TAS HSM Administrator Guide S-2540-201

Contents

1 About the TAS HSM Administrator Guide....6
2 TAS HSM System Introduction....9
2.1 TAS HSM Software Components....11
2.1.1 Versity Storage Management (VSM) Software....12
2.1.2 TAS HSM Software....13
2.1.3 CIMS Node Software Introduction....19
2.2 TAS HSM System Hardware Components....21
2.3 TAS HSM System Networks....22
3 TAS HSM High Availability....25
3.1 Manual MDC Failover....25
3.2 Automatic MDC Failover....26
4 Versity Storage Manager (VSM) Software....28
4.1 VSM Licensing....28
4.2 High Capacity and Volume Management....29
4.3 Paged and Direct I/O....29
4.4 VSM Metadata Storage....30
4.5 Fast File Recovery....30
4.6 Shared File System Support....30
4.7 VSM Configuration File System....31
4.8 VSM Disk Allocation Units....31
4.9 VSM File System Types....31
4.10 Set VSM File System Stripe Width....32
5 Manage a TAS System with Bright....33
5.1 Use `cmgui` to Manage TAS....33
5.1.1 Run CMGUI from the CIMS Node....35
5.1.2 Install and Run `cmgui` on a Remote System....37
5.2 Bright Node Categories for TAS....38
5.3 Bright Node Groups for TAS....40
5.4 TAS Software Image Management....40
5.4.1 TAS DM Node Category....41
5.4.2 TAS MDC Node Category....42
5.4.3 Change TAS Node Category Settings....42
5.4.4 TAS Software Image Properties....43
5.4.5 Clone a Service Node Software Image....44
5.4.6 Enable Boot Record in Software Image....45

5.5 Configure TAS Finalize Scripts to Provision Node-specific Files....47
5.6 Bright Exclude Lists....48
5.6.1 Set Up Exclude Lists....49
5.6.2 Exclude List Defaults....50
5.6.3 Check Exclude Lists....53
5.6.4 Change an Exclude List....53
5.7 Configure kdump on TAS Nodes....54
6 Manage the TAS HSM....58
6.1 Configure the Archiver....58
6.1.1 Schedule Queue....63
6.1.2 Drive/Volume Size and Selection....63
6.1.3 Archive Request Entered in Scheduling Queue....64
6.1.4 Archive Directives....64
6.1.5 Control the Size of Archive Files Using `archmax`....64
6.1.6 Set Archiver Buffer Size with `bufsize`....64
6.2 Configure the Stager....65
6.2.1 The stager.cmd File....65
6.2.2 Preview Request....66
6.2.3 Set the Global VSN and Age Directives....66
6.2.4 Set Global or File System Specific Water Marks....67
6.2.5 Priority Scheme....68
6.3 Configure the Recycler....68
6.3.1 Control the Recycler Process....70
6.3.2 Removable Media Cartridges....70
6.3.3 Prevent Recycling with the no_recycle Directive....70
6.3.4 Specify Recycling on an Automated Library....70
6.3.5 recycler.sh File....71
6.3.6 Configure Recycling for Disk Archive Volumes....72
6.4 Configure the Releaser....72
6.4.1 The Releaser Process....73
6.4.2 Partial Releasing and Partial Staging....74
6.5 Change the VSM Configuration....74
6.6 Configure VSM Archive Devices....76
6.7 VSM Master Configuration File (MCF)....77
6.8 Configure a VSM File System....80
6.8.1 Create a VSM File System....80
6.8.2 Configure a Shared VSM File System in Bright....82
6.8.3 Configure a VSM File System on a Client Node....84

6.8.4 Add and Remove Client Hosts....86
6.8.5 Export a VSM File System....88
6.8.6 Mount or Unmount a VSM File System on the MDC Node....91
6.9 Quotas....91
6.9.1 Soft and Hard Limits....91
6.9.2 Disk Blocks and File Limits....92
6.9.3 Configure Quotas for a File System....92
6.9.4 Accounting and Infinite Quotas....94
6.9.5 Enable Default Quota Values....94
6.9.6 Enable or Change Quota limits....94
6.9.7 Check Quotas Using `samquota`....95
6.9.8 Remove or Change Quotas....95
6.10 Archive Daemons and Processes....96
6.11 Trace Files....97
6.12 Operator Utility samu....97
6.12.1 Keyboard Shortcuts....98
6.12.2 Archiver Status Display (a)....99
6.12.3 The Device Configuration Display (c)....99
6.12.4 Daemon Trace Controls Display (d)....100
6.12.5 File Systems and Archive Parameters Display (f)....101
6.12.6 Shared Clients Display (g)....104
6.12.7 Help Information Display (h)....104
6.12.8 Usage Information Display (l)....105
6.12.9 Mass Storage Status Display (m)....105
6.12.10 Staging Status Display....105
6.12.11 Removable Media Load Requests Display....106
6.12.12 Removable Media Display....106
6.12.13 Tape Drive Status Display....107
6.12.14 Stage Queue Display....108
6.12.15 Robot Catalog Display (v)....108
6.12.16 Pend Stage Queue Display....110
6.12.17 Memory Display....110
6.12.18 Disk Volume Dictionary Display....110
6.12.19 Inode Display....111
6.12.20 Preview Shared Memory Display....112
6.12.21 File System Parameters....112
6.12.22 Active Services (P)....113
6.12.23 Sector Data Display....113

6.12.24 SCSI Sense Data....113
6.12.25 Device Table Display (U)....114
7 TAS Man Pages....115
7.1 mcp(1)....115
7.2 msum(1)....118
7.3 tasdefaults.conf(4)....120
7.4 tasarchive(8)....121
7.5 tasbundle(8)....122
7.6 taschart(8)....122
7.7 tasclean(8)....123
7.8 tasdump(8)....124
7.9 tasha(8)....125
7.10 tashwm(8)....127
7.11 taskeyscan(8)....128
7.12 tasrw(8)....128
7.13 tassynchadmin(8)....129
7.14 tastapeq(8)....130

1 About the TAS HSM Administrator Guide

This publication includes administration information for Cray TAS HSM software release TAS-XX-2.0.1.

TAS-XX-2.0.1 Release Features

- The `tasbundle` script is now run from the CIMS node which also runs `samexplore` to gather VSM file system information for Cray Service.
- A new `tasarchive` command automatically manages CIMS administrative file system backups.
- A Pacemaker resource was added that defines the hostname `mdc` to always be the active VSM MDC. Simply SSH to `mdc` from the CIMS node to login to the active MDC.
- Place license keys in a `/etc/opt/vsm/VSM.license.`*`hostname`* file. License files must be named `VSM.license.`*`hostname`*, the Pacemaker VSM resource copies the correct license in place when an MDC becomes active.
- A NO_VSMMFS_MOUNT option is included in the `tasdefaults.conf` file to prevent file systems from automatically mounting via the high-availability software after a failover.
- Additional volumes can be easily added to a pool to simplify volume management by reserving a volume to be used exclusively with an archive set. This simplifies the volume configuration, because there is no need to distribute the volumes between all the archive sets.

  NOTE: With release TAS-XX-2.0.1, all of the VSM configuration files (which includes the `archiver.cmd` and `mcf` files) are now stored in a single location in the shared TAS administrative file system. They are under the `/tas_admin/VSM/etc` directory and bind mounted to `/etc/opt/vsm` on the active MDC node. This simplifies administration and eliminates the need to synchronize the configuration files on an active MDC node back to the MDC software image in Bright, after modifications. This also eliminates the need to `chroot` into the MDC software images on the CIMS node, then update the running MDC images using Bright.

Scope of this Content

This content assumes administrators have previous experience with operating similar SAM-QFS file and tape archive systems. Cray implementation of bright cluster manager (Bright) software leverages its capability to manage the following specialized nodes:

- CIMS node
- DM node
- MDC node

Bright is not used to manage a TAS system as traditional compute cluster. It follows that some Bright features, such as cloudbursting, are not implemented in Cray TAS systems.

This document provides an overview of Bright. It also provides some examples of how to use Bright to manage the system. Refer to the *Data Management Practices (DMP) Administrator Guide, S-2327* for additional administration procedures and examples for using Bright to manage the CIMS node and service nodes.

Related Content

Table 1. Related Content

Document	Description
Cray Data Management Practices (DMP) Administrator Guide, S-2327	Provides software administration procedures for the Cray TAS CIMS node and Cray service nodes.
Versity Storage Manager (VSM) Administrator Guide	VSM file system administrator information.
Cray Integrated Management Services (CIMS) Software Installation Guide, S-2522	Provides software installation and configuration procedures for the Cray TAS CIMS node.
Lustre File System by Cray (CLFS) Software Installation Guide, S-2521	Describes the initial software installation and configuration procedures for TAS service nodes.
Bright® Cluster Manager 6.1 Administrator Manual	Includes information about the Bright software. Describes how to use the user interface (`cmgui`) and management shell (`cmsh`) to perform common administrative tasks. A PDF of this document is stored on the CIMS node in the `/cm/shared/docs/cm` directory.
Cray Tiered Adaptive Storage Hardware Guide, HR85-8500	Provides: • A high-level introduction • An architectural overview • Site planning and system specifications • Hardware implementation • Details for the Dell servers • NetApp block storage devices for the administrative block storage (ABS) and data-cache block storage DCBS) devices
DELL® R630, R720, and R730 Hardware Owners Guides, Dell Ethernet Switch and KVM software documentation, available from *www.dell.com/support*.	Dell equipment documentation.
NetApp® storage device software documentation is available from *www.netapp.com*	NetApp storage equipment documentation.

Typographic Conventions

`Monospace`	Indicates: • Program code

	• Reserved words • Library functions • Command-line prompts • Screen output • File/path names • Key strokes (e.g., `Enter` and `Alt-Ctrl-F`) • Other software constructs
`Monospaced Bold`	Indicates commands that must be entered on a command line or in response to an interactive prompt.
`Oblique` or *`Italics`*	Indicates user-supplied values in commands or syntax definitions.
Proportional Bold	Indicates a graphical user interface window or element.
\ (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

Visit the Cray Publications Portal at *http://pubs.cray.com*. Make comments online using the **Contact Us** button in the upper-right corner or Email pubs@cray.com. Comments are important to Cray and will responded to within 24 hours.

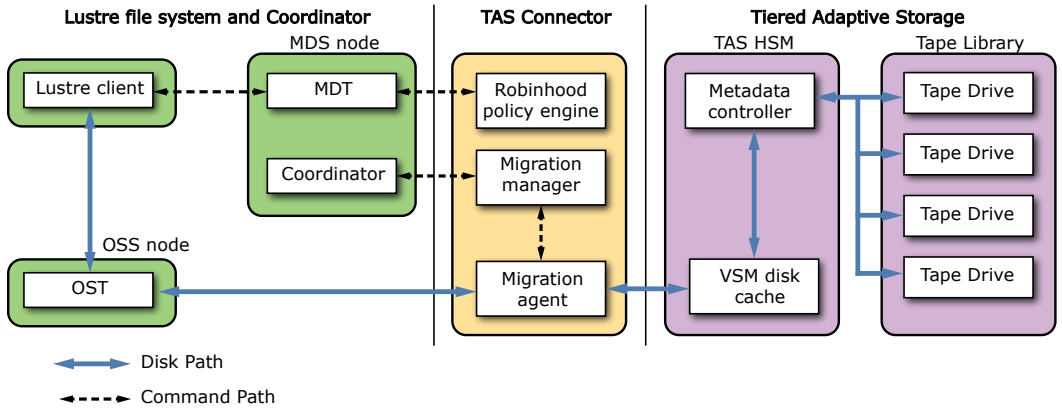
2 TAS HSM System Introduction

TAS HSM

Cray Tiered Adaptive Storage (TAS) systems implement hierarchical storage management (HSM) features. These use multiple storage tiers to manage and archive large file systems. TAS systems automatically move—based on customer policies—data between high-cost, high-bandwidth storage devices and low-cost, low-bandwidth storage devices.

TAS systems interconnect a Lustre® file system with a tape-archive backend.

Figure 1. TAS HSM and TAS Connector Block Diagram



TAS systems run Cray proprietary TAS-XX-2.0.1 software and Versity® Storage Manager (VSM) software. VSM is a proprietary Linux version of SAM-QFS® developed by Versity, Inc. The TAS system uses an open architecture and Versity open-file format.

The Cray TAS system has two major functional capabilities:

- Provides a POSIX shared file system with good support for small files and support for:
  - Hundreds of native file system clients
  - Native interfaces to storage tiers
  - Integrated volume management
- Provides an archive capability that manages data between disk and/or tape storage tiers via a policy engine that can manage up to four copies of each file.

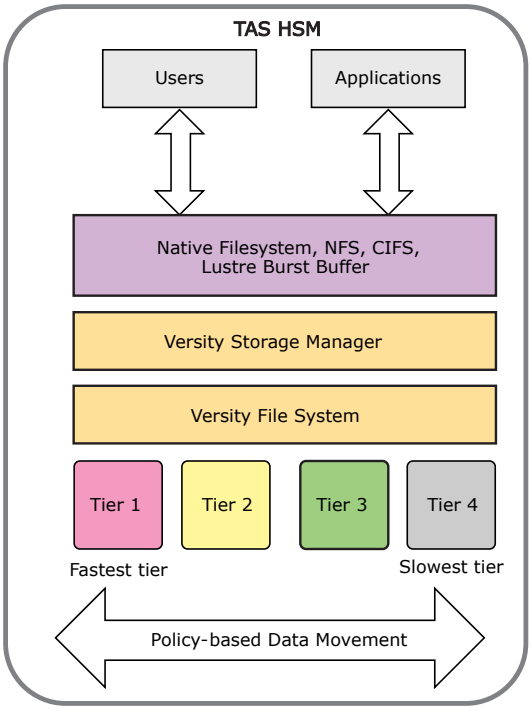
Each storage system is defined as a tier in VSM, based on storage capacity and/or I/O bandwidth. The TAS system stores data on slower devices (lower tiers) and copies the data to faster storage devices (higher tiers) based on site policies. TAS systems are built on:

- The foundation of standard Data Management Practices (DMP) management and server node hardware and software
- OEM high-performance storage systems
- OEM tape archive systems

Up to four physical tiers are supported:

- Tier 0 – Performance-optimized for high I/O and throughout (disk or SSD)
- Tier 1 – Primary storage where live data lives most of time (disk or SSD)
- Tier 2 – Capacity-optimized nearline storage (disk or tape)
- Tier 3 – Extreme capacity- and cost-optimized for deep archives (usually tape)

Figure 2. TAS System HSM Architecture



TAS HSM major components are:

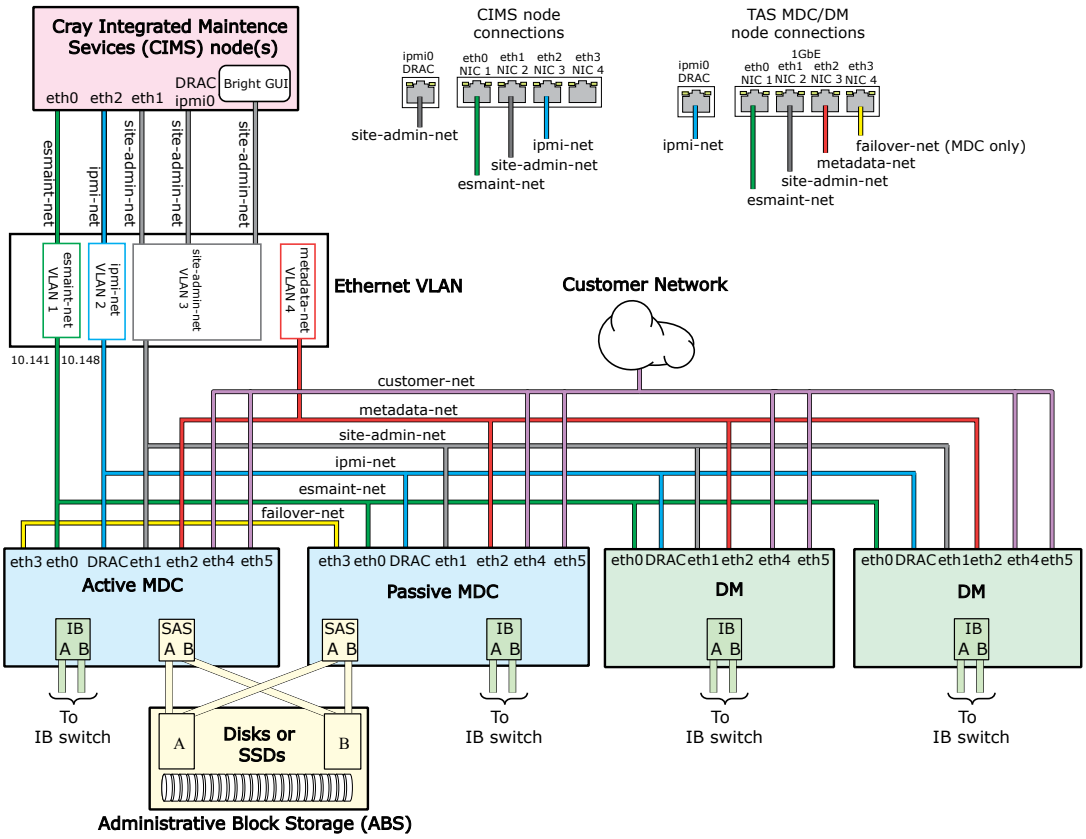
- A Cray integrated management services node (CIMS)
- A metadata management system that includes two metadata controller (MDC) nodes
- A fast data disk cache block storage (DCBS) device
- An optional capability to support one or more disk or tape archive tiers
- A GigE management network (which is usually a separate VLAN on the metadata network)
- A Gigabit Ethernet (GigE) or 10 Gigabit Ethernet (10GbE) metadata network VLAN
- An InfiniBand (IB) quad-data rate (QDR) storage network
- An optional FC tape archive network

- Connectivity to a customer network
- A policy engine controls movement of data between storage tiers

There are two types of switches—storage area network (SAN) switches and network switches. The SAN switches may implement Fibre Channel or InfiniBand standards. They provide a data storage fabric to connect the MDC and DM nodes to the cache block storage and to connect the MDCs to the robotic tape library.

The network VLAN Ethernet switches provide administrative communication between the CIMS node, and the MDC and DM nodes. The VLAN provides access for system administrators and all the servers (CIMS, MDC, and DM). In addition, the VLAN provides metadata communication between the MDC and DM nodes.

Figure 3. Cray TAS HSM Gateway Networks



2.1 TAS HSM Software Components

CIMS Node Software

The base operating system for the CIMS node—also referred to as *head node* or *master node* in Bright Cluster Manager (Bright) documentation—runs the SUSE Linux Enterprise Server (SLES) Service Pack 3 (SP3) base operating system (SLES11SP3). Bright software runs a daemon process on the CIMS node and service nodes to manage all system nodes.

IMPORTANT: Service node software images used to boot and configure MDC and DM nodes are stored on CIMS node in `/cm/images` directory.

Cray Proprietary TAS Software

The TAS HSM software release includes:

- Versity storage manager (VSM) software
- The TAStoolkit
- Customized Bright software images for each system service node

Versity Storage Manager (VSM) Software

Service Node Software

All service nodes run:

- Customized CentOS 6.5 Cray ESF software images
- TAS software
- InfiniBand or Fibre Channel software
- Any other software needed for their role in the system

2.1.1 Versity Storage Management (VSM) Software

Versity storage management (VSM) software is proprietary hierarchical storage management (HSM) software. It is constructed around open source core technologies (SAM-QFS). SAM-QFS. It was released as open source software by Sun Oracle under the Common Development and Distribution License (CDDL) license. VSM incorporates portions of the open source SAM-QFS technology under CDDL terms. Versity Software utilizes an open source file format—GNU TAR. Files written with VSM can be read directly from the media without the use of any additional software, either open source or proprietary. VSM media is also self describing. This means that all file metadata is stored on the media with the files in an open standard format.

VSM provides the Versity archiving file system and a user data archiving and management application. The system presents a POSIX file system interface to a global namespace. It spans all storage infrastructure tiers. All users of the file system use the VSM interface to read and write files to the storage devices as if they were doing so on primary disk storage. The system may be configured to back up all work-in-progress instantaneously in the background—or periodically—based on time or site policy. All VSM file systems are shared in a TAS system to enable file systems to migrate between the nodes.

VSM storage tiers are implemented by copying or moving files from the data cache block storage (DCBS), the data archive storage tier (DAST), or tape-archive storage tier (TAST). Archive media may consist of a VSM file system, another file system, robotic library tape storage, disk storage, or manually loaded tape storage devices. The administrator configures policies that classify data for the archive. VSM releases disk space associated with archived data according to policies. It then restores (stages) the files to DCBS as needed. The process of staging and releasing files is transparent to applications.

The VSM software includes the following components:

NOTE: In release TAS-XX-2.0.1 the `/tas_admin/VSM/etc` directory stores the system configuration files and is bind mounted to `/etc/opt/vsm` on the active MDC node.

VSM Archiver

The *archiver* copies online disk cache files to archive media (tape or disk). The archiver is configured with the `archiver.cmd` file, which defines what to archive. The archiving process is initiated when files match a site-defined set of criteria.

VSM Releaser

The *releaser* maintains the file system online disk cache at a site-specified high limit and low limit. Releasing is when primary (disk) storage used by an archived file's data is made available. The high-water mark (HWM) and low-water mark (LWM)—expressed as a percentage of total disk space—are used to manage free space in the online disk cache. When online disk consumption exceeds the high-water mark, the system begins to release the disk space occupied by eligible archived files until the low-water mark is reached. Files are selected for release depending on their size and age.

VSM Stager

The VSM *stager* restores file data to the disk cache. When a user or process requests file data that has been released from disk cache, the stager automatically copies the file data from archive media to the online disk cache. The read operation immediately follows the staging operation. This enables the file to be immediately available to an application before the entire file is completely staged.

VSM Recycler

As files are modified, archive copies of older versions are expired. Expired copies no longer needed are purged from the system. The *recycler* identifies the archive volumes with the largest proportions of expired archive copies. It preserves the unexpired copies by moving them to separate volumes.

The Operator Utility samu

Use the operator utility to monitor and control the VSM file system and archiving daemons. Start `samu` by entering the `samu` command from the MDC node. The default help screen displays. Type the letter assigned to the various menus listed below. Enter `Ctrl-f` to page through all of the `samu` command menus and displays.

2.1.2 TAS HSM Software

The TAS HSM software release includes Versity Storage Manager (VSM) software, the TAStoolkit, and customized Bright software images.

The TAS MDC and DM node software images are configured initially from a Lustre® File System by Cray (CLFS) node software image. CLFS node software images (for example, `ESF-XX-2.2.0-201404151111)` run the 6.5 operating system and are customized for TAS systems by Cray.

Customized Bright software images are pre-installed on the CIMS node. They are used to boot and configure the MDC nodes and DM nodes.

2.1.2.1 TAStoolkit Software

The TAStoolkit software is installed on the MDC node software image. It includes:

- System configuration utilities
- Monitoring tools

- A fast copy data mover
- A perl module to assist with scripting on the TAS system

The TAStoolkit version is stored in `/opt/cray/tas/VERSION`. The TAStoolkit configuration file is installed in the shared TAS administration file system `/tas_admin/config/tasdefaults.conf` on the MDC node. This defines the site-specific variables. This file is configured during software installation. It can be updated as needed. TAStoolkit commands are installed in `/opt/cray/tas/bin` on the MDC nodes. They are typically executed from `crontab`. A `quiet` option is available to disable all output to `STDOUT`.

Refer to the `man` page on the MDC node for more information about each command. Use the `-h` command line option to display the command usage. The command status can then be reported via `syslog` or email.

Script Notification and Execution Methods

The monitoring commands can be run from a Linux shell with a simple status to `STDOUT` or executed by Bright cluster manager (Bright) and report any errors via `syslog` or email.

2.1.2.1.1 TAS Toolkit Man Pages

TAS `man` pages are available online from the MDC nodes or CIMS node using the `man` command.

Table 2. TAS Toolkit Man Pages

Command (Man Page)	Description
`mcp(1)`	Copy `SOURCE` to `DEST`, or multiple `SOURCE`(s) to `DIRECTORY`.
`msum(1)`	Print or check checksums.
`tasdefaults.conf(4)`	The `/tas_admin/config/tasdefaults.conf` file alters various configuration tunables for the environment. The `tasdefaults.conf` file is referenced by the various TAStoolkit commands and alters their behavior. The `tasdefaults.conf` file can be updated at anytime.
`tasbundle(8)`	`tasbundle` has been deprecated for individual nodes. Use `sosreport` or run `tasbundle` on the CIMS node to create a package for a support representative. Collects pertinent information on the TAS system in a compressed tar archive file: `/tmp/tasbundle/`*`hostname_timestamp`*`.tar.xv`.
`taskeyscan(8)`	Removes and then adds the SSH host keys for all TAS nodes to the `known_hosts` file.
`tasarchive(8)`	Automatically manages CIMS adminstrative file system backups. The archive files are stored accordingly to the appropriate settings in the `tasdefaults.conf` configuration file. The archive files are rotated each time `tasarchive` runs.
`tasclean(8)`	Maintains a number of VSM file system dumps/logs and TAS chart (`taschart`) reports. The number of dumps and reports retained is defined with the VSM_DUMP_COUNT and VSM_CHART_COUNT variables in the `tasdefaults.conf` file.
`taschart(8)`	TAS performance reporting tool.

Command (Man Page)	Description
tasdump(8)	Automatically manages VSM file system dumps. The dumps are stored accordingly to the appropriate settings in the tasdefaults.conf configuration file.
tasha(8)	Functions as the main interface for managing and monitoring the TAS cluster environment. The main functions of the command are cluster node: ● Start ● Stop ● Convert ● Status The tasha command works in two modes—clustered and manual. The various tasha commands available are specific to the clustered or manual mode.
tashwm(8)	Monitors VSM high-water mark.
tasrw(8)	Reads and writes a small file on each read/write mounted VSM file system. Verifies basic functionality of each file system. On exit, the file is removed.
tassyncadmin(8)	Manages VSM file system dumps.
tastapeq(8)	Returns warning if tape request in queue exceeds defined wait time.
tasvolumes(8)	Tape volume reporting utility that returns status counts on the Auditing, Expired, Non-SAM formatted, unavailable, and duplicate VSN volume attributes.

2.1.2.1.2 tasdefaults.conf File

The TAS /tas_admin/config/tasdefaults.conf file on the MDC node enables various system-configuration setting adjustment for TAStoolkit utilities. The tasdefaults.conf file can be updated anytime.

An example tasdefaults.conf is located in /opt/cray/tas/examples on the MDC node. This can be copied to /tas_admin/config. Edit this configuration file to configure TAStoolkit commands. The tasdefaults.conf file can be updated anytime.

The following example tasdefaults.conf file lists the parameters that can be modified. A NO_VSMMFS_MOUNT option is included in the tasdefaults.conf file to prevent file systems from automatically mounting via the high-availability software after a failover. Refer to the man page for tasdefaults.conf on the MDC node for more information.

```
# Location of VSM configuration files and directories
VSM_SBIN=/opt/vsm/sbin
VSM_BIN=/opt/vsm/bin
VSM_CONF=/etc/opt/vsm

###
# Notification
###
# Admin email address for info/warning/error messages
EMAIL_ADDR=root
```

```
# TAS Syslog facility
TAS_SYSLOG_FACILITY=local2
TAS_SYSLOG_LOG=-/tas_admin/logs/tas-log

# VSM Syslog facility
VSM_SYSLOG_FACILITY=local4
VSM_SYSLOG_LOG=-/tas_admin/logs/sam-log

###
# TASdump VSM Dump parameters
###
# Directory to place VSM dumps
VSM_DUMP_DIR=/tas_admin/vsm_dumps
# Directory to place VSM dump logs
VSM_DUMP_LOG_DIR=/tas_admin/logs/vsm_dumps
# Compress dump files? NO=0, YES=1
VSM_DUMP_COMPRESS=1
# Dump archive count
VSM_DUMP_COUNT=10

###
# TASchart Reporting
###
# Data Devices to report performance statistics on
DATA_DEVICES=md0:dc0000:dc0001:dc0002:dc0003:dc0004:dc0005
REPORT_DIR=/tas_admin/reports
SAR_DIR=/var/log/sa
TAS_CHART_COUNT=31

###
# TAStapeq Tape wait queue
###
# Max tape request queue wait time in minutes
WAITQ_TIME=30
# Supported tape types (seperated by '|');
TAPE_TYPES=lt|li

###
# TAS Admin File System Sync
###
TAS_SYNC_DST=/vsm/tasfs1/.tas/tas_admin_sync
TAS_SYNC_LOG=/var/log/tas_admin_sync.log

###
# TAS CIMS archive sync
###
CIMS_ARCHIVE_DIR=/vsm/tasfs1/.tas/cims
CIMS_ARCHIVE_LIMIT=90
CIMS_WEEKLY_FILES=3

###
# TAS Usage Reporting
###
TAS_USAGE_LOG=/tas_admin/logs/vsm_usage.log

###
# TAS Cluster
###
# 0 = mount VSM file systems when cluster starts
# 1 = don't mount VSM file system when cluster starts
NO_VSMFS_MOUNT=0
```

2.1.2.1.3 Samfsdumps

The administrative script, `tasdump`, manages VSM file system dumps (`samfsdump`). The dumps are stored according to the settings in the `tasdefaults.conf` configuration file. This command is be run periodically via `cron` to provide continuous dump management. By default, the `tasdump` command is run once daily.

Refer to the `tasdump(8)` man page on the MDC node for more information about managing VSM dumps.

2.1.2.1.4 TAS Watch Scripts

tashwm Monitor high-water mark. Returns and reports error if any VSM mounted file system has exceeded its defined high-water mark.

tasrw Reads and writes a number of small files on each read/write mounted VSM file system. This verifies basic functionality of each file system.

tastapeq Returns warning if tape request in queue exceeds defined wait time.

2.1.2.1.5 TAS Reporting Scripts

taschart TAS performance reporting tool.

tasvolumes Retrieves tape volume status.

tasusage Collects capacity information and library usage information from VSM. This information is logged in the `/tas_admin/logs/vsm_usage.log` file. This command should be run via `crontab` daily without any arguments.

2.1.2.1.6 TAS Maintenance Scripts

tasclean Maintains a number of VSM file system dumps/logs and TAS chart reports. The number of dumps and reports retained is defined with the `VSM_DUMP_COUNT` and `VSM_CHART_COUNT` variables in the `tasdefaults.conf` file.

tasdump Reads the following options from the `tasdefaults.conf` file:

VSM_DUMP_DIR	Location of dump directory
VSM_DUMP_LOG_DIR	Location of dump log directory.
VSM_DUMP_COMPRESS	Compress dump files? 0=NO, 1=YES

tasha Invokes a manual MDC failover from the active MDC node to the passive MDC node.

tassyncadmin Automatically manages VSM adminstrative file system backups via `rsync` and stores the files based on settings in the `tasdefaults.conf` configuration file.

TAS_SYNC_DST	Location of destination `rsync` directory.
TAS_SYNC_LOG	Location of `rsync` log file.

tasarchive Automatically manages CIMS adminstrative file system backups based on the settings in the `tasdefaults.conf` configuration file. The archive files are rotated each time `tasarchive` runs. A `cron` job must be configured on the CIMS node to enable tasarchive. An example `crontab` file is provided in `/opt/cray/tas/examples`.

CIMS_ARCHIVE_DIR	Location of archive files.
CIMS_ARCHIVE_LIMIT	`tasarchive` will not run if the target file system usage is greater than specified limit.

CIMS_WEEKLY_FILES	Number of archive files to keep in addition to latest archive.
taskeyscan	Removes and then adds the SSH host keys for all TAS nodes to the known_hosts file.

2.1.2.1.7 TAS Bundle Script

The TAS bundle script (`tasbundle`) packages service and support files on the CIMS node in a compressed tar archive file: `/tmp/tasbundle/`*`hostname_timestamp`*`.tar.xv`
or `/tmp/tasbundle/`*`hostname_timestamp`* directory.

The `tasbundle` script also runs the `samexplorer` command to collect VSM information. The information collected by this command should be included in bugs and provided to Cray Service.

2.1.2.1.8 TAS `tasfunc` Perl Module

The perl library `tas::tasfunc` is installed in `/opt/cray/tas/lib` on the MDC nodes.

```
use lib '/opt/cray/tas/lib';
use tasfunc;
```

`tas::tasfunc` is a perl module used to write perl scripts. The subroutines included in this module are common tasks associated with VSM file system and TAS infrastructure monitoring and managing. The following functions are included in the `tas:tasfunc` module.

admin_mounted ()	Returns true if the TAS administrative file system: `/tas_admin` is mounted. Example: `print "Admin FS is mounted\n" if admin_mounted();`
get_hostname()	Returns hostname of system calling the function. Example: `$hostname=get_hostname();`
check_mcf()	Validates the MCF file. Returns non-zero value if `/etc/opt/vsm/mcf` is not configured properly. Example: `print "MCF configuration error!\n" if check_mcf();`
get_date()	Returns date string in the "YY/MM/DD HH:MM:SS" format. Example: `$date = get_date;`
get_timestamp()	Returns timestamp string in the "YYMMDDHHMMSS" format. Example: `$date = get_timestamp;`
get_version()	Returns version of the TAStoolkit. Example: `my $version = get_version;`
is_active_node()	Returns true if MDC is configured and acting as MDC. Example: `print "I'm the active node\n" if is_active_node();`
is_mdc()	Returns true if node is configured as MDC (not necessarily active/running MDC). Example: `print "I'm the MDC\n" if is_mdc();`

is_vsm_running()	Returns `true` if VSM is running. Example: `print "Admin FS is mounted\n" if admin_mounted();`
logmsg()	Print message to the VSM log file via the `syslog` facility. Example: `my $msg = "important message here..."; logmsg(info, $msg);`
vsm_configured()	Returns array of configured file system family names `samfs1`, `samfs2`, `archiver1`. Example: `my @Filesystems = vsm_configured();`
vsm_mounted()	Returns array of mount VSM file systems. Example: `my @Filesystems = vsm_mounted();`

2.1.2.1.9 Fast Copy Data Mover

The TAStoolkit provides the `mcp` and `msum` commands, which replace `cp` and `md5sum` (renamed `mcp` and `msum`). These commands use parallelism to achieve maximum copy and checksum performance on clustered file systems.

2.1.3 CIMS Node Software Introduction

The Cray Integrated Management Services (CIMS) node software release (ESM) includes CIMS software, the SUSE Linux Enterprise Server (SLES™) Service Pack 3 (SP3) operating system, Bright Cluster Manager 6.1 software (Bright), Cray Lustre® control and Cray tools and utilities. CIMS software is provided with the Cray ESM XX-3.1.0 software release.

Bright software provides:

- A management shell (`cmsh`)
- A graphical user interface (`cmgui`)
- A cluster management daemon (`cmd` command, or CMDaemon). CMDaemon runs on all nodes in the system. CMDaemon on a service node responds to requests from `cmsh` or `cmgui` on the CIMS node and communicates with the CMDaemon processes running on each service node. The CMDaemon processes on each service node communicate only with the CMDaemon processes running on the node.

Bright manages the hardware and software for all the devices and nodes in a system through the Bright CMDaemon process (`cmd`). Bright supports a GUI (`cmgui`) and command line shell (`cmsh`) interface. Either the `cmgui` or `cmsh` can be used to manage the system and there may be certain tasks are more easily visualized using `cmgui`, and other tasks are more efficient using the `cmsh`.

System administration may also be performed using the Bright GUI (`cmgui`). The `cmsh` command prompt displays an asterisk (*) when changes have not been committed. Configuration changes are queued until they are committed (saved). Be sure to commit changes using the commit command before exiting `cmsh`, or configuration changes are not saved to the Bright database.

Refer to the *Bright Cluster Manager 6.1 Administrator Manual* for detailed information about Bright software management. PDF files for the Bright manuals are stored on the CIMS node in the `/cm/shared/docs/cm` directory, and linked to from the `/root` directory.

2.1.3.1 DM Node Software

TAS DM nodes are installed with Cray ESF software (release ESF-XX-2.2.0), which are built on CentOS 6.5 and include:

- Lustre® client software
- InfiniBand software
- `esfsprogs`
- Other storage device tools and utilities

Cray manufacturing customizes the software image for DM nodes, configures the image in Bright, and distributes it with the TAS HSM and TAS Connector software releases.

The DM node software images are stored in /cm/images on the CIMS node and named accordingly (`TAS-XX-2.0.1-201510011105-dm` for example).

> **IMPORTANT:** Always backup customized Cray software images configured for service nodes located in the CIMS node `/cm/images` directory. Clone these software images to a new software image. Then make modifications to the new image.

TAS DM nodes may include (depending on system configuration):

- InfiniBand or Fibre Channel software and drivers
- Cray Cluster Connect™ (C3) Lustre® Client Release software (*C3™ Lustre Client*)

2.1.3.2 MDC Node Software

TAS MDC node software is distributed with the TAS software release. MDC node software images are built from a complete installation of the Cray ESF release media. This includes CentOS 6.5, TAS software, and storage management software. TAS DM nodes are installed with Cray ESF software (release ESF-XX-2.2.0), which are built on CentOS 6.5 and include:

- Lustre® client software
- InfiniBand software
- `esfsprogs`
- Other storage device tools and utilities for:
  - Administrative block storage (ABS) devices
  - Data cache block storage (DCBS) devices

Cray manufacturing customizes the software image for MDC nodes which includes:

- VSM software
- The TAStoolkit
- Other TAS customizations

The MDC node software images are stored in /cm/images on the TAS node and named accordingly (`TAS-XX-2.0.1-201510011105-mdc` for example).

> **IMPORTANT:** Always backup customized Cray software images configured for service nodes located in the CIMS node `/cm/images` directory. Clone these software images to a new software image and make modifications to the new image.

TAS MDC node software images may also include InfiniBand or Fibre Channel software and drivers depending on system configuration.

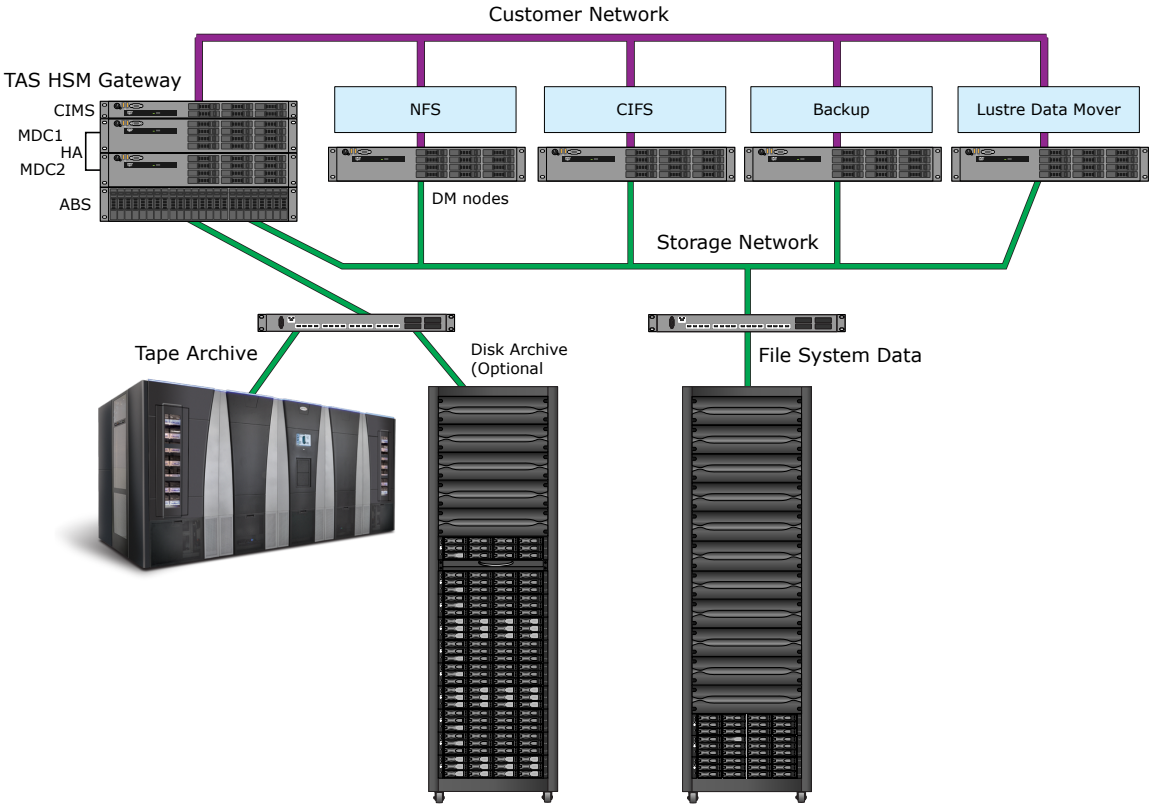
2.2 TAS HSM System Hardware Components

A Cray TAS system includes:

- A Cray Integrated Management Services (CIMS) node
- Redundant metadata controller (MDC) nodes
- Two data mover (DM) nodes
- Ethernet
- InfiniBand or Fibre Channel (FC) switch infrastructure (as shown below)

Storage infrastructure uses NetApp high-performance storage systems, and supports FC attached tape archive systems.

Figure 4. TAS HSM System Hardware Components



- The Cray Integrated Management Services (CIMS) node is a 2U rack-mounted server that manages all system hardware and software components and runs Bright Cluster Manager (Bright) software (CMDaemon or `cmd`).
- The CIMS node also stores, manages, and configures the *service* node software images. Service nodes used in Cray systems include:
  - Metadata controllers (MDC)
  - Data movers (DM)

  - Migration Managers (CMM)
  - Migration agents (CMA)

  The service nodes PXE boot from the CIMS node. Therefore, software modifications made to service nodes must be made using the `chroot` shell in the software image on the CIMS node. Then they must be pushed out to the node(s) using Bright cluster management (Bright).

- The Metadata Controller (MDC) nodes are redundant high-availability 2U rack-mounted servers. They manage file locking, space allocation, and data access authorization. They run the Versity Storage Manager (VSM) hierarchical storage management (HSM) software. MDC nodes PXE boot the 6.5 operating system provisioned from the CIMS node and run CMDaemon. MDC nodes also include a software stack to support the required I/O connectivity such as InfiniBand, Fibre Channel, or GigE software.
- The Data Mover (DM) nodes are 2U rack-mounted servers. They move data between the Cray TAS system and site file systems (NFS™, CIFS, or Lustre®). Data movers PXE boot the 6.5 operating system provisioned from the CIMS node. They run CMDaemon and a software stack to support the required I/O connectivity such as InfiniBand, Fibre Channel, or GigE software.
- One or more Ethernet switches to support the: maintenance network (`esmaint-net` and `ipmi-net`), metadata network (`metadata-net`), customer administration (`site-admin-net`) and user networks.
- One or more InfiniBand (IB) switches to support high-speed data transfers from the MDC and DM nodes to the NetApp storage devices or other high-performance storage systems.
- One or more Fibre Channel (FC) switches to support FC connections from the MDC nodes and DM nodes to the tape archive or other FC storage systems.
- High-performance storage systems for administrative block storage (ABS) and data cache block storage (DCBS).
- Large-capacity storage systems for the disk archive storage tier (DAST) or tape archive storage tier (TAST).

2.3 TAS HSM System Networks

There are several required networks for a Cray TAS system. TAS networks are configured on the CIMS node in an XML configuration file. They are customized during the installation process.

Bright software uses *internal* and *external* designations to classify networks. The `esmain-net`, `ipmi-net`, for example are internal networks accessible only to the CIMS node. External networks in a TAS system are *`site-user-net`*, and *`site-admin-net`*, which enable users from outside the system to gain access. There are other network classifications within Bright, such as cloud and global, but these are not used. Additional networks may be defined, depending on the requirements of the system.

The primary networks used in a TAS system are:

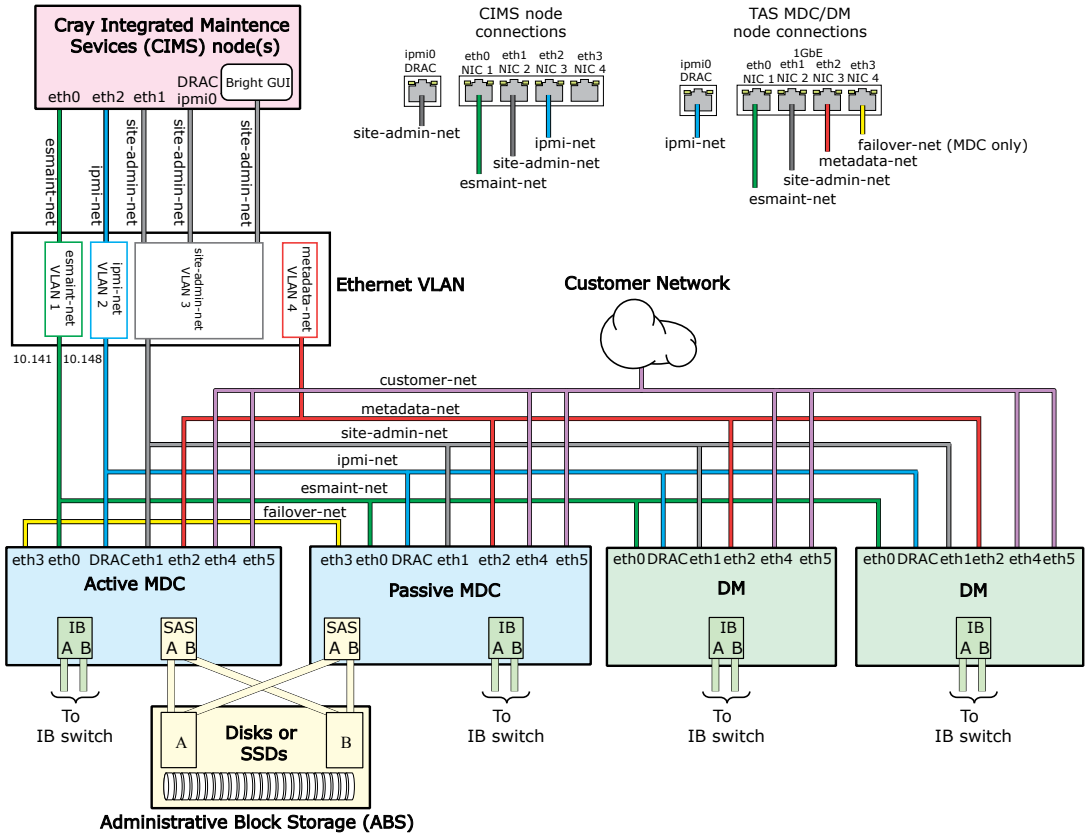
`metadata-net`	Metadata network for TAS system.
`site-admin-net`	External administration network used by site administrators to log into the CIMS server. The name and IP address of this network are customized during installation. The CIMS IPMI interface (iDRAC) may also be on this network to provide remote console and power management of the system.
`ipmi-net`	Internal Intelligent Platform Management Interface (IPMI) connected to Dell™ Remote Access Controller (DRAC). Provides remote console and power management of the nodes.
`esmaint-net`	An internal management network that connects the CIMS node(s) with the service nodes, switches, and RAID controllers. This network enables Bright to manage and provision the

service nodes and other devices in the system. When using the Bright GUI (`cmgui`) or Cluster Management Shell (`cmsh`) this network is classified as the internal management network.

`ib-net` Internal InfiniBand® network used by the service nodes for Lustre® LNET traffic.

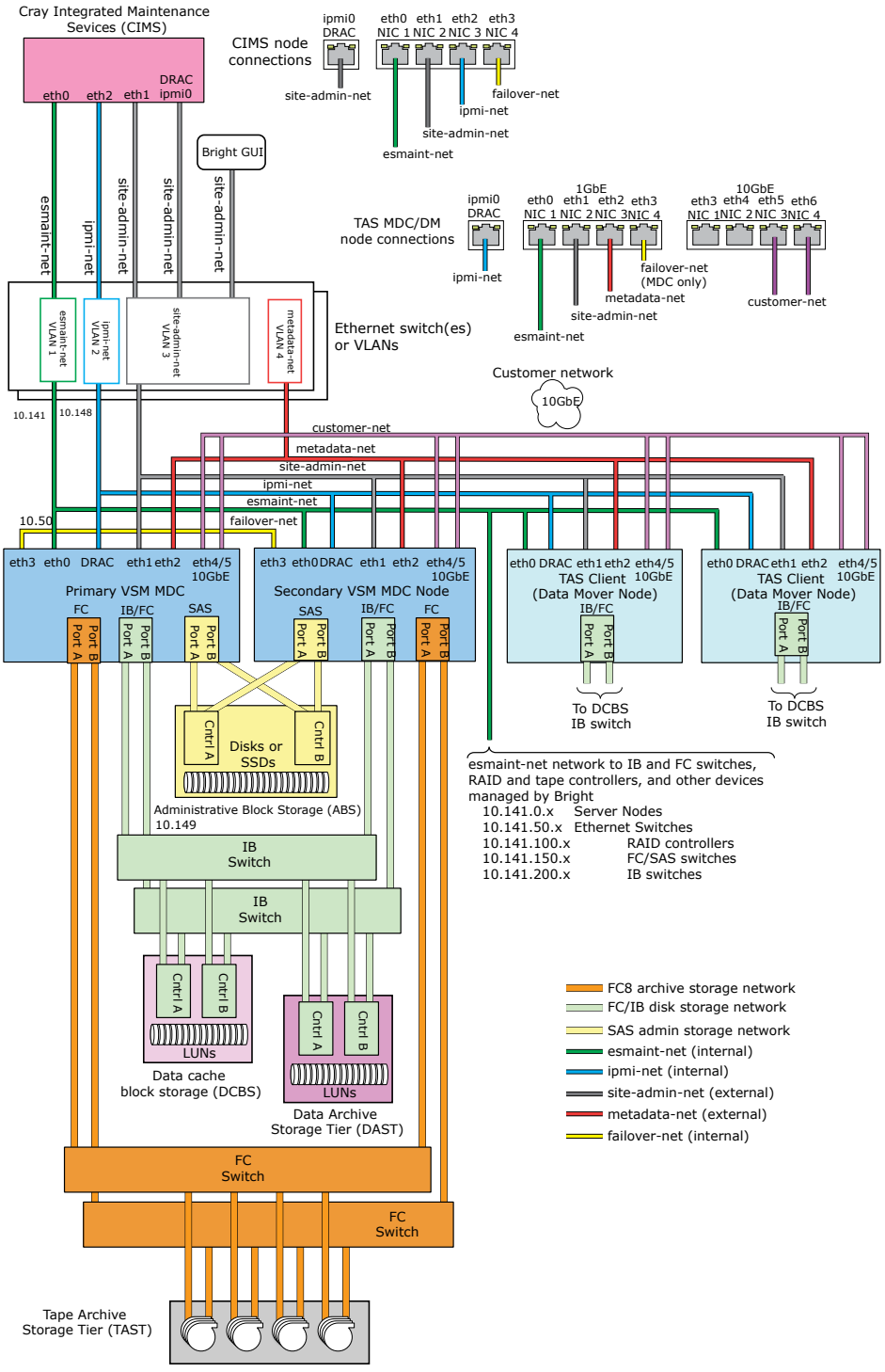
`failover-net` Internal failover network used between two servers in an HA configuration for heartbeats between the active/passive CIMS nodes. This network does not connect to a managed switch.

Figure 5. TAS HSM Gateway Networks



The networks used in a complete TAS HSM system are shown in the figure.

Figure 6. TAS HSM System Components and Networks



3 TAS HSM High Availability

The TAS system metadata controller (MDC) nodes are configured in an active/passive high-availability configuration. There is a manual or automatic failover process to move services between the nodes. The normal node operation mode has one active servicing the VSM file systems and managing the archive processes. The second MDC (passive) is powered up but does not have VSM file systems mounted. It provides no archive services. With both MDC nodes powered on, the CIMS node can monitor the health of these servers. It can provide advance notification of any hardware, networking, or file system issues.

Manual Failover

The manual failover process is managed through the `tasha` command. The `tasha` command is typically run during a failover scenario where the active node is stopped. The standby node is then converted to the active MDC and started. Refer to the `man` page on the TAS MDC node for more information.

Automatic Failover

Pacemaker software and Corosync cluster engine software provide automated failover TAS capabilities for systems. Pacemaker software detects and recovers from node and resource-level failures by making use of Corosync software messaging and membership capabilities.

`mdc` Hostname

The TAS-XX-2.0.1 release add a Pacemaker resource that defines the hostname `mdc` to always be the active VSM MDC. Simply SSH to `mdc` from the CIMS node to login to the active MDC.

```
cims# ssh -l root mdc
```

3.1 Manual MDC Failover

Procedure

1. Log in to the CIMS node as `root` and use SSH to log in to the active MDC (`tas-mdc1`).

   ```
   tas-cims1# ssh tas-mdc1
   tas-mdc1#
   ```

2. Stop all VSM services on the active node (`tas-mdc1` in this procedure). Then unmount all VSM and TAS administrative file systems (`/tas_admin`), and stop VSM services.

   ```
   tas-mdc1# tasha stop
   ```

   The passive MDC (`tas-mdc2`) is then given control of the VSM file systems.

The passive MDC then mounts the VSM file systems and TAS administrative file systems. The VSM services are then started. After this point, the passive MDC completed its conversion to the active MDC.

```
tas-mdc2# tasha start
```

3.2 Automatic MDC Failover

Pacemaker

TAS HSM high-availability (HA) is implemented using the Pacemaker cluster stack. Pacemaker starts and stops services. It contains logic for ensuring serivces are running and that monitored services are running only on one node.

Pacemaker relies on the Corosync messaging layer for cluster communications. Corosync implements the Totem single-ring ordering and membership protocol. It also provides Pacemaker with UDP- and InfiniBand-based messaging, quorum, and cluster membership. Pacemaker and Corosync software use the Open Cluster Framework (OCF) standard cluster APIs.

Pacemaker is composed of five key components:

- Cluster Information Base (CIB)
- Cluster Resource Management daemon (CRMd)
- Local Resource Management daemon (LRMd)
- PE or Policy Engine
- STONITHd (shoot the other node in the head)

Use the Pacemaker command `crm_mon -n` to check the state of the cluster:

```
tas-mdc1# crm_mon -n

    Last updated: Mon Feb  9 13:08:39 2015
    Last change: Fri Jan 30 13:48:05 2015 via crm_attribute on tas-mdc2
    Stack: classic openais (with plugin)
    Current DC: tas-mdc2 - partition with quorum
    Version: 1.1.10-14.el6-368c726
    2 Nodes configured, 2 expected votes
    5 Resources configured

    Node tas-mdc1: online
            fenceMDC2       (stonith:fence_ipmilan):        Started
            VSM               (ocf::heartbeat:vsm):         Started
            TAS_Admin    (ocf::heartbeat:Filesystem):       Started
            VSMFS             (ocf::heartbeat:vsmfs):       Started
    Node tas-mdc2: online
            fenceMDC1       (stonith:fence_ipmilan):        Started
```

This display shows the two nodes, `tas-mdc1` and `tas-mdc2` in the cluster. The node status, `online`, is shown next to the node name. The monitored resources are listed below the node where they are running.

- `TAS_Admin` - manages the `/tas_admin` file system
- `VSM` - manages the VSM service
- `VSMFS` - manages the VSM file system mount

Each node runs a STONITH fencing agent.

The following example shows the 2 MDC nodes in standby mode:

```
tas-mdc1# crm_mon -n
Last updated: Sat Mar 28 11:19:48 2015
Last change: Tue Mar 17 13:09:27 2015 via crmd on tas-mdc2
Stack: classic openais (with plugin)
Current DC: tas-mdc2 - partition with quorum
Version: 1.1.10-14.el6-368c726
2 Nodes configured, 3 expected votes
5 Resources configured

Node tas-mdc1: standby
Node tas-mdc2: standby
```

Corosync

The Corosync Cluster Engine is a group communication system. It has additional features for implementing HA within applications. Corosync is started as a regular system service. To check Corosync connectivity and display a summary of communication-ring health, use the `corosync-cfgtool -s` utility from the TAS MDC node:

```
tas-mdc1# corosync-cfgtool -s
Printing ring status.
Local node ID 16790026
RING ID 0
        id      = 10.50.0.1
        status  = ring 0 active with no faults
RING ID 1
        id      = 10.141.0.1
        status  = ring 1 active with no faults
tas-mdc1#
```

4 Versity Storage Manager (VSM) Software

Versity Storage Manager (VSM) software is a proprietary software product constructed around SAM-QFS core technology. VSM software retains the same command set and administration methods used for SAM-QFS file system administration.

Versity Software utilizes an open source file format—GNU TAR. Files written with VSM can be read directly from the media without the use of any additional software, either open source or proprietary. VSM media is also self describing. This means that **all** file metadata is stored on the media with the files in an open standard format.

The Versity file system is a multi-threaded, storage-management system. The VSM archiving capability provides continuous file-system data protection. Data is archived across all tiers of storage. This means that all the data is available, but not necessarily on high-cost storage. Multiple file systems are used to take advantage of full system throughput capability. It:

- Is based on Open-source SAM-QFS
- Runs on Linux
- Has a flexible feature set for defining archive storage policies
- Has highly tunable system parameters to ensure optimal performance
- Has powerful tools to manage primary disk utilization
- Has a dynamic ability to grow or shrink file systems as needed
- Has an open-source file format, open APIs
- Supports virtually unlimited archive
- Has a simple approach that minimizes management overhead
- Automatically migrates data to media
- Efficiently utilizes storage media—releasing obsolete files and repacking moveable ones

4.1 VSM Licensing

To activate the software, first obtain a license key from Versity. After obtaining a license key, place the license key in the `/etc/opt/vsm/VSM.license.`*`hostname`* file. License files must be named `VSM.license.`*`hostname`*, the Pacemaker VSM resource copies the correct license in place when an MDC becomes active.

```
devmdc1# ls /etc/opt/vsm/VSM.license*
/etc/opt/vsm/VSM.license
/etc/opt/vsm/VSM.license.tas-mdc1
/etc/opt/vsm/VSM.license.tas-mdc2
```

The key is generated with the system UUID and is tied to a specific host. Use the following command from the MDC node to obtain the system UUID.

```
tas-mdc1# /usr/sbin/dmidecode -s system-uuid
```

4.2 High Capacity and Volume Management

The VSM file system uses a true 64-bit address space. This allows large files to be striped and easily spread across various disks and RAID devices. VSM imposes no limit on the number of file systems. Each file system can specify up to 1020 device partitions (LUNs). Each device partition can range in size up to 4.5 Petabytes of data. VSM supports up to a maximum of 4.5 Exabytes of data per file system.

The Versity file system supports two kinds of disk allocation—striped and round robin. The volume management features are specified in the master configuration file (MCF) and the mount parameters.

VSM does not require an external volume management application. Such applications, however, may supplement the VSM system.

The inode space is dynamically allocated. This allows for a virtually unlimited number of VSM files. Disk storage space availability is the only limitation. Each inode entry is a 1024-byte block of descriptive information about directory or file characteristics.

Inodes are located in the `.inodes` file under the mount point. Inodes can be separated or included on the file data devices. If inodes are separated, they are located on the dedicated metadata devices. The number of files may be increased by adding additional metadata devices. The sizes of the metadata devices restricts the number of files in the VSM file system. The number of files can be increased by dynamically growing the metadata devices.

> **NOTE:** For optimal performance, Versity currently recommends no more than 300 million files per file system.

The following table displays standard inode times. The first three times POSIX times and the last three times are VSM times.

Table 3. Standard Inode Times

`modification time`	The time when the file was last modified
`access time`	The time when the file was last accessed
`changed time`	The time when the inode information was last changed
`attributes time`	The time the attributes specific to the file were last changed
`residence time`	The time the file changed from online to offline or offline to online
`creation time`	The time the file was created

4.3 Paged and Direct I/O

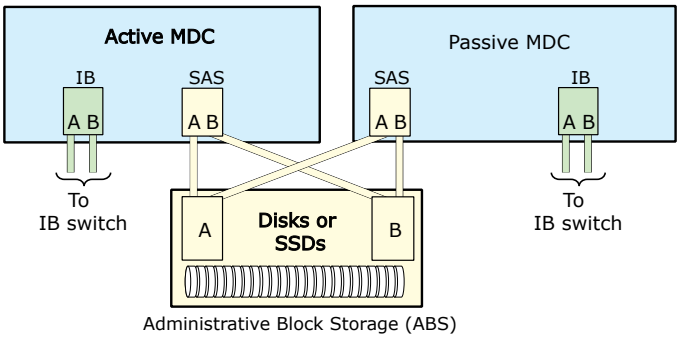
Paged refers to cached and buffered I/O. User data is stored or cached in memory pages to be used by the kernel and written to disk. Direct I/O writes user data directly from user memory to disk. To use direct I/O instead of the default paged I/O, set the `O_DIRECT` flag in the file mode on the `open()` system call.

> **NOTE:** For direct I/O, the buffer must be aligned to 4k and each buffer size must be a multiple of 4k.

4.4 VSM Metadata Storage

File systems use metadata to reference files and directory information. VSM enables administrators to separate file system metadata and file data by storing it on different devices. Multiple metadata devices may be defined in order to help reduce data-device head movement and to reduce rotational latency. Defining multiple devices can also improve the RAID cache utilization. The Cray TAS system uses NetApp block storage or similar devices using SAS for Administrative Block Storage (ABS).

Figure 7. TAS Administrative Block Storage



4.5 Fast File Recovery

Fast file recovery is a key function of VSM. The system allows for quick recovery after an unexpected outage. VSM dynamically recovers after a failure using:

1. Identification records
2. Error checking on all critical I/O operations
3. Serial writes

After a file system outage, execute a `samfsck -F` to repair the file system. With fast file recovery, however, it is possible to schedule the `samfsck` at a later time if necessary.

If there are errors on the first `samfsck -F`, Cray recommends running the `samfsck -F` process again—until there are no errors.

4.6 Shared File System Support

The TAS-VSM shared file system has performance advantages over a network file system. File data in an NFS mounted file system is accessed over the network. File data on disks in a shared VSM file system is directly accessed by the hosts. File data travels via direct-access I/O (local path I/O) as VSM moves data directly between the disks and the hosts.

Multiple host systems can be mounted in a distributed file system. One host is the metadata server for the file system. Other hosts are designated as the clients. Only one host can serve as the metadata controller (MDC) at

any given moment. More than one host, however, may be configured to potentially function as a metadata server in high-availability (HA) configurations.

4.7 VSM Configuration File System

The VSM configuration uses the `/tas_admin` file system.

> **NOTE:** In release TAS-XX-2.0.1 the `/tas_admin/VSM/etc` directory stores the system configuration files and is bind mounted to `/etc/opt/vsm` on the active MDC node.

Library Catalog Definition

The library catalog is defined in the VSM `mcf` file as `/tas_admin/VSM/var/catalog/`*`catalog_name`*, for example `T200`. Cray recommends that an absolute library catalog path name be included in the `mcf` file. The configuration example below is the best practice and includes the library catalog definition:

```
 #       T200 with 2 LTO6
#
# Equipment                       Eq      Eq      Fam.    Dev.  Additional
# Identifier                      Ord     Type    Set     State Parameters
#-----------                      ---     --      ------ ------ ------------------
-
/dev/tape/by-id/scsi-1SPECTRA_PYTHON_9410005694 100     rb      T200    on
/dev/tape/by-id/scsi-321130090a5005694-nst      102     tp      T200    on
/dev/tape/by-id/scsi-321140090a5005694-nst      103     tp      T200    on
/dev/tape/by-id/scsi-321120090a5005694-nst      104     tp      T200    on
/dev/tape/by-id/scsi-321110090a5005694-nst      105     tp      T200    on
```

4.8 VSM Disk Allocation Units

VSM supports a wide range of block sizes, called disk allocation units (DAUs). The DAU is set when the file system is initialized (`sammkfs`). This is an important feature which, when set correctly, can reduce overhead associated with read-modify-write operations. It is especially useful when dealing with applications that access large files.

4.9 VSM File System Types

VSM supports `ms` and `ma` file systems. The file system type is defined in the master configuration file (`mcf`). The `ms` file system maintains data and file system metadata on the same devices. The `ma` file system maintains data and file system metadata on different devices.

Table 4. `ms` and `ma` File Systems

`ms` File System	**`ma` File System**
The `ms` file system maintains data and file system metadata on the same devices	The `ma` file system maintains data and file system metadata on different devices

ms File System	ma File System
A ms file system type must use the md device type. The metadata and the data are both written to the md device(s).	A ma file system type writes the metadata to mm devices while the data is written to either md or mr devices.

The mr device type can be adjusted to have different DAU sizes, but the size is in power of two increments. The DAU size can be set to 8 kilobytes up to a maximum 32,768 kilobytes.

The default mr DAU is 64 kilobytes. When the md device type is used in a ma file system, it is used to store only data—never metadata. When the md device type is used in an ms file system, it stores both metadata and data. The md and mm device types use a dual allocation scheme (small and large allocation is used).

On the md device, the small DAU is 4k and the large DAU is 16, 32, or 64 kilobytes. On the mm device, the small DAU is 4k and the large DAU is 16k. The default DAU size is 64 kilobytes. Different DAU sizes, however, may be specified at file-system-initialization time by using the sammkfs command from the MDC node:

```
tas-mdc1# sammkfs -a allocation-size family_set
```

4.10 Set VSM File System Stripe Width

A stripe width of 0 (stripe=0) indicates that file system allocation is round robin. The VSM file system allocates each file on a different device in a round robin fashion. A stripe width greater than 0 indicates that the VSM file system allocates *n* DAUs for each file on one device before switching to the next device.

The stripe width should be set based on workload. If the machine is used in a general-purpose environment, then stripe=0 is optimal. If the workload is large file streaming I/O access, then stripe>0 is optimal. The stripe width mount option has a different default for the stand-alone file system versus the shared file system. The default stripe width on stand-alone ms and ma file systems is 128K. The default stripe width on shared ms and ma file systems is round robin (stripe=0). To set the stripe width, use the -o stripe=*n* mount option where *n* is a numeric value representing the stripe width. If the stripe width is set to 0 then the default round robin allocation is used. A stripe width=*n* (>0), indicated *n* DAUs are allocated on a device for one file before switching to the next device.

Table 5. Default VSM Stripe Widths for a Stand-alone File System

DAU Size	Stripe Width (default)	Amount Allocated
16 kilobytes	8 DAUs	128 kilobytes
32 kilobytes	4 DAUs	128 kilobytes
64 kilobytes	2 DAUs	128 kilobytes
>=128 kilobytes	1 DAU	1 DAU

5 Manage a TAS System with Bright

Tiered Adaptive Storage (TAS) systems are managed using Bright Cluster Manager (Bright) software from the CIMS node.

Refer to the *Bright Cluster Manager 6.1 Administrator Manual* for detailed information about Bright software management. PDF files for the Bright manuals are stored on the CIMS node in the `/cm/shared/docs/cm` directory. They are linked to from the `/root` directory.

A CIMS node runs Bright software. It provides a system management interface for all system service nodes, RAID controllers, and switches. Bright communicates with each device over the `esmaint-net` maintenance network and over the `ipmi-net` IPMI network. The Bright management interfaces are:

- `cmsh` — The Cluster-management shell or command line interface (CLI).
- `cmgui` — A graphical user interface (GUI)
- `cmd` or CMDaemon — A process that runs on all TAS system nodes. On the CIMS node, CMDaemon responds to requests from cmsh or cmgui. It communicates with the CMDaemon processes running on each service node. The CMDaemon processes on each service node communicate only with the CMDaemon processes running on the CIMS node.

Either the `cmgui` or `cmsh` can be used to manage the system. There may be certain tasks that are more easily visualized using `cmgui`. Other tasks may be more efficient using `cmsh`.

Be aware that Bright software runs a database to manage the system. Therefore, modifications to the system are not invoked until they are committed in `cmsh` or saved in `cmgui`. All service nodes PXE boot from software images stored in the `/cm/images` directory on the CIMS node.

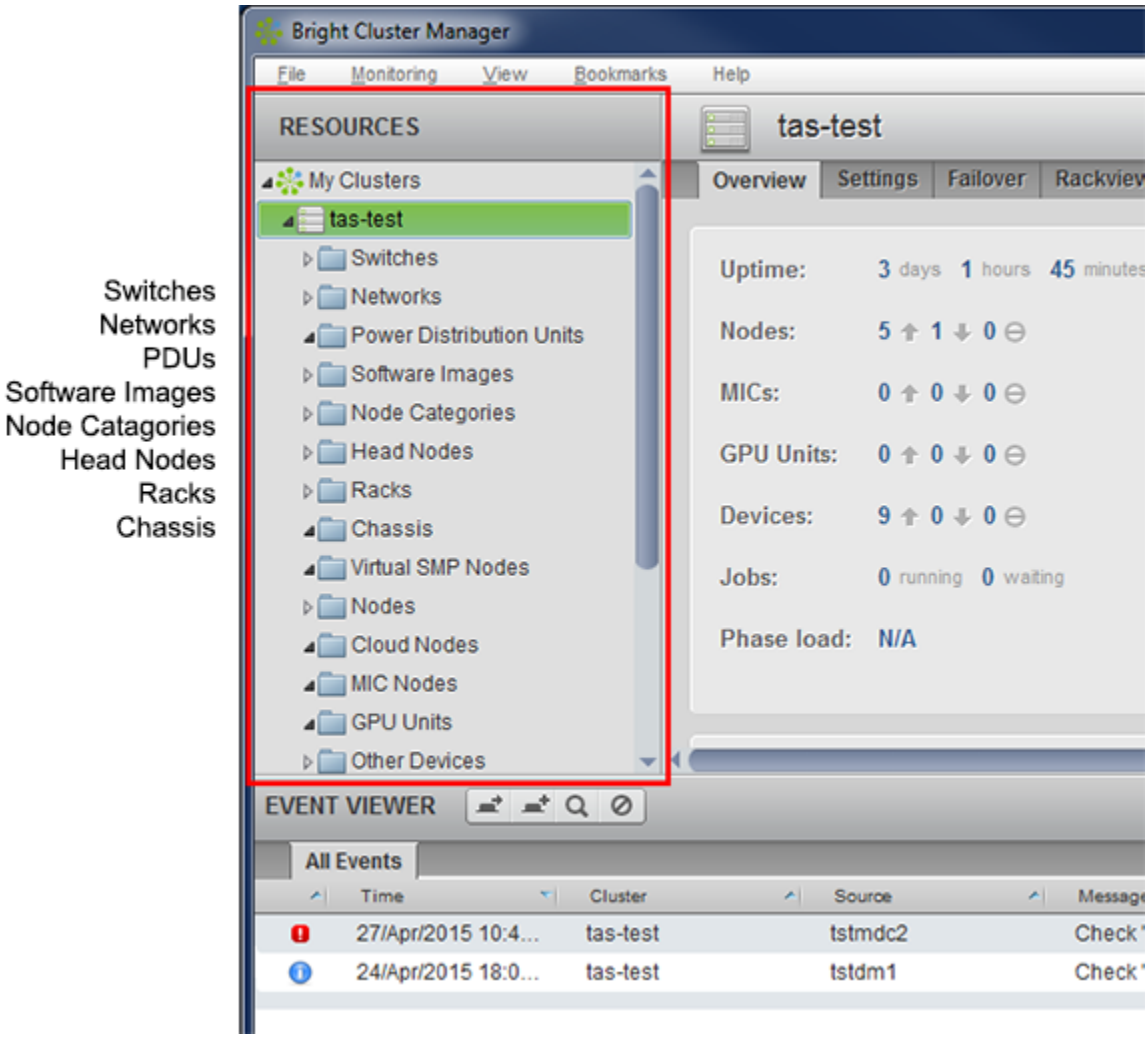
> **NOTE:** With release TAS-XX-2.0.1, all of the VSM configuration files (which includes the `archiver.cmd` and `mcf` files) are now stored in a single location in the shared TAS administrative file system. They are under the `/tas_admin/VSM/etc` directory and bind mounted to `/etc/opt/vsm` on the active MDC node. This simplifies administration and eliminates the need to synchronize the configuration files on an active MDC node back to the MDC software image in Bright, after modifications. This also eliminates the need to `chroot` into the MDC software images on the CIMS node, then update the running MDC images using Bright.

Cray recommends that software-configuration changes to service node software are made to the software image on the CIMS node using the `chroot` environment, and pushed out to the running node (updating a node). Note that changes to the MDC configuration in `/etc/opt/vsm` are made on the active MDC and do not need to be sync'd. Alternatively, the software image on the running node can be captured or "grabbed" by Bright and stored on the CIMS node—but only if the Bright exclude lists are configured properly. Grabbing a software image from a running node could inadvertently add user file systems or other unwanted data to the software image.

5.1 Use `cmgui` to Manage TAS

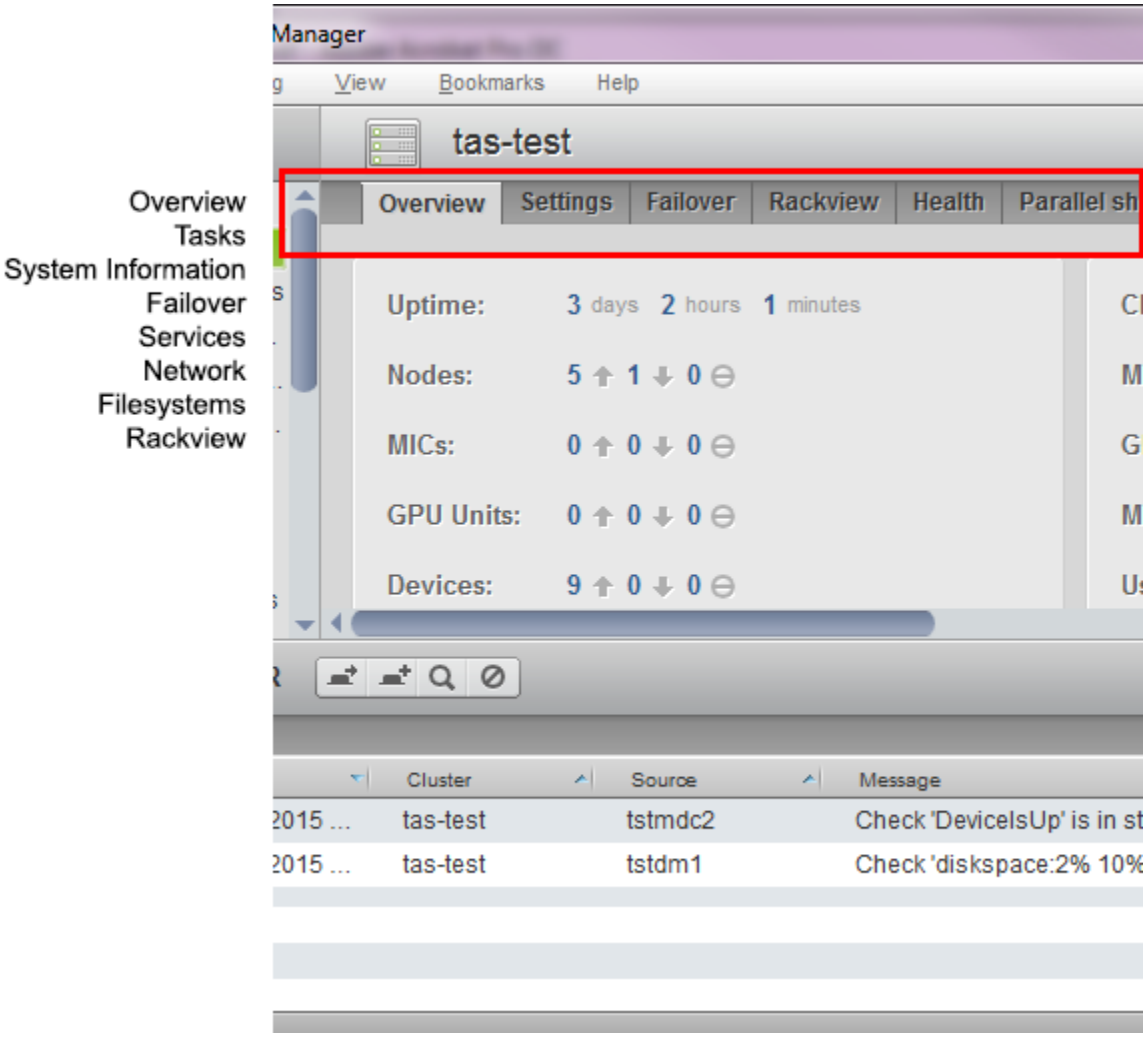
The `cmgui` RESOURCES pane lists all the devices managed by Bright and Bright categories, software images, node groups, and networks. Each of these items can be selected and managed using the `cmgui`. The `cmgui` enables the **Save** button to commit changes to the system.

Figure 8. `cmgui` RESOURCES Pane



When an object is selected in the RESOURCES pane, the tabs at the top of the GUI change. For more information about using `cmgui` and how to perform common administrative tasks for Cray system.

Figure 9. `cmgui` Tabs



5.1.1 Run CMGUI from the CIMS Node

About this task

The `cmgui` program may be run on the CIMS node and displayed to a remote X Window System running on a Linux®, Windows®, or Mac OS® desktop or other platform.

Procedure

1. On a remote system such as a Linux desktop or PC, start an X-server application such as Xming or Cygwin/X.

2. Enter the following command to log in to the CIMS (in this example, `cims`) with SSH X forwarding.

```
remote% ssh -X root@cims
cims#
```

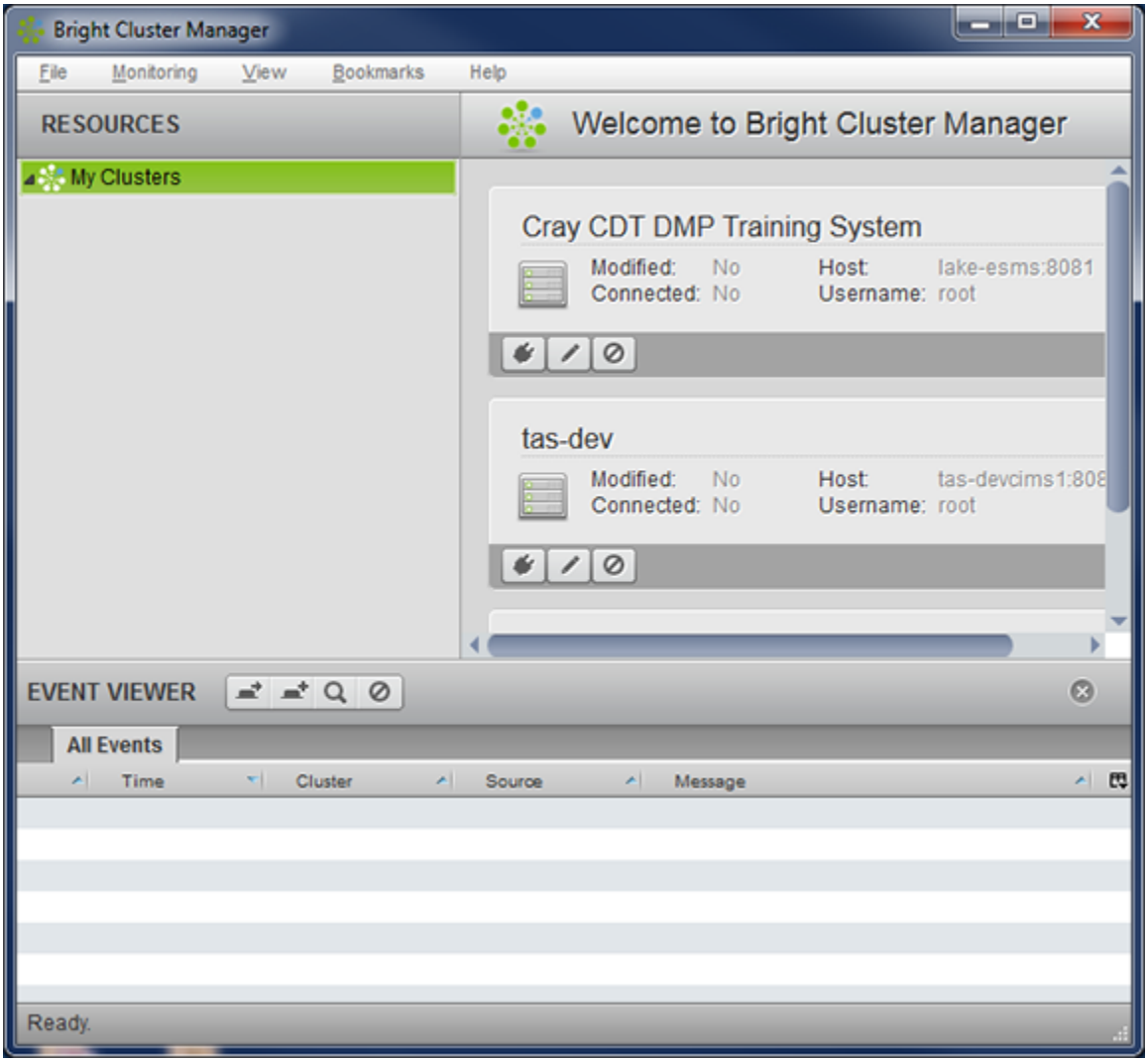
3. Start the `cmgui` program.

```
cims# /cm/shared/apps/cmgui/cmgui &
```

4. Select **Add a new cluster**. Enter the CIMS node hostname, username, and password and click OK.

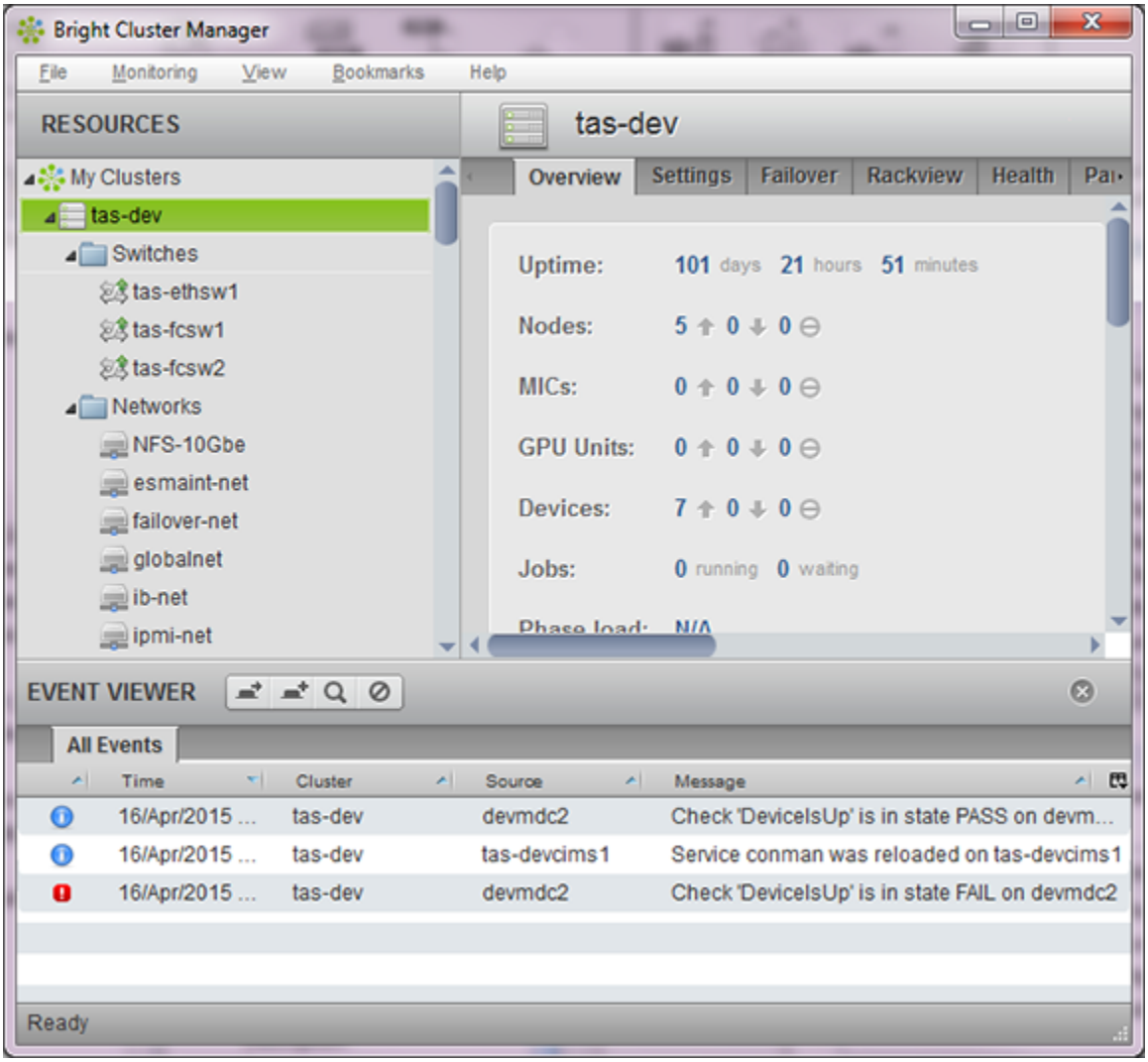
5. Select the power plug icon and enter the system password to connect to the system.

Figure 10. `cmgui` Window



The `cmgui` window displays the system configuration.

Figure 11. `cmgui` Connect-to-Cluster



5.1.2 Install and Run `cmgui` on a Remote System

About this task

The Bright GUI (`cmgui`) can be installed on a Linux®, Windows®, or Mac OS® platform and supports a virtual network computing (VNC) server for remote connections.

> **IMPORTANT:** Communication between the remote computer and the CIMS node should be encrypted.

Cray recommends SSH port forwarding or SSH tunneling. When running the `cmgui` program from the remote computer, `cmgui` connects to the CIMS node using SSL. Cray also recommends using SSH port forwarding when using VNC. The Linux, Windows, or Mac OS installation software for the cluster management GUI (`cmgui`) is located in `/cm/shared/apps/cmgui/dist` on the CIMS node.

> **IMPORTANT:** Whenever you update the CIMS node with the latest ESM software, always reinstall the updated `cmgui` software on the remote systems.

Procedure

1. Copy the Windows `.exe` file, (`install.cmgui.6.1.`*`revision`*`.exe`), Linux compressed `TAR` file (`cmgui.6.1.`*`revision`*`.tar.bz2`, or Mac OS package file (`install.cmgui.macosx.6.1.`*`revision`*`.pkg` from the `/cm/shared/apps/cmgui/dist` directory on the CIMS node to a `tmp` directory on the remote system.

   ```
   remote% scp root@tas-cims1:/cm/shared/apps/cmgui/dist/* /tmp
   ```

2. Copy the PFX certificate file (`admin.pfx`) from the `root` directory of the CIMS node to a secure location on the remote system so that it can be used for authentication purposes. Rename the file so that you can identify the system it authorizes (*`systemname`*`-admin.pfx` for example).

   ```
   remote%  scp root@tas-cims1:/root/admin.pfx /securelocation/systemname-admin.pfx
   ```

3. Install the software.

   a. On Windows systems, execute the installer `.exe` file and follow the instructions.

   b. On Linux systems, extract the files using the `tar` command:

      ```
      remote%  tar -xvjf cmgui.6.1-revision.tar.bz2
      ```

   c. On Mac OS systems, click on the `.pkg` file and follow the instructions.

4. Start `cmgui` and select the power plug icon and enter the PFX certificate password to connect to the system.

5.2 Bright Node Categories for TAS

Bright software implements the concept of service *node categories* and *node groups* to manage groups of nodes.

Categories specify a number of parameters that are common to all members of the node category such as software image, finalize script, and disk setup. The node installer configures each node image during provisioning from the CIMS node. A service node must be associated with a single node category. Category parameters can be overridden on a per node basis, if desired, by setting configuration parameters for the node, instead of the node category.

Bright software configures a separate interface for each node because the IP addresses that Bright uses are specific to each node. Software images are common across multiple nodes, so the Bright interface files must reside in the Bright database and be placed on service nodes at boot time.

Service nodes can belong to several different node groups, and there are no parameters associated with node groups. Node groups are typically used to invoke commands across several nodes simultaneously.

Table 6. Bright Node Categories for TAS

Category	Description
`default`	Default category configured by installation software for service nodes. **Do not delete the `default` category**.
`tas-dm`	Default category for data mover (DM) nodes.
`tas-mdc`	Default category for metadata controller (MDC) nodes.

Category	Description
`tas-cmm`	Default category for migration manager (CMM) nodes.
`tas-cma`	Default category for migration agent (CMA) nodes.

Node categories provide control over several node parameters such as:

revision Object revision.

bmcpassword Password used to send `ipmi`/`ilo` commands to nodes. The baseboard management controller (BMC or iDRAC) password is inherited from the `base` partition and is not set for the node category in Cray TAS systems.

bmcusername User name used to send `ipmi`/`ilo` commands to nodes. Inherited from the `base` partition and is not set for the node category in Cray TAS systems.

defaultgateway Default gateway for the category.

filesystemexports Configure the entries placed in `/etc/exports`.

filesystemmounts Configure the entries placed in `/etc/fstab`.

installbootrecord Install boot record on service node local disk to enable booting without a CIMS node.

installmode Specifies software install mode. Typically set to `auto`. Can be `auto`, `full`, `main`, or `nosync`.

ipmipowerresetdelay Delay used for `ipmi`/`ilo` power reset, default is `0`.

managementnetwork Specifies the network used for management traffic. Always set to `esmaint-net`.

name Name of category.

nameservers List of name servers the category will use.

newnodeinstallmode Default install mode for new nodes. Typically set to `full`.

roles Assign the roles the node should play.

searchdomain Search domains for the category.

services Manage operating system services.

softwareimage Software image the category will use.

timeservers List of time servers the category will use.

usernodelogin `ALWAYS` or `NEVER` allow a user to log in to the node.

disksetup Disk setup for nodes.

excludelistfullinstall	Exclude list for full install.
excludelistgrab	Exclude list for grabbing the image running on the node to an existing image.
excludelistgrabnew	Exclude list for grabbing to a new image.
excludelistsyncinstall	Exclude list for a sync install. Specifies what files and directories to exclude from consideration when copying parts of the filesystem from a known good software image to the node.
excludelistupdate	Exclude list for updating a running node.
finalizescript	Finalize script for category.
initializescript	Initialize script for category.
notes	Administrator notes.

5.3 Bright Node Groups for TAS

Node groups simplify management and control activities and enable administrators to perform commands on a group of nodes simultaneously. Typical node groups are listed below:

Table 7. TAS Node Groups

Node Group	Description
DM	All DM service nodes
MDC	All MDC service nodes
CMA	All CMA service nodes
CMM	All CMM service nodes

5.4 TAS Software Image Management

A TAS software image is a blueprint for the contents of the local file systems on service nodes. Software images are stored in the `/cm/images` directory on the CIMS. They contain a full Linux™ CentOS 6.5 file system and other customizations. Software images are typically named with prefixes such as ESF or TAS. They include the release version and date in the image name.

When a service node boots, the CIMS provisions it with a copy of the assigned software image configured in the node category. Software images for service nodes are modified on the CIMS using Linux tools and commands such as `rpm` and `chroot`. They are then pushed out to the node using `cmgui` Update node button or `cmsh imageupdate` command from `device` mode.

NOTE: In release TAS-XX-2.0.1 the `/tas_admin/VSM/etc` directory stores the `mcf` file and is bind mounted to `/etc/opt/vsm` on the active MDC node.

IMPORTANT:

The software images shipped with the system (`TAS-XX-2.0.1-201510011105-dm` or `TAS-XX-2.0.1-201510011105-mdc` for example), have been configured with a base CentOS 6.5 operating system, VSM software, and TAS software tools and utilities by Cray manufacturing. They should not be corrupted. Always backup the factory configured software images. Always clone the factory configured software image to a new software image and make modifications to that new image. All software images in the `/cm/images` directory on the CIMS must be backed-up regularly and clearly named for the site.

Keep a record of the changes made to each software image configuration, so that future software updates can be implemented more easily.

Note that `default-image` and `default-image.previous` (created when the CIMS software is updated) are default service node images created by Cray. They use the SLES11SP3 operating system. Do not clone these images to create DM node or MDC node images.

When a node boots, the node provisioning system sets up the node with a copy of the software image that is configured for the node. Software images are assigned to a node `category` in Bright. Nodes are then assigned to a node `category` to configure their operation. This enables administrators to have flexibility in controlling specific software images for various node types. Use the `clone` command to create a copy of the software image and configure it for testing using the Bright.

Software images should be modified only in the `/cm/images` directory from the CIMS `chroot` environment. Grabbing a running software image from a node to a software image file can be problematic if `excludeslistgrab` and `excludelistgrabnew` exclude lists are not configured correctly for the site. Cray recommends that administrators modify a cloned known-good software image, test it on a single node. If the configuration changes are satisfactory, then update the node category or node group to minimize user interruptions.

To list the system software images stored on the CIMS node, login and list the contents of `/cm/images`.

```
tas-cims1:/cm/images# ls -l
total 4944780
dr-xr-xr-x 27 root root       4096 Jun  5  2014 ESF-XX-2.2.0-201404151111
dr-xr-xr-x 27 root root       4096 Jun  5  2014 ESF-XX-2.2.0-201404151111.backup
dr-xr-xr-x 27 root root       4096 Jun  5  2014 TAS-1.0.2-201406051030-dm
dr-xr-xr-x 30 root root       4096 Jun 23  2014 TAS-1.0.2-201406051030-mdc
dr-xr-xr-x 27 root root       4096 Jan 21 09:33 TAS-XX-2.0.1-201510011105-DM
dr-xr-xr-x 30 root root       4096 Mar  9 15:38 TAS-XX-2.0.1-201510011105-MDC
drwxr-xr-x 24 root root       4096 Jun  4  2014 default-image
```

5.4.1 TAS DM Node Category

A TAS DM node category (`tas-dm` for example) in Bright Cluster Manager (Bright) is configured by Cray manufacturing to boot and configure DM nodes. The `tas-dm` category is configured with the DM node software image (`TAS-XX-2.0.1-201510011105-dm` for example). The category exclude list for each type of install mode is configured for this category. This determines which files are updated or left untouched when sync'd with the software image on the CIMS node.

The DM node category file system mounts include `/vsm/tasfs1` and other shared file systems from the CIMS node.

Use the DM node category to make simple configuration changes to all DM nodes (all DM nodes assigned to the category).

5.4.2 TAS MDC Node Category

A TAS MDC node category (`tas-mdc` for example) is configured by Cray manufacturing to boot and configure MDC nodes. The `tas-mdc` category is configured with the MDC node software image (`TAS-XX-2.0.1-201510011105-mdc` for example). The category exclude list for each type of install mode is configured for this category. This determines which files are updated or left untouched when sync'd with the software image on the CIMS node.

The MDC node category configures file system mount points (`/vsm/tasfs1` and the TAS administration file system `/tas_admin`). Other shared file systems are mounted from the CIMS node.

Use the MDC node category `tas-mdc` to make simple configuration changes to all MDC nodes (all MDC nodes assigned to the category).

5.4.3 Change TAS Node Category Settings

Prerequisites

Bright software is operational.

About this task

Node categories group similar nodes together so that they can load a specific software image when the nodes PXE boot. Bright enables administrators to make changes—either to the software image or to the node category.

Procedure

1. Log in to the CIMS node as `root` and start `cmsh`.

```
cims#  cmsh
[cims]%
```

2. Switch to `category` mode and list the Bright categories configured for the system.

```
cims#  category
[cims->category]% list
Type   Name (key)               Software image
------ ------------------------ ------------------------
Node   default                  default-image
Node   tas-cmm                  TAS-XX-2.0.1-201510011105-CMM
Node   tas-cmm-6.5              TAS-XX-2.0.0-rc1
Node   tas-cma                  TAS-XX-2.0.1-201510011105-CMA
```

   The `tas-cmm` category loads the `TAS-XX-2.0.1-201510011105-CMM` software image from the CIMS node `/cm/images` directory.

3. To display the settings for a category, use the `category show` subcommand from `category` mode.

```
[cims->category]% show tas-cmm

[tas-devcims1->category]% show tas-mdc
Parameter                        Value
-------------------------------- ------------------------------------------------
BMC Password                     < not set >
BMC User ID                      -1
BMC User name
Default gateway                  172.30.86.1
Disk setup                       <2436 bytes>
```

```
Exclude list full install          <312 bytes>
Exclude list grab                  <1024 bytes>
Exclude list grab new              <1024 bytes>
Exclude list sync install          <1441 bytes>
Exclude list update                <2892 bytes>
Filesystem exports                 <0 in submode>
Filesystem mounts                  <8 in submode>
Finalize script                    <4162 bytes>
GPU Settings                       <0 in submode>
Initialize script                  <0 bytes>
Install boot record                yes
Install mode                       AUTO
Ipmi power reset delay             0
Management network                 esmaint-net
Name                               tas-mdc
Name servers
New node install mode              FULL
Notes                              <0 bytes>
Provisioning associations          <1 internally used>
Require FULL Install Confirmation  no
Revision
Roles                              <0 in submode>
Scaling governor
Search domain
Services                           <0 in submode>
Software image                     TAS-XX-2.0.1-201510011105-CMM
Time servers
Type                               Node
User node login                    ALWAYS
```

4. Enter the use command to set the category. Then use the `get defaultgateway` subcommand from `category` mode to list all parameters configured for that category.

```
[cims->category]% use tas-cmm
[cims->category[tas-cmm]% get softwareimage
TAS-XX-2.0.1-201510011105-CMM
```

TIP: Press the `Tab` key to display a list of valid subcommands whenever working in `cmsh`.

5. To configure a new software image for the `tas-cmm` category for example, use the `set` command.

```
[cims->category[tas-cmm]% set softwareimage TAS-XX-2.0.1-201510011105-CMM_Test
[cims->category*[tas-cmm*]%
```

IMPORTANT: The asterisk (`*`) symbols on the `cmsh` command line indicates that the current changes have not been saved (committed) into the Bright database.

6. Commit the changes to the Bright database so that the `tas-cmm` category loads the new software image.

```
[cims->category*[tas-cmm*]% commit
[cims->category[tas-cmm]%
```

7. Reboot the node to load the new software image.

5.4.4 TAS Software Image Properties

The `cmgui` lists the system software images in the **Software Images** section of the **RESOURCES** pane. Use `cmsh softwareimage` mode, and enter `list` to list the software images on the system.

To select an image and show its properties, first `use` the software image. Then enter the `show` command to display its properties from `cmsh softwareimage` mode.

To display a specific property such as the software image kernel parameters, enter `get` from `cmsh softwareimage` mode, (press the `Tab` key to display a list of valid options). Then provide the `kernelparameters` option as shown.

```
tas-cims1# cmsh
[tas-cims1]% softwareimage
[tas-cims1->softwareimage]% list
 Name (key)                          Path                                          Kernel version
 ----------------------------------  --------------------------------------------  ----------------------
 ESF-XX-2.2.0-201510252041           /cm/images/ESF-XX-2.2.0-201510252041          2.6.32-358.6.1.el6.x86_64
TAS-XX-2.0.1-201510011105-dm         /cm/images/TAS-XX-2.0.1-201510011105-dm       2.6.32-358.6.1.el6.x86_64
TAS-XX-2.0.1-201510011105-mdc        /cm/images/TAS-XX-2.0.1-201510011105-mdc      2.6.32-358.6.1.el6.x86_64
TAS-XX-2.0.1-201510011105-rev1-dm    /cm/images/TAS-XX-2.0.1-201510011105-rev1-dm  2.6.32-358.6.1.el6.x86_64
TAS-XX-2.0.1-201510011105-rev1-mdc   /cm/images/TAS-XX-2.0.1-201510011105-rev1-mdc 2.6.32-358.6.1.el6.x86_64
 default-image                       /cm/images/default-image                      3.0.80-0.5-default
```

Show TAS Software Image Properties

```
[tas-cims1->softwareimage]% use TAS-XX-2.0.1-201510011105-rev1-dm
[tas-cims1->softwareimage[TAS-XX-2.0.1-201510011105-rev1-dm]]% show

Parameter                      Value
------------------------------ ------------------------------------------------
Boot FSPart                    98784247918
Creation time                  Mon, 05 March 2015 09:15:51 CDT
Enable SOL                     yes
FSPart                         98784247918
Kernel modules                 <34 in submode>
Kernel parameters              rdloaddriver=scsi_dh_rdac, pci=bfsort
Kernel version                 2.6.32-358.6.1.el6.x86_64
Locked                         no
Name                           TAS-XX-2.0.1-201510011105-rev1-dm
Notes                          <19 bytes>
Path                           /cm/images/TAS-XX-2.0.1-201510011105-rev1-dm
Revision
SOL Flow Control               no
SOL Port                       ttyS1
SOL Speed                      115200
```

Show TAS Software Image Kernel Parameters

```
[tas-cims1->softwareimage[TAS-XX-2.0.1-201510011105-rev1-dm]]% get kernelparameters
rdloaddriver=scsi_dh_rdac, pci=bfsort%
```

5.4.5 Clone a Service Node Software Image

About this task

Always clone software images to a test image and make modifications or install updates on the test software image. This method avoids corrupting production software images, initial service node software images provided with the system, or software image updates created by update software,

This procedure clones a service node software image (`TAS-XX-2.0.1-201510011105-dm`) to a new software image named `TAS-XX-2.0.1-201510011105-dm-test`.

Procedure

1. Log into the CIMS as `root` and run the `cmsh` command.

```
remote% ssh root@esms1
cims1# cmsh
[cims1]%
```

2. Enter `softwareimage` mode and list the available images.

```
tas-cims1# cmsh
[tas-cims1]% softwareimage
[tas-cims1->softwareimage]% list
Name (key)                          Path                                          Kernel version
----------------------------------  --------------------------------------------  ----------------------
 ESF-XX-2.2.0-201510252041           /cm/images/ESF-XX-2.2.0-201510252041           2.6.32-358.6.1.el6.x86_64
TAS-XX-2.0.1-201510011105-dm        /cm/images/TAS-XX-2.0.1-201510011105-dm       2.6.32-358.6.1.el6.x86_64
TAS-XX-2.0.1-201510011105-mdc       /cm/images/TAS-XX-2.0.1-201510011105-mdc      2.6.32-358.6.1.el6.x86_64
TAS-XX-2.0.1-201510011105-rev1-dm   /cm/images/TAS-XX-2.0.1-201510011105-rev1-dm  2.6.32-358.6.1.el6.x86_64
TAS-XX-2.0.1-201510011105-rev1-mdc  /cm/images/TAS-XX-2.0.1-201510011105-rev1-mdc 2.6.32-358.6.1.el6.x86_64
 default-image                       /cm/images/default-image                       3.0.80-0.5-default
```

3. Create a new software image named `TAS-XX-2.0.1-201510011105-dm-test` by cloning `TAS-XX-2.0.1-201510011105-dm`.

   IMPORTANT: The time to clone a software image using Bright depends on the image size. Cloning a minimal image (operating system only) completes in 5 to 10 minutes. A fully configured image can take longer. The Bright clone operation spawns a background process that does not prevent you from rebooting a node or performing other configuration changes to software image before the image is fully cloned. Cray recommends that you copy the software image from the Linux prompt in the `/cm/images` directory on the CIMS node to a new image name, wait for the prompt to return, then clone the image using Bright. Because the new image structure already exists, the clone operation in Bright occurs instantly (only the image attributes in the database are cloned).

   Exit `cmsh` and copy `TAS-XX-2.0.1-201510011105-dm` to `TAS-XX-2.0.1-201510011105-dm-test` from the Linux prompt.

```
[cims1->softwareimage]% quit
cims1# cp -pr /cm/images/TAS-XX-2.0.1-201510011105-dm /cm/images/TAS-XX-2.0.1-201510011105-dm-test
cims1# cmsh
cims1% softwareimage
[cims1->softwareimage]% clone TAS-XX-2.0.1-201510011105-dm TAS-XX-2.0.1-201510011105-dm-test
[cims1->softwareimage*[TAS-XX-2.0.1-201510011105-dm-test*]]% commit
[cims1->softwareimage[TAS-XX-2.0.1-201510011105-dm-test]]%
```

4. Display all the software images.

```
[tas-cims1->softwareimage[TAS-XX-2.0.1-201510011105-dm-test]]% list
Name (key)                          Path                                          Kernel version
----------------------------------  --------------------------------------------  -------------------------
 ESF-XX-2.1.0-201310252041           /cm/images/ESF-XX-2.1.0-201310252041           2.6.32-358.6.1.el6.x86_64
TAS-XX-2.0.1-201510011105-dm        /cm/images/TAS-XX-2.0.1-201510011105-dm       2.6.32-504.3.3.el6.x86_64
TAS-XX-2.0.1-201510011105-dm-test   /cm/images/TAS-XX-2.0.1-201510011105-dm-test  2.6.32-504.3.3.el6.x86_64
TAS-XX-2.0.1-201510011105-mdc       /cm/images/TAS-XX-2.0.1-201510011105-mdc      2.6.32-504.3.3.el6.x86_64
TAS-XX-2.0.1-201510011105-rev1-dm   /cm/images/TAS-XX-2.0.1-201510011105-rev1-dm  2.6.32-504.3.3.el6.x86_64
TAS-XX-2.0.1-201510011105-rev1-mdc  /cm/images/TAS-XX-2.0.1-201510011105-rev1-mdc 2.6.32-504.3.3.el6.x86_64
 default-image                       /cm/images/default-image                       3.0.80-0.5-default
```

5. Exit `cmsh`.

```
[cims1->softwareimage[TAS-XX-2.0.1-201510011105-dm]]% quit
cims1#
```

5.4.6 Enable Boot Record in Software Image

Prerequisites

The service node must be fully configured in Bright.

About this task

Enable the boot record setting for software images so that service nodes can boot from their local hard drive if the CIMS node is not available.

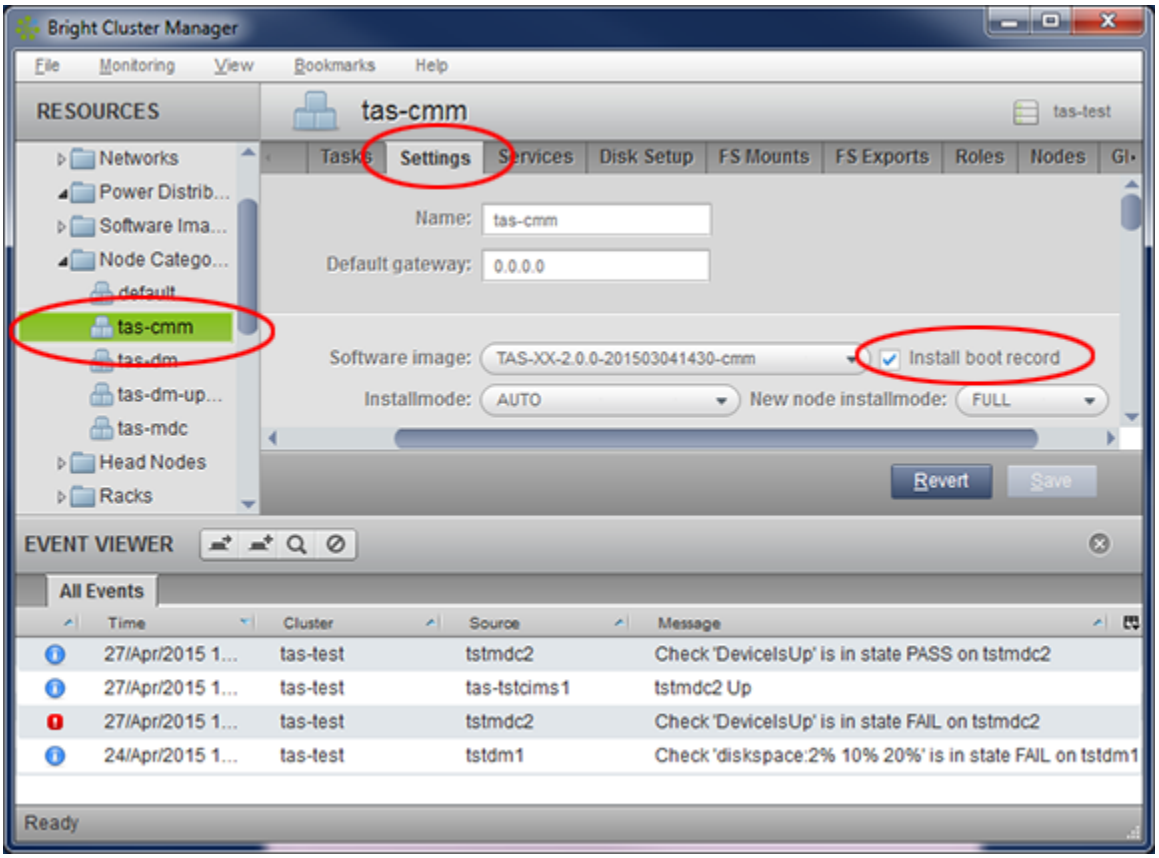
Set service node software images configured in Bright so that the node-installer installs a boot record on the local drive. This enables service nodes to reboot without access to the CIMS node.

Set the `installbootrecord` property for the service node settings and service node category to `on` so that the node can boot from the local drive. Booting from the hard drive must be set to a higher priority than network (PXE) booting in the BIOS settings for the node. Otherwise PXE booting occurs, despite the value set for `installbootrecord`.

Procedure

1. Use`cmgui` to select the service node category or device in the **RESOURCES** pane.
2. Select the **Settings** tab for the category or device.
3. Set the `Install boot record` checkbox and save the setting in the service node device or category as shown in the figure.
4. Unplug the cable for the `esmaint-net` network (the provisioning network or BOOTIF network) on the service node and reboot the node to determine if it can boot successfully without access to the CIMS node.

Figure 12. Install Boot Record



Using `cmsh`:

5. Log in to the CIMS node as `root`.

6. Enable the `installbootrecord` setting for either the device, or category:
   - To set the `installbootrecord` setting for a single service node, enter:

     ```
     cims1 # cmsh -c "device use device_name; set installbootrecord yes; commit"
     ```

   - To set the `installbootrecord` setting for a service node category, enter:

     ```
     cims1 # cmsh -c "category use category_name; set installbootrecord yes; commit"
     ```

7. Unplug the cable for the `esmaint-net` network (the provisioning network or BOOTIF network) on the service node and reboot the node to determine if it can boot successfully without access to the CIMS node.

5.5 Configure TAS Finalize Scripts to Provision Node-specific Files

Configure TAS finalize scripts in Bright to set up a node. Provision node-specific files before the node switches to local root.

The only way to setup node-specific files using the Bright node-installer prior to the node switching to local root, is through a finalize script mechanism for either the Bright service node category or device setting. The finalize script is a mechanism to perform node specific configuration actions for the node installer. From `cmsh`, enter

```
cims1# cmsh
[cims1]% category
[cims1->category]% get default finalizescript
#!/bin/bash
#
# This is the example finalize script. The node-installer runs this script when
# provisioning is complete, but before switching to local hard drive. The local
# hard drive is mounted under /localdisk. Node specific customizations can be
# made from this script. Some extra environment variables can be used by the
# script. All variable names are prefixed with CMD_. Below is a list with some
# example values:
#
# CMD_ACTIVE_MASTER_IP=10.141.255.254
# CMD_CATEGORY=idefault
.
.
.
# Some data, like interfaces, fsmounts and fsexports are stored in multiple
# variables. The code example below explains how they can be used:
#
# echo "These are the interfaces:" > /localdisk/env
# for interface in $CMD_INTERFACES
# do
#   eval type=\$CMD_INTERFACE_${interface}_TYPE
#   eval ip=\$CMD_INTERFACE_${interface}_IP
#   eval mask=\$CMD_INTERFACE_${interface}_NETMASK
#
#   echo "$interface type=$type" >> /localdisk/env
#   echo "$interface ip=$ip" >> /localdisk/env
#   echo "$interface netmask=$mask" >> /localdisk/env
# done
```

To do this at the node-installer level, copy the files into the image storage area. Use the hostname variable ($CMD_HOSTNAME) in the finalize script to copy the files from the area they are stored in (inside the image) to the proper location for the system. This is done so that it can reference them when it boots.

Another method would be to use the node-data area and the NFS mounts. This requires that the files be placed in `/cm/shared/node-data` and that `chkconfig on` the necessary services are `chkconfig on` in the node-finalize service (`/etc/init.d/node-finalize`). This service is configured to be the last service started in the `rc` start-up process. Manage the permissions on the files per site policy. Either option should be run at the root level. This enables the permissions on the files to allow only root access.

5.6 Bright Exclude Lists

Exclude lists are configured for a node category such as, `esLogin-XC`, `esLogin-XE`, `esfs-odd-`*`filesystem`*, `esfs-even-`*`filesystem`*, or `esfs-failed-`*`filesystem`* categories. TAS systems use separate exclude lists for `tas-mdc` and `tas-dm` categories.

Software images are synchronized by either pushing files from software image on the CIMS node to the service node, or pulling files from the service node to the software image on the CIMS node.

Three exclude lists control which files are pushed from the CIMS node to the service node. These are: `excludelistfullinstall`, `excludelistsyncinstall`, and `excludelistupdate`. These lists contain files that are not pushed to the service node during software image installation.

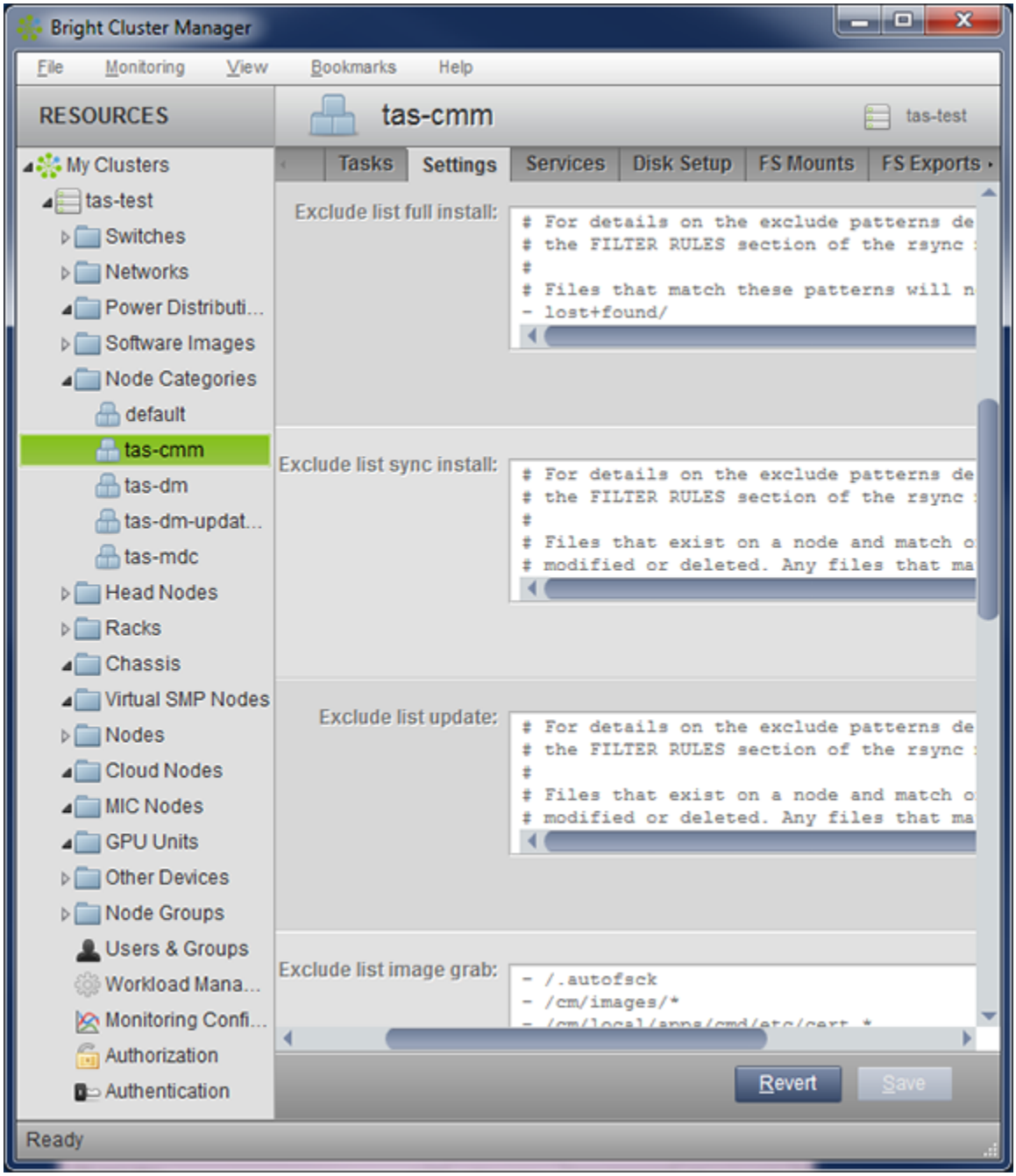
Two exclude lists control how files are pulled from the service node to the software image on the CIMS node, these are `excludelistgrab`, `excludelistgrabnew`.

> **IMPORTANT:** Cray recommends administrators use the `chroot` environment to edit software images on the CIMS node, then update the service node with the modified image. Grabbing a service node image back to the CIMS node can inadvertently store user file systems or other unwanted files in the software image on the CIMS if the Bright exclude lists are not configured properly for the service node category.

Files created or modified by a finalize script must be listed in the `excludelistupdate` exclude list for the category. Software updates will overwrite customized files if the files are not specified in an exclude list for the category. Customized files must also be specified in the `excludelistgrab`, and `excludelistgrabnew` exclude lists to prevent customized files from being copied to the CIMS node. Be sure to configure the `excludelistgrab` and `excludelistgrabnew` exclude lists to exclude all network file systems such as NFS®, Lustre®, or GPFS™ file systems.

The figure shows the exclude lists configured for a node category under the `cmgui` **Node Category->Settings** tab.

Figure 13. Set Up Bright Exclude Lists



Refer to the *Bright Cluster Manager 6.1 Administrator Manual* PDF file in the `/root` directory on CIMS node for more information.

5.6.1 Set Up Exclude Lists

Exclude lists are configured for the Bright service node categories. Software images are synchronized by either pushing files from software image on the CIMS node to the service node, or pulling files from the service node to the software image on the CIMS node.

Three exclude lists control which files are pushed from the CIMS node to the service node. These are: `excludelistfullinstall`, `excludelistsyncinstall`, and `excludelistupdate`. These lists contain files that are **not** pushed to the service node during software image installation.

Two exclude lists control how files are pulled from the service node to the software image on the CIMS which are `exludelistgrab`, `excludelistgrabnew`.

Files created or modified by a finalize script must be listed in the `excludelistupdate` exclude list for the category. Software updates will overwrite customized files if the files are not specified in an exclude list for the category. Customized files must also be specified in the `excludelistgrab`, and `excludelistgrabnew` exclude lists to prevent customized files from being copied to the CIMS. Be sure to configure the `excludelistgrab` and `excludelistgrabnew` exclude lists to exclude all network file systems such as NFS™, Lustre®, or GPFS™ file systems.

If user home directories in `/home/users` are mounted from an NFS server and also mount a Lustre® file system in `/lus/scratch`, then add `/home/users/*` and `/lus/scratch` to the `excludelistgrab` and `excludelistgrabnew` exclude lists. This prevents the `cmsh grabimage` command from copying all of the files from a remote file server to a software image on the CIMS node and potentially fill up the `/cm/images` file system on the CIMS node.

5.6.2 Exclude List Defaults

Default exclude lists are configured for the service node categories in Bright when software is installed. During an `imageupdate` command, the synchronization process uses the `excludelistupdate` list, which is a list of files and directories. One of the cross checking actions that may run during the synchronization is that the items on the list are excluded when copying parts of the file system from a known good software image to the node.

> **NOTE:** In release TAS-XX-2.0.1 the `/tas_admin/VSM/etc` directory stores the system configuration files and is bind mounted to `/etc/opt/vsm` on the active MDC node.

Exclude List	Purpose
`excludelistfullinstall`	Exclude list for full install
`excludelistgrab`	Exclude list for grabbing to an existing image
`excludelistgrabnew`	Exclude list for grabbing to a new image
`excludelistsyncinstall`	Exclude list for sync install
`excludelistupdate`	Exclude list for update

Make sure the string `no-new-files:` is prepended to each entry in the `excludelistsyncinstall` and `excludelistupdate`.

Descriptions of the default Bright exclude lists follow. Each list below indicates whether the exclude list is used to push files from the CIMS node to service nodes, or pull files from the service node to the CIMS node software image.

To display the exclude lists, use Bright `cmsh category` mode from the CIMS node:

> **TIP:** Pressing the tab key in `cmsh` mode displays a list of available arguments.

```
cims1# cmsh
[cims1]% category
[cims1->category]]% use tas-mdc
[cims1->category[tas-mdc]]% get excludelistfullinstall
[cims1->category[tas-mdc]]% get excludelistgrab
```

Use the `cmsh category` mode `set excludelistgrab` command open a `vim` editor, and enables you to edit the exclude list. Commit your changes using the `commit` command before existing `cmsh`.

`excludelistfullinstall` — Push

When the full software image installation occurs at boot time, all files from the software image in the CIMS are pushed to the service node unless they are included in the `excludelistfullinstall` exclude list. Files that match these patterns are not installed onto the service node.

```
# For details on the exclude patterns defined here please refer to
# the FILTER RULES section of the rsync man page.
#
# Files that match these patterns will not be installed onto the node.
- lost+found/
- /proc/*
- /sys/*
no-new-files: - /cgroup/*
no-new-files: - /ipathfs/*
no-new-files: - /etc/ssh/ssh_host_*
```

`excludelistsyncinstall` — Push

When software image is synchronized at boot time, all files from the software image on the CIMS node are pushed to the service node unless they are entered in the `excludelistsyncinstall` exclude list. Any files that match one of these patterns and that exist in the image but are absent on the node, are copied to the service node. Files that exist on a service node and match one of these default patterns are not modified or deleted. If necessary, add the string `no-new-files:` to every line in `excludelistsyncinstall` to exclude files that are needed for a full install but are troublesome for an `imageupdate`.

The `/tas_admin/*`, `/vsm/*`, `/etc/opt/vsm/*`, `/var/opt/vsm/*`, and `/var/lib/pacemaker` entries are included to preserve the VSM license information and system configuration information.

```
# For details on the exclude patterns defined here please refer to
# the FILTER RULES section of the rsync man page.
#
# Files that exist on a node and match one of these patterns will not be
# modified or deleted. Any files that match one of these patterns and that
# exist in the image but are absent on the node, will be copied to the node.
- /.autofsck
- /cm/local/apps/pbspro/*/spool/*
- /cm/local/apps/sge/*/spool/*
.
.
.
- /tas_admin/*
- /vsm/*
- /etc/opt/vsm/*
- /var/opt/vsm/*
- /var/lib/pacemaker
.
.
.
```

`excludelistupdate` — Push

If the node is already booted, running the `cmsh imageupdate` command pushes all files from the service node software image on the CIMS node to the software image on the running node, except those entered on the `excludelistupdate` exclude list. The `exludelistupdate` list is in the form of two sublists. Both sublists are lists of paths, except that the second sublist is prefixed with the string `no new files:`. When a node is updated, all of its files are examined during `imageupdate` synchronization. The logic used is as follows: Files that exist on a node and match one

of the patterns below will not be modified or deleted. Any files that match one of these default patterns and that exist in the image but are absent on the node are copied to the node.

If an excluded path from `excludelistupdate` exists on the node, then no files from that path are copied over from the software image to the node. If an excluded path from `excludelistupdate` does not exist on the node, then:

- If the path is on the first, non-prefixed list, then the path is copied over from the software image to the node.
- If the path is on the second, prefixed list, then the path is not copied over from the software image to the node. That is, no new files are copied over, like the prefix text implies.

To work around this logic, prepend `no-new-files:` to each line. This prevents new files and paths being created on the nodes. If necessary, add `no-new-files:` to every line in `excludelistupdate` to exclude files that are needed for a full install but are troublesome for a `imageupdate`.
The `/tas_admin/*`, `/vsm/*`, `/etc/opt/vsm/*`, `/var/opt/vsm/*`, and `/var/lib/pacemaker` entries are included to preserve the VSM license information and system configuration information.

```
- /.autofsck
- /.autorelabel
- /boot/grub/device.map
- /boot/grub/grub.conf
- /boot/grub/menu.lst
.
.
.
- /var/spool/*
- /var/tmp/*
- /var/crash/*
- /tas_admin/*
- /vsm/*
- /etc/opt/vsm/*
- /var/opt/vsm/*
- /var/lib/pacemaker
.
.
.
no-new-files: - /var/run/*
no-new-files: - /cgroup/*
no-new-files: - /ipathfs/*
no-new-files: - /etc/ssh/ssh_host_*
```

`excludelistgrabnew` — Pull

Run the `cmsh grabimage -n newimage` command to synchronize files from the service node to a new software image on the CIMS node, unless the files or directories are entered in the `excludelistgrabnew` exclude list. The default list follows.

The `/var/lib/pacemaker` is included in the `excludelistsyncinstall` exclude list to retain the pacemaker cluster information base (CIB) configuration so that node states (online/standby/maintenance) are preserved if both MDC nodes are rebooted. This is not an issue when a single MDC node is rebooted while the other node is running, because the configuration is restored from the other node via Corosync Cluster Engine.
The `/tas_admin/*`, `/vsm/*`, `/etc/opt/vsm/*`, `/var/opt/vsm/*`, and `/var/lib/pacemaker` entries are included to preserve the VSM license information and system configuration information.

```
- /.autofsck
- /cm/images/*
- /cm/local/apps/cmd/etc/cert.*
- /cm/local/apps/pbspro/*/spool/*
.
.
.
- /tas_admin/*
- /vsm/*
- /etc/opt/vsm/*
- /var/opt/vsm/*

no-new-files: - /cgroup/*
no-new-files: - /ipathfs/*
no-new-files: - /etc/ssh/ssh_host_*
```

5.6.3 Check Exclude Lists

About this task

Each of the exclude lists has specific comments about the preconfigured exclusions.

Procedure

1. Log in to the CIMS node as `root`.
2. Use the following command to check the exclude list settings for the `excludelistfullinstall` exclude list. Substitute the appropriate node category and exclude list name in the example below.

   ```
   cims# cmsh -c "category use tas-mdc; get excludelistfullinstall"
   # For details on the exclude patterns defined here please refer to
   # the FILTER RULES section of the rsync man page.
   #
   # Files that match these patterns will not be installed onto the node.
   - lost+found/
   - /proc/*
   - /sys/*
   ```

   Using `cmgui`, select a node category from the Bright **RESOURCE** tree, then select the **Settings** tab to check exclude lists.

5.6.4 Change an Exclude List

About this task

Change Bright exclude lists to contol what files are grabbed from a running node to the software image, or what files are updated on the node during a reboot or synchronize.

Procedure

1. Log in to the CIMS node as `root`.
2. The following command opens a text editor (`vim`) that modifies the `excludelistfullinstall` exlude list for the Bright service node category `tas-mdc`. Substitute the appropriate service node category and exclude list name in the example.

   ```
   cims# cmsh -c "category use tas-mdc; set excludelistfullinstall; commit"
   ```

5.7 Configure kdump on TAS Nodes

Procedure

1. Log in to the CIMS as `root`.

2. To configure kdump for TAS service nodes, clone the service node software image to a test image. This example clones `TAS-XX-2.0.1-201510011105-dm` to `TAS-XX-2.0.1-201510011105-dm-kdump`.

```
cims# cd /cm/images
cims# cp -pr  TAS-XX-2.0.1-201510011105-dm TAS-XX-2.0.1-201510011105-dm-kdump
cims# cmsh
cims%] softwareimage
[cims->softwareimage]% clone  TAS-XX-2.0.1-201510011105-dm TAS-XX-2.0.1-201510011105-dm-kdump
```

3. Commit your changes.

```
[cims->softwareimage*[TAS-XX-2.0.1-201510011105-dm-kdump*]]% commit
[cims->softwareimage[TAS-XX-2.0.1-201510011105-dm-kdump]]%
```

4. Clone the existing DM node category to a test category to configure kdump.

```
[cims->softwareimage[TAS-XX-2.0.1-201510011105-dm-kdump]]% category
[cims->category]% clone tas-dm tas-dm-test
[cims->category*[tas-dm-test*]]% commit
```

5. Use the `tas-dm-test` category.

```
[cims->category]% use tas-dm-test
[cims->category*[tas-dm-test*]%
```

6. Assign the kdump software image (`TAS-XX-2.0.1-201510011105-dm-kdump`) to the `tas-dm-test` category.

```
[cims->category*[tas-dm-test*]% set softwareimage TAS-XX-2.0.1-201510011105-dm-kdump
```

7. If needed, add `/var/crash` to the exclude lists for the `TAS-XX-2.0.1-201510011105-dm-kdump` image category. The `vim` editor launches and enables the exclude list file to be edited.

```
[cims->category*[tas-dm-test*]% set excludelistsyncinstall
[cims->category*[tas-dm-test*]% set excludelistupdate
[cims->category*[tas-dm-test*]% set excludelistgrab
[cims->category*[tas-dm-test*]% set excludelistgrabnew
```

8. Save each file and commit your changes.

```
[tas-cims1->category*[tas-dm-test*]%  commit
[tas-cims1->category[tas-dm-test]%
```

9. Assign the `tas-dm-test` category to a DM node (`tas-dm1`) and commit your changes.

```
[cims->category[tas-dm-test]]% device use tas-dm1
[cims->device[tas-dm1]]% set category tas-dm-test
[cims->device*[tas-dm1*]]% commit
[cims->device[tas-dm1]]%
```

10. If you are saving kdump crash files to the service node local disk, add the following lines to the finalize script for the `tas-dm-test` category. This command opens the `vim` editor. Scroll down and add the lines before `exit 0`.

```
[cims->device[tas-dm1]]% category use tas-dm-test
[cims->category[tas-dm-test]]% set finalizescript
DEV=$( awk -- '{ if ($2 == "/localdisk/var/crash") { print $1; exit 0 } }' < /proc/mounts )
[ -n "$DEV" ] && e2label $DEV crash
```

11. Commit your changes.

```
[cims->category*[tas-dm-test*]]% commit
```

12. Set the storage location for crash dumps. To save crash dumps to the CIMS node, proceed to *13a*. To save crash dumps to the service node local disk, proceed to *13b*.

13. To save crash files to the `/var/crash` directory on the CIMS node:

a. Use `fsexports` to determine whether the CIMS is exporting `/var/crash`.

```
[cims->category[tas-dm-test]]% device use tas-cims1
[cims->device[tas-cims1]]% fsexports
[cims->device[tas-cims1]->fsexports]% list
Name (key)                                      Path                              Hosts
----------------------------------------------  --------------------------------  --------------------------
/cm/shared@esmaint-net                          /cm/shared                        esmaint-net (10.141.0.0/16)
/home@esmaint-net                               /home                             esmaint-net (10.141.0.0/16)
/var/spool/burn@esmaint-net                     /var/spool/burn                   esmaint-net (10.141.0.0/16)
/cm/node-installer/certificates@esmaint-net     /cm/node-installer/certificates   esmaint-net (10.141.0.0/16)
/cm/node-installer@esmaint-net                  /cm/node-installer                esmaint-net (10.141.0.0/16)
```

b. If `/var/crash` is not exported from the CIMS, then configure and export it to service nodes.

```
[[cims->device[cims]->fsexports] add /var/crash
[cims->device*[cims*]->fsexports*[/var/crash*]]% set name /var/crash@esmaint-net
[cims->device*[cims*]->fsexports*[/var/crash*]]% set extraoptions no_subtree_check
[cims->device*[cims*]->fsexports*[/var/crash*]]% set hosts esmaint-net
[cims->device[cims]*]->fsexports*[/var/crash*]]% set write yes
[cims->device*[cims*]->fsexports*[/var/crash*]]% commit
```

c. Exit `/var/crash` submode.

```
[cims->device[cims]->fsexports[/var/crash]]% exit
[cims>device[cims]->fsexports]%
```

d. Verify that the CIMS is exporting `/var/crash`.

```
[cims>device[tas-cims1]->fsexports]% list
Name (key)                                      Path                              Hosts
----------------------------------------------  --------------------------------  --------------------------------
/cm/shared@esmaint-net                          /cm/shared                        esmaint-net (10.141.0.0/16)
/home@esmaint-net                               /home                             esmaint-net (10.141.0.0/16)
/var/spool/burn@esmaint-net                     /var/spool/burn                   esmaint-net (10.141.0.0/16)
/cm/node-installer/certificates@esmaint-net     /cm/node-installer/certificates   esmaint-net (10.141.0.0/16)
/cm/node-installer@esmaint-net                  /cm/node-installer                esmaint-net (10.141.0.0/16)
/var/crash@esmaint-net                          /var/crash                        esmaint-net (10.141.0.0/16)
```

e. Exit `cmsh`.

f. Update the exports.

```
cims# exportfs -a
```

14. Use the `chroot` shell to edit the `/boot/pxelinux.cfg/default` file in kdump test image (`TAS-XX-2.0.1-201510011105-dm-kdump`).

a. Use `chroot` to edit the `/boot/pxelinux.cfg/default` file.

```
cims# chroot /cm/images/ TAS-XX-2.0.1-201510011105-dm-kdump
cims:/> vi /boot/pxelinux.cfg/default
```

b. Scroll down and locate the following line:

```
# End of documentation, configuration follows:
```

c. Enter the following lines in the `default` configuration file:

```
LABEL kdump
KERNEL vmlinuz
IPAPPEND 3
APPEND initrd=initrd crashkernel=512M CMDS console=tty0 console=ttyS1,115200n8 CMDE
MENU LABEL ^KDUMP      - Normal boot mode with kdump
MENU DEFAULT
```

d. Examine the other `LABEL` entries in the `default` configuration file and remove the line: `MENU DEFAULT`.

e. Exit and save the file.

15. Edit the `/etc/kdump.conf` file and add or modify the following lines:

```
cims:/> vi /etc/kdump.conf
```

a. Add the following lines at the end of the `/etc/kdump.conf` file.

```
path /var/crash
core_collector makedumpfile -c --message-level 1 -d 27
link_delay 60
default reboot
```

If you want to save crash dump files to the local add this line to the `kdump.conf` file. If the file system type is `ext4`, then replace `ext3` with `ext4`.

```
ext3 LABEL=crash
```

Create a persistent partition (`/var/crash`) in the disk setup XML file for the kdump test category (`tas-dm-test`). Creating a separate partition for crash dumps on the service node software image prevents `/var` from filling up and causing problems for the operating system.

b. Exit and save the file.

16. Enable the `kdump` service.

```
cims:/> chkconfig kdump on
```

17. Exit the `chroot` shell.

```
cims:/> exit
cims#
```

18. Reboot the `tas-dm1` test node and run kdump.

a. Start a console window on the test service node (`tas-dm1`).

```
cims# cmsh
[cims]% device; use tas-dm1
[cims->device[tas-dm1]]% rconsole
```

b. Reboot the test node (`tas-dm1`).

c. When the node reboots, initiate kdump.

```
cims#  ssh tas-dm1
tas-dm1# echo c > /proc/sysrq-trigger
```

If dumping over NFS to the CIMS, the dump file is created in `/var/crash` on the CIMS node. If dumping to the service node's local disk, the dump file is created in `/var/crash` on the service node's local disk.

19. Assign the kdump software image to the production TAS DM node category (`tas-dm`). Switch to `category` mode and configure the production TAS DM node category to use the kdump software image.

```
[cims->device[tas-dm1]]% category
[cims->category]% use tas-dm
[cims->category[tas-dm]]% set softwareimage  TAS-XX-2.0.1-201510011105-dm-kdump
```

20. Reboot all of the nodes in the `tas-dm` category, so that they use the kdump software image.

```
[cims->category[tas-dm]]% device
[cims->device]% reboot -c tas-dm
tas-dm1: Reboot in progress ...
```

21. Exit `cmsh`.

```
[cims->device]% quit
```

6 Manage the TAS HSM

Cray TAS HSM systems are managed from a CIMS node.

CIMS Node

CIMS node, is the head node for the system. It runs the SLES11SP3 operating system and includes Bright Cluster Manager 6.1 management software (Bright). All service nodes, metadata controller (MDC), and data mover (DM) nodes, PXE boot from the CIMS node.

Each service node is assigned to a node category in Bright. It boots a specific software image for that category. TAS node categories in Bright are typically `tas-dm` and `tas-mdc`. Cray customizes each software image for both TAS MDC and DM nodes.

High Availability (HA)

Each MDC node is in an HA configuration implemented using PaceMaker and Corosync software.

Service Nodes

Service nodes run the CentOS 6.5 software. MDC nodes run Versity Storage Manager (VSM) software. This provides the high-performance tape archiving file system.

6.1 Configure the Archiver

Disk archives or tape archives in a robotic tape library are used for data storage in the Cray TAS system. The archiver automatically makes copies of files on the online disk cache and places them onto the archive media. Configure the `/etc/opt/vsm/archiver.cmd` file to specify:

- Which kinds of files are archived
- When they are archived
- How many copies of each file to make and place on a variety of archive media

The archiver can archive at least four archive copies at a time.

> **NOTE:** With release TAS-XX-2.0.1, all of the VSM configuration files (which includes the `archiver.cmd` and `mcf` files) are now stored in a single location in the shared TAS administrative file system. They are under the `/tas_admin/VSM/etc` directory and bind mounted to `/etc/opt/vsm` on the active MDC node. This simplifies administration and eliminates the need to synchronize the configuration files on an active MDC node back to the MDC software image in Bright, after modifications. This also eliminates the need to `chroot` into the MDC software images on the CIMS node, then update the running MDC images using Bright.

The archiving component of VSM copies one file from the file system to another archive media—either a volume in a library or another file system. The `archiver.cmd` file may be configured to specify which files are to be archived immediately, at a later time, or never archived at all. Every file within the file system becomes a member to only one archive set. File characteristics determine the archive set in which it is placed. Archive files are written in the open-source GNU tar format.

The `archiver.cmd` is the archiver command file. It is critical that this file has no errors. If there are errors, the archiver will not run. Use the `archiver -lv` command to check the file for errors.

Archive Reserve Set

The VSM reserve set feature allows a single volume pool so that when an archive set uses a tape, it is used only by that archive set in the future. This enables a single scratch pool be used for all archive sets which makes administering tapes much easier.

The following is an example of the `-reserve set` in the `archiver.cmd` example file.

```
############################
### Archive set section ###
############################
# startage -
# startsize -
# startcount -
# drives -
# drivemin -
# (see man archiver.cmd for more information)
params
allsets -reserve set -sort path -offline_copy stageahead

allsets.1 -startage 2h -startsize 250G -startcount 1000 -drives 3
allsets.2 -startage 2h -startsize 250G -startcount 1000 -drives 3

tas_admin.1 -startage 1h -startsize 1G -startcount 1000 -drives 1
tas_admin.2 -startage 1h -startsize 1G -startcount 1000 -drives 1
endparams
```

An archive set simply pulls a volume from the pool as needed and that volume is then used strictly with that archive set. With TAS-XX-2.0.1, additional volumes can now be easily added to a pool to simplify volume management by reserving a volume to be used exclusively with an archive set. This simplifies volume configuration, because there is no need to distribute volumes between all the archive sets.

```
############################
###     VSN pools      ###
############################
# Define scratch pool
vsnpools
scratch li 09305[0-9] 09375[0-9] 09382[0-9] 0000[1-2][0-9] 19243[0-9]
endvsnpools

############################
###     VSN section     ###
############################
vsns
allsets.1       li      -pool scratch
allsets.2       li      -pool scratch
tas_admin.1     li      -pool scratch
tas_admin.2     li      -pool scratch
endvsns
```

Configure the Archiver

The following is a guide to configure the `/etc/opt/vsm/archiver.cmd` for optimal performance.

Carefully consider how many file systems users will be accessing concurrently. Using multiple file systems can increase the performance of the archiver. Compared to a single file system, scan time is reduced when using multiple file systems.

Archive file logs are used to recover data—either in the event of a catastrophic failure or when the VSM file system is unavailable. Save the archive logs to a secure location. Make sure data can be recovered in the event of a failure.

The archive media contains one or more tar files. Each tar file contains different file data. Each tar file is assigned to a volume serial name (VSN) or disk volume. Use this VSN or disk volume to identify the different tar files on the archive media.

A file is eligible to be archived after the data inside the file has been altered and closed. The archive age is the time period that has lapsed since the file was modified last. This archive age can be predefined for each archive copy.

The archiver completes four steps for each file it is prepared to copy.

1. Identify each file to be archived
2. Create the archive request
3. Schedule the request for each file
4. Archive each file listed in the request

The first step the archiver performs is to identify each file to be archived. There is one `sam-arfind` process per mounted file system. The process `sam-arfind` monitors that mounted file system to identify and determine which files need to be archived. This is the process that also determines when a file needs to be re-archived due to a change in data. Accessing the file does not cause re-archiving of the file. The file must be modified in some way to start the re-archiving process. The `sam-arfind` process also determines which archive set the file will belong to by scanning:

- The file directory path
- Minimum file size
- Maximum file size
- The user/group name of the owner of the file

If the age of the archive for the file and its copies has been exceeded, the archiver moves to step two—adding the file to the archive request set based on the file classification. There can be many different archive request sets. This allows administrators to schedule the archiver independently for many different classifications. If the age of the file has not been exceeded, then the file is added to the scan list along with the time when the age will be met. The scan list continuously runs. When a file has reached its specified age, it will then be added to the archive request set.

A file may be offline at the time the archiver tries to access it. If this happens, the `sam-arfind` process selects a volume(s) that should be used as the archive copy source. The `sam-arcopy` process then requests a stage of the file back to the disk cache in order to make another archive copy.

The archiver uses two methods to complete its actions—continuous archiving and scanned archiving. Continuous archiving works with the file system to determine which files to be archived and which files do not need to be archived. Continuous archiving is set by default. To change this, edit the `archiver.cmd` file by changing the following line from `examine=noscan` to `examine=scan`. When the default continuous method is initiated, it begins with the following default start conditions.

- The archiver runs every two hours (2:00 HRS)

- The archiver not run until at least 90% of the `archmax` value of data is modified and ready for archiving
- The archiver waits for 500,000 files to be ready until it archives

  These are all scheduling conditions that need to be met before the archiver starts; when any one of them is met, the archiver begins. Start conditions may be set for each archive set which override the defaults above by using the command parameters below. These parameters allow administrators to take full advantage of the archiver by configuring the timeliness, in relation to the work load that must be performed by the archiver.

  `-startage`
  `-startcount`
  `-startsize`

  Scanned archiving is used to check the file system periodically in order to select files to be archived. At midnight, the `sam-arfind` process systematically checks all the files to determine which need to be archived.

  The second step the archiver performs is to create an archive request set. An archive request set is a group of files that all belong to one archive set. The outstanding archive-request log is located at `/var/opt/vsm/archiver/`*`filesystem`*`/ArchReq`. The `ArchReq` file is a binary file. Use the `showqueue` command to display the archive requests in a readable format.

```
tasmdc1# showqueue
Filesystem tasfs1:
Files waiting to start:          0
Files being scheduled:          50
Files archiving:                 0
Events processed:              213
    archive             0
    change              1
    close              86
    create             57
    hwm                 0
    modify              0
    rearchive           0
    rename             12
    remove             57
    unarchive           0
    internal       19,787
Exam list: 0 entries

Scan list  Examine: noscan
  0 2014-08-02 00:00:00 background       ---- inodes
Archive requests
tasfs1.samfs1.1.28 schedule 2014-08-01 00:28:03 drives:3
    files:25 space: 737.843G flags:
 (min:   30.500k)    priority: 0 0
    No drives available

tasfs1.samfs1.2.29 schedule 2014-08-01 00:28:03 drives:3
    files:25 space: 737.843G flags:
 (min:   30.500k)    priority: 0 0
    No drives available
```

  The `sam-archiverd` daemon oversees the archive requests, regardless of whether scanned or continuous archiving is used. The `sam-archiverd` daemon composes the archive sets by selecting certain files from the `ArchReq`. Depending on individual parameters, not all files in the request list will be archived simultaneously. In this instance, the `sam-archiverd` will archive any remaining files in a different archive set.

  The `sam-archiverd` daemon uses specific criteria to determine which way to place files into the archive request. By default, the daemon places all files in the archive request using their full path name. That way all files and their directories are placed together on the same archive media. The daemon also uses site-specific

criteria. This delivers control over the order in which files are archived and how to distribute them across the volumes:

- `-reserve`
- `-sort`
- `-rsort` (reverse sort)
- `-drives`

When the archive set parameters are evaluated, they are done so in this order—starting at the top of the list and moving down. The `-reserve` parameter is evaluated first. When it is specified, the `sam-archiverd` daemon orders the files on a “reserved” volume in accordance with file:

- Directory path
- User name
- Group name

The `-reserve` parameter specifies that the volumes used for archiving are reserved. If users opt out of using the `-reserve` parameter, then the archive sets are mixed on the volumes assigned to the archive set.

When either the `-sort` or `-rsort` parameters are specified, the `sam-archiverd` daemon orders the files in accordance to a specified sort method such as file:

- Age
- Size
- Directory location

The files in any archive set can be sorted to keep them together according to the property associated with the method specified. If no method is specified, then the path-sorting method is used. If `-rsort` is used, the sort is performed in the reverse order of what is normal for the specified method.

The following sort methods can be used:

Age Each archive file will be sorted by ascending modification time where the oldest files are archived first.

None There will be no sorting performed for the archive file. Files will be archived in the order they are encountered on the file system.

Path Each archive file will be sorted by the full file pathname. This method keeps the files from the same directories together on the archive media.

Priority Each archive file will be sorted by descending archive priority—where the highest priority files are archived first.

Size Each archive file will be sorted by ascending file size—where the smallest files are archived first and the largest files archived last.

In the instance that no sort method has been specified, the offline files are ordered by the volume to which archive copies reside. This ensures that every file in every archive set on the same volume is staged simultaneously in the order in which they were originally stored on the archive media. If there is more than one archive copy of an offline file that is being made, the offline file will not be released until all the required copies are finished.

The third step consists of scheduling the archive requests. The `sam-archiverd` daemon controls the scheduling of the archive request. The following conditions cause the `sam-archiverd` daemon to schedule archive requests.

1. The archiving for an archive request has completed.
2. The state of the archiver changed.
3. The state of the media changed.
4. An archive request is entered into the scheduling queue.

6.1.1 Schedule Queue

The scheduling queue is ordered by priority. Each time the scheduler runs, it examines the archive requests and determine whether or not they can be assigned to a `sam-arcopy` process and have their files copied to the archive media. In order for the archive request to be scheduled, the drive usage must be available for making file copies. The volume usage must be available with enough space to hold the files in the archive request.

6.1.2 Drive/Volume Size and Selection

If the `-drives` parameter has been specified, then the selected files are divided among the multiple drives. The `-drivemin` value can be defined by either the `-drivemin` parameter or the `archmax` value. If the collective size of all the files in the archive request is lower than the `-drivemin` value, only one drive will be used. If the collective size of all the files in the archive request is larger than the `-drivemin` value, then the number of drives used will be determined by the collective size of all files divided by the `-drivemin` value.

Depending on a variety of variables, the drives can take different amounts of time to archive files. The `-drivemax` parameter specifies the maximum number of bytes that will be written to the drive before that drive is rescheduled for more files to be archived. The `-drivemax` parameter, when used correctly, optimizes drive utilization.

Archiving needs at least a single volume to exist with enough space to hold the files in the archive request. If there is not enough space for at least part of the files in the archive request, then archiving cannot occur. When the specified volume is busy, then another volume is selected. This feature can be disabled with the `-fillvsns` parameter. If the `-fillvsns` parameter has been specified, then the archive request cannot be scheduled.

If the archive request is too large to fit on one single volume, the archiver selects the files that will fit on the drive and places them there. When the files in the archive request are too large for the volume, the files are not written to it. To help prevent this, have the volume overflow selected for the archive request. Specify volume overflow for the archive set by using the `-ovflmin` parameter. Specify the volume overflow for the media by using the `ovflmin=directive`.

The `-ovflmin` parameter sets the minimum size for a file that needs more than one volume for the media. When files are small enough to fit on the volume, they will be placed accordingly. When a file is too large, however, the overflow will be written to multiple volumes. When `-ovflmin` values for both the archive set and the media are specified, the value for the archive set takes precedence.

If the size of the files is larger than the value specified in the `-ovflmin`, additional volumes will be assigned for use. The volumes will be selected in the order of decreasing size to help minimize the number of volumes required. The archive request will wait for a volume to become available. If all volumes are busy, it will archive files to the first available volume.

6.1.3 Archive Request Entered in Scheduling Queue

The `sam-archived` daemon computes the scheduling priority for each archive request. It does this by adding the archive priority to multiples that are associated with various system resource properties. The properties are associated with the number of time (in seconds) for which the archive request has been in queue. The `sam-archived` daemon uses the now adjusted priorities to assign each ready archive request to be copied.

The final task is archiving the files in the archive request set. The `sam-archived` daemon marks the boundaries for each file when the archive request is ready to be archived. These boundaries help keep each archive file sized less than the specified `-archmax` value. If a single file is larger than this value, the file becomes the only one in the archive file.

6.1.4 Archive Directives

The `archiver.cmd` file consists of two main areas—global directives and file system-specific directives. These two parts specify the general archive operations. The global directives are the top part of the `/etc/opt/vsm/archiver.cmd` file. These directives affect all the file systems defined in the `mcf` file. The bottom part of the `archiver.cmd` file consists of the file system-specific directives. These directives need to come after the global directives. The system specific directives can override the global directives for their given file system. They start with the `fs = name` directive that identifies the file system.

6.1.5 Control the Size of Archive Files Using `archmax`

The `archmax` directive is used to define the maximum size an archive file can be. User files are combined to create the archive file. Once the target-size has been reached, no more user files can be added to the archive file. User files larger than `archmax` are written to a single archive file.

To edit the default values, use the following directive:

```
archmax = media target-size
```

```
# Larger sizes provide better tape streaming I/O, but
# wastes more tape capacity
archmax = li 100G
archmax = ti 300G
```

Media is the media type the files are located on. Target-size is the maximum size desired for the archive file. The target-size is media dependent. Depending on individual file system needs, setting large or small archive file sizes will have advantages and disadvantages. As a good practice, the `archmax` directive should be set at no more than five percent of the media capacity.

6.1.6 Set Archiver Buffer Size with `bufsize`

A memory buffer is used by default to copy a file to archive media. Use the `bufsize` directive to specify a non-default buffer size. By default, a file being archived is stored in memory in a buffer of a default size for the media type before being written to archive media. Use the `bufsize` directive to specify a buffer size. A custom size can improve performance. This parameter has the following format:

```
-bufsize=buffer-size
```

```
# Lock tape buffers in memory
# default 8, not locked
bufsize = li 64 lock
```

The default buffer size is four, indicating that the actual buffer size is four multiplied by the *dev*_blksize value for the media type. Specify a number from 2 to 32. The *dev*_blksize value is specified in the `/etc/opt/vsm/defaults.conf` file.

> **NOTE:** In release TAS-XX-2.0.1 the `/tas_admin/VSM/etc` directory stores the system configuration files and is bind mounted to `/etc/opt/vsm` on the active MDC node.

6.2 Configure the Stager

The stager automatically copies the file data back to the online disk cache when a user or process requests the data. Staging is the process of copying file data that has been released back to online storage to make it available to the user. After the stager runs, it attempts to use all the drives in the library. The `stager.cmd` file can be configured to customize stager operations by inserting directives into the `/etc/opt/vsm/stager.cmd` file.

The stage buffer size is determined by the media type. It defaults to 16 * the media block size. The stage buffer is unlocked by default. The stage buffer size and lock status can be set in the `stager.cmd` file. The number of outstanding stage requests is based on memory size. It can vary from 5,000 up to 500,000. It can be set in the `stager.cmd` by setting the `maxactive` parameter.

If an offline file is required by a user or application, its archive copy is staged to the disk cache. The application read operation tracks directly behind the stager operation so that if a file is needed immediately it can be accessed before the entire file is staged. If a stage error is returned, the system switches to another copy of the file. It returns an error only after trying to stage all archived copies.

Refer to the `man` page for `stager.cmd` on the MDC node for more information about stager configuration file.

6.2.1 The `stager.cmd` File

Directives may be used in the `/etc/opt/vsm/stager.cmd` to override default behaviors. Each directive in the `stager.cmd` file must appear on its own line.

The following directives are accepted by the `stager.cmd` file:

directio = on/off	Off causes paged I/O to be used. If this directive is on, the directive sets direct I/O for all staging when the file size is equal or greater than the `dio_min_size` value. `directio` is set to `on` for shared.
dio_min_size = *n*	If the file size is less than the value of the `dio_min_size`, the stager defaults to paged I/O for non-shared VSM file systems. By default *n* = 8MBs.
drives = library *count*	Defines the number of drives that are to be used for staging on the media library. By default the count is equal to the actual number of drives in the library.
bufsize = media buffer_size [lock]	Sets the stage buffer size for specific media types. Lock the buffer if lock is set.

logfile = filename [event]	The stager log contains reports for each file that has been staged. This directive defines the file that is to be used for the log file. By default there is no log file, this directive must be used to create one.
maxactive = *number*	Uses an integer number to set the maximum number of stage requests that can be active at any given moment in the stager. The minimum integer that can be set is 5000. The maximum integer is 500,000. By default the number is based on the size of available memory, 5000 for every gigabyte.
maxretries = *number*	Limits the number of stage retries capable by the archive copy when errors arise to an integer number. The minimum integer that can be used is 0 and the maximum is 20. By default the integer is three.
fs = *family_set_name*	Specifies the following directives apply only to the defined *family_set_name* until another `fs` is specified.

6.2.2 Preview Request

Processes request that media be loaded/unloaded. When there are more requests for tapes than available drives, the remaining requests are sent to the preview queue. The `previews=value` directive in the `/etc/opt/vsm/defaults.conf` file controls the number of entries that can be in the preview queue.

The `defaults.conf` file is read when the `sam-fsd` daemon starts. Changes may be made at any time, but the changes do not take place until `sam-fsd` rereads the files and `sam-amld` is restarted.

```
tas-mdc1# samd stop
tas-mdc1# samd config
tas-mdc1# samd start
```

By default, the preview requests are completed in a first-in-first-out order (FIFO). This order can be changed in the `/etc/opt/vsm/preview.cmd` file.

The `sam-amld` daemon reads the `preview.cmd` file at startup. The order in which this file queues the requests depends on whether the request is meant for staging or archiving. Administrators have control over the priority of VSNs and can determine the priority of preview requests for specific file systems.

Only one directive can be placed per line in the preview file. Comment lines must start with a `#`. Only two types of directives can be used in the preview file—global directives and file-system directives. Global directives are placed at the top of the file and are applied to the whole file system. When more than one directive is used for a file system, the directives specific to that file system take precedence over the global directives.

File system directives start with the `fs = directive`. This names the file system to which all the following directives apply. A single file can contain multiple blocks of directives. When one directive is read, that directive continues in effect until another directive is read. Or until the end of the file.

6.2.3 Set the Global VSN and Age Directives

The VSN and age priority directives, should be placed before any file system specific directives inside the `preview.cmd` file. It is necessary to update the `vsn_priority` directive.

```
vsn_priority = value
```

This is a static priority factor directive. It indicates its value by which the total priority increases when there exists a higher-priority volume. In order for this priority factor to be used, the volume needs to have its priority flag set before it is scheduled as a preview request. Use the `chmed` command to set the priority flag with the `-p` option. By default the value is set to 1000.0.

> **IMPORTANT:** Use `chmed` to clear or set flags and values in the library catalog entry. These values are crucial to correct operation of the VSM file system. They should only be modified by administrators in particular circumstances. Be cautious when using this command. There is no way to check before hand to make sure the catalog remains consistent.
>
> After setting the `vsn_priority` directive, update the `age_priority` directive.
>
> ```
> age_priority = factor
> ```
>
> This is a static priority factor, though it acts like a dynamic priority factor. This factor is multiplied by the number of seconds for which a request is in preview-request mode. The sum of this multiplication is then added to the overall request priority. In this context, the longer a request waits the higher its priority becomes. Enabling this setting ensures that older requests are not pushed back by newer high-priority requests.
>
> When setting this factor to more than 1.0, the time factor becomes more important to the total-priority calculation. Setting this value below 1.0 decreases the time-factor importance. To completely eliminate the time factor, set this value to 0.0. If a volume does not have a priority flag, it sits in the queue—its priority increases the longer it waits there. The volume priority can keep increasing and, eventually, become higher than a VSN that has entered the queue later with the priority flag.

6.2.4 Set Global or File System Specific Water Marks

Water mark directives have the following format:

```
water-mark-type_priority = value
```

Water mark directives are described in the following table.

Table 8. Water Mark Directives

Directive	Description
`lwm_priority = value`	Defines the amount by which—when the file system is below the lower water mark (LWM) level—the water mark priority factor changes for archiving request . The default value is 0.0.
`lhwm_priority = value`	Defines the amount by which—when the file system crosses from below the low water mark to above the low water mark but remains below the high water mark—the water mark priority factor changes for archiving request. This indicates to the system that it is filling up. The default value is 0.0.
`hlwm_priority = value`	Defines the amount by which—when the file system has previously exceeded the high water mark but has now fallen back below this mark yet still remains above the low water mark—the water mark priority factor changes for archiving request. This may occur when the releaser is not able to free enough space to drop the disk space below the low water mark. The default value is 0.0.

Directive	Description
`hwm_priority = value`	Defines the amount by which—when the file system exceeds the high water mark level—the water mark priority factor changes for archiving request . The default value is 0.0.

The following equation determines the water mark priority of the preview requests:

`lwm_priority` + `lhwm_priority` + `hlwm_priority` + `hwm_priority` = The water mark priority

Water mark preview request are either global directives or file system specific directives. Together these four water mark priorities create a dynamic priority factor. This includes a percentage value that indicates how full the file system is and at which levels the high and low water marks are set. The value that gets assigned to the preview request is determined on whether that factor is global, specific to a file system, or not set at all.

Water mark priorities are not used to calculate media requests for staging. They are used to calculate requests for archiving. If the water mark priority factor is a positive number, it will raise the archive request above the staging request. Similarly when the water mark priority is negative, the overall priority for archive request is reduced. This usually favors staging requests over archive requests.

When the file system crosses conditions that cause the system to change the conditional definition, the priority of each volume associated with that file system is—based on the appropriate water mark priority setting—recalculated.

6.2.5 Priority Scheme

`vsn_priority` + `wm_priority` + (`age_priority` * *`time_in_seconds_in_queue`*) = Total priority

The `wm_priority` value is the condition that was valid at the time: `hwm_priority`, `lhwm_priority`, `lwm_priority`, or `lhwm_priority`. The total priority of a preview request is the sum of all the priority factors. Since the data on the file system is not directly modified or affected by the settings in the `/etc/opt/vsm/preview.cmd` file, it is safe to experiment and change the directives to find the best settings.

The VSN drives can be limited. File archiving can also be performed in the background. Administrators can customize the settings in the `preview.cmd` file to help support any scenarios that influence the file system.

Refer to the man page for `preview.cmd` on the MDC node for more information.

6.3 Configure the Recycler

The directives in the `/etc/opt/vsm/recycler.cmd` file control the recycling process using the archive set method.

The recycler finds old or expired archive copies. It removes them from the system, clearing up volumes for new copies. Archive copies are considered expired when a user modifies a file. This leaves the old archive copies associated with that file out of date. New archive copies of that file will be made. This allows the old ones to be purged from the system. The recycler will move all unexpired archive copies to a new volume and purge the remaining expired volumes. This process is transparent to the end users. The recycler process identifies and moves expired copies to a separate volume. This frees up space for new copies on the current volume. If a volume contains nothing but expired files, configure the system to take different actions. The volume can be

relabeled for immediate reuse. The volume can also be exported to off-site storage to keep historical records of all file changes.

The `/etc/opt/vsm/recycler.cmd` file holds the general directives for the recycler process. It may also hold directives for every library in the VSM environment. Root access is required to edit and add directives to the `/etc/opt/vsm/recycler.cmd` file.

When creating the log file, specify it in the `recycler.cmd` file. The file name is the location of the file that is to be used for the log. The following is an example of this line:

```
logfile = /var/log/vsm/recycler.log
```

Archive volumes consist of the following:

- Current, relative data, comprised of archive images
- Expired data, comprised of archive images
- Unused space, comprised of free space not being used by any archive images

The recycler can be configured with site-specific parameters to keep a certain amount of space occupied by expired data. It also can specify when to start purging data once that amount has been met. As the files in the file system are changed or deleted, the corresponding archive images expire. Their classification changes from relative data to expired data. The actual physical space consumed by the archive images, expired or not, does not change.

The recycler helps keep the system up to date and relative by transforming used space from expired data into free space without losing any current data. The recycler may be run as often as desired. It is designed to automatically run periodically without having to be invoked by the user. The recycler keeps logs of state information in the library catalogs and the inodes. The `sls -D` command displays information about the file copies—such as whether that file copy is scheduled for re-archiving or purging.

The first recycler-process step has to finish before the archiver can re-archive the files.

The recycler process consists of three steps:

1. The recycler starts the process by marking all the current archive images with the re-archive attribute. The files with this attribute will be re-archived on a different volume so the current volume can be used again.
2. If tape is being used for archiving, the recycler marks the archive volume with the recycle attribute to prevent any further writing to the volume.
3. Finally, the archiver moves all the marked images to another separate volume and places all the current images on separate volumes other than the expired volumes (this is referred to as "re-archiving"). The volume with the expired images can now be relabeled so it can be reused and new data can be written on it.

The `sam-recycler` command is used to initiate the recycler process. The recycler uses two different methods to achieve its goal. Depending on the media type being used, this goal will be achieved either by automated library or archive set.

Table 9. Recycle Methods, Media Type, and Configuration File

Media Type	Method Used	Configuration File
Disks	Archive set	`archiver.cmd`
Removable media cartridges	Automated library	`recycler.cmd` and `recycler.sh`
Removable media cartridges	Archive set	`recycler.cmd`, `recycler.sh`, and `archiver.cmd`

When preparing to configure a recycler process, use the following rules as a guide.

- Do not recycle volumes that contain removable media files. A volume that contains removable media files can never be fully drained. Removable media files are created by using the `request` command.
- The directives in the `/etc/opt/vsm/archiver.cmd` control the recycling process using the archive set method. The directives in the `/etc/opt/vsm/recycler.cmd` control the recycling process using the automated library method.
- The `/etc/opt/vsm/recycler.cmd` file controls the recycler behavior.
- Do not run the recycler simultaneously while performing maintenance on the file system. This can cause inaccurate information in the `.inodes` file, which the recycler uses to determine which files are expired. If this happens, current files may be confused for expired files and they will be removed from the system prematurely.
- Before the recycler is run, make sure all the file systems are mounted correctly. If the recycler is running on an online disk, the file system containing the disk volumes must be mounted with the host system available.

CAUTION: When using disk archiving in a multiple VSM server environment, take extreme care when configuring the recycler. Ensure that the `diskvols.conf` file for each VSM server is pointing to a unique set of disk archiving directories. If any directories are shared by the VSM servers, the recycler will destroy the disk archive data.

6.3.1 Control the Recycler Process

To run the recycler, edit command files to enable or disable the recycler. Depending on the recycle method, either edit the `/etc/opt/vsm/archiver.cmd` file (for archive set method), or the `/etc/opt/vsm/recycler.cmd` file (for automated library method). Edit the appropriate file then use the `sam-recycler` command to initiate the recycler.

```
tas-mdc1# sam-recycler -dxv
```

When this command is issued, the recycler reads the `recycler.cmd` file. If any errors occur, they can be viewed in the VSM log located in `/var/log/vsm/sam-log`. Create a `crontab` entry to periodically run the recycler.

6.3.2 Removable Media Cartridges

Create a `recycler.cmd` file to recycle archive copies on cartridges in a library. To recycle using the archive set method, configure each library in the `recycler.cmd` file. Doing this will ensure the VSNs that do not fit into an archive set can still be recycled if necessary. To finish this operation, create a `recycler.sh` file.

6.3.3 Prevent Recycling with the no_recycle Directive

The `no_recycle` is a directive that disables the archiver from recycling volumes. The `no_recycle` directive has the following format:

```
no_recycle media_type VSN_regexp
```

`media_type` is the type of media the archiver is using (tape or disk) and *`VSN_regexp`* specifies a regular expression for the VSNs or disk volumes.

6.3.4 Specify Recycling on an Automated Library

Various recycling parameters for the VSNs may be enabled with the library directive. The library directive has the following format *library_name parameter*, where the *library_name* specifies the library name as defined in the family set (Fam. Set) field in the mcf file, specifies parameter keywords separated by spaces.

Table 10. Recycling Library Directive Parameters

Parameter	Description
-dataquantity *size*	Specifies the maximum amount of data—in the process of clearing volumes for reuse—the recycler is able to schedule for re-archiving. The default maximum value is 1GB.
-ignore	This directive prevents specified volumes from being recycled. It may be used for testing.
-hwm *percent*	This sets the high-water mark for the library. By default, it is set to 95%.
-vsncount *count*	This directive sets the maximum number of volumes that are scheduled to be recycled. By default this number is set to one.
-mail *email*	Set the recycler to send email reports to a specified email address. By default this feature is disabled.
-mingain *value*	This directive set the minimum VSN gain. By default it is set to 60% for volumes with less than 200GB of capacity and 90% for volumes with more than 200GB.

6.3.5 recycler.sh File

When archiving on removable media cartridges, create a /etc/opt/vsm/scripts/recycler.sh file. If archiving to only a single disk, then it is not necessary to create the recycler.sh file. The recycler.sh script will run when all current images from a VSN have been re-archived to a new VSN and have finished draining the cartridge of all active archive images. The recycler.sh script accepts the following arguments in this order:

```
# recycler.sh generic_media_type VSN slot eq media_type family_set_name
```

Table 11. recycler.sh Arguments

Argument	Description
generic_media_type	Used to specify the name of the appropriate media labeling command. This name must be tp.
VSN	The volume serial name (VSN) of the cartridge.
slot	Represents the slot location used for the media in the library.
eq	Represents the equipment number for the library where the media cartridge is located.
media_type	The actual media type. This may be used by chmed. For example, *media_type* = li for LTO media.

Argument	Description
family_set	Represents the family set name of the library or historian.

6.3.6 Configure Recycling for Disk Archive Volumes

Edit the `archiver.cmd` file to enable recycling if archiving to disk. When recycling to archive set, add the archive set directives to allow recycling between the `params` and `endparams` directives.

The following table describes the recycling directives that can be used in the archive set.

Table 12. Recycling Directives

Directive	Description
`-recycle_ignore`	Prevents the specified archive set from being recycled. Useful for testing.
`-rearch_stage_copy` *copynumber*	Specifies the staging for re-archiving to take place from specified copies.
`-recycle_dataquantity` *size*	Puts a limit on how much data the recycler schedules for re-archiving. By default the limit is ignored.
`-recycle_ignore`	Stops the archive set from being recycled.
`-recycle_mingain`*percent*	Sets the minimum gain mark limiting the recycling of volumes for disk volumes. The `mingain` is a percentage of all expired data associated with the volume. When the expired data exceeds the `mingain` value, the recycler begins recycling the expired data. The default `mingain` setting is 50%.
`-recycle_minobs`*percent*	This directive puts a limit on the recycler's selected tar files in a volume by setting a threshold limit for re-archiving processes of the disk archive volumes. When this percentage of all expired files in the archive tar file reaches this limit, the recycler will start moving current files into a new tar file. When this moving process is complete and all the current files have been moved, the old tar file is marked as a candidate and removed from the disk archive. The default threshold setting is 50%.

6.4 Configure the Releaser

The releaser component maintains the file system online disk cache automatically at the site-specified percentage usage thresholds. This is done by freeing disk blocks that are currently occupied by eligible archived files. Releasing is the process of freeing the primary disk storage that is currently being used by the archived file data. The online disk cache uses "water marks" to determine when to release disk space.

The `/etc/opt/vsm/releaser.cmd` file on the active MDC node controls this process.

There is a high-water mark and a low-water mark. The high-water mark is reached when the online disk consumption reaches a certain level. The `sam-releaser` daemon then releases disk space that is being

occupied by eligible archived files until the low-water mark is reached. Alternately, use the `release` command to immediately release file disk space. Configure the system to release files after they are archived or specify that the files are never to be released. File size and age are both considered when the system is determining which files are to be selected for release.

Configure the high-water and low-water marks by setting the `high=percent` and `low=percent` when setting the file system mount options. The default high is 80% and default low is 70%. The settings can be dynamically changed by changing the high and low parameters with `samu`. The `samu` settings are not persistent across mounts, so set the mount parameters at the same time.

Refer to the `man` page for `releaser.cmd` on the MDC node for more information about releaser configuration file.

6.4.1 The Releaser Process

When the high-water mark is reached, VSM file system invokes the releaser. The releaser reads the `releaser.cmd` file and collects the directives that control the release process. The releaser process then scans the file system to collect information about each file. At this point, the releaser begins releasing files in the priority order it has determined best suited. A file system can consist of thousands of files—of all different types. It can be wasteful for the system to keep track of the release priority for every single one of these files. This is because, if the system releases only a few large files, the file system might fall below the low-water mark. It is important for the releaser to examine the priority of each file to determine the best candidate for release. The releaser does this by identifying the first candidates and—if they do not have a priority greater than the lowest-priority amongst the candidate list—discarding subsequent candidates. The size of this candidate list is 100,000 if the number of inodes is >= 1 million or 30,000 if the number of inodes is < 1 million. The candidate list size can be changed by setting `list_size` in the `releaser.cmd` file.

The releaser selects the files with the highest priority to be released after the candidates have been selected and all other lower-level priority candidates ignored. As each file is released, the releaser checks to see if the low-water mark has been reached. This process continues until the low-water mark is reached. The releaser then stops releasing files. In some instances, all original candidate files in the candidate list can be released and the file system will still be over the high-water mark. When this happens, the releaser identifies an additional candidate list and starts the process all over. The releaser exits and shuts down if there are no viable candidates. This can happen, for example, if the files in question do not yet have archive copies. If this does happen, the releaser exits. After one minute passes, the releaser then starts again rechecking the files.

The age of a file is the amount of time that has elapsed from a certain event in the file history to the present time. The file inode tracks the following events which alter the file age:

- The time when residence has changed
- The time data has been accessed
- The time when the data has been modified

The `sls -D` option displays these times.

The candidate is a file eligible to be released. The following is a list of circumstances that will exclude a file from being a candidate for release. If the file:

- Is so small releasing it will have no substantial impact on space
- Was staged in the past at a time that makes it less than the minimum residence-time setting
- Has been flagged for partial release, through the release command `-p` option, causing it to already be partially released
- Has been marked as "never to be released" (this is done using the `release -n` command

- Has been marked as damaged
- Has not been archived
- Is a directory, pipe, block, or a character-special file, or removable media file
- Has an age that is negative (this can happen when the network file system client has inaccurate clock settings)
- Is being staged for additional copies (when the copy is complete the file will become eligible for release)
- If the `archiver.cmd` command file specifies the `-norelease` parameter for the file and the required copies have not yet been made
- Is currently offline

The priority of a file is represented by a numeric value. This priority indicates the rank of a candidate file based on user-supplied weights. The overall ranking is achieved through the sum of the age priority and the size priority. Files that have a larger numeric value for their priority are released before the files with lower numeric values.

The weight is a numeric based value that evens out the priority calculation. Thus, it includes file attributes that are of interest and excludes file attributes that are not important. The weight is a floating-point value on a scale from 0.0 to 1.0. Exclude certain attributes by setting their weight to 0 in the `/etc/opt/vsm/releaser.cmd` file.

6.4.2 Partial Releasing and Partial Staging

Releasing and staging are complementary processes. When a file is partially released, a portion at the beginning of the file remains in the disk cache while the remaining parts of the file are released. Partially releasing a file is useful because it provides immediate access to data in the file stub without having to stage the file.

The default partial size and the maximum size of the stub to remain online can be configured when mounting the file system using the following mount options:

`-o partial=`*`n`* — Sets the default kilobyte size (*n*) of a file stub that will remain online.

`-o maxpartial=`*`n`* — Sets the maximum kilobyte size (*n*) for the file stub that will remain online. This value must be larger than or equal to the `-o partial=n` value.

File stub size can be set using the `-p` option on the release command. Users may also specify different stub sizes for individual files. This is achieved with the `-s` option on the release command. These values need to be less than the value set for the mount option `maxpartial` setting.

The `-o partial_stage=n` option can be used to set how much of a file must be read before the rest of that file is staged of a partial release stub at the time of the system mount.

6.5 Change the VSM Configuration

About this task

If modifying the `/etc/opt/vsm/archiver.cmd` file, edit the file then validate the configuration on active MDC using the `archiver -lv` command, and restart archiver to enable changes using the `samcmd arrestart` command.

The `samcmd arrestart` command stops all active archive operations and force the archiver to re-read the `archiver.cmd` without stopping all of the storage archive manager.

If modifying the `mcf`, `defaults.conf`, `hosts.fs`, or `samfs.cmd` files, run the `samd config` command as shown below.

Procedure

1. Log in to the CIMS node as `root`.
2. SSH to the active MDC node and change directories to `/etc/opt/vsm`.

```
cims# ssh mdc
```

3. Edit the configuration file, for example, `archiver.cmd`.

```
tas-mdc1# cd /etc/opt/vsm
tas-mdc1# vi archiver.cmd
```

4. Exit the editor and save the file.
5. Validate the configuration from the active MDC node using the `archiver -lv` command.

```
tas-mdc1# archiver -lv
Reading '/etc/opt/vsm/archiver.cmd'.
1: #
2: # Example archiver.cmd
3: #
4:
5: #Wait for arrun before archiving.
6: #Do no automatically start archive
7: # wait
8:
9: ### Global directives
10: # Maximum size of tar archive
11: # Larger sizes provide better tape streaming I/O, but
12: # wastes more tape capacity
13: archmax = li 100G
14: archmax = ti 300G
```

6. Restart the archiver from the active MDC node.

```
tas-mdc1# samcmd arrestart
```

7. If modifying the `mcf`, `defaults.conf`, `hosts.fs`, or `samfs.cmd` files, run the `samd config` command to activate the new VSM configuration on the active MDC node.

```
tas-mdc1# samd stopsam
tas-mdc1# samd config
```

6.6 Configure VSM Archive Devices

Configure the persistent archive devices in the MCF (`/etc/opt/vsm/mcf`). The supported devices are listed in the `/etc/opt/vsm/inquiry.conf` file.

Network attached robots are not supported. Use the ACSLS SCSI Media Changer for the SL8500 robot. ACSLS 8.2 or 8.3 is required for the SCSI Media Changer option.

- Use the command `lsscsi -g` to list the SCSI devices.
- Use the `mtx` command to access the robot.
- Use the `mt` command to access the tape drives.

```
tas-mdc1# lsscsi -g
[4:0:0:0]    cd/dvd  TSSTcorp DVD-ROM SN-108DN D150  /dev/sr0   /dev/sg0
[6:2:0:0]    disk    DELL     PERC H710        3.13  /dev/sda   /dev/sg1
[7:0:0:0]    tape    IBM      ULTRIUM-TD6      E4J0  /dev/st0   /dev/sg2
[8:0:0:0]    tape    IBM      ULTRIUM-TD6      E4J0  /dev/st1   /dev/sg3
[9:0:0:0]    tape    IBM      ULTRIUM-TD6      E4J0  /dev/st2   /dev/sg14
[10:0:0:0]   tape    IBM      ULTRIUM-TD6      E4J0  /dev/st3   /dev/sg15
[10:0:0:1]   mediumx SPECTRA  PYTHON           2000  /dev/sch0  /dev/sg16
[11:0:0:0]   disk    NETAPP   INF-01-00        0810  /dev/sdb   /dev/sg4
[11:0:0:1]   disk    NETAPP   INF-01-00        0810  /dev/sdc   /dev/sg5
[11:0:0:2]   disk    NETAPP   INF-01-00        0810  /dev/sdd   /dev/sg6
. . .
```

In the MCF file (`/etc/opt/vsm/mcf`) sets up the robot and tape devices using `/dev/mapper/`.

```
tas-mdc1# more /etc/opt/vsm/mcf
#
#
#
# Equipment                     Eq      Eq      Fam.    Dev.  Additional
# Identifier                    Ord     Type    Set     State Parameters
#-----------                    ---     --      ------ ------ ------------------
tasfs1                          20      ma      tasfs1  on      shared
/dev/mapper/md0                 21      mm      tasfs1  on      -
/dev/mapper/dc0000              22      mr      tasfs1  on      -
/dev/mapper/dc0001              23      mr      tasfs1  on      -
/dev/mapper/dc0002              24      mr      tasfs1  on      -
/dev/mapper/dc0003              25      mr      tasfs1  on      -
/dev/mapper/dc0004              26      mr      tasfs1  on      -
/dev/mapper/dc0005              27      mr      tasfs1  on      -
#
#       T200 with 2 LTO6
#
# Equipment                      Eq      Eq      Fam.    Dev.  Additional
# Identifier                     Ord     Type    Set     State Parameters
#-----------                     ---     --      ------ ------ -----------------
-
/dev/tape/by-id/scsi-1SPECTRA_PYTHON_9410005694 100     rb      T200    on
/dev/tape/by-id/scsi-321130090a5005694-nst      102     tp      T200    on
/dev/tape/by-id/scsi-321140090a5005694-nst      103     tp      T200    on
/dev/tape/by-id/scsi-321120090a5005694-nst      104     tp      T200    on
/dev/tape/by-id/scsi-321110090a5005694-nst      105     tp      T200    on
```

If necessary, rebuild the symlinks if they are missing, reboot, and verify that they exist. Use the following command to rebuild the symlinks:

```
tas-mdc1# /sbin/udevadm trigger
```

Device mapper (DM) multipath is the standard configuration for all nodes. Configure DM multipath if there are multiple paths to devices from service nodes. All nodes configured for DM multipath have a connection to each storage array controller. Specific WWID values must be added to the `multipaths {}` section of the `/etc/multipath.conf` file so that devices have persistent names each time the node boots.

```
## Example of alias entries
#
multipaths {
multipath {
  wwid   360080e50001f76e400000c6554fde05c
  alias   dc0000
}
multipath {
  wwid   360080e50001f769c00000abe54453445
  alias  dc0001
}
multipath {
  wwid  360080e50001f76e400000b675445345f
  alias  dc0002
}
multipath {
  wwid   360080e50001f769c00000ac0544534a2
  alias  dc0003
}
multipath {
  wwid   360080e50001f76e400000b69544534b5
  alias  dc0004
}
multipath {
  wwid   360080e50001f769c00000ac2544534ec
  alias  dc0005
}
multipath {
  wwid   360080e5000341d18000004f7539f8bbc
  alias  md0
}
multipath {
  wwid   360080e5000341ab10000016552e9afc9
  alias  adm0
}
multipath {
  wwid   360080e5000341d180000017a52e9b0a7
  alias  dsk_archive0
}
multipath {
  wwid   360080e5000341ab10000016752e9b03a
  alias  dsk_archive1
}
multipath {
  wwid   360080e5000341d180000067253e57a5f
  alias  quorum_dsk
}
}
```

6.7 VSM Master Configuration File (MCF)

The master configuration file (MCF) resides in `/etc/opt/vsm/mcf`.

The MCF holds information for how devices managed by VSM are interrelated and how they communicate with other parts of the system. The MCF holds specific information about the devices and file systems that are included in the environment and enables administrators to identify and organize the disk partitions for the file system.

The MCF (example below) is processed by the system when `sam-fsd` is started. Changes may be made to the MCF at any time. They will not, however, take effect until `sam-fsd` is restarted or until the command `samd` configuration is entered. There are six fields inside the `mcf`—four that are required and two that are optional.

```
tas-mdc1# more /etc/opt/vsm/mcf
#
#
#
# Equipment                     Eq      Eq      Fam.    Dev.  Additional
```

```
# Identifier                    Ord     Type    Set     State Parameters
#-----------                    ---     --      ------ ------ ------------------
tasfs1                          20      ma      tasfs1  on      shared
/dev/mapper/md0                 21      mm      tasfs1  on      -
/dev/mapper/dc0000              22      mr      tasfs1  on      -
/dev/mapper/dc0001              23      mr      tasfs1  on      -
/dev/mapper/dc0002              24      mr      tasfs1  on      -
/dev/mapper/dc0003              25      mr      tasfs1  on      -
/dev/mapper/dc0004              26      mr      tasfs1  on      -
/dev/mapper/dc0005              27      mr      tasfs1  on      -
#
#       T200 with 2 LTO6
#
# Equipment                      Eq      Eq      Fam.    Dev.  Additional
# Identifier                     Ord     Type    Set     State Parameters
#-----------                     ---     --      ------ ------ -----------------
-
/dev/tape/by-id/scsi-1SPECTRA_PYTHON_9410005694 100     rb      T200    on
/dev/tape/by-id/scsi-321130090a5005694-nst      102     tp      T200    on
/dev/tape/by-id/scsi-321140090a5005694-nst      103     tp      T200    on
/dev/tape/by-id/scsi-321120090a5005694-nst      104     tp      T200    on
/dev/tape/by-id/scsi-321110090a5005694-nst      105     tp      T200    on
```

Follow these rules when entering data into the MCF:

- Use space/tab between the fields of the file
- Use the number sign character (#) to begin a comment line
- Use the dash character (-) to indicate the field has no meaningful information when entering data in optional fields

The following fields are required:

- Equipment identifier
- Equipment number
- Equipment type
- Family set
- The following fields are optional:
  - Additional parameters
  - Device state

MCF Equipment Identifier

The `Equipment Identifier` field identifies a file system or an automated library. This field names a file system along with the file system disk devices or it names an automated library with the library tape devices. When the equipment identifier field identifies the name of a file system, it is restricted to 31 characters. For other content, however, it can display up to 127 characters. File system names must begin with a letter. They can only contain alphabetical characters and numeric characters or the underscore character (_).

The file system name can only contain 31 characters. It must also be the same as the name in the family set (`Fam. set`) field. The master configuration file (MCF) must define the disks/devices that are included in the file system. More than one file system may be included in the MCF.

MCF Disk Partition or Slice Description

The disk partition or slice description may contain up to 127 characters. To identify a disk partition or a slice, use a `/dev/mapper` device. Set the `nodev` keyword for the `mm` device on a shared client if the client is not connected to the metadata device.

MCF Automated Library Description

The automated library description may contain up to 127 characters. To identify an automated library, use a `/dev/mapper` device.

MCF Tape Drive Description

The tape drive description may contain up to 127 characters. To identify a tape drive entry, use a `/dev/mapper` device.

Equipment Number

The equipment number field (*eq*) may contain numerous rows. Each row in this field must have its own unique identifier for every device that is defined. This number must be an integer between one and 65,534. The number will represent every disk and removable media device that is configured for the system. The equipment number is used to specify that device in the VSM commands. Use low numbers in the equipment number field to identify each device. Keep the internal archive tables small.

Equipment Type

The equipment type field specifies how VSM determines the interaction between specific devices. Two or three characters represent different device types. The following are acceptable character combinations:

Table 13. MCF Equipment Type Field

`ms`	A file system that stores data and metadata together on the same device.
`ma`	A file system that stores metadata on a separate device. The metadata is stored on `mm` devices and the data is stored on a `md` or `mr` device.
`md`	A round-robin or a striped disk device using dual allocation to store the file data. In a `ms` file system, data and metadata is stored on this `md` device. In a `ma` file system, only data is stored on this device.
`mm`	A device that stores the metadata information (inodes, directories, etc.). More than one metadata device can be defined. Round-robin is the default allocation used for multiple metadata devices. All metadata on a `ma` file system is stored on separate devices from the data devices.
`mr`	A round-robin or striped device using single allocation to store file data.
`rb`	A SCSI attached library which is automatically configured by VSM. It is also a STK library configured in SCSI emulation mode by ACSLS.
`tp`	A generic tape drive which is automatically configured by VSM.
`hy`	The historian, the on-the-shelf media catalog.

MCF Family Set Field

The family set is a user-defined name for a group of devices. The name must begin with a letter. It can only contain numeric, alphabetic, and the underscore (_) characters. The family set name is used to associate related groups of related devices. These names can be a file system name, an automated library identifier, or a dash (–) character. The dash character must be used only for a standalone removable tape device.

When the name of an automated library identifier is established, then the library (and all the drives associated with that library) must use the same identifier. When a family system name is established, every disk device attached to the file system must—in the family set field—share the same file system name. VSM uses this field to group all the devices together as a file system or an automated library. Use the `sammkfs` command to record the family set name for all of the devices connected to the file system. The `samfsck -F -R` command can also be used to rename the family set name.

MCF Device State and Additional Parameters Field

When the file system is first initialized it sends the state of the device to be displayed in the device state field. The device `state` field displays either `on` or `off`. All devices are set to be `on`, by default. The device state field is optional and can be any of the following:

- `on` (default)
- `off`
- `unavail`
- `down`
- `idle`
- `readonly`
- `noalloc`

This field is used by:

- Disk devices
- Libraries
- Drives
- Any other system-attached devices

The `Additional Parameters` field contains information that is not vital to the system. It may display the path to a library catalog or it might display an interface file. This field cannot exceed 127 characters. This field can also be left blank using the dash character (–). Use this field to specify a new path to the library catalog file in the `mcf` file (`/tas_admin/vsm/var/catalog/`*`catalog_name`*). In a shared file system, the path must include the keyword shared in the additional parameters field.

6.8 Configure a VSM File System

6.8.1 Create a VSM File System

About this task

The master configuration file (MCF) in `/etc/opt/vsm/mcf` stores the information about all the devices that are controlled by VSM. The MCF contains information the file system requires to organize RAID/disk devices into file systems. The MCF also contains the entries for every automated library or standalone tape device in the file system. A sample `mcf` file is located in `/opt/vsm/examples/mcf` on the MDC node.

Procedure

1. Log in to the CIMS node as `root`.

2. Get list of disk devices in `/dev/mapper` to include in the VSM file system from active MDC. Devices named md*num* are SSD metadata devices attached to the MDC nodes. Devices named dc*num* are disk-based data devices attached to both MDC and client nodes. Devices in bold will be used for the VSM file system in this example. Use the Bright `cmsh -c` command to run a command in the `cmsh` environment. To list disk devices to include in the VSM file system, enter:

```
cims# cmsh -c "device ; use tas-mdc1 ; sysinfo"

System Information
--------------------------------------------------------------------------------
BIOS Version 2.2.2
BIOS Vendor
BIOS Date 01/16/2014
Motherboard Manufacturer
Motherboard Name 0H5J4J
System Manufacturer
<...SNIP...>
Disk sdw INF-01-00 (15998895783936 bytes, 16TB)
Disk sdx INF-01-00 (15998895783936 bytes, 16TB)
Disk sdy INF-01-00 (15998895783936 bytes, 16TB)
Disk sdz INF-01-00 (199512555520 bytes, 199.5GB)
Disk dm-0 LVM2 Logical Volume /dev/mapper/dc0000 (32001998454784 bytes, 32TB)
Disk dm-1 LVM2 Logical Volume /dev/mapper/dc0001 (15998895783936 bytes, 16TB)
Disk dm-2 LVM2 Logical Volume /dev/mapper/dc0002 (15998895783936 bytes, 16TB)
Disk dm-3 LVM2 Logical Volume /dev/mapper/dc0003 (15998895783936 bytes, 16TB)
Disk dm-4 LVM2 Logical Volume /dev/mapper/dc0004 (15998895783936 bytes, 16TB)
Disk dm-5 LVM2 Logical Volume /dev/mapper/dc0005 (15998895783936 bytes, 16TB)
Disk dm-6 LVM2 Logical Volume /dev/mapper/md0 (199512555520 bytes, 199.5GB)
Disk dm-7 LVM2 Logical Volume /dev/mapper/adm0 (1798765019136 bytes, 1.799TB)
Disk dm-8 LVM2 Logical Volume /dev/mapper/dsk_archive0 (1798765019136 bytes, 1.799TB)
Disk dm-9 LVM2 Logical Volume /dev/mapper/dsk_archive1 (1798765019136 bytes, 1.799TB)
```

3. SSH to the active MDC node.

```
cims# ssh -l root mdc
Last login: Wed Oct  7 16:12:31 2015 from tas-cims1.cm.cluster
```

4. Edit the `/etc/opt/vsm/mcf` file and complete each of the fields.

5. Configure the mount point for the `tasfs1` file system in Bright, and assign the mount point to the `tas-mdc` MDC node category.

6. Create the host file for the `tasfs1` file system in `/etc/opt/vsm/hosts.tasfs1`. For details on the configuration of the host file see the `hosts.fs(4)` man page on the MDC node.

```
#
# Host file for family set 'tasfs1'
#
# Version: 4 Generation: 4 Count: 4
# Server = host 1/mdc1, length = 106
#Example
#
tas-mdc1 10.143.0.1 1 0 server
tas-mdc2 10.143.0.2 2 0
tas-dm1 10.143.0.3 0 0
tas-dm2 10.143.0.4 0 0
```

7. Verify the configuration on the MDC.

```
tas-mdc1# sam-fsd
Non-autogenerated mcf file exists, leaving it alone
```

```
Would start sam-sharefsd(tasfs1)
Trace file controls:
sam-amld          /var/opt/vsm/trace/sam-amld
                  cust err fatal misc proc date
                  size 10M age 0
sam-archiverd     /var/opt/vsm/trace/sam-archiverd
                  cust err fatal misc proc date
                  size 10M age 0
sam-catserverd    /var/opt/vsm/trace/sam-catserverd
                  cust err fatal misc proc date
                  size 10M age 0
sam-fsd           /var/opt/vsm/trace/sam-fsd
                  cust err fatal misc proc date
                  size 10M age 0
sam-rftd          /var/opt/vsm/trace/sam-rftd
                  cust err fatal misc proc date
                  size 10M age 0
sam-recycler      /var/opt/vsm/trace/sam-recycler
                  cust err fatal misc proc date
                  size 10M age 0
sam-sharefsd      /var/opt/vsm/trace/sam-sharefsd
                  cust err fatal misc proc date
                  size 10M age 0
sam-stagerd       /var/opt/vsm/trace/sam-stagerd
                  cust err fatal misc proc date
                  size 10M age 0
sam-shrink        /var/opt/vsm/trace/sam-shrink
                  cust err fatal misc proc date
                  size 10M age 0
Would start sam-archiverd()
Would start sam-stagerd()
Would start sam-amld()
```

8. Configure the mount point for the `tasfs1` file system in Bright. File system mounting options are configured in Bright for service node categories (`tas-mdc` for example). Refer to the *Data Management Practices (DMP) Administrator Guide, S-2327* for more information about file system mounting using the Bright `cmgui` or `cmsh` tools.

9. Use SSH to log in to the active MDC node as `root`.

```
cims# ssh mdc
```

10. Activate the new VSM configuration.

```
tas-mdc1# samd config
```

11. Initialize the `tasfs1` file system.

```
tas-mdc1# sammkfs -S tasfs1
```

12. Mount the file system. Bright service node categories determine which file systems are mounted on a service node. File systems for DM and MDC node categories are configured in Bright.

6.8.2 Configure a Shared VSM File System in Bright

About this task

File systems for DM and MDC node categories are configured in Bright. This procedure configures a file system for the MDC node category (`tas-mdc`). It can also be used to configure a file system for the DM node (`tas-dm`) or a client node category.

Procedure

1. Log in to the CIMS node as `root` and start `cmsh`.

2. Switch to `category` mode, and select (`use`) a service node category (`tas-mdc` in this example).

```
tas-cims1# cmsh
[tas-cims1]% category use tas-mdc
[tas-cims1->category[tas-mdc]]%
```

3. List the file systems for the `tas-mdc` category.

```
[tas-cims1->category[tas-mdc]]% fsmounts
[tas-cims1->category[tas-mdc]->fsmounts]% listd
Device                       Mountpoint (key)                 Filesystem
---------------------------- -------------------------------- -----------
devpts                       /dev/pts                         devpts
proc                         /proc                            proc
sysfs                        /sys                             sysfs
$localnfsserver:/cm/shared   /cm/shared                       nfs
$localnfsserver:/home        /home                            nfs
/dev/mapper/adm0             /tas_admin                       ext4
```

4. Add a new VSM file system (`tasfs1`) to the `tas-mdc` category.

```
[tas-cims1->category[tas-mdc]->fsmounts]% add /vsm/tasfs1d
[tas-cims1->category*[tas-mdc*]->fsmounts*[/vsm/tasfs1*]]%
```

5. Set the device name for the file system.

```
[tas-cims1->category[tas-mdc]->fsmounts]% set device tasfs1
[tas-cims1->category*[tas-mdc*]->fsmounts*[/vsm/tasfs1*]]%
```

6. Specify the type of file system (`samfs`).

```
[tas-cims1->category*[tas-mdc*]->fsmounts*[/vsm/tasfs1*]]% set filesystem samfs
```

7. Specify the mount options for the file system. Separate each option with a comma as shown.

```
[tas-cims1->category*[tas-mdc*]->fsmounts*[/vsm/tasfs1*]]% set mountoptions default,shared,noauto
```

8. Commit the changes.

```
[tas-cims1->category*[tas-mdc*]->fsmounts*[/vsm/tasfs1*]]% commit
[tas-cims1->category[tas-mdc]->fsmounts[/vsm/tasfs1]]%
```

9. List the file systems for the `tas-mdc` category.

```
[tas-cims1->category[tas-mdc]]% fsmounts
[tas-cims1->category[tas-mdc]->fsmounts]% list
Device                       Mountpoint (key)                 Filesystem
---------------------------- -------------------------------- ----------
devpts                       /dev/pts                         devpts
proc                         /proc                            proc
sysfs                        /sys                             sysfs
$localnfsserver:/cm/shared   /cm/shared                       nfs
$localnfsserver:/home        /home                            nfs
tasfs1                       /vsm/tasfs1                      samfs
/dev/mapper/adm0             /tas_admin                       ext4
```

In the examples below, the file system `vsmfs1` is a stand-alone file system with archiving. `tasfs1` is a shared file system with archiving. The disk archive file system is `tasfs2` which has no archiving capability due to the mount option `nosam`. The mount parameter `stripe=2` means the file system is running with striped allocation.

```
tasfs1 /vsm/tasfs1 samfs rw,auto,shared 0 0
tasfs2 /vsm/tasfs2 samfs rw,auto,nosam 0 0
```

Always mount the disk archiving file systems first. Always unmount the disk archiving file system last.

6.8.3 Configure a VSM File System on a Client Node

Procedure

1. Log in to the CIMS node as `root`.

2. Determine the software image name used by the client node or DM node (`TAS-XX-2.0.1-201510011105-dm` in this example). The `tas-dm` category is used in the example.

```
cims1# cmsh -c 'category ; show tas-dm'
Parameter                      Value
------------------------------ ------------------------------------------------
BMC Password                   < not set >
BMC User ID                    -1
BMC User name
Default gateway                172.30.86.1
Disk setup                     <2436 bytes>
Exclude list full install      <312 bytes>
Exclude list grab              <996 bytes>
Exclude list grab new          <996 bytes>
Exclude list sync install      <1414 bytes>
Exclude list update            <2864 bytes>
Filesystem exports             <1 in submode>
Filesystem mounts              <6 in submode>
Finalize script                <4164 bytes>
Initialize script              <0 bytes>
Install boot record            yes
Install mode                   AUTO
Ipmi power reset delay         0
Management network             esmaint-net
Name                           tas-dm
Name servers
New node install mode          FULL
Notes                          <0 bytes>
Provisioning associations      <1 internally used>
Revision
Roles                          <1 in submode>
Search domain
Services                       <0 in submode>
Software image                 TAS-XX-2.0.1-201510011105-dm
Time servers
User node login                ALWAYS
```

3. Change directories to `/cm/images/TAS-XX-2.0.1-201510011105-dm` on the CIMS.

```
tas-cims1# cd /cm/images/TAS-XX-2.0.1-201510011105-dm
```

4. Generate an MCF file for the client node with `samfsconfig`. Redirect output to the `mcf` file in the `TAS-XX-2.0.1-201510011105-dm` software image. The `nodev` device in the following example is a special device used on client nodes. It displays where a metadata device typically displays on an MDC node. The `nodev` device is required because client nodes use the shared file system protocol to access the metadata disks. They do not access the metadata disks directly.

```
tas-cims1# ssh tas-dm1 'samfsconfig /dev/mapper/*' | tee etc/opt/vsm/mcf
#
# Family Set 'tasfs1' Created Mon Jul 21 14:09:13 2014
# Generation 0 Eq count 7 Eq meta count 1
#
# zoned-off or missing metadata device
```

```
#
tasfs1                20    ma   tasfs1  - shared
nodev                 21    mm   tasfs1  -
/dev/mapper/dc0000    22    mr   tasfs1  -
/dev/mapper/dc0001    23    mr   tasfs1  -
/dev/mapper/dc0002    24    mr   tasfs1  -
/dev/mapper/dc0003    25    mr   tasfs1  -
/dev/mapper/dc0004    26    mr   tasfs1  -
/dev/mapper/dc0005    27    mr   tasfs1  -
```

5. Configure the mount point for the `tasfs1` file system in Bright.

6. Create the host file for the `tasfs1` file system in `etc/opt/vsm/hosts.tasfs1`. For details on host file configuration see the `hosts.fs(4)` man page on the MDC node.

```
#
# Host file for family set 'tasfs1'
#
# Version: 4 Generation: 4 Count: 4
# Server = host 1/mdc1, length = 106
#Example
#
tas-mdc1 10.143.0.1 1 0 server
tas-mdc2 10.143.0.2 2 0
tas-dm1 10.143.0.3 0 0
tas-dm2 10.143.0.4 0 0
```

7. Synchronize the changes to the MCF file with each DM node. Use the `cmsh imageupdate` command to update the `tas-dm` category.

```
tas-cims1#  cmsh -c 'device ; imageupdate -w -c tas-dm'
```

8. Verify the configuration on the DM node (`tas-dm1`).

```
tas-cims1#  ssh tas-dm1 sam-fsd
Non-autogenerated mcf file exists, leaving it alone
Would start sam-sharefsd(tasfs1)
Trace file controls:
sam-amld      /var/opt/vsm/trace/sam-amld
              cust err fatal misc proc date
              size    10M  age 0
sam-archiverd /var/opt/vsm/trace/sam-archiverd
              cust err fatal misc proc date
              size    10M  age 0
sam-catserverd /var/opt/vsm/trace/sam-catserverd
              cust err fatal misc proc date
              size    10M  age 0
sam-fsd       /var/opt/vsm/trace/sam-fsd
              cust err fatal misc proc date
              size    10M  age 0
sam-rftd      /var/opt/vsm/trace/sam-rftd
              cust err fatal misc proc date
              size    10M  age 0
sam-recycler  /var/opt/vsm/trace/sam-recycler
              cust err fatal misc proc date
              size    10M  age 0
sam-sharefsd  /var/opt/vsm/trace/sam-sharefsd
              cust err fatal misc proc date
              size    10M  age 0
sam-stagerd   /var/opt/vsm/trace/sam-stagerd
              cust err fatal misc proc date
              size    10M  age 0
sam-shrink    /var/opt/vsm/trace/sam-shrink
              cust err fatal misc proc date
              size    10M  age 0
```

9. From the CIMS node, use SSH to log in to the data mover node (`tas-dm1` in this example).

```
cims# ssh tas-dm1
tas-dm1#
```

10. Activate the new VSM configuration on the `tas-dm1` node.

```
tas-dm1# samd config
```

11. Mount the VSM file system on the DM nodes.

File system mounting options are in Bright for service node categories (`tas-dm` or `tas-mdc` for example).

6.8.4 Add and Remove Client Hosts

About this task

After a client host has been added to the Bright system configuration, use the `samsharefs` command to fetch the shared file system host file. Then write it to a file so that it can be edited. If the file system has not yet been mounted, use the `-R` option. It is possible to add and remove client hosts after creating the file system and mounting all the participants.

Procedure

1. Log in to the CIMS node as `root`.

2. Determine the software image name used by the MDC node `tas-mdc` category (`TAS-XX-2.0.1-201510011105-mdc` in this example).

```
cims1# cmsh -c 'category ; show tas-mdc'
Parameter                      Value
------------------------------ ------------------------------------------------
BMC Password                   < not set >
BMC User ID                    -1
BMC User name
Default gateway                172.30.86.1
Disk setup                     <2436 bytes>
Exclude list full install      <312 bytes>
Exclude list grab              <996 bytes>
Exclude list grab new          <996 bytes>
Exclude list sync install      <1414 bytes>
Exclude list update            <2864 bytes>
Filesystem exports             <1 in submode>
Filesystem mounts              <6 in submode>
Finalize script                <4164 bytes>
Initialize script              <0 bytes>
Install boot record            yes
Install mode                   AUTO
Ipmi power reset delay         0
Management network             esmaint-net
Name                           tas-mdc
Name servers
New node install mode          FULL
Notes                          <0 bytes>
Provisioning associations      <1 internally used>
Revision
Roles                          <1 in submode>
Search domain
Services                       <0 in submode>
Software image                 TAS-XX-2.0.1-201510011105-mdc
Time servers
User node login                ALWAYS
```

3. Change directories to `/cm/images/TAS-XX-2.0.1-201510011105-mdc` on the CIMS node.

```
cims1# cd /cm/images/TAS-XX-2.0.1-201510011105-mdc
```

4. Edit the host file for the `tasfs1` file system in `etc/opt/vsm/hosts.tasfs1`. The example below adds the client host `tas-dm3`. For details on the configuration of the host file, see the `hosts.fs` man page on the MDC node. Always add the new clients at the end of the `hosts.`*`filesystem`* file.

```
cims1# vi hosts.tasfs1
#
# Host file for family set 'tasfs1'
#
#
# Version: 4    Generation: 4    Count: 4
# Server - host 1/mdc1, length - 106
#Example
#
tas-mdc1    10.143.0.1 1 0 server
tas-mdc2    10.143.0.2 2 0
tas-dm1     10.143.0.3 0 0
tas-dm2     10.143.0.4 0 0
tas-dm3     10.143.0.5 0 0
```

5. Synchronize the change to the MDC nodes.

```
cims1# cmsh -c 'device ; imageupdate -w -c tas-mdc'
```

6. Verify the configuration on the MDC.

```
cims1#  ssh tas-dm1 sam-fsd
Non-autogenerated mcf file exists, leaving it alone
Would start sam-sharefsd(tasfs1)
Trace file controls:
sam-amld      /var/opt/vsm/trace/sam-amld
              cust err fatal misc proc date
              size    10M  age 0
sam-archiverd /var/opt/vsm/trace/sam-archiverd
              cust err fatal misc proc date
              size    10M  age 0
sam-catserverd /var/opt/vsm/trace/sam-catserverd
              cust err fatal misc proc date
              size    10M  age 0
sam-fsd       /var/opt/vsm/trace/sam-fsd
              cust err fatal misc proc date
              size    10M  age 0
sam-rftd      /var/opt/vsm/trace/sam-rftd
              cust err fatal misc proc date
              size    10M  age 0
sam-recycler  /var/opt/vsm/trace/sam-recycler
              cust err fatal misc proc date
              size    10M  age 0
sam-sharefsd  /var/opt/vsm/trace/sam-sharefsd
              cust err fatal misc proc date
              size    10M  age 0
sam-stagerd   /var/opt/vsm/trace/sam-stagerd
              cust err fatal misc proc date
              size    10M  age 0
sam-shrink    /var/opt/vsm/trace/sam-shrink
              cust err fatal misc proc date
              size    10M  age 0
Would start sam-archiverd()
Would start sam-stagerd()
Would start sam-amld()
```

7. Update the host configuration. If the file system is mounted, then use the `samsharefs` command with the `-u` option as follows

```
cims1# ssh tas-mdc1 samsharefs -u tasfs1
#
# Host file for family set 'tasfs1'
#
#
# Version: 4    Generation: 2    Count: 4
# Server = host 0/tas-mdc1, length = 122
#
```

```
tas-mdc1    10.143.0.1 1 0 server
tas-mdc2    10.143.0.2 2 0
tas-dm1     10.143.0.3 0 0
tas-dm2     10.143.0.4 0 0
tas-dm3     10.143.0.5 0 0
```

8. Add the `-R` option if the file system is not mounted.

```
cims1# ssh tas-mdc1 samsharefs -R -u tasfs1
```

6.8.5 Export a VSM File System

Procedure

1. Log in to the CIMS node as `root` and start `cmsh`.

```
tas-cims1# cmsh
[tas-cims1]%
```

2. Verify the DM node supports a 10GbE connection to the customer 10GbE network

```
[tas-cims1]% network
[tas-cims1->network]% use nfs-10gbe-net
[tas-cims1->network[nfs-10gbe-net]]% show
Parameter                      Value
------------------------------ ------------------------------------------------
Base address                   10.2.0.0
Broadcast address              10.2.255.255
Domain Name                    esmaint-net.cluster
Dynamic range end              10.2.0.40
Dynamic range start            10.2.0.0
Gateway                        0.0.0.0
IPv6                           no
Lock down dhcpd                no
MTU                            9000
Management allowed             no
Netmask bits                   16
Node booting                   no
Notes                          <0 bytes>
Revision
Type                           Internal
name                           nfs-10gbe-net
```

3. In another window, run a `ping` test to verify that the DM node and sever being setup for the NFS mount are communicating.

4. Switch to device mode to add an interface to the DM node.

```
[tas-cims1->network[nfs-10gbe-net]]% device
[tas-cims1->device]%
```

5. Select (`use`) the DM node and list its interfaces.

```
[tas-cims1]% device use tas-dm1
[tas-cims1->device[tas-dm1]]% interfaces
[tas-cims1->device[tas-dm1]->interfaces]% list
Type         Network device name  IP              Network
------------ -------------------- --------------- ----------------
bmc          ipmi0                10.148.0.3      ipmi-net
physical     BOOTIF [prov]        10.141.0.3      esmaint-net
physical     eth1                 172.30.86.136   site-admin-net
physical     eth2                 10.143.0.3      metadata-net
```

6. Add the physical 10GbE interface for `eth4` and configure the settings.

```
[tas-cims1->device[tas-dm1]->interfaces]% add physical eth4
[tas-cims1->device*[tas-dm1*]->interfaces*[eth4*]]% set ip 10.2.0.3
[tas-cims1->device*[tas-dm1*]->interfaces*[eth4*]]% set network nfs-10gbe-net
[tas-cims1->device*[tas-dm1*]->interfaces*[eth4*]]% show
Parameter                      Value
------------------------------ ------------------------------------------------
Additional Hostnames
Card Type
DHCP                           no
IP                             10.2.0.3
MAC                            00:00:00:00:00:00
Network                        NFS-10Gbe-net
Network device name            eth4
Revision
Speed
Type                           physical
```

7. Commit the changes in Bright.

```
[tas-cims1->device*[tas-dm1*]->interfaces*[eth4*]]% commit
[tas-cims1->device[tas-dm1]->interfaces[eth4]]%
```

8. In another window, log in to the CIMS node `root` and SSH to the DM node. Reboot the DM node.

```
tas-cims1# ssh tas-dm1
[root@tas-dm1 ~]# reboot
```

9. From the remote server window, use `ping` to test the IP address the `eth4` interface on the DM node. Verify it is functioning.

```
remote# ping 10.2.0.3
PING 10.2.0.3 (10.2.0.3) 56(84) bytes of data.
64 bytes from 10.2.0.3: icmp_seq=1 ttl=64 time=0.036 ms
64 bytes from 10.2.0.3: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 10.2.0.3: icmp_seq=3 ttl=64 time=0.095 ms
^C
--- 10.2.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.036/0.055/0.095/0.029 ms
```

10. In the Bright `cmsh` window, configure the DM node category (`tas-dm`) to export the `/vsm/tasfs1` file system. Switch to `category` mode and select (`use`) the `tas-dm` category

```
[tas-cims1->device[tas-dm1]->interfaces[eth4]]% category
[tas-cims1->category]% use tas-dm
[tas-cims1->category[tas-dm]]%
```

11. Switch to `roles` submode assign the storage role to the `tas-dm` category. This starts the required services for the all DM nodes using the `tas-dm` category.

```
[tas-cims1->category[tas-dm]]% roles
[tas-cims1->category[tas-dm]->roles]% assign storage
[tas-cims1->category*[tas-dm*]->roles*[storage*]]%
```

12. Commit the changes in Bright. Note that the NFS service should start for DM nodes assigned to the `tas-dm` category.

```
[tas-cims1->category*[tas-dm*]->roles*[storage*]]% commit
[tas-cims1->category[tas-dm]->roles[storage]]%
Mon Aug  4 10:49:19 2014 [notice] tas-dm1: Service nfs was started
[tas-cims1->category[tas-dm]->roles[storage]]%
Mon Aug  4 10:49:19 2014 [notice] tas-dm2: Service nfs was started
```

13. Exit the storage submode and return to `tas-dm` in `category` mode.

```
[tas-cims1->category[tas-dm]->roles[storage]]% exit
[tas-cims1->category[tas-dm]->roles]% exit
[tas-cims1->category[tas-dm]]%
```

14. Configure the `tas-dm` category to export the `/vsm/tasfs1` file system. List the file systems that the `tas-dm` category is exporting.

```
[[tas-cims1->category[tas-dm]]% fsexports tas-dm; list
Name (key)               Path                     Hosts                              Write
------------------------ ------------------------ ---------------------------------- ------
[tas-cims1->category[tas-dm]->fsexports]]%
```

15. Exit `fsexports` submode.

```
[tas-cims1->category[tas-dm]->fsexports]]% exit
[tas-cims1->category[tas-dm]]%
```

16. List the file systems mounted by the `tas-dm` category.

```
[tas-cims1->category[tas-dm]]% fsmounts tas-dm; list
Device                       Mountpoint (key)                 Filesystem
---------------------------- -------------------------------- ----------------
devpts                       /dev/pts                         devpts
proc                         /proc                            proc
sysfs                        /sys                             sysfs
$localnfsserver:/cm/shared   /cm/shared                       nfs
$localnfsserver:/home        /home                            nfs
tasfs1                       /vsm/tasfs1                      samfs

[tas-cims1->category[tas-dm]->fsmounts]% exit
```

17. Exit `fsmounts` mode.

```
[[tas-cims1->category[tas-dm]->fsmounts]% exit
[[tas-cims1->category[tas-dm]]%
```

18. Add the shared `/vsm/tasfs1` file system export to the `tas-dm` category.

```
[tas-cims1->category[tas-dm]]% fsexports
[tas-cims1->category[tas-dm]->fsexports]]%
```

19. Configure the network settings.

```
[tas-cims1->category[tas-dm]->fsexports]]% add /vsm/tasfs1
[tas-cims1->category*[tas-dm*]->fsexports*[/vsm/tasfs1*]]% set name /vsm/tasfs1/@nfs-10gbe-net
[tas-cims1->category*[tas-dm*]->fsexports*[/vsm/tasfs1/@nfs-10gbe-net*]]% set extraoptions no_subtree_check
[tas-cims1->category*[tas-dm*]->fsexports*[/vsm/tasfs1/@nfs-10gbe-net*]]% set hosts nfs-10gbe-net
[tas-cims1->category*[tas-dm*]->fsexports*[/vsm/tasfs1/@nfs-10gbe-net*]]% set write yes
```

20. Commit the changes in Bright and exit the submode.

```
[tas-cims1->category*[tas-dm*]->fsexports*[/vsm/tasfs1/@nfs-10gbe-net*]]% commit
[tas-cims1->category[tas-dm]->fsexports[/vsm/tasfs1/@nfs-10gbe-net]]% exit
[tas-cims1->category[tas-dm]->fsexports]]%
```

21. List the file system exports for the `tas-dm` category.

```
[tas-cims1->category[tas-dm]->fsexports]]% list
Name (key)                 Path                     Hosts                              Write
-------------------------- ------------------------ ---------------------------------- ------
/vsm/tasfs1/@NFS-10Gbe-net  /vsm/tasfs1              NFS-10Gbe-net (10.2.0.0/16)       yes
```

22. Enter **`quit`** to exit `cmsh`.

```
[tas-cims1->category[tas-dm]->fsexports]]% quit
tas-cims1#
```

23. From the remote server window, create a mount point for the file system.

```
remote# mkdir -p /vsm/tasfs1
```

24. Mount the exported file system on the remote server.

```
remote# mount -t nfs -o nfsvers=3 10.2.0.3:/vsm/tasfs1 /vsm/tasfs1/
remote# mountd
/dev/sda2 on / type ext3 (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
debugfs on /sys/kernel/debug type debugfs (rw)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,mode=1777)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
fusectl on /sys/fs/fuse/connections type fusectl (rw)
securityfs on /sys/kernel/security type securityfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
10.2.0.3:/vsm/tasfs1 on /vsm/tasfs1 type nfs (rw,nfsvers=3,addr=10.2.0.3)
```

25. Verify `/vsm/tasfs1` file system access from the remote server.

```
remote# cd /vsm/tasfs1/
remote# ls
.archive  .inodes  .quota_g  .quota_u  .stage  lost+found  tas-dm1  tas-dm2
```

6.8.6 Mount or Unmount a VSM File System on the MDC Node

On the active metadata controller node (MDC), VSM shared file systems are mounted using the `tasha` utility. The shared VSM filesystem must be mounted on an active MDC node prior to manually mounting the filesystem on the data mover (DM) nodes.

To unmount the file system work in reverse from when mounting the file system. First, unmount from the client hosts that the file system was mounted on – the order of the client hosts does not matter. If necessary, force the file system to unmount by using the `-f` option.

6.9 Quotas

File system quotas are used to control all space used by the users. Every user is allocated a certain amount of space and a certain number of inodes. The user cannot exceed that number. After quotas are enabled, they can easily be adjusted as needed. Quotas are specified by ID. Each user or group has an ID that is used to set the quota for that user.

By default, the quota mount option is enabled. If the file system detects the presence of quota files in the root directory at mount time, it enables quotas. Quotas can be disabled by setting the `noquota` mount option.

6.9.1 Soft and Hard Limits

Limits can be set to keep users from exceeding an allotted amount of system resources. Soft limits can—for a small amount of time—be reached or exceeded by a users. Users, however, must quickly make sure that they drop down below the soft limit. When users exceed the soft limit, a timer is triggered. Users have a limited time to drop their usage below the soft limit. The timer is set by the administrator using the `samquota` command.

The system will—if the timer reaches zero and users have not brought their usage below the soft limit—prevent users from adding additional data or files. It will do this until the soft limit is met. The hard limit cannot be

exceeded. Even if users exceed the soft limit, they still cannot exceed the hard limit. When users do reach the hard limit, the `EDQUOT` error will be returned and their operation will not complete. Users can view information about their quotas by using the `squota` command.

Administrators must determine the disk space and the number of inodes that should be allocated to each user. A simple rule of thumb—to be certain that the group does not exceed the total space—is to divide the space allocated by the number of users in the group. For example if 100Gbs has been allocated for a group of four, each user would receive a limit of 25Gbs. In some circumstances not every user will need the full 25Gbs of allotted space. It is okay to add extra space for each user if there is certainty that not every user will require the full amount.

6.9.2 Disk Blocks and File Limits

All quotas in VSM are specified in blocks of 512-bytes. Disk space is expressed in disk allocation units (DAUs). The DAUs for a given file system may be discovered with the `samfsinfo` command. It is best to set block quotas in terms of multiples of the file system DAU for easier management. If multiples of the DAU are not used, then the user can only utilize space that is rounded down from the nearest DAU.

There are two block limits, online and total. Online blocks are DAUs allocated to the user on the disk cache, that is, all online files. Total blocks are DAUs allocated to the user on all files, whether they are online or offline. The user can exceed the number of inodes they were allocated without using any blocks if the user created all empty files. Similarly, the user could use a single inode and exceed the block quota if they created a file that was so large it consumed all data blocks in the quota.

6.9.3 Configure Quotas for a File System

Procedure

1. Determine how many users and how many groups are using the system.
2. Log in to the CIMS node as `root`.
3. Use `ssh` to log in to the MDC node.
4. Change directories to the root directory of the file system where quotas will be configured. Determine if quotas are enabled for the file system (`quota_u`, `.quota_g`, or `.quota_a` files are present).

```
tas-mdc1# cd /vsm/tasfs1
tas-mdc1# ls -la
total 7476
drwxr-xr-x 7 root root    4096 Jul 30 10:20 .
drwxr-xr-x 3 root root    4096 Feb 27 09:16 ..
drwx------ 2 root root    4096 Jul 21 14:09 .archive
-r-------- 1 root root 7536640 Jul 23 14:12 .inodes
drwx------ 2 root root    4096 Jul 21 14:09 .stage
drwx------ 2 root root   16384 Jul 21 14:09 lost+found
drwxr-xr-x 2 root root   24576 Jul 28 07:20 tas-dm1
drwxr-xr-x 2 root root   24576 Jul 28 07:21 tas-dm2
```

5. If the files do not exist, use the `dd` command to create the quota files.

   Determine the size of the `.quota_u` file based on the number of users. Determine the size of the `.quota_g` file based on the number of groups. For example, for 1024 , the size of `.quota_u` file should be (1024 * 128). For 600 groups, the size of the `.quota_g` file should be (600 * 128), rounded up the next 4k increment.

If quota files are present and if the file system is mounted with quotas disabled, the quota records become inconsistent with actual usages when blocks or files are allocated or freed. If a file system with quotas is mounted and run with quotas disabled, run the `samfsck -F` command to update the quota file usage counts before again remounting the file system with quotas enabled.

```
tas-mdc1# dd if=/dev/zero of=/vsm/tasfs1 /.quota_u bs=4096 count=32
32+0 records in
32+0 records out
131072 bytes (131 kB) copied, 0.00555971 s, 23.6 MB/s
```

```
tas-mdc1#  dd if=/dev/zero of=/vsm/tasfs1 /.quota_g bs=4096 count=19
19+0 records in
19+0 records out
77824 bytes (78 kB) copied, 0.00646215 s, 12.0 MB/s
```

```
tas-mdc1# ls -la
total 9524
drwxr-xr-x 7 root root    4096 Jul 30 10:43 .
drwxr-xr-x 3 root root    4096 Feb 27 09:16 ..
drwx------ 2 root root    4096 Jul 21 14:09 .archive
-r-------- 1 root root 7536640 Jul 23 14:12 .inodes
-rw-r--r-- 1 root root   77824 Jul 30 10:43 .quota_g
-rw-r--r-- 1 root root  131072 Jul 30 10:41 .quota_u
drwx------ 2 root root    4096 Jul 21 14:09 .stage
drwx------ 2 root root   16384 Jul 21 14:09 lost+found
drwxr-xr-x 2 root root   24576 Jul 28 07:20 tas-dm1
drwxr-xr-x 2 root root   24576 Jul 28 07:21 tas-dm2
```

6. Unmount the file system so that it can be remounted later with the quota files already created.

```
tas-mdc1# samd stopsamd
tas-mdc1# umount /vsm/tasfs1
```

7. Check the file system using the `samfsck` command. This sets up the blocks used and files used by the users and groups. Note, `samfsck -F` is used to correct quotas.

```
tas-mdc1# samfsck -F /vsm/tasfs1d
name:     tasfs1       version:     3    shared
First pass
Second pass
Third pass
samfsck: Updating quota file group, indices 0 - 8191
samfsck: Updating quota file user, indices 0 - 8191

Inodes processed: 7360

total data kilobytes       = 140627607552
total data kilobytes free  = 140627599360
total meta kilobytes       = 194836480
total meta kilobytes free  = 194808448

tas-mdc1#
```

8. Mount the file system to complete the process. The file system reads the quota files at the time of mount. If the `noquota` mount option is not preset and either or both quota files exist, quotas are enabled.

```
tas-mdc1# mount /vsm/tasfs1
```

9. Start the VSM archive daemons.

```
tas-mdc1# samd startsam
Configuring VSM
Non-autogenerated mcf file exists, leaving it alone
Starting sam-fsd:
Starting VSM archive daemons
Starting VSM sam-amld daemon
tas-mdc1#
```

6.9.4 Accounting and Infinite Quotas

When a user or group is granted an infinite quota, they are never denied access to resources on the file system. Enable infinite quotas by setting both the hard limit and the soft limit to zero. After specifying zero for the quota level, the system will track accounting information on all of the users or groups without enforcing limits. The following example illustrates how to enable an infinite quota for the group `audio`.

```
tas-mdc1# samquota -G audio -b 0: s,h -f 0:s,h /vsm/tasfs1
tas-mdc1# samquota -G audio /vsm/tasfs1

                                Online Limits                Total Limits
       Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/vsm/tasfs1
Files  group   105       298        0        0       298        0        0
Blocks group   105       314        0        0      5065        0        0
Grace period                    0s                           0s
---> Infinite quotas in effect.
```

6.9.5 Enable Default Quota Values

Using the `samquota` command, it is possible to enable default quotas for the user or group. `-U 0` sets the default quota for all users and `-G 0` sets the default quota for all groups. The following example illustrates how to enable default quotas:

```
tas-mdc1# samquota -U 0 -b 14000:s -b 16000:h -b 14G:s:t -b 17G:h:t -f 2000:s -f 2400:h -t 2w /vsm/
tasfs1d

-b limit:s   --- sets the soft online block limit - Set to 14000
-b limit:h   --- sets the hard online block limit - Set to 16000
-b limit:s:t --- sets the soft total block limit - Set to 14GB
-b limit:h:t --- sets the hard total block limit - Set to 17GB
-f limit:s   --- sets the soft file limit - Set to 2000 files
-f limit:h   --- sets the hard file limit - Set to 2400 files
-t_limit_    --- sets the timer grace period - Set to two weeks
```

6.9.6 Enable or Change Quota limits

It is best to enable default quotas for one user, then use that file as a template to enable quotas for the other users. The following example illustrates how to edit an existing quota file to create a template. Use the `-e` option and either the `-U` for user or `-G` for group option.

```
tas-mdc1# samquota -G group  -e /vsm/tasfs1 > /tmp/quota_group_template
tas-mdc1# cat /tmp/quota_group_template
# Type  ID
#               Online Limits                          Total   Limits
#             soft            hard                   soft            hard
# Files
# Blocks
# Grace Periods
#
samquota -G group \
      -f      550:s:o -f      300:h:o         -f      600:s:t -f      300:h:t \
      -b    55000:s:o -b    60000:h:o         -b 40000000:s:t -b  6000000:h:t \
                -t  1w:o                                -t 1w:t   /vsm/tasfs1
```

The first line lists the file limit, the second line lists the blocks, and the third line sets the grace period for the timer if the soft limit is breached. After retrieving the quota group file template:

1. Edit it to change the group
2. Save the file

3. Exit the editor

Next, apply the changes by using the shell to execute the file.

```
tas-mdc1# sh -x /tmp/quota_group_template
```

The `-x` option directs the shell to echo the commands it executes. You can omit the `-x` option if desired.

6.9.7 Check Quotas Using `samquota`

Check quotas using the `samquota` command. `samquota` will generate a report on any specified user or group.

```
tas-mdc1# samquota -U user filesystem
tas-mdc1# samquota -G group filesystem
```

These commands display the quotas in effect for a specific user or group. Replace *`user`* or *`group`* with the numeric ID or the name of the user or group to be viewed.

The *`filesystem`* should be the file system for that user or group. It is also possible to use the name of any file on the file system. Generally, this file will be the name of the root directory of the file system.

The following example shows that a user is within their quota limits.

A *`user`* within their quota limits

```
tas-mdc1# samquota -U user /vsm/tasfs1
                                Online Limits                Total Limits
        Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/vsm/tasfs1
Files  User    198700     15      300      600        15      300      600
Blocks User    198700    200     3000     5000       250     4000     5000
Grace period                 0s                          0s
```

The following example shows that *`user`* has exceeded quota limits.

```
tas-mdc1# samquota -U user /vsm/tasfs1
                                Online Limits                Total Limits
        Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/vsm/tasfs1
Files  User    198702     56      300      600        56      300      600
Blocks User    198702   3100*    3000     5000     35000    40000    50000
Grace period                 0s                          0s
```

The plus sign (+) is used when the soft limit is being exceeded. It will be enforced after the grace period. In this example, the plus size was replaced with the asterisk (*). This means the one week grace period was exhausted and online soft limits are under enforcement.

6.9.8 Remove or Change Quotas

First, retrieve the quota for a specific user or group. If the user or group needs more time to manage their account to lower their disk usage, change the soft limit grace period using the `samquota` command.

```
tas-mdc1# samquota -U userid -t interval filesystem
tas-mdc1# samquota -G groupid -t interval filesystem
```

`userid` Numeric ID or user name for the desired user.

groupid Numeric ID or group name for desired group.

interval The grace period. To change the duration, specify the new time interval. By default, the time is set to seconds (s) but it can be weeks (w), days (d), hours (h), or minutes (m).

filesystem The name of the file system or any single file on the file system.

6.10 Archive Daemons and Processes

The `sam-fsd` daemon is the master daemon. It is the parent for all other VSM daemons. When the system is first started, the `sam-fsd` daemon runs automatically. When running any VSM file system, the `sam-fsd` daemon is always active. If the `sam-fsd` is not running, it can be restarted with the `samd config` command. If a change is made to any configuration file, the `samd config` command must be executed. This causes `sam-fsd` to read the configuration files.

Each time a shared file system is mounted a `sam-sharefsd` starts and remains active. The `sam-sharefsd` daemon's parent is `sam-fsd`. The `samfsd` starts a new `sam-sharefsd` daemon each time a shared file system is mounted.

The system uses TCP sockets between the server and the host to communicate. A listener socket is opened on the metadata server on TCP port 7105 `sam-qfs`. The `sam-qfs` port entry is added to the `/etc/services` file during installation. Do not remove this port entry as it is needed by the file system to function properly. The following list describes the daemons and processes that can run on a VSM system.

sam-amld This daemon initializes the automated daemons: `sam-catserverd`, `sam-scannerd`, `sam-robotsd`

sam-catserverd This daemon tracks all media in the libraries cataloged in the file system

sam-robotsd This daemon starts the automated media changer and monitors the control daemons

sam-scannerd This daemon monitors all the manually mounted media devices, the scanner checks each device for inserted archive media cartridges

sam-archiverd This daemon controls all archiving, this daemon is the parent of `sam-arfind` and `sam-arcopy`

sam-fsd This is the master daemon

sam-rftd This daemon transfers data between multiple host systems

sam-sharefsd This daemon controls a shared file system

sam-releaser This is an important process that releases disk space that is held by archived files on the file system

sam-stagealld This daemon controls all associative staging on the file system

sam-stagerd	This daemon controls all staging of the file system
sam-rpcd	This daemon controls the remote procedure call (RPC) application programming interface (API) server processes
	The `samd` command is used to control the daemons and to read, check, and reset the following configuration files:
samd config	Read, check, and reset the configuration files
samd buildmcf	Builds the MCF from an existing file system
samd start	Start up the robotic daemon, `sam-amld`
samd stop	Stop the robotic daemon, `sam-amld`
samd startsam	Start up the archiver and stager daemons
samd stopsam	Stop the archiver and stager daemons
samd umount *family_set*	Stop the archiver and stager on this *`family_set`* and `umount` the file system
samd reload	Reload the VSM kernel module and restart `sam-fsd`
samd unload	Stop `sam-fsd` and unload the VSM kernel module

6.11 Trace Files

By default trace files are disabled. Trace files may be enabled by editing the `defaults.conf` file. Trace files are not necessary to run the file system. They are primarily used by Cray support for debugging purposes. Messages are written to trace files. These files contain information pertaining to the task performed by the different daemons.

Trace files are written to the `/var/opt/vsm/trace` directory. Each trace file is named for the specific daemon from which the trace file was derived (for example, `sam-archiverd`, `sam-fsd`, `sam-rftd`, etc...). This can be changed in the `defaults.conf` file.

Trace files contain information about each daemon. They hold the time and source and event from which they were created. There are default events. Special events can be created by using the directives in the `defaults.conf` file. The default files contain the program name, the process ID, and the time. This cannot be changed. More information can be included such as the date, the source file, and the event type. See man `defaults.conf` for a list of the default events and how to change or add events.

VSM prevents trace files from growing too large and consuming too many resources. The `sam-fsd` daemon monitors the size of the trace files. It also periodically executes the `/opt/vsm/sbin/trace_rotate` command. This script moves the trace files into sequentially numbered files. The script can easily be modified to accommodate site policies.

6.12 Operator Utility samu

Use the operator utility to monitor and control the TAS HSM VSM file system and archiving daemons. Start `samu` by entering the `samu` command from the MDC node. The default help screen displays, then type the letter assigned to the various menus listed below. Enter `Ctrl-f` to page through all of the `samu` command menus and displays.

```
Help information            page 1/15   samu vsm-1.1.7-0 18:30:40 Oct 9 2015

Displays:
    a   Archiver status                 v       Robot catalog
    c   Device configuration            w       Pending stage queue
    d   Daemon trace controls           C       Memory
    f   File systems                    D       Disk volume dictionary
    g   Shared clients                  I       Inode
    h   Help information                J       Preview shared memory
    l   Usage information               L       Shared memory tables
    m   Mass storage status             M       Shared memory
    n   Staging status                  N       File system parameters
    p   Removable media load requests   P       Active Services
    r   Removable media                 S       Sector data
    s   Device status                   T       SCSI sense data
    t   Tape drive status               U       Device table
    u   Staging queue

    more (Ctrl-f)

Versity Software Inc                                  VSM on tas-mdc1
```

6.12.1 Keyboard Shortcuts

To navigate in the `samu` utility, enter:

Ctrl-B	Return to the previous page
Ctrl-F	Go to the next page
Ctrl-D	Move 1/2 page forward
Ctrl-U	Moves 1/2 page backward
Ctrl-K	Moves the display format forward

Use the following keyboard shortcuts to control `samu`:

q	Quit `samu`
:	Enter the command mode
:h	Display the help screen
:q	Exit `samu`
Space bar	Refresh the display

Ctrl-f	Displays next page
Ctrl-b	Displays previous page
Ctrl-d	Displays 1/2 of next page
Ctrl-u	Displays 1/2 of previous page
Ctrl-l	Shows more detail in selected displays
Ctrl-k	Shows advanced display format
Ctrl-l	Clears the display
Ctrl-r	Toggle refresh
/	Search for volume serial name (VSN)
%	Search for barcode
$	Search for slot

6.12.2 Archiver Status Display (a)

The following example illustrates the use of the `:a` command and calls for the status of the archiver on the file system. Enter the name of the file system after the `a` command.

```
Archiver status                          samu vsm-1.1.7-0 18:30:40 Oct 9 2015

Archiver Status
samu 5.0 18:30:40 Mar 28 2014
sam-archiverd: Waiting for resources
sam-arfind: tasfs1 mounted at /vsm
Waiting until 2014-05-08 07:54:02 to scan .inodes
sam-arfind: tasfs2 mounted at /vsm
Waiting until 2014-05-08 07:52:57 to scan .inodes
sam-arfind: tasfs3 mounted at /qfs1
Waiting until 2014-05-08 07:44:33 to scan .inodes
. . .

Versity Software Inc                                  VSM on tas-mdc1
```

6.12.3 The Device Configuration Display (c)

The `:c` command displays all devices connected to the file system and their corresponding equipment numbers.

```
c
```

```
Device configuration:                    samu vsm-1.1.7-0 18:30:40 Oct 9 2015

ty   eq state   device_name                                        fs  family_set
sp  100 on      /dev/tape/by-id/scsi-1SPECTRA_PYTHON_9110004FC9  100 T200
li  101 on      /dev/tape/by-id/scsi-321110090a5004fc9-nst       100 T200
li  102 on      /dev/tape/by-id/scsi-321120090a5004fc9-nst       100 T200
```

```
li  103 on      /dev/tape/by-id/scsi-321130090a5004fc9-nst       100 T200
li  104 on      /dev/tape/by-id/scsi-321140090a5004fc9-nst       100 T200
li  105 on      /dev/tape/by-id/scsi-321210090a5004fc9-nst       100 T200
li  106 on      /dev/tape/by-id/scsi-321220090a5004fc9-nst       100 T200
li  107 on      /dev/tape/by-id/scsi-321230090a5004fc9-nst       100 T200
li  108 on      /dev/tape/by-id/scsi-321240090a5004fc9-nst       100 T200
hy  109 on      historian                                       109

Versity Software Inc                                VSM on tas-mdc1
```

Table 14. Device Configuration Display Field Descriptions

Field	Description
ty	Defines the type of device
eq	Displays the equipment number for each connected device
state	Displays the current state of each connected device
device_name	Displays the name of the device used in its path
fs	Displays the family set equipment number
family_set	Displays name of the family set or the library of which the device belongs

The device state field displays one of the following states:

Table 15. Device State Field Descriptions

Field	Description
on	The device is on and can be accessed.
off	The device is off and cannot be accessed.
ro	The device is on but only available on read-only access.
down	The device can only be accessed for maintenance.
idle	Displays the family set equipment number.
nalloc	The nalloc flag has been set and no new allocations can be made on this device.

6.12.4 Daemon Trace Controls Display (d)

The :d command displays all events and information for the daemons being traced as specified in the defaults.conf file. The information contains the events, the size, the age, and the paths to the events being traced.

```
Daemon trace controls                    samu vsm-1.1.7-0 18:30:40 Oct 9 2015

sam-amld       /var/opt/vsm/trace/sam-amld
               cust err fatal misc proc date
               size    10M  age 0

sam-archiverd  /var/opt/vsm/trace/sam-archiverd
```

```
               cust err fatal misc proc date
               size    10M  age 0

sam-catserverd /var/opt/vsm/trace/sam-catserverd
               cust err fatal misc proc date
               size    10M  age 0

sam-fsd        /var/opt/vsm/trace/sam-fsd
               cust err fatal misc proc date
               size    10M  age 0

sam-rftd       /var/opt/vsm/trace/sam-rftd
               cust err fatal misc proc date
               size    10M  age 0

sam-recycler   /var/opt/vsm/trace/sam-recycler
               cust err fatal misc proc date
               size    10M  age 0

sam-sharefsd   /var/opt/vsm/trace/sam-sharefsd
               cust err fatal misc proc date
               size    10M  age 0

sam-stagerd    /var/opt/vsm/trace/sam-stagerd
               cust err fatal misc proc date
               size    10M  age 0

sam-shrink     /var/opt/vsm/trace/sam-shrink
               cust err fatal misc proc date
               size    10M  age 0

Versity Software Inc                                VSM on tas-mdc1
```

6.12.5 File Systems and Archive Parameters Display (f)

From `samu`, enter `:f` to display the components of the file system and equipment numbers (`eq` column).

```
File systems                              samu vsm-1.1.7-0 18:30:40 Oct 9 2015


ty       eq   state          device_name      status high low mountpoint server
 ma      20      on              tasfs1  m----3c--r- 80% 70% /vsm/tasfs1 tas-mdc1
 mm      21      on      /dev/mapper/md0
 mr      22      on   /dev/mapper/dc0000
 mr      23      on   /dev/mapper/dc0001
 mr      24      on   /dev/mapper/dc0002
 mr      25      on   /dev/mapper/dc0003
 mr      26      on   /dev/mapper/dc0004
 mr      27      on   /dev/mapper/dc0005


Versity Software Inc                                VSM on tas-mdc1
```

Table 16. File Systems Display Field Descriptions

Field	Description
`ty`	The type of device

Field	Description
eq	The equipment number
state	Current operating state of the device. Valid device states are: • on The device is on, and the disk or tape is loaded in the transport available for access • off The device is not available for access • ro The device is on, but read only • down The device is available only for maintenance access • idle The device is not available for new connections. Operations in progress continue until completion • nalloc The nalloc flag has been set, which prohibits any future allocation to this device
high	The threshold percentage of the high disk use
low	The threshold percentage of the low disk use
mountpoint	The path name where the file system is mounted
device_name	The name of the device
status	The device status code
sever	The hostname that the shared file system is mounted on

Device status code field descriptions are listed in the following table:

Table 17. Device Status Code Field Descriptions

Field	Description
s---------	Media is currently being scanned
m---------	Automated library is operational
M---------	Maintenance mode
-E--------	The device received an unrecoverable error
-a--------	The device is in audit mode
--l-------	Media is labeled.
--L-------	Media is currently being labeled
--N-------	Foreign media
---I------	Waiting for the device to idle
---A------	Device requires attention
----C-----	Device requires cleaning
----U-----	Unloading requested
-----R----	The device has been reserved

Field	Description
------w---	A process is busy writing to the media
-------o--	The device is now open
--------P-	The device is positioning
--------F-	Media is full
---------R	The media is read only
---------r	The device is spun up and ready
---------p	The device is present
---------W	The device is write protected

The file system and archiving parameter commands are listed in the following table. Specify the equipment number for *eq* in the table.

Table 18. Archiving Parameter Command Descriptions

Command Parameter	Description
:idle *eq*	Stops all activity on the device
:off *eq*	Turns off the device
:on *eq*	Turns on the device
:unavail *eq*	Selects device *eq* and makes it unavailable for use with the file system (Users might, for example, set a drive state to unavail in a disaster-recovery situation where they are trying to load media to restore a file system and they do not want the VSM software to attempt to use this drive)
:unload *eq*	Unloads specified mounted media device *eq*
:ackmsg *eq*	Acknowledges and clears the critical message in the device display
:noalloc *eq*	Stops new allocations for the specified device
:alloc *eq*	Reverses the noalloc command and allows allocation to device
:rdlease *eq interval*	Regulates the allowable time for the read leases (for *interval*, specify an integer number between 15 and 600 seconds)
:wrlease *eq interval*	Regulates the allowable time for the write leases (for *interval*, specify an integer number between 15 and 600 seconds)
:aplease *eq interval*	Regulates the allowable time for append leases (for *interval*, specify an integer number between 15 and 600 seconds)
:mm_stripe *eq value*	Regulates the disk allocation unit (DAU) size for metadata stripe width
:suid *eq*	Allows running programs to change their owner IDs
:nosuid *eq*	Prevents running programs from changing their owner IDs

Command Parameter	Description
:stripe *eq value*	Changes the stripe width for the file system to the number of DAUs specified
:sync_meta *eq value*	Regulates the time when metadata is written to disk
:trace *eq*	Enables the trace feature for the file system
:notrace *eq*	Disables the trace feature for the file system
:clear *eq*	Clears the removable media mount display of the specified volume
:devlog *eq option*	Logs one or multiple events
:diskvols *volume* +*flag* | -*flag*	Regulates all flags that are placed into the disk volume dictionary
:fs *eq*	Allows for a new default file system
:mount *eq*	Specifies the mount system
:open *eq*	Opens the disk device for access
:read *hex_addr*	Reads a specific 1K sector of the disk device
:refresh *n*	Regulates the screen refresh time for the samu. Specify *n* in seconds
:snap *filename*	Copies the operator display to a file (the default file name is snapshots)

6.12.6 Shared Clients Display (g)

The :g command lists all of the hosts (shared clients) in a shared file system along with their status.

```
Shared clients                              samu vsm-1.1.7-0 18:30:40 Oct 9 2015
tasfs1 is shared, server is tas-mdc1, 2 clients 4 max
ord hostname  seqno     nomsgs  status  config   conf1  flags
  1 tas-mdc1  14212198  0       91      838740d  31     0     MNT SVR
  2 tas-mdc2  12343     0       a1      838740d  21     20    MNT CLI BLK
  3           34007066  0       e0      838740c  21     0         CLI
  4           33968245  0       e0      838740c  21     0         CLI
Versity Software Inc                                 VSM on tas-mdc1
```

6.12.7 Help Information Display (h)

The :h command displays the main help screen. Enter Ctrl-f to page through all of the help screens and display information about all the commands and parameters.

```
Help information            page 1/15   samu vsm-1.1.7-0 18:30:40 Oct 9 2015

Displays:

    a   Archiver status                 v       Robot catalog
    c   Device configuration            w       Pending stage queue
    d   Daemon trace controls           C       Memory
    f   File systems                    D       Disk volume dictionary
    g   Shared clients                  I       Inode
    h   Help information                J       Preview shared memory
```

```
    l   Usage information                L       Shared memory tables
    m   Mass storage status              M       Shared memory
    n   Staging status                   N       File system parameters
    p   Removable media load requests    P       Active Services
    r   Removable media                  S       Sector data
    s   Device status                    T       SCSI sense data
    t   Tape drive status                U       Device table
    u   Staging queue

    more (Ctrl-f)

Versity Software Inc                                VSM on tas-mdc1
```

6.12.8 Usage Information Display (l)

The `:l` command shows the usage information for the file system, including the capacity and space used for each library and file system.

```
Usage information                          samu vsm-1.1.7-0 18:30:40 Oct 9 2015


uuid: 4C4C4544-0032-5910-804E-B6C04F395A31  OS name: tas-mdc1
Architecture: x86_64 CPUs: 16 (16 online)
License: Features: 20000000  Mon Jun  1 00:00:00 2015

library       100: capacity   80.5T bytes space   31.3T bytes, usage  61%
library    totals: capacity   80.5T bytes space   31.3T bytes, usage  61%

filesystem tasfs1: capacity  131.0T bytes space  131.0T bytes, usage   1% server
filesystem totals: capacity  131.0T bytes space  131.0T bytes, usage   1%

Versity Software Inc                                VSM on tas-mdc1
```

6.12.9 Mass Storage Status Display (m)

The `:m` command shows the status of mass storage file systems and their member drives. This display shows only mounted file systems. Member drives are indented one space and appear directly below the file system to which they belong.

```
Mass storage status                        samu vsm-1.1.7-0 18:30:40 Oct 9 2015

 ty      eq  status       use state  ord  capacity      free    ra  part high low
 ma      20  m----3c--r-  1%  on          130.970T  130.969T    1M    16  80% 70%
 mm      21               1%  on       0  185.811G  185.783G [194807216 inodes]
 mr      22               1%  on       1   21.828T   21.828T
 mr      23               1%  on       2   21.828T   21.828T
 mr      24               1%  on       3   21.828T   21.828T
 mr      25               1%  on       4   21.828T   21.828T
 mr      26               1%  on       5   21.828T   21.828T
 mr      27               1%  on       6   21.828T   21.828T

saVersity Software Inc                              VSM on tas-mdc1
```

6.12.10 Staging Status Display

The `:n` command shows the status of the stager for all media and a list of outstanding stage requests. To display the staging status for a specific media type enter:

```
Staging status                          samu vsm-1.1.7-0 18:30:40 Oct 9 2015

Log output to: /var/log/vsm/stager.log
Stage request 1: li.064200
 Resources not available VSN 064200

Stage request 2: li.053652
 Loading VSN 053652

Stage request 3: li.001005
 Positioning for file /vsm1/data3/dir223/file803

Staging queues starting at 1
ty pid   user        status       wait   files  vsn
li 2795  root        pending     15:43   2076   064200
li 28627 root        active       0:09      1   053652
li 17833 root        active       0:02      2   001005

Versity Software Inc                              VSM on tas-mdc1
```

6.12.11 Removable Media Load Requests Display

The `:p` command lists information about pending load requests for removable media. You can use the *`mt`* argument to select either a specific type of media, such as DLT tape, or a family of media, such as tape. The priority display lists the priority in the preview queue, rather than the user queue, and sorts the entries by priority. For *`mt`*, specify an equipment type listed in the `mcf` man page (for example, specify `s9` for StorageTek™ 97xx series libraries).

The display shows mount requests in the following formats:

- Both manual and automated library requests by user
- Both manual and automated library requests by priority
- Manual requests only
- Automated library requests only

```
Removable media load requests all both samu vsm-1.1.7-0 18:30:40 Oct 9 2015
                                                 count: 1
index type pid    user        rb      flags        wait count   vsn
   0  li   28971  root        50      ---f---      0:09         053652


Versity Software Inc VSM on exp-mdc1
--------------------------------------------------------------------------------
-------------
```

6.12.12 Removable Media Display

The `:r` command monitors the activity on removable media devices such as tape drives. You can monitor either a specific type of device, such as video tape, or a family of devices such as all tape devices. For *`eq`*, specify the equipment number for the device. To display the status for a specific device:

```
----------------------------------------------------------------------------------------------------
Removable media status: all                samu vsm-1.1.7-0 18:30:40 Oct 9 2015
ty eq     status       act use state    vsn
li 101    ---------p   0   0%  notrdy   empty
li 102    ---------p   0   0%  notrdy   empty
li 103    ---------p   0   0%  notrdy   empty
li 104    ---------p   0   0%  notrdy   empty

Versity Software Inc VSM on exp-mdc1
----------------------------------------------------------------------------------------------------
```

Table 19. Device Status Field Descriptions

Field	Description
`ty`	The type of device
`eq`	The equipment number
`status`	The device status code
`act`	Activity count
`use`	Percentage of cartridge space used
`state`	Current operating state of the removable media—valid device states are: • `ready` - The device is on, and the disk or tape is loaded in the transport available for access • `notrdy` - The device is on, but no disk or tape is present in the transport • `idle` - The device is not available for new connections. Operations in progress continue until completion • `off` - The device is not available for access • `down` - The device is available only for maintenance access • `nalloc` - The `nalloc` flag has been set, which prohibits any future allocation to this device
`vsn`	Volume serial name assigned to the volume, or the keyword `nolabel` if the volume is not labeled, blank if no volume is present in the transport or device is `off`

6.12.13 Tape Drive Status Display

The `:t` command shows the status of all tape drives configured within the environment.

```
Tape drive status                      samu vsm-1.1.7-0 18:30:40 Oct 9 2015


ty   eq  status      act  use  state          vsn
li  101  ---------p    0   0%  notrdy        empty
li  102  ---------p    0   0%  notrdy        empty
li  103  ---------p    0   0%  notrdy        empty
li  104  ---------p    0   0%  notrdy        empty
li  105  ---------p    0   0%  notrdy        empty
```

```
li  106  ---------p    0   0%  notrdy        empty
li  107  ---------p    0   0%  notrdy        empty
li  108  ---------p    0   0%  notrdy        empty

Versity Software Inc                              VSM on tas-mdc1
```

6.12.14 Stage Queue Display

The `:u` command lists all files in the staging queue.

```
Staging queue by media type: all        samu vsm-1.1.7-0 18:30:40 Oct 9 2015
    volumes 1 files 22
ty     length  fseq        ino      position      offset  vsn
dt   451.611k    20       1030         207cc         473  DAT001
dt   341.676k    20       1031         207cc         7fc  DAT001
dt   419.861k    20       1032         207cc         aa9  DAT001
dt   384.760k    20       1033         207cc         df2  DAT001
dt   263.475k    20       1034         207cc       10f5  DAT001
dt   452.901k    20       1035         207cc        1305  DAT001
dt   404.598k    20       1036         207cc        1690  DAT001
dt   292.454k    20       1037         207cc        19bb  DAT001
dt   257.835k    20       1038         207cc        1c05  DAT001
dt   399.882k    20       1040         207cc        1e0b  DAT001
. . .

Versity Software Inc                              VSM on tas-mdc1
```

Table 20. Staging Queue Display Field Descriptions

Field	Description
`ty`	The type of device
`length`	File length
`fseq`	File system equipment number
`ino`	The position of the archive file on the specific medium
`offset`	Offset of the archive file on the specific medium
`vsn`	Volume serial name of the volume

6.12.15 Robot Catalog Display (v)

The `:v` command shows the location and volume serial name (VSN) of all disks or tapes currently catalogued in the automated library. For *eq*, specify the equipment number of the device. Type the keyword historian to view the historian catalog. To display catalog information for a specific device:

```
Robot VSN catalog by slot       : eq 100   samu vsm-1.1.7-0 18:30:40 Oct 9 2015
                                                              count 64
slot          access time count use flags         ty vsn

   0     2014/07/21 12:01    1  66% -il-o-b--c--  li 044215
   1     2014/07/30 10:44    2 100% -il-o-b----f  li 192432
   2     2014/07/30 10:46    2 100% -il-o-b----f  li 192433
   3     2014/07/30 10:47    2  36% -il-o-b-----  li 192434
```

```
   4     2014/07/21 12:02    1  43% -il-o-b-----  li 192435
   5     2014/07/21 12:02    1  43% -il-o-b-----  li 192436
   6     none               50   0% -il-oCb-----  li CLNWC6
   7
  10     2014/07/21 12:02    1  66% -il-o-b--c--  li 044219
  11     2014/07/30 10:44    4   0% -il-o-b-----  li 044218
  12     none                0   0% -il-o-b-----  li 044217
  13
  15     2014/07/21 10:43    1  66% -il-o-b--c--  li 044214
. . .
```

Table 21. Robot Catalog Display Field Descriptions

Field	Description
`slot`	Slot number within the specified library
`access time`	Time the volume was last accessed
`count`	Number of accesses to this volume since the last audit operation
`use`	Percentage of space used for the volume
`flags`	Flags for the device
`ty`	Device type
`vsn`	Volume serial name of the volume

Table 22. Robot Catalog Flag Field Descriptions

Field	Description
`A----------`	Volume needs audit
`-i----------`	Time the volume was last accessed
`--l--------`	Labeled. Overrides `N`
`---N--------`	Unlabeled—this volume is foreign to the environment
`----E-------`	Media error—set when the software detects a write error on a cartridge
`----o-------`	Slot occupied
`-----C------`	Volume is a cleaning tape. Overrides `p`
`-----p------`	Priority volume serial name (VSN)
`------b-----`	Barcode detected
`--------W---`	Write protect (set when the physical write protection mechanism is enabled on a cartridge)
`---------R--`	Read only
`---------c--`	Recycle
`----------d-`	Duplicate volume serial name (VSN)—overrides `U`
`----------U-`	Volume unavailable
`-----------f`	Archiver found volume full

Field	Description
`----------X`	Export slot

6.12.16 Pend Stage Queue Display

The `:w` command displays queued stage requests for volumes that have not loaded. To display the pending stage queue for a specific media type:

```
Pending stage queue by media type: all  samu vsm-1.1.7-0 18:30:40 Oct 9 2015
                                                 volumes 0 files 0

ty      length  fseq        ino   position     offset vsn
at      1.383M     1         42      3a786       271b 000002
at      1.479M     1         56      3a786       5139 000002
at   1018.406k     1         60      3a786       6550 000002
at      1.000M     1         65      3a786       7475 000002
at      1.528M     1         80      3a786       99be 000002

Versity Software Inc                                VSM on tas-mdc1
```

Table 23. Pending Stage Queue Display Field Descriptions

Field	Description
`ty`	Device type
`length`	File length
`fseq`	File system equipment number
`ino`	The inode number
`position`	The position (in decimal format) of the archive file on the specific medium
`offset`	Offset of the archive file on the specific medium
`vsn`	Volume serial name of the volume

6.12.17 Memory Display

The `C` (uppercase C) command is used for debugging. This display shows information about a specific memory address. Enter the hexadecimal number of the memory address to view the memory address:

Type `:C` *`hex_addr`*.

6.12.18 Disk Volume Dictionary Display

The disk volume dictionary tracks the disk media used for archiving. The `diskvols.conf` file contains information about the volume serial name (VSN)—such as its capacity, status flag, and how much space has already been used. If it is necessary to clear or change the dictionary flag, use the `samu diskvols` command. The flags include `l`, `read only`, and `bad media`.

```
Disk volume dictionary                    samu vsm-1.1.7-0 18:30:40 Oct 9 2015

header
version 460

volumes
magic 340322 version 9 nkeys 2 ndata 2
index         space          capacity         used       flags    volume
    0         12882411520    12887785472      10291200   -----    diskar01
    1          6443827200     6443892736         70656   -----    diskar02

clients
magic 340322 version 9 nkeys 0 ndata 0


Versity Software Inc                                  VSM on tas-mdc1
```

Table 24. Disk Volume Dictionary Flag Descriptions

Flag	Description
`l`	Volume is labeled—*`seqnum`* file has been created (this is set by the administrator to prevent the software from creating a new *`seqnum`* file)
`r`	Volume is defined on a remote host
`U`	Volume is unavailable
`R`	Volume is read only
`E`	Media error, indicating the software detects a write error on the disk archive directory

6.12.19 Inode Display

The inode display shows information on the running inodes. The inode display is for debugging the system. It should only be used by Cray Service. The `samu` utility prompts the user for the mount point.

The `:I` command shows the content of inodes.

To display inodes for an entire *`filesystem`*:

To display a specific inode:

Specify *`inode-number`* in either hexadecimal or decimal.

```
Inode     0x1 (1) format: file            samu vsm-1.1.7-0 18:30:40 Oct 9 2015
   incore: y

00008100 mode       -r--------            53da473b access_time
00000001 ino        (1)                   0bcf77d0
00000001 gen        (1)                   53d00922 modify_time
00000002 parent.ino (2)                   20211130
00000002 parent.gen (2)                   53d00922 change_time
00000000 size_u                           20211130
00730000 size_l     (7536640)             53cd6559 creation_time
01000000 rm:media/flags                   00000000 attribute_time
00000000 rm:file_offset                   53d93102 residence_time
00000000 rm:mau                           00000000 unit/cs/arch/flg
```

```
00000000 rm:position                        00000000 ar_flags
00000000 ext_attrs  --------                00000000 stripe/stride/sg
00000000 ext.ino    (0)                     00000000 media  -- --
00000000 ext.gen    (0)                     00000000 media  -- --
00000000 uid        root                    00000000 psize      (0)
00000000 gid        root                    00000730 blocks     (1840)
00000001 nlink      (1)                     00001cc0 free_ino   (7360)
00010800 status ---------  ----- -- --      00000000 stage ahd  (0)
00000000 obty/-/-/p2flg                     00000000 xattr.ino    (0)
00000000 projid                             00000000 xattr.gen    (0)

Extents (4k displayed as 1k):
00_ 00000004810.00 00000004820.00 00000004830.00 00000004840.00 00000004850.00
05_ 00000004860.00 00000004870.00 00000004880.00 00000004890.00 000000048a0.00
000000048b0.00 0000000480
. . .d
```

6.12.20 Preview Shared Memory Display

The `:J` command displays the shared memory segment for the preview queue. This display is designed for debugging and is intended to be used only by Cray technical support.

```
Preview shared memory   size: 184320    samu vsm-1.1.7-0 18:30:40 Oct 9 2015
00000000  00000000 00000000 00000000 00000000  ................
00000010  80000000 00000000 00000000 00000000  ................
00000020  00000000 00000000 73616d66 73202d20  ........samfs -
00000030  70726576 69657720 6d656d6f 72792073  preview memory s
00000040  65676d65 6e740000 00d00200 00000000  egment...P......
00000050  ffcf0200 00000000 00000000 00000000  .O..............
00000060  00000000 00000000 80000000 00000000  ................
00000070  00000000 00000000 00000000 00000000  ................

00000080  40eb0000 64000000 00000000 00000000  @k..d...........
00000090  0000803f 00007a44 08eb0000 00000000  ...?..zD.k......
000000a0  01000000 00000000 00000000 00000000  ................
000000b0  00000000 00000000 80000000 00000000  ................
000000c0  00000000 00000000 00000000 00000000  ................
000000d0  00000000 00000000 00000000 00000000  ................
000000e0  00000000 00000000 00000000 00000000  ................
000000f0  00000000 00000000 00000000 00000000  ................

00000100  00000000 00000000 00000000 00000000  ................
00000110  00000000 00000000 00000000 00000000  ................
00000120  00000000 00000000 00000000 00000000  ................


Versity Software Inc                                VSM on tas-mdc1
```

6.12.21 File System Parameters

The `:N` command displays all mount point parameters, the superblock version, and other file system information.

```
File system parameters                  samu vsm-1.1.7-0 18:30:40 Oct 9 2015

mount_point    : /vsm/tasfs1            partial        : 16k
server         : tas-mdc1                maxpartial     : 16k
filesystem name: tasfs1                 partial_stage  : 16384
```

```
eq  type        : 20 ma                 flush_behind    : 0
state version   :    0    3             stage_flush_beh: 0
(fs,mm)_count   :    7    1             stage_n_window : 8388608
sync_meta       : 1                     stage_retries   : 3
atime           : default               stage timeout   : 0
stripe          : 0                     dio_consec r,w :    0    0
mm_stripe       : 1                     dio_frm_min r,w:  256  256
high  low       :  80%  70%             dio_ill_min r,w:    0    0
readahead       : 1048576               ext_bsize       : 4096
writebehind     : 524288                def_retention   : 43200
wr_throttle     : 2697754624            minallocsz      : 8388608
rd_ino_buf_size: 4096                   maxallocsz      : 134217728
wr_ino_buf_size: 4096                   min_pool        : 64
maxphys         : a0cc7800              meta_timeo      : 3
                                        lease_timeo     : 0
                                        (rd,wr,ap)lease:   30   30   30
config          : 0x0838740d            config1         : 0x00000031
status          : 0x00000091            mflag           : 0x00000000


Device configuration:
ty   eq state   device_name                       fs family_set
mm   21 on      /dev/mapper/md0                  20 tasfs1
mr   22 on      /dev/mapper/dc0000               20 tasfs1
mr   23 on      /dev/mapper/dc0001               20 tasfs1
mr   24 on      /dev/mapper/dc0002               20 tasfs1
mr   25 on      /dev/mapper/dc0003               20 tasfs1
mr   26 on      /dev/mapper/dc0004               20 tasfs1
mr   27 on      /dev/mapper/dc0005               20 tasfs1
```

6.12.22 Active Services (P)

The `:P` command displays the services registered with the VSM single-port multiplexer.

```
Active Services                         samu vsm-1.1.7-0 18:30:40 Oct 9 2015


Registered services for host 'tas-mdc1':
    sharedfs.tasfs1
  1 service(s) registered.


Versity Software Inc                                VSM on tas-mdc1
```

6.12.23 Sector Data Display

The `:S` command displays the SCSI status of a SCSI device. This display is designed for debugging and is intended to be used only by Cray technical support.

6.12.24 SCSI Sense Data

The `:T` command displays raw device data. This display is designed for debugging and is intended to be used only by Cray technical support.

6.12.25 Device Table Display (U)

The `:U` command displays the device table in a human-readable form. This display is designed for debugging and is intended to be used only by Cray Service. To display the device table for a specific device:

```
Device table: eq: 20    addr: 00000578  samu vsm-1.1.7-0 18:30:40 Oct 9 2015

message:

0000000000000000 0000000000000000       00000000 delay
0000000000000080 mutex                  00000000 unload_delay
00000db8 next
66736174 set:  tasfs1
00003173
00000000
00000000
00140014 eq/fseq
08020802 type/equ_type
0000     state
00000000 st_rdev
00000000 ord/model
00000000 mode_sense
00000000 sense
00000000 space
00000000 capacity
00000000 active
00000000 open
00000000 sector_size
00000000 label_address
00000000 vsn:
00000000
00000000
00000000
00000000 status: ----------
00000000 dt
66736174 name: tasfs1

Versity Software Inc                                VSM on tas-mdc1
```

7 TAS Man Pages

7.1 mcp(1)

Name

`cp` - copy files and directories.

Synopsis

```
cp [OPTION]... [-T] SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
cp [OPTION]... -t DIRECTORY SOURCE...
```

Description

Copy `SOURCE` to `DEST`, or multiple `SOURCE`(s) to `DIRECTORY`.

`mcp`-specific options (defaults in brackets):

`--buffer-size=MBYTES` Read/write buffer size [4].

`--check-tree` Print hash subtrees to pinpoint corruption.

`--fadvise-read` Enable use of `posix_fadvise` during reads.

`--hash-leaf-size=KBYTES` Granularity of hash tree [1048576].

`--hash-type=TYPE` hash type [MD5], with TYPE one of:

- `md5`
- `sha1`
- `sha256`
- `sha384`
- `sha512`
- `sha224`
- `crc32`
- `crc32rfc1510`
- `crc24rfc2440`

--length=LEN	Copy LEN bytes beginning at `--offset` (or 0 if `--offset` not specified).
--listen-port=PORT	Listen on port PORT for requests from cooperating hosts
--manager-host=HOST	Host name or IP address of management thread for multi-node/multi-host copies
--manager-port=PORT	Port on which to contact management thread.
--mpi	Enable use of MPI for multi-node copies.
--no-direct-read	Disable use of direct I/O for reads.
--no-direct-write	Disable use of direct I/O for writes.
--no-double-buffer	Disable use of double buffering during file I/O.
--offset=POS	Copy `--length` bytes beginning at POS (or to end if `--length` not specified).
--password-file=FILE	File to use for passwords (will be created if does not exist).
--print-hash	Print hash of each file to stdout similar to `md5sum`, with sum of the source file computed, but destination file name printed so that `md5sum -c` can be used on the output to check that the data written to disk was what was read.
--print-stats	Print performance per file to `stderr`.
--print-stripe	Print striping changes to `stderr`.
--read-stdin	Perform a batch of operations read over `stdin` in the form 'SRC DST RANGES' where SRC and DST must be URI-escaped (RFC 3986) file names and RANGES is zero or more comma-separated ranges of the form 'START-END' for 0 <= START < END.
--skip-chmod	Retain temporary permissions used during copy.
--split-size=MBYTES	Size to split files for parallelization [1024].
--threads=NUMBER	Number of OpenMP worker threads to use [4].

Standard options (mandatory arguments to long options are mandatory for short options too):

-a, --archive	Same as `-dR`, `--preserve=all`.
--backup[=CONTROL]	Make a backup of each existing destination file.
-b	Like `--backup` but does not accept an argument.
--copy-contents	Copy contents of special files when recursive.
-d	Same as `--no-dereference`, `--preserve=links`.
-f, --force	If an existing destination file cannot be opened, remove it and try again (redundant if the `-n` option is used).
-i, --interactive	Prompt before overwrite (overrides a previous `-n` option).
-H	Follow command-line symbolic links in SOURCE.
-l, --link	Link files instead of copying.
-L, --dereference	Always follow symbolic links in SOURCE.

-n, --no-clobber	Do not overwrite an existing file (overrides a previous `-i` option).
-P, --no-dereference	Never follow symbolic links in SOURCE.
-p	Same as `--preserve=mode`, ownership, timestamps.
--preserve[=ATTR_LIST]	Preserve the specified attributes (default: mode, ownership, timestamps), if possible additional attributes: context, links, xattr, all.
--no-preserve=ATTR_LIST	Don't preserve the specified attributes.
--parents	Use full source file name under DIRECTORY.
-R, -r, --recursive	Copy directories recursively.
--reflink[=WHEN]	Control clone/CoW copies. See below.
--remove-destination	Remove each existing destination file before attempting to open it (contrast with `--force`).
--sparse=WHEN	Control creation of sparse files. See below.
--strip-trailing-slashes	Remove any trailing slashes from each SOURCE argument
-s, --symbolic-link	Make symbolic links instead of copying.
-S, --suffix=SUFFIX	Override the usual backup suffix.
-t, --target-directory=DIRECTORY	copy all SOURCE arguments into DIRECTORY
-T, --no-target-directory	Treat DEST as a normal file.
-u, --update	Copy only when the SOURCE file is newer than the destination file or when the destination file is missing.
-v, --verbose	Explain what is being done.
-x, --one-file-system	Stay on this file system.
--help	Display this help and exit.
--version	Output version information and exit.

By default, sparse SOURCE files are detected by a crude heuristic. The corresponding DEST file is made sparse as well. That is the behavior selected by `--sparse=auto`. Specify `--sparse=always` to create a sparse DEST file whenever the SOURCE file contains a long enough sequence of zero bytes. Use `--sparse=never` to inhibit creation of sparse files. When `--reflink[=always]` is specified, perform a lightweight copy, where the data blocks are copied only when modified. If this is not possible the copy fails, or if `--reflink=auto` is specified, fall back to a standard copy. The backup suffix is `~`, unless set with `--suffix` or `SIMPLE_BACKUP_SUFFIX`. The version control method may be selected via the `--backup` option or through the `VERSION_CONTROL` environment variable. Here are the values:

none, off	Never make backups (even if `--backup` is given).
numbered, t	Make numbered backups.
existing, nil	Numbered if numbered backups exist, simple otherwise.
simple, never	Always make simple backups.

As a special case, `cp` makes a backup of SOURCE when the force and backup options are given and SOURCE and DEST are the same name for an existing, regular file.

Author

Written by Torbjorn Granlund, David MacKenzie, Jim Meyering, and Paul Kolano.

Reporting Bugs

Report `cp` bugs to `bug-coreutils@gnu.org` GNU coreutils home page:

`http://www.gnu.org/software/coreutils/`

General help using GNU software: `http://www.gnu.org/gethelp/`

Copyright

Copyright © 2009 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later `http://gnu.org/licenses/gpl.html`. This is free software—users are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

See Also

The full documentation for `cp` is maintained as a Texinfo manual. If the `info` and `cp` programs are properly installed at your site, the command `info coreutils 'cp invocation'` should give you access to the complete manual.

7.2 msum(1)

Name

`md5sum`- Compute and check MD5 message digest.

Synopsis

```
md5sum [OPTION]... [FILE]...
```

Description

Print or check checksums. With no FILE, or when FILE is –, read standard input. `msum`-specific options (defaults in brackets):

--buffer-size=MBYTES	Read/write buffer size [4].
--check-tree	Print/check hash subtrees to pinpoint corruption.
--fadvise-read	Enable use of posix_fadvise during reads.
--hash-leaf-size=KBYTES	Granularity of hash tree [1048576].
--hash-type=TYPE	Hash type [MD5], with TYPE one of:

- `md5`
- `sha1`
- `sha256`
- `sha384`
- `sha512`
- `sha224`
- `crc32`
- `crc32rfc1510`
- `crc24rfc2440`

`--length=LEN` Hash LEN bytes beginning at `--offset` (or `0` if `--offset` not specified).

`--listen-port=PORT` Listen on port PORT for requests from cooperating hosts.

`--manager-host=HOST` Host name or IP address of management thread for multi-node/multi-host copies.

`--manager-port=PORT` Port on which to contact management thread.

`--mpi` Enable use of MPI for multi-node checksums.

`--no-direct-read` disable use of direct I/O for reads

`--no-double-buffer` Disable use of double buffering during file I/O.

`--offset=POS` Hash-length bytes beginning at POS (or to end if `--length` not specified).

`--password-file=FILE` File to use for passwords (will be created if does not exist).

`--read-stdin` Perform a batch of operations read over stdin in the form 'FILE RANGES' where FILE must be a URI-escaped (RFC 3986) file name and RANGES is zero or more comma-separated ranges of the form 'START-END' for 0 <= START < END.

`--split-size=MBYTES` Size to split files for parallelization [1024]

`--threads=NUMBER` Number of OpenMP worker threads to use [4].

Standard options:

`-b, --binary` Read in binary mode.

`-c, --check` Read sums from the FILEs and check them.

`-t, --text` Read in text mode (default).

The following three options are useful only when verifying checksums:

`--quiet` Do not print OK for each successfully verified file.

`--status` Don't output anything, status code shows success.

`-w, --warn` Warn about improperly formatted checksum lines.

`--help` Display this help and exit.

--version Output version information and exit.

When checking, the input should be a former output of this program. The default mode is to print a line with checksum, a character indicating type (`*' for binary, ` ' for text), and name for each FILE.

Author

Written by Ulrich Drepper, Scott Miller, David Madore, and Paul Kolano.

Reporting Bugs

Report `md5sum` bugs to `bug-coreutils@gnu.org`GNU coreutils home page:

`http://www.gnu.org/software/coreutils/`

General help using GNU software: `http://www.gnu.org/gethelp/`

Copyright

Copyright ™ 2009 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <`http://gnu.org/licenses/gpl.html`>. This is free software—users are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

See Also

The full documentation for `md5sum` is maintained as a Texinfo manual. If the info and `md5sum` programs are properly installed at your site, the command info coreutils `md5sum invocation` should provide access to the complete manual.

7.3 tasdefaults.conf(4)

Name

`tasdefaults.conf` - Default values for Tiered Adaptive Storage (TAS) configuration.

Synopsis

`/tas_admin/config/tasdefaults.conf`

Description

The `tasdefaults.conf` file allows the end user to alter various configuration tunables to meet their environmental needs. The `tasdefaults.conf` file is referenced by the various TAS commands that alter their behavior. The `tasdefaults.conf` file can be updated at anytime.

General Parameters

MCF Location of VSM mcf configuration file. Default: `/etc/opt/VSM- samfs/mcf`.

VSM_SBIN	Location of VSM admin commands. Default: `/opt/VSMsamfs/sbin`.
EMAIL_ADDR	Email address for TAS notifications. Default: `"root" tasdump` PARAMETERS
VSM_DUMP_DIR	Location of dumps. Default: `/vsm_admin/vsm_dumps`
VSM_DUMP_LOG_DIR	Location of dump logs. Default: `/vsm_admin/logs/vsm_dumps`
VSM_DUMP_COMPRESS	Compress dump files? NO=0, YES=1. Default: 1 `taschart` PARAMETERS
DATA_DEVICES	List of data devices to report performance statistics, separated by `|`. Example `sdd|sde|sdf`. Default: Empty List.
REPORT_DIR	Location of reports. Default: `/vsm_admin/reports` Location of report directory.
SAR_DIR	Location of raw SAR data files. Default: `/var/log/sa tastapeq` PARAMETERS
WAITQ_TIME	Max tape request queue wait time in minutes. Default: 30 minutes.
TAPE_TYPES	Acceptable tape types (seperated by `|`). Default: `lt|li` (LTO and T10K) `tassyncadmin` PARAMETERS
TAS_SYNC_DST	Location of of destination `rsync` directory.
TAS_SYNC_LOG	Location of of `tassyncadmin` log. `tasarchive` PARAMETERS.
CIMS_ARCHIVE_DIR	Location of archive files.
CIMS_ARCHIVE_LIMIT	`tasarchive` will not run if the target file system usage is greater than specified limit.
CIMS_WEEKLY_FILES	Number of archive files to keep in addition to latest archive.

TAS Cluster Parameters

NO_VSMFS_MOUNT	Mount files cluster start? YES=0, NO=1.

Author

Scott Donoho (sdonoho@cray.com)

7.4 tasarchive(8)

Name

`tasarchive` - TAS administrative file archive

Synopsis

```
tasarchive [OPTION]
```

Description

`tasarchive` automatically manages CIMS adminstrative file system backups. The archive files are stored accordingly to the appropriate settings in the `tasdefaults.conf` configuration file. The archive files will be rotated each time `tasarchive` is run.

Options

The `tasarchive(8)` command uses the following parameters from the `tasdefaults.conf` file:

CIMS_ARCHIVE_DIR	Location of archive files.
CIMS_ARCHIVE_LIMIT	`tasarchive` will not run if the target file system usage is greater than specified limit.
CIMS_WEEKLY_FILES	Number of archive files to keep in addition to latest archive.

See Also

`tasdefaults.conf(4)`

Bugs

No known bugs.

7.5 tasbundle(8)

Name

`tasbundle`

Synopsis

```
tasbundle
```

Description

`tasbundle` has been deprecated for individual nodes. Please use `sosreport` or run `tasbundle` on the management node to create a package for a support representative.

See Also

`sosreport(1)`

7.6 taschart(8)

Name

`taschart(8)` - TAS report tool.

Synopsis

```
tasdump [OPTION]
```

Description

`taschart` is a TAS performance reporting tool.

Options

The `tasdump(8)` command accepts the following options:

`-h, --help` Help (list command usage).

`-m, --mail` Mail option (email report to admin email address defined in `tasdefaults.conf`).

`-q, --quiet` Quiet mode.

Do not print messages to STDOUT.

Besides the command line options, the following configuration settings in the `tasdefaults.conf` file can be altered:

`DATA_DEVICES` List of data devices to report performance statistics, separated by `|`. Example `sdd|sde|sdf`. Default: Empty List.

`REPORT_DIR` Location of reports Default: `/tas_admin/reports` (location of report directory).

`SAR_DIR` Location of raw SAR data files. Default: `/var/log/sa`

See Also

`tasdefault.conf(4)`

Bugs

No known bugs.

Author

Scott Donoho (sdonoho@cray.com)

7.7 tasclean(8)

Name

`tasclean` - Manage VSM dumps and TAS chart reports

Synopsis

```
tasclean [OPTION]
```

Description

`tasclean` maintains a number of VSM file system dumps/logs and TAS chart reports. The number of dumps and reports retained is defined with the `VSM_DUMP_COUNT` and `VSM_CHART_COUNT` variables in the `tasdefaults.conf` file.

Options

The `tasclean(8)` command accepts the following options:

`-h, --help` Help. List command usage.

`-m, --mail` Mail option. Email `tasclean` report to admin email address defined in `tasdefaults.conf`.

`-q, --quiet` Quiet mode.

Do not print messages to STDOUT.

See Also

`samfsdump(8)` and `samfsrestore(8)`

Bugs

No known bugs.

Author

Scott Donoho (sdonoho@cray.com)

7.8 tasdump(8)

Name

`tasdump`- Manage VSM file system dumps.

Synopsis

```
tasdump [OPTION]
```

Description

`tasdump` automatically manages VSM file system dumps. The dumps are stored accordingly to the appropriate settings in the `tasdefaults.conf` configuration file.

Options

The `tasdump(8)` command accepts the following options:

`-h, --help`	Help. List command usage.
`-l, --last`	List latest dumps of mounted VSM file systems.
`-m, --mail`	Mail option. Email dump log to admin email address defined in`tasdefaults.conf`.
`-q, --quiet`	Quiet mode. Do not print messages to STDOUT.

Besides the command line options, the following configuration settings in the `tasdefaults.conf` file can be altered:

`VSM_DUMP_DIR`	Location of dump directory. Defaults to `/tas_admin/vsm_dumps`.
`VSM_DUMP_LOG_DIR`	Location of log dump directory. Defaults to `/tas_admin/logs/vsm_dumps`.
`VSM_DUMP_COMPRESS`	Directive to compress VSM dumps. Defaults to 1 (compress dumps).

See Also

`tasdefaults.conf(4)`, `samfsdump(1M)`, `samfsrestore(1M)`.

Bugs

No known bugs.

Author

Scott Donoho (sdonoho@cray.com)

7.9 tasha(8)

Name

`tasha` - TAS MDC cluster manual failover.

Synopsis

```
tasha [OPTION] [COMMAND]
```

Description

`tasha` functions as the main interface for managing and monitoring the TAS cluster environment. The main functions of the command are starting, stopping, converting and statusing a cluster node. The `tasha` command works in two modes clustered and manual. The various `tasha` commands available are specific to the clustered or manual mode.

Options

The `tasha(8)` command accepts the following options:

-h, --help	Help. List command usage.
-q, --quiet	Quiet mode. Do not print messages to STDOUT.
-r, --retries	Retries. Number of start command retries. May be required at boot time if device creation needs more time at boot time.

Commands (clustered)

The `tasha` command in a clustered configuration requires one of following options:

failover *nodename*	Help. List command usage.
standby *nodename*	Quiet mode. Do not print messages to STDOUT.
online *nodename*	Retries Number of start command retries. May be required at boot time if device creation needs more time at boot time.
maint	Toggle cluster maintenance node.

Commands (manual)

The `tasha` command in a non-clustered configuration requires one of following options:

start	Starts the VSM services, mounts file systems on node. Runs only on active node.
stop	Stops the VSM services, unmounts file systems on node. Runs only on active node.
makemdc	Converts standby node to active node in the TAS cluster.
status	Returns TAS status of node.

Examples (clustered)

In clustered mode, the `tasha` command is typically run during a fail over scenario where the active node is failed over to the standby node. This command can be run on either the active or passive node.

```
tas-mdc1# tasha failover tas-mdc2
```

Examples (manual)

In manual mode, the `tasha` command is typically run during a fail-over scenario where the active node is stopped. This followed by the standby node being converted to the active MDC and then started.

```
active# tasha stop
```

```
standby# tasha makemdc
```

```
standby# tasha start
```

See Also

samu(1), samsharefs(1)

Bugs

No known bugs.

Author

Scott Donoho (sdonoho@cray.com)

7.10 tashwm(8)

Name

tashwm - Monitor VSM high water mark

Synopsis

```
tashwm [OPTION]
```

Description

tashwm returns error and reports error if any VSM mounted file system has exceeded its defined HWM.

Options

The tashwm(8) command accepts the following options:

-h, --help Help. List command usage.

-m, --mail Mail option. Email error message to admin email address defined in tasdefaults.conf.

-q, --quiet Quiet mode. Do not print messages to STDOUT.

-s, --syslog Log mode. Print error message to message log.

See Also

samu(1M)

Bugs

No known bugs.

Author

Scott Donoho (sdonoho@cray.com)

7.11 taskeyscan(8)

Name

`taskeyscan`- Update `~/.ssh/known_hosts`.

Synopsis

```
taskeyscan
```

Description

`taskeyscan` removes and then adds the SSH host keys for all tas nodes to the `known_hosts` file.

Bugs

No known bugs.

See Also

`ssh-keygen(1)`, `ssh-keyscan(1)`

7.12 tasrw(8)

Name

`tasrw [OPTION]`

Synopsis

```
tasrw
```

Description

`tasrw` reads and writes a small file on each read/write mounted VSM file system, verifies the basic functionality of each file system. The file is removed on exit of `tasrw`.

Options

The `tasrw(8)` command accepts the following options:

-h, --help Help. List command usage.

-m, --mail `tasarchive` Mail option. Email error message to admin email address defined in `tasdefaults.conf`.

-q, --quiet Quiet mode. Do not print messages to STDOUT.

See Also

`samu(1M)`

Bugs

No known bugs.

Author

Scott Donoho (sdonoho@cray.com)

7.13 tassynchadmin(8)

Name

`tassyncadmin`- Manage VSM file system dumps.

Synopsis

```
tassyncadmin [OPTION]
```

Description

`tassyncadmin` automatically manages VSM adminstrative file system back-ups via `rsync`. The `rsync`s are stored accordingly to the appropriate settings in the`tasdefaults.conf` configuration file.

Options

The `tassyncadmin(8)` command accepts the following options:

-h, --help Help. List command usage.

-m, --mail Mail option. Email `rsync` log to admin email address defined in `tasdefaults.conf`.

-q, --quiet Quiet mode. Do not print messages to STDOUT.

Besides the command line options, the following configuration settings in the `tasdefaults.conf` file can be altered:

TAS_SYNC_DST Location of of destination `rsync` directory.

TAS_SYNC_LOG Location of of `tassyncadmin` log.

See Also

`tasdefaults.conf(4)`, `rsync(1M)`

Bugs

No known bugs.

Author

Scott Donoho (sdonoho@cray.com)

7.14 tastapeq(8)

Name

`tastapeq` - Monitor VSM tape queue.

Synopsis

```
tastapeq [OPTION]
```

Description

`tastapeq` returns warning if tape request in queue exceeds defined wait time.

Options

The `tastapeq(8)` command accepts the following options:

`-h, --help` Help. List command usage.

`-m, --mail` Mail option. Email dump log to admin email address defined in `tasdefaults.conf`.

`-q, --quiet` Quiet mode. Do not print messages to STDOUT.

`-s, --syslog` Log mode. Log messages to admin log.

Besides the command line options, the following configuration settings in the `tasdefaults.conf` file can be altered:

`WAITQ_TIME` Max tape request queue wait time in minutes. Default 30 minutes.

`TAPE_TYPES` Acceptable tape types (seperated by `|`); Default: `lt|li` (LTO and T10K).

See Also

`samu(1M)`,`showqueue(1M)`

Bugs

No known bugs.

Author

Scott Donoho (sdonoho@cray.com)