, the International Electrotechnical Commission (IEC) adopted a standard of prefixes for binary multiples for use in information technology. The table below compares the SI and IEC prefixes, symbols, and values.

SI decimal vs IEC binary prefixes for multiples					
SI decimal standard			IEC binary standard		
Prefix (Symbol)	Power	Value	Value	Power	Prefix (Symbol)
kilo- (kB)	10^3	1000	1024	2^{10}	kibi- (KiB)
mega- (MB)	10^6	1000000	1048576	2^{20}	mebi- (MiB)
giga- (GB)	10^9	1000000000	1073741824	2^{30}	gibi- (GiB)
tera- (TB)	10^{12}	1000000000000	1099511627776	2^{40}	tebi- (TiB)
peta- (PB)	10^{15}	1000000000000000	1125899906842624	2^{50}	pebi- (PiB)
exa- (EB)	10^{18}	1000000000000000000	1152921504606846976	2^{60}	exbi- (EiB)
zetta- (ZB)	10^{21}	1000000000000000000000	1180591620717411303424	2^{70}	zebi- (ZiB)
yotta- (YB)	10^{24}	1000000000000000000000000	1208925819614629174706176	2^{80}	yobi- (YiB)

For a detailed explanation, including a historical perspective, see <http://physics.nist.gov/cuu/Units/binary.html>.

10 Checklists for Migration using a Physical Migration SMW

The process of migrating configuration and image data from a CLE 5.2 / SMW 7.2 system to a CLE 6.0 / SMW 8.0 system is not necessarily sequential: some steps are optional, and some can be performed in parallel by different people. Cray recommends using the provided checklists to track progress through the migration process.

[Master Checklist for Migration using a Physical Migration SMW](#) on page 431 lists all of the high-level tasks needed to perform a migration using a physical migration SMW. It includes links to more detailed checklists for some tasks.

10.1 Master Checklist for Migration using a Physical Migration SMW

Table 112. Master Checklist for Migration using a Physical Migration SMW

✓	Migration Phase	Task	Notes
	1. Training	Read Ansible and Python resources	
		Get trained on Cray System Administration for CLE 6.0 / SMW 8.0	
		Review Cray technical publications	
	2. Planning	Determine need for additional hardware	
		Obtain additional hardware, if needed	
	Preparation of Physical Migration SMW and Boot Raid	Prepare for a fresh install on the migration SMW	
		Installation Checklist 1: Install the Base Operating System on the SMW on page 433	
		Installation Checklist 2: Install the SMW and CLE Software on page 434	
		Migration Checklist 1.1-P: Configure Other Features and Services on page 434	
	3. Preparation of Configuration Data and Images	Start a typescript file	
		Migration Checklist 2.1: Extract Configuration Data on page 435	

✓	Migration Phase	Task	Notes
		Migration Checklist 2.2: Transfer Global Configuration Data on page 435	
		Migration Checklist 2.3: Transfer CLE Configuration Data on page 436	
		Migration Checklist 2.4: Load and Validate Configuration Data on the Migration SMW on page 438	
		Migration Checklist 2.5-V: Update Non-config-set Configuration Files on the Migration SMW	
		Choose image recipes to build	
		Migration Checklist 2.6: Build Image Roots and Boot Images from Recipes on page 439	
		Assign kernel parameters	
		Check NIMS information	
		Identify and port site-local scripts	
		Install Cray Programming Environment (PE) software	
		4. Preservation of Other Data	Run final accounting reports
	Save operational data		
	Save site user data		
	Drain workload manager (WLM) queues		
	5. Shutdown and Switch	Shut down CLE system	
		(SMW HA only) Put SMW HA cluster into maintenance mode	
		Switch cabling to migration SMW and boot RAID	
		Migration Checklist 3.2-P: Discover XC System Hardware on page 440	
		Migration Checklist 3.3-P: Complete CLE Configuration on page 440	
		Migration Checklist 3.4: Boot the CLE System on page 441	
		Migration Checklist 3.5: Complete Post-boot Configuration of Config Services on page 442	
		Migration Checklist 3.6-P: Install and Configure Additional Software on page 443	

✓	Migration Phase	Task	Notes
		Back up newly installed software	
		Restore operational data	

10.2 Installation Checklist 1: Install the Base Operating System on the SMW

Table 113. Installation Checklist 1: Install the Base Operating System on the SMW

✓	Task	Notes
<i>Prepare to Install the Base Linux Distribution</i> on page 26		
	(Dell R815 SMW only) <i>R815 SMW: Change the BIOS and iDRAC Settings</i> on page 26	
	(Dell R630 SMW only) <i>Configure the Dell R630 SMW RAID Virtual Disks</i> on page 32	
	(Dell R630 SMW only) <i>R630 SMW: Change the BIOS and iDRAC Settings</i> on page 36	
Install the base OS		
	<i>Install the SLES 12 Base Linux Distribution on the Migration SMW</i> on page 46	
<i>Configure Boot RAID Devices</i> on page 51		
	<i>Install SANtricity Storage Manager for NetApp, Inc. Devices</i> on page 53	
	<i>Set Up Boot RAID Space for Direct-attached Lustre</i>	
	<i>Create Boot RAID Volume Group and Volumes for NetApp, Inc. devices</i> on page 54	
	Zone the SAS (Serial Attached SCSI) or FC (Fibre Channel) switch using one of these procedures: <ul style="list-style-type: none"> • <i>Zone the QLogic FC Switch</i> • <i>Zone the Brocade FC Switch</i> • <i>Zone the LSI SAS Switch</i> 	
	<i>Reboot the SMW and Verify LUNs are Recognized</i> on page 57	
	<i>Make a Snapshot Manually</i> on page 57	

10.3 Installation Checklist 2: Install the SMW and CLE Software

Table 114. Installation Checklist 2: Install the SMW and CLE Software

✓	Task	Notes
	Start a Typescript File on page 59	
Prepare to bootstrap the installation		
	Prepare to Bootstrap the SMW Installation on page 60	
	Determine the Persistent Device Name for a LUN on page 62	
	RAID Disk Space Requirements on page 63	
Bootstrap and install the SMW and CLE software		
	Bootstrap the SMW Installation on page 65	
	Provision SMW Storage on page 71	
	Run the Installer for an Initial Installation on page 72	
	Set Default Snapshot and Boot the SMW on page 73	

10.4 Migration Checklist 1.1-P: Configure Other Features and Services

Table 115. Migration Checklist 1.1-P: Configure Other Features and Services

✓	Task	Notes
	Set or Change the HSS Data Store (MariaDB) Root Password on page 75	
	Start a Typescript File on page 59	
	Make a Post-install Snapshot using snaputil on page 77	
	Update install.cle.conf for Software Updates on page 77	
	Configure Power Management on page 78	
	Reduce Impact of Btrfs Periodic Maintenance on SMW Performance on page 82	
	(if using SEC) Configure the Simple Event Correlator (SEC) on page 82	

✓	Task	Notes
	(if using SEDC) <i>Enable System Environmental Data Collections (SEDC)</i> on page 364	
	<i>Prevent Unintentional Re-creation of Mail Configuration Files</i> on page 83	
	<i>Install the Dell Systems Management Tools and Documentation DVD</i> on page 83	

10.5 Migration Checklist 2.1: Extract Configuration Data

Checklist items are based on this procedure: *Extract Configuration Data from the CLE 5.2 / SMW 7.2 System* on page 86.

Table 116. Migration Checklist 2.1: Extract Configuration Data

✓	Task	Notes
	<i>Save configuration files from the sharedroot</i>	
	<i>Save configuration files from the SMW</i>	
	<i>Save installer files from the SMW</i>	
	<i>Save configuration files from the boot node</i>	
	<i>Save files from the UFS (or syslog or SDB) node</i>	
	<i>Actively probe the SMW of the running system</i>	
	<i>Actively probe the boot node of the running system</i>	
	<i>Actively probe all non-boot service nodes of the running system</i>	
	(SMW HA only) <i>Probe the active SMW and record the cluster configuration</i>	

10.6 Migration Checklist 2.2: Transfer Global Configuration Data

Table 117. Migration Checklist 2.2: Transfer Global Configuration Data

✓	Task	Notes
	Generate global worksheets and move copies to <code>/var/adm/cray/release/global_worksheet_workarea</code>	

✓	Task	Notes
	Update <code>cray_bootraid</code> Worksheet in Global Config Set on page 101	
	Update <code>cray_firewall</code> Worksheet in Global Config Set on page 105	
	Update <code>cray_global_net</code> Worksheet in Global Config Set on page 106	
	Update <code>cray_ipforward</code> Worksheet in Global Config Set on page 115	
	Update <code>cray_liveupdates</code> Worksheet in Global Config Set on page 116	
	Update <code>cray_logging</code> Worksheet in Global Config Set on page 116	
	Update <code>cray_multipath</code> Worksheet in Global Config Set on page 120	
	Skip <code>cray_network_boot_packages_worksheet.yaml</code>	
	Update <code>cray_time</code> Worksheet in Global Config Set on page 135	

10.7 Migration Checklist 2.3: Transfer CLE Configuration Data

Table 118. Migration Checklist 2.3: Transfer CLE Configuration Data

✓	Task	Notes
	Generate CLE worksheets and move copies to <code>/var/adm/cray/release/p0_worksheet_workarea</code>	
	Update <code>cray_alps</code> Worksheet on page 138	
	Update <code>cray_auth</code> Worksheet on page 148	
	Update <code>cray_batchlimit</code> Worksheet on page 156	
	Update <code>cray_boot</code> Worksheet on page 158	
	Update <code>cray_ccm</code> Worksheet on page 159	
	Update <code>cray_cnat</code> Worksheet on page 161	
	Update <code>cray_drc</code> Worksheet on page 164	
	Update <code>cray_dvs</code> Worksheet on page 165	

✓	Task	Notes
	Update <i>cray_dw_wlm</i> Worksheet on page 170	
	Update <i>cray_dws</i> Worksheet on page 171	
	Update <i>cray_elogin_inet</i> Worksheet (Update Cray eLogin Service Worksheets on page 175)	
	Update <i>cray_elogin_motd</i> Worksheet (Update Cray eLogin Service Worksheets on page 175)	
	Update <i>cray_elogin_networking</i> Worksheet (Update Cray eLogin Service Worksheets on page 175)	
	Update <i>cray_eswrap</i> Worksheet (Update Cray eLogin Service Worksheets on page 175)	
	Update <i>cray_firewall</i> Worksheet on page 176	
	Update <i>cray_image_binding</i> Worksheet on page 176	
	Update <i>cray_ipforward</i> Worksheet on page 178	
	Update <i>cray_liveupdates</i> Worksheet on page 178	
	Update <i>cray_lmt</i> Worksheet on page 179	
	Update <i>cray_inet</i> Worksheet on page 180	
	Update <i>cray_local_users</i> Worksheet on page 187	
	Update <i>cray_logging</i> Worksheet on page 192	
	Update <i>cray_login</i> Worksheet on page 193	
	Update <i>cray_lustre_client</i> Worksheet on page 194	
	Update <i>cray_lustre_server</i> Worksheet on page 199	
	Update <i>cray_multipath</i> Worksheet on page 200	
	Update <i>cray_munge</i> Worksheet on page 202	
	Update <i>cray_net</i> Worksheet on page 203	
	Update <i>cray_netroot_preload</i> Worksheet on page 235	
	Update <i>cray_node_groups</i> Worksheet on page 236	
	Update <i>cray_node_health</i> Worksheet on page 242	
	Update <i>cray_persistent_data</i> Worksheet on page 266	
	Update <i>cray_rsip</i> Worksheet on page 269	
	Update <i>cray_rur</i> Worksheet on page 273	
	Update <i>cray_scalable_services</i> Worksheet on page 280	
	Update <i>cray_sdb</i> Worksheet on page 282	

✓	Task	Notes
	Update <i>cray_service_node</i> Worksheet on page 285	
	Update <i>cray_shifter</i> Worksheet on page 286	
	Update <i>cray_simple_shares</i> Worksheet on page 288	
	Update <i>cray_simple_sync</i> Worksheet on page 290	
	Update <i>cray_ssh</i> Worksheet on page 291	
	Update <i>cray_storage</i> Worksheet on page 292	
	Update <i>cray_sysconfig</i> Worksheet on page 293	
	Update <i>cray_sysenv</i> Worksheet on page 295	
	Update <i>cray_time</i> Worksheet on page 295	
	Update <i>cray_user_settings</i> Worksheet on page 296	
	Update <i>cray_wlm_detect</i> Worksheet on page 298	
	Update <i>cray_wlm_trans</i> Worksheet on page 298	
	Update <i>cray_zonesort</i> Worksheet on page 300	

10.8 Migration Checklist 2.4: Load and Validate Configuration Data on the Migration SMW

Table 119. Migration Checklist 2.4: Load and Validate Configuration Data on the Migration SMW

✓	Task	Notes
	Disable Pre- and Post-configuration Scripts on page 301	
	Update Global Config Set from Worksheets on page 302	
	Create New CLE Config Set from Worksheets on page 302	
	Update CLE Config Set on page 303	
	Update <i>/etc/motd</i> for Nodes on page 304	
	Copy Files for External Lustre Fine-grained Routing on page 305	
	Configure Files for Cray Simple Sync Service on page 305	

✓	Task	Notes
	<ul style="list-style-type: none"> • Configure Simple Sync for DVS Server Nodes on page 307 • Configure Simple Sync for TCP Wrappers on page 309 • (if Slurm used) Configure Simple Sync for Slurm on page 310 	
	Validate Config Sets on page 312	
	Ensure Time Zone Setting Accessible by Cabinet and Blade Controllers on page 313	
	Continue Initial DataWarp Configuration on page 314	

10.9 Migration Checklist 2.5-P: Update Non-config-set Configuration Files on the Migration SMW

Table 120. Migration Checklist 2.5-P: Update Non-config-set Configuration Files on the Migration SMW

✓	Task	Notes
	Move settings to non-config-set files in CLE 6.0 / SMW 8.0 from counterpart files on the CLE 5.2 / SMW 7.2 system. See step 1 of Update Non-config-set Configuration Files on the Migration SMW on page 319.	
	Create Hardware Test Configuration with xtdiscover on page 319	
	Assign Service Nodes Manually and Disable Components on page 320	
	Check for Site Modifications in SMW xtrim.conf on page 321	
	Check for Site Modifications to SEDC Files on page 322	
	Check for Site Modifications to SMW Firewall and IP Tables on page 322	

10.10 Migration Checklist 2.6: Build Image Roots and Boot Images from Recipes

Table 121. Migration Checklist 2.6: Build Image Roots and Boot Images from Recipes

✓	Task	Notes
	Bootstrap NIMS with imgbuilder on page 329	
	Install SMW/CLE Patches on the Migration SMW on page 330	
	Prepare Cray Image Groups and Custom Recipes on page 332	
	Assign Nodes to New NIMS Groups on page 335	
	Build Images and Map Them to NIMS Groups on page 339	

10.11 Migration Checklist 3.2-P: Discover XC System Hardware

Table 122. Migration Checklist 3.2-P: Discover XC System Hardware

✓	Task	Notes
	Bootstrap Hardware Discovery on page 357	
	Update Firmware on page 359	
	Discover Hardware and HSN Routing, Prepare STONITH on page 360	
	(Optional) Configure Partitions on page 362	
	Repurpose Compute Nodes on page 363	
	Finish Configuring the SMW for the CLE System Hardware on page 363	

10.12 Migration Checklist 3.3-P: Complete CLE Configuration

Table 123. Migration Checklist 3.3-P: Complete CLE Configuration

✓	Task	Notes
	Update and Validate Global Config Set after Migration Switch on page 365	
	Update and Validate CLE Config Sets after Migration Switch on page 366	
	Check CLE Hostnames in /etc/hosts File on page 366	
	Display and Capture all Config Set Information on page 367	
	Make a Post-config Snapshot using snaputil on page 368	
	Make a Post-config Backup of Current Global and CLE Config Sets on page 369	
	Check NIMS Information on page 342	

10.13 Migration Checklist 3.4: Boot the CLE System

Table 124. Migration Checklist 3.4: Boot the CLE System

✓	Task	Notes
	Boot the Boot and SDB Nodes on page 371	
	Restore ALPS Files to /alps_shared on page 371	
	Push Diag Image to Boot Node on page 372	
	Push Netroot Images to Boot Node on page 373	
	Push PE Image Root to Boot Node on page 374	
	Boot the Rest of the System using a Boot Automation File on page 376	
	Run Tests after Boot is Complete on page 378	
	Prepare Site and Software Revision Information Reporting using xtgetrev and xtshowrev on page 379	
	Test xtdumpsys and cdump on page 380	
	Make a Post-boot Snapshot using snaputil on page 382	

✓	Task	Notes
	Make a Post-boot Backup of Current Global and CLE Config Sets on page 383	

10.14 Migration Checklist 3.5: Complete Post-boot Configuration of Config Services

Table 125. Migration Checklist 3.5: Complete Post-boot Configuration of Config Services

✓	Task	Notes
	Prepare fresh worksheets for editing (see Complete Post-boot Configuration of Config Services on page 384).	
	Configure custom node health checker (NHC) plugins, as needed (see Complete Post-boot Configuration of Config Services on page 384).	
	Configure custom Resource Utilization Reporting (RUR) data_plugins or output_plugins, as needed (see Complete Post-boot Configuration of Config Services on page 384).	
	Configure a complex RSIP configuration, as needed (see Complete Post-boot Configuration of Config Services on page 384).	
	Configure Shifter, as needed (see Complete Post-boot Configuration of Config Services on page 384).	
	Restore contents of WLM spool directory, as needed (see Complete Post-boot Configuration of Config Services on page 384).	
	Apply Site Firewall and IP Tables Configuration via Config Set and Ansible Play on page 386	
	Update and Validate CLE Config Set for Post-boot Changes on page 387	
	(if using DAL) Configure Direct-attached Lustre (DAL) on page 388	
	(optional if using DAL) LMT Configuration for DAL on page 395 (Lustre Monitoring Tool for direct-attached Lustre)	

10.15 Migration Checklist 3.6-P: Install and Configure Additional Software

Table 126. Migration Checklist 3.6-P: Install and Configure Additional Software

✓	Task	Notes
	Complete DataWarp Configuration on page 401	
	Install and Configure a Workload Manager (WLM) on page 403	
	(SMW HA only) The migration of the first SMW in an SMW HA system is complete. To complete migration of the full SMW HA system, proceed with the standard SMW HA fresh install process, beginning with section 2.3 "Prepare to Install SMW HA Software" in <i>XC™ Series SMW HA Installation Guide (SLEHA12.SP0.UP03) S-0044</i> .	