SGI® InfiniteStorage Cluster Manager for Linux® Administrator's Guide

007–3800–001

CONTRIBUTORS

Written by Lori Johnson

Engineering contributions by Dale Brantly, Jeff Cech, Susheel Gokhale, Ron Kerry, LaNet Merrill, Nate Pearlstein, Kevan Rehm, Paddy Sreenivasan

Illustrated by Chrystie Danzer

Production by Karen Jacobson

Record of Revision

Version	Description
001	May 2004 Original publication to support SGI Cluster Manager 3.0 for Linux

Contents

Figures

Tables

About This Guide

This guide provides information about SGI Cluster Manager 3.0 for Linux, which provides high-availability services for SGI Altix servers. It is based on the Red Hat Cluster Manager product. An optional product provides high-availability services for CXFS clustered filesystems, the Data Migration Facility (DMF), and the Tape Management Facility (TMF).

Related Publications

The following publications contain additional information that may be helpful:

- *Red Hat Cluster Suite: Configuring and Managing a Cluster*, which is available on the SGI cluster manager CD 1 and at the following website:

  https://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/cluster-suite/

- SGI ProPack for Linux 64-bit and SGI Altix documentation:
  - *NIS Administrator's Guide*
  - *Personal System Administration Guide*
  - *SGI ProPack for Linux Start Here*
  - *SGI Altix 3000 User's Guide*
  - *Performance Co-Pilot for IA-64 Linux User's and Administrator's Guide*
  - *SGI L1 and L2 Controller Software User's Guide*

Obtaining Publications

You can obtain SGI documentation as follows:

- See the SGI Technical Publications Library at http://docs.sgi.com. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.

- If it is installed on your SGI system, you can use InfoSearch, an online tool that provides a more limited set of online books, release notes, and man pages. With an IRIX system, enter `infosearch` at a command line or select **Help > InfoSearch** from the Toolchest.
- On IRIX systems, you can view release notes by entering either `grelnotes` or `relnotes` at a command line.
- On Linux systems, you can view release notes on your system by accessing the `README.txt` file for the product. This is usually located in the `/usr/share/doc/`*productname* directory, although file locations may vary.
- You can view man pages by typing `man` *title* at a command line.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
variable	Italic typeface denotes variable entries and words or concepts being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
[ ]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.

GUI	This font denotes the names of graphical user interface (GUI) elements such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, fields, and lists.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this publication, contact SGI. Be sure to include the title and document number of the publication with your comments. (Online, the document number is located in the front matter of the publication. In printed publications, the document number is located at the bottom of each page.)

You can contact SGI in any of the following ways:

- Send e-mail to the following address:

  techpubs@sgi.com

- Use the Feedback option on the Technical Publications Library Web page:

  http://docs.sgi.com

- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.

- Send mail to the following address:

  Technical Publications
  SGI
  1500 Crittenden Lane, M/S 535
  Mountain View, California 94043–1351

SGI values your comments and will respond to them promptly.

Introduction

The SGI Cluster Manager for Linux provides a *highly available* system that survives a single point of failure. It uses redundant components and special software to provide *highly available services* for a cluster that contains multiple machines/partitions (*members*). This release supports a cluster of two members.

SGI Cluster Manager is based on Red Hat Cluster Manager, which is part of Red Hat Cluster Suite 3.0. Therefore, this guide will refer to the Red Hat documentation whenever possible. You must have access to *Red Hat Cluster Suite: Configuring and Managing a Cluster*, which is available on the SGI Cluster Manager CD and at the following website:

https://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/cluster-suite/

This book provides additional information needed to use the SGI product for SGI Altix servers in a high-availability cluster environment. You should read through this guide before you begin configuring the cluster.

Although SGI Cluster Manager for Linux provides similar functionality to IRIX FailSafe, there are differences; see Appendix A, "FailSafe and SGI Cluster Manager" on page 73.

This chapter discusses the following:

Base Product

The SGI Cluster Manager base product provides failover support for the following:

- Filesystems (including XFS)
- NFS
- Samba
- IP addresses
- User applications that you define

Optional Plug-In Product

An optional value-add product supplies highly available services for the following plug-ins:

- CXFS clustered filesystems
- Data Migration Facility (DMF)
- XVM volume manager in local mode

This product also provides a failover script for the Tape Management Facility (TMF). You can modify your application to use this script to provide highly available services for TMF.

Highly Available Services

A highly available service consists of the following:

- Disks
- IP address
- Filesystem
- NFS (if used)
- Samba (if used)
- User application (if used)

All highly available services are owned by one member at a time. Highly available services are monitored by the SGI Cluster Manager software. If one member fails, the other member restarts the highly available applications of the failed member, known as the *failover* process.

To clients, the services on the replacement member are indistinguishable from the original services before failure occurred. It appears as if the original member has crashed and rebooted quickly. The clients notice only a brief interruption in the highly available service.

Differences Between Red Hat Cluster Manager and SGI Cluster Manager

Table 1-1 summarizes the differences between Red Hat Cluster Manager and SGI Cluster Manager. (You should not install Red Hat Cluster Manager on an SGI Altix system. It is intended only for Linux 32–bit machines. It will result in conflicts with SGI Cluster Manager.)

Table 1-1 Red Hat Cluster Manager and SGI Cluster Manager

Topic	Red Hat Cluster Manager	SGI Cluster Manager for Linux
Service timeouts	Not supported.	Supported.
Check/monitor interval	Supports a check interval, which does not include execution time. That is, the check starts at time *t*, and the next check will be at *t+check_interval*, irrespective of execution time.	Supports a monitor interval. The next check will be done after *execution time + monitor interval*.
Software/hardware watchdog timers	Supported.	Not supported. Do not enable the software watchdog when configuring a member. If you enable software watchdog, you will see error messages when cluster daemons are started. For more information, see "Watchdog Errors" on page 66.

Topic	Red Hat Cluster Manager	SGI Cluster Manager for Linux
Samba service	Does not keep Samba files on shared storage. Locations: • PID directory: `/var/run/samba/`*netbiosname* • Log directory: `/var/log/samba` • Lock directory: `/var/cache/samba` • Password file: `/etc/samba/smbpasswd`	Keeps Samba files in the shared disk and in the log file on the local disk. The lock directory is not removed during failover. Locations: • PID directory: *mountpoint*`/.samba/`*sharename*`/pid` • Log directory: `/var/log/samba` • Lock directory: *mountpoint*`/.samba/`*sharename*`/locks` • Password file: `/.samba/`*sharename*`/private/smbpasswd` For more information, see Chapter 7, "Samba Plug-In" on page 43.
Power control	Network– or serial-based power controllers. There can be multiple controllers defined for a given member.	SGI L2 system controllers on SGI Altix servers There can be only one controller defined for a given member.
Shared quorum partition errors	Reboots the member.	Local cluster daemons exit and wait for the member to be reset by other members in the cluster.

Hardware Requirements

SGI Cluster Manager for Linux requires the following:

- A cluster of up to two members. The following servers are supported:
  - SGI Altix 350 servers with IO10 and serial-port dongle (see "Power Controllers" on page 10)
  - SGI Altix 3700 servers or Altix 3300 servers (may be partitioned; each partition is an individual member)
- At least one shared disk

- Network cabling: you can connect private network or cross-over cables between members. You have a choice between an Ethernet cable from server to hub or a 20-ft cross-over Ethernet cable.
- Serial cabling: You must use the serial ports on the IX brick.

Figure 1-1 shows an example configuration using CXFS.

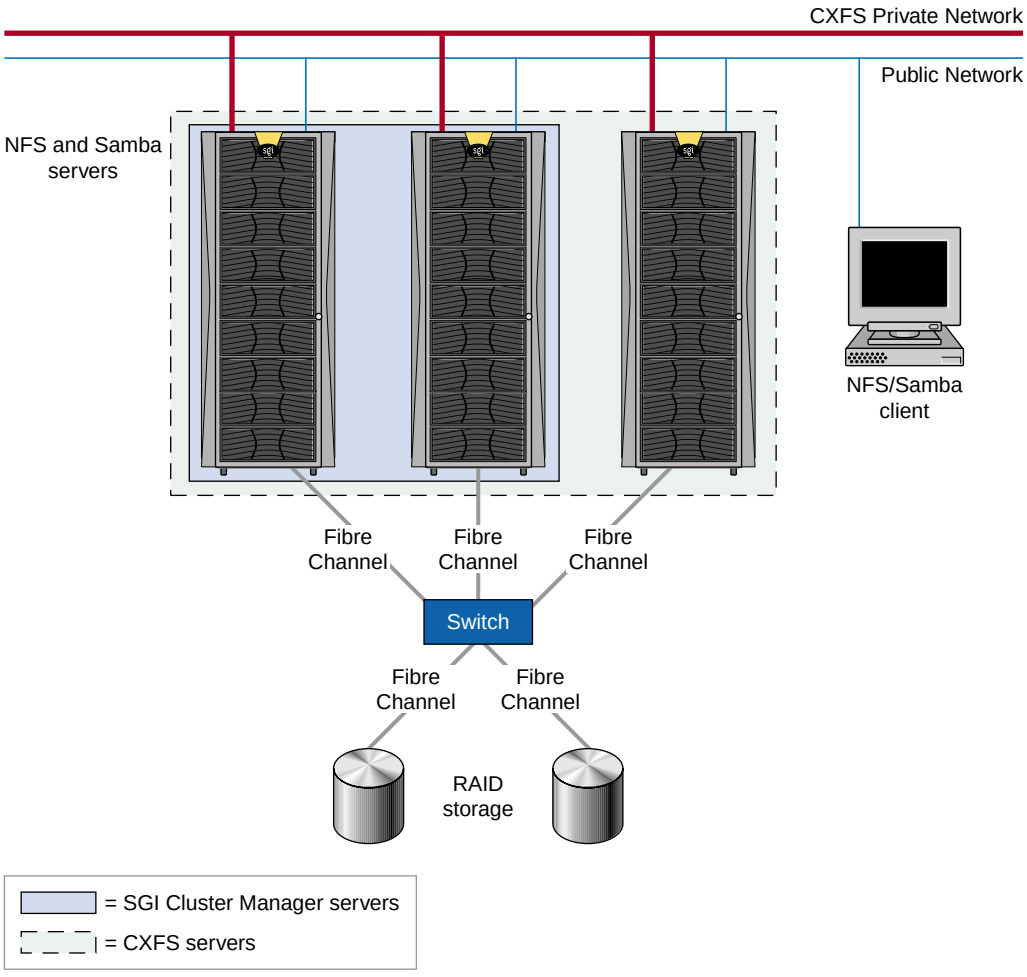

Figure 1-1 An Example CXFS and SGI Cluster Manager Configuration

Software Requirements

SGI Cluster Manager requires the following:

- SGI ProPack 3
- Red Hat Enterprise Advanced Server 3.0.

Note: The Linux virtual server (load-balancing software) that is part of Red Hat Cluster Suite is not supported on SGI Altix systems.

This release also supports the following releases:

- Samba 3.0 as shipped with Red Hat Enterprise Advanced Server 3.0
- CXFS 3.1 (or later) Altix server/client release

Note: Use of clustered XVM volumes with SGI Cluster Manager requires the CXFS plug-in. The SGI Custer Manager base product supports local XVM volumes.

- DMF 3.0.1 (or later)
- TMF 1.4.1 (or later)

See "Software Packages" on page 13 and the `README` file for a list of the RPMs included on CDs.

Failover Domains

The *failover domain* is the list of members in the cluster where a service can be online. The failover domain is not ordered.

When a member boots, all other members determine if service should be relocated to the new member if the failover domain is correctly defined for the service.

If there is a failback, member crash, or initial membership, the service will be started on the local member as follows:

- If no failover domain is defined
- If the member is not part of the domain (if the domain is not restricted) and no members of the domain are in membership
- If the member is part of the domain and no ordering is specified

If an administrative command is used to enable a service, the service will be started on the local member as follows:

- If the member is part of the domain and is the lowest-ordered

---

Note: *Lowest-ordered* means a higher preference for a service to be started on that member.

- If the member is part of a domain and all lower-ordered members are unavailable
- If the member is not part of the domain (if the domain is not restricted) and at least one member of the domain is in membership

The service **will not** be started on the local member under the following circumstances:

- If the member is not configured
- If the member is not in the membership
- If the member is not in the domain (for a restricted domain)

Cluster Daemons

Following is an overview of the cluster daemons:

- `clumembd` is the cluster membership daemon. It performs network heartbeats and checks the liveliness of other members in the cluster
- `cluquorumd` is the cluster quorum daemon. It computes new membership and implements quorum. `cluquorumd` also enforces I/O fencing (resets) and reads/writes membership information to the shared partitions.

- `clurmtabd` is the cluster remote NFS mount table daemon. It synchronizes NFS mount point entries by polling the `/var/lib/nfs/rmtab` file.
- `clusvcmgrd` is the cluster service manager daemon. It starts/stops and checks the status of services running in the cluster.
- `clulockd` is the cluster global lock manager daemon. The locks are stored on shared disk.

For more information, see the Red Hat documentation.

Hardware Installation

This chapter discusses the following:

Shared Disks

SGI Cluster Manager for Linux **requires** two 10–MB partitions to keep membership quorum partitions in the shared disk. You should use the XSCSI raw device driver to access these partitions (do not use the Linux raw device driver).

SGI Cluster Manager supports SAN configurations using TP9300, TP9500, and TP9100 RAIDs. Each member in the cluster should be connected to storage using dual paths so that there is no single point of failure. All disks must be in RAID 0/1 mirrored, RAID3, or RAID5 configurations.

The device names for the shared disks must be identical on all cluster members. Use the `/usr/lib/clumanager/create_device_links` script to create the same device name on each member.

For more information, see:

- *SGI® InfiniteStorage TP9400 and SGI® InfiniteStorage TP9500 and TP9500S RAID User's Guide*
- *SGI® InfiniteStorage TP9300 and TP9300S RAID User's Guide*
- *SGI TPSSM Administration Guide*

Heartbeat Network

SGI Cluster Manager for Linux uses hostnames for sending heartbeat and control messages. To use the private network, you must configure hostnames as private network interface names. Ethernet cables are provided that will allow the members to be connected directly or using a network hub.

You can use 10/100baseT or 1–Gb ports in the system for heartbeat communication. For more information, see *SGI Altix Systems Dual-Port Gigabit Ethernet Board User's Guide*.

Heartbeats are either broadcast on all networks or multicast on the network interface that hostname configured.

Power Controllers

You must use SGI system controllers for power control. SGI Cluster Manager supports only one system controller per member.

Network-based, and serial-based power controllers are not supported for SGI Cluster Manager on SGI Altix servers.

For more information, see the following:

- *SGI Altix 3000 User's Guide*
- *SGI Altix 350 System User's Guide*

For information about configuring the power controller, see "Step 4: Add Power Controller Configuration" on page 22.

L2 Power Controller

The L2 system controller and the required USB cables are optional equipment available for purchase.

Use the following:

- DB9 serial ports on an IX brick on Altix 3700
- Serial ports on Altix 350 with IO10 and serial-port dongle

Use the remote modem port on the L2 system controller. Connect the serial cable to the remote modem port on one end and the `tty` port on the other end.

Testing Connectivity

To test the serial lines, do the following:

1. Create symlinks to the serial devices:

```
# ln -s /dev/ttyIOC4/0 /dev/ttyI0
# ln -s /dev/ttyIOC4/1 /dev/ttyI1
```

2. Use the `cu`(1) command to test the connection to the controller. The `cu` command requires the `tty` names to be in the following format:

   `/dev/tty`*XXX*

   You can use `/dev/ttyIOC4/0` to define power controllers in SGI Cluster Manager.

   Note: You must use `parity=even`.

   For example:

```
# cu -l /dev/ttyI0 -s 38400 --parity=even
Connected.

Jackhammer-001-L2>cfg
L2 163.154.17.133: - 001 (LOCAL)
L1 163.154.17.133:0:0    - 001c04.1
L1 163.154.17.133:0:1    - 001i13.1
L1 163.154.17.133:0:5    - 001c07.2
L1 163.154.17.133:0:6    - 001i02.2
```

   For more information, see the `cu`(1) man page.

You can also use the `clufence`(8) command to test serial connectivity.

Software Installation

This chapter describes how to install the SGI Cluster Manager for Linux software.

Note: The package names in this chapter are examples; the names in the released version may differ slightly.

This chapter includes the following sections:

Software Packages

The following packages are provided:

- Base product:
  - `clumanager-1.2.3-AS3.0sgi300`*XX*`.ia64.rpm`, which contains the basic monitoring and failover capabilities
  - `redhat-config-cluster-1.2.3-AS3.0sgi300`*XX*`.noarch.rpm`, which contains the configuration tools

    Note: The `redhat-config-cluster` RPM is dependent upon the `clumanager` RPM. You must uninstall `clumanager` first.

  - `rh-cs-en-3-AS3.0sgi300`*XX*`.noarch.rpm`, which contains the Red Hat documentation
  - `sgi-cluster-manager-docs-3.0-1.noarch.rpm`, which contains this SGI guide
- Optional high-availability plug-ins for CXFS, DMF, TMF, and local XVM: `clumanager-sgi-1.0.0`

Installing the Software

Do the following:

1. If necessary, upgrade to SGI ProPack 3 according to the directions in *SGI ProPack for Linux Start Here*.
2. Insert the *SGI Cluster Manager 3.0 for Linux - Base Product* CD and do the following to mount the CD and see its contents:

```
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom
# ls
COPYING  README  RPM_MD5_SUMS  SGI  TRANS.TBL
```

Read the README file to learn about any late-breaking changes in the installation procedure.

3. Install the software from the CD using the rpm(8) command:

```
# rpm -Uvh clumanager-1.2.3*.rpm redhat-config-cluster-1.0.0*.rpm rh-cs-en-3*rpm \
sgi-cluster-manager-docs*.rpm
      Preparing...                ########################################### [100%]
         1:clumanager             ########################################### [ 25%]
         2:redhat-config-cluster  ########################################### [ 50%]
         3:rh-cs-en               ########################################### [ 75%]
         4:sgi-cluster-manager-docs
```

For more information, see the rpm(8) man page.

4. If you have purchased the optional high-availability product for CXFS, DMF, TMF, and local XVM plug-ins, insert the *SGI Cluster Manager 3.0 for Linux - Storage Software Plug-ins* CD. Do the following to mount the CD and see its contents:

```
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom
# ls
COPYING  README  RPM_MD5_SUMS  SGI  TRANS.TBL
```

Read the README file to learn about any late-breaking changes in the installation procedure.

5. Install the software from the CD:

```
# rpm -Uvh clumanager-sgi-1.0.0-AS3.0sgi300a1.ia64.rpm
```

Uninstalling the Software

To uninstall the software, use the following command:

`rpm -e` *rpm_name*

Uninstalling the `clumanager` RPM will attempt to stop cluster daemons in the local node.

Note: The `redhat-config-cluster` RPM is dependent upon the `clumanager` RPM. You must uninstall `clumanager` first.

For more information, see the `rpm`(8) man page.

Configuration

This chapter provides an overview of the basic configuration process, plus specific information about SGI Cluster Manager for Linux that differs from the Red Hat Cluster Manager. For details, follow the information in Chapter 2, "Cluster Configuration," in the *Red Hat Cluster Suite: Configuring and Managing a Cluster* manual.

Cluster Configuration Tools

SGI Cluster Manager supports the following tools to configure the cluster:

- `redhat-config-cluster` Cluster Status graphical user interface (GUI)
- `redhat-config-cluster-cmd` command-line interface (CLI)

At any given time, you must use only one of these tools to perform configuration tasks. The GUI and the CLI supply similar functionality, although there are a few exceptions.

Figure 4-1 shows an example of the GUI for a cluster named `test-cluster`.

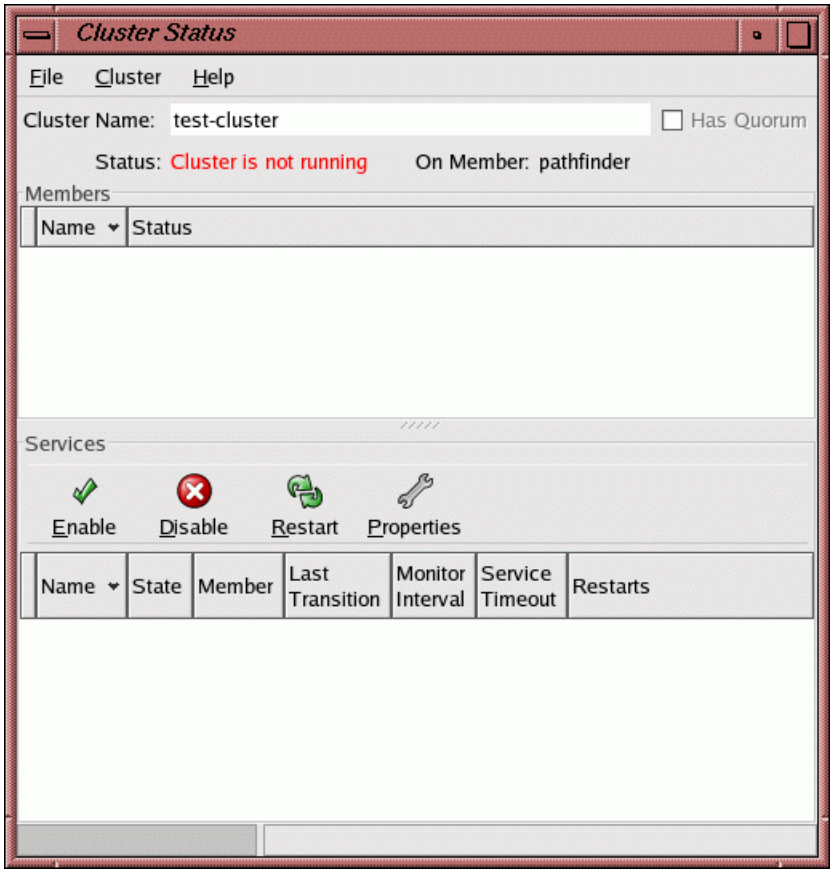

Figure 4-1 Cluster Status `redhat-config-cluster` GUI

Caution: After making modifications to the configuration using the GUI, you should save the information using the following selections:

File
> Save

The information is written to a local `/etc/cluster.xml` file as well as to the shared partition.

Configuration Steps

This section discusses the configuration steps:

Step 1: Define Shared State

The names of the device files for filesystems and raw partitions to store quorum information must be the same on all cluster members. You must do one of the following:

- Ensure that the members have their disks attached identically.
- Create symlinks on each member as appropriate. You must re-create the symlinks every time the machine reboots because `/dev` files are re-created. Therefore, you should modify the `/usr/lib/clumanager/create_device_links` script to add the device symlinks that are required.

SGI recommends that the two shared partitions should be on different FC controllers. They should be at least 10 MB in size and the partition type must be `linux`. For example, suppose you have the following output from the `hinv` command:

```
Integral SCSI controller pci04.01.0: Version Fibre Channel QLA2300 (Rev 1) pci04.01.0
    Disk Drive: unit   2 lun  0 on SCSI controller pci04.01.0  0
Integral SCSI controller pci05.01.0: Version Fibre Channel QLA2300 (Rev 1) pci04.01.0
    Disk Drive: unit   3 lun  0 on SCSI controller pci05.01.0  0
```

Partition 1 from disk on FC target 2 and FC target 3 are used for storing shared state. You could create symlinks to the raw XSCSI device names as follows (you must use character special devices):

```
# ln -s /dev/xscsi/pci04.01.0/target2/lun0/rpart1 /dev/shared_raw1
# ln -s /dev/xscsi/pci05.01.0/target3/lun0/rpart1 /dev/shared_raw2
```

You should add these symlink commands to the `/usr/lib/clumanager/create_device_links` script on the cluster member.

The shared raw partitions are `/dev/shared_raw1` and `/dev/shared_raw2`.

To define the shared raw partitions in the GUI configuration window, select:

Cluster
> Shared State

In the CLI:

```
# redhat-config-cluster-cmd --sharedstate --type=raw --rawprimary=/dev/shared_raw1 \
                                --rawshadow=/dev/shared_raw2
```

You should perform this step before defining the cluster ("Step 2: Create the Cluster" on page 21).

Note: Do not use the Red Hat Linux `raw`(8) interface for storing shared state. Disregard the information provided in sections 1.2.5, 1.4.4, and 1.4.6 of *Red Hat Cluster Suite: Configuring and Managing a Cluster*.

Instead, you should use the XSCSI raw partition device file and the `create_device_links` script. Use symlinks if the raw partition XSCSI device names on both members are not identical. You should add the symlink commands to the `/usr/lib/clumanager/create_device_links` script.

Files under `/dev` are re-created when the member reboots. If you are creating symlinks from devices to files under the `/dev` directory, you must also do so in the `/etc/init.d/clumanager` scripts.

Step 2: Create the Cluster

To create the cluster in the GUI, type the cluster name in the **Cluster Name** field in cluster configuration window. The default cluster name is `SGI High Availability cluster`.

In the CLI:

```
redhat-config-cluster-cmd --cluster --name "clustername"
```

Step 3: Define the Members

To define a member in the GUI, do the following:

- Select the following:

  Cluster
  > Configure

- Click **Member** tab.
- Click **New**.
- Provide the hostname to be used for communication.

- Disable the software watchdog.

Note: The software watchdog is not supported by the SGI Cluster Manager.

In the CLI:

`redhat-config-cluster-cmd --add_member` *membername*

Step 4: Add Power Controller Configuration

For each member, you must provide information about its power controller. The SGI Cluster Manager supports the use of L2. You can specify only one power controller for each member.

In the GUI, select the member and click on **Add Child**. The fields are as follows:

- **Type** is the power controller type of the node being defined (the local member). It must be `l2`. `l2` is the default type in the GUI; in the CLI, there is no default. (This is the `type` field in the CLI).
- **Peer's TTY device file name** is the `tty` device filename on the **peer member** to which the local system controller is connected using a serial cable. The default value in the GUI is `/dev/ttyIOC4/0`. (This is the `device` field in the CLI.)
- **Altix partition** is the local member's Altix partition ID. If there are no partitions, partition ID is `0`. The default value in the GUI is `0`. (This is the `partion` field in the CLI).

Figure 4-2 shows an example in the GUI.

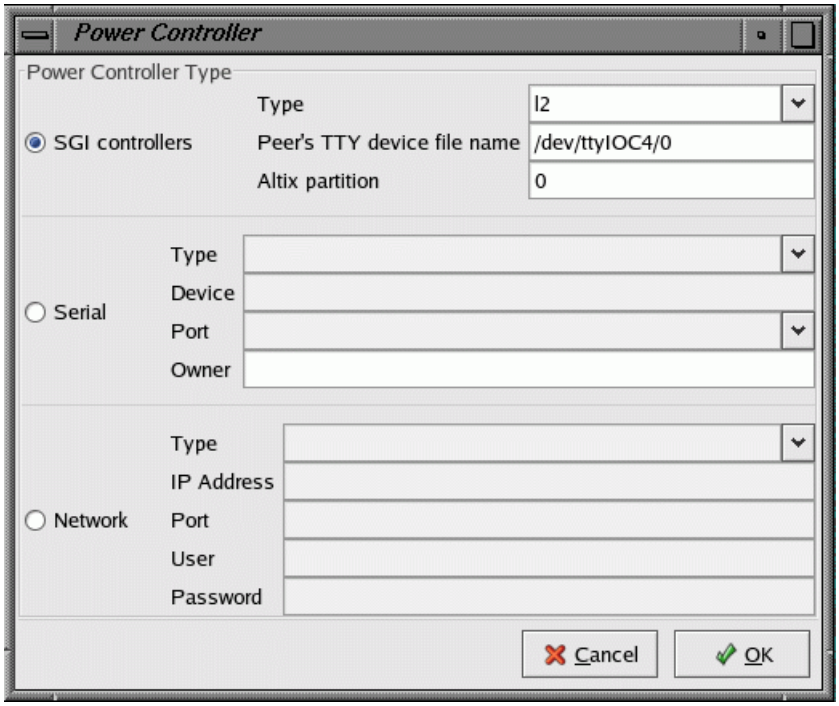

Figure 4-2 Configuring the Power Controller Information

In the CLI:

```
redhat-config-cluster-cmd --member=membername --add_powercontroller --type=l2 \
     --device=/dev/ttyIOC4/0 --partition=n
```

For hardware information, see "Power Controllers" on page 10.

Step 5: Change the Heartbeat Interval, Timeout, and Failover Speed

You can modify the time it takes to detect a member failure, known as the *failover speed*.

Note: The default failover speed differs depending upon which tool (GUI or CLI) you use to define the cluster.

Failover Speed and the GUI

In the GUI, you can supply the failover speed directly by using the following selections:

Cluster
> Daemon properties
> clumembd

The GUI does not let you display or set the `interval` and `tko_count` values.

The GUI provides 15 seconds as the default failover speed value.

Use the sliding bar to adjust failover speed, as shown in Figure 4-3. You cannot change the value for failover speed while the cluster daemons are running.

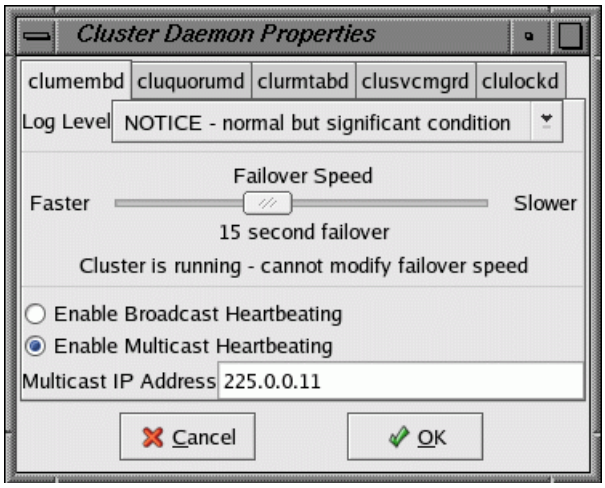

Figure 4-3 Adjusting Failover Speed

Failover Speed and the CLI

The `clumembd` daemon lets you specify the failover speed indirectly by defining the heartbeat interval and the timeout:

- `interval` specifies the heartbeat interval, which is the number of microseconds before a heartbeat is sent to all other members in the cluster. The default value is 500000 (0.5 seconds).

- `tko_count` specifies the heartbeat timeout, which is the number of heartbeats missed before a member is declared as failed. The default value is 20.

The resulting failover speed is:

failover_speed = *interval_value* * *tko_count_value*

Therefore, the default member failure detection time is 10 seconds (0.5 * 20).

Table 4-1 shows the failure detection times and parameter values that are supported.

Table 4-1 Supported Failure Detection Times and Parameter Values

Failover Speed (in seconds)	`interval` (in microseconds)	`tko_count`
30	1000000	30
25	1000000	25
20	1000000	20
15	750000	20
10	500000	20
5	330000	15

For example, the following command displays the heartbeat `interval` and `tko_count` values:

```
# redhat-config-cluster-cmd --clumembd

clumembd:
  loglevel = 5
  interval = 500000
  tko_count = 20
  thread = yes
  broadcast = no
  multicast = yes
  multicast_ipaddress = 225.0.0.11
```

The following command changes the failover speed from 10 seconds to 15 seconds:

```
# redhat-config-cluster-cmd --clumembd --interval=750000 --tko_count=20
```

Note: You cannot change the values for `interval` and `tko_count` while the cluster daemons are running.

For more information about using the command-line interface, see the "D.1. Using `redhat-config-cluster-cmd`" section of the Red Hat manual.

Step 6: Set the Tiebreaker

The tiebreaker is used to avoid a *split-brain* scenario, in which both members attempt to form individual clusters. The tiebreaker ensures that only the member that can contact the tiebreaker IP address can form a cluster. The tiebreaker is the IP address of a machine or a router that **does not participate** in the cluster. Usually, it is the IP address of a network router that connects the members to the external world (clients).

In the GUI:

Cluster
> Daemon properties
> cluquorumd

In the CLI:

```
redhat-config-cluster-cmd --cluquorumd --tiebreaker_ip=IPaddress
```

Step 7: Create the Failover Domain

The failover domain is optional; if a failover domain is not defined, the service will be started on any member. For more information, see "Failover Domains" on page 6.

Note: Defining `--ordered=yes` will cause the service to start on the first member defined if it is available. A side effect of this is that the service will automatically failback from the peer node to the primary node when the primary node is rebooted after a failure or maintenance period.

In the GUI, click on **New** with focus on **Failover Domains** in the configure window. You must click on **OK** to actually add the nodes to the domain.

In the CLI:

```
redhat-config-cluster-cmd --add_failoverdomain --name=domainname  \
                          --restricted=yes|no  --ordered=yes|no

redhat-config-cluster-cmd --failoverdomain=domainname --add_failoverdomainnode \
                          --name=membername
```

Step 8: Set the Service Monitoring Levels and Timeouts

You can specify service timeouts (in seconds) for each highly available service. This timeout is common for all actions (start, stop, and status check) that apply to the service. (You cannot specify timeouts for each resource within the service.)

NFS and Samba support multiple monitoring levels:

- `Check for processes`

  NFS checks for `nfsd` processes.

  Samba checks for `smb` and `nmb` processes

- `Check as client`

  NFS sends null RPCs to the NFS server.

  Samba sends `smb` and `nmb` queries to the samba server.

In the GUI, click **New** with the focus on **Services** in the configure window. Figure 4-4 shows an example of configuring a Samba high-availability service.

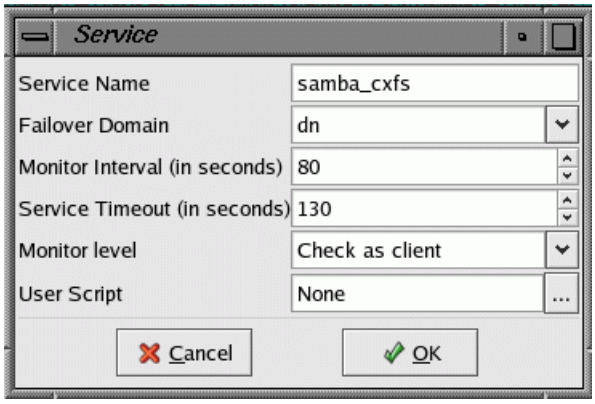

Figure 4-4 Configuring a High-Availability Service

In the CLI:

```
redhat-config-cluster-cmd --add_service --name=servicename --checkinterval=interval \
     --servicetimeout=interval --monitorlevel="level" \
     --failoverdomain=domainname
```

Note: The monitoring-level string values are case-sensitive and should be either of the following:

```
"Check for processes"
"Check as client"
```

Step 9: Add a Service IP Address

In the GUI, click on **Add Child** with the focus on the particular service. Choose **Add service IP address** in the new window.

In the CLI:

```
redhat-config-cluster-cmd --service=servicename --add_service_ipaddress \
     --ipaddress=IPaddress --netmask=netaddress \
     --broadcast=broadcastaddress
```

Step 10: Add the Disk and Filesystem Information to the Service (Optional)

In the GUI configuration window, click on **Add Child** with the focus on the particular service. Choose **Add Device** in the new window.

In the CLI:

```
redhat-config-cluster-cmd --service=servicename --add_device --name=path

redhat-config-cluster-cmd --service=servicename --device=path --mount   \
                          --mountpoint=mountpt --fstype=xfs|cxfs --options=mountoptions \
                          --forceunmount=yes
```

Step 11: Add a Samba Share (Optional)

Samba share names must be unique within the cluster.

In the GUI:

Add Exports
> Samba

In the CLI:

```
redhat-config-cluster-cmd --service=servicename --device=path \
                          --sharename=sharename
```

Step 12: Define the NFS Information

Define the NFS export point and NFS client information.

In the GUI:

Add Exports
> NFS

In the CLI:

```
redhat-config-cluster-cmd --service=servicename --device=path \
                          --add_nfsexport --name=exportdirectory

redhat-config-cluster-cmd --service=servicename --device=path \
                          --nfsexport=exportpath --add_client \
                          --name=* --options=options
```

Step 13: Save the Cluster Configuration

If you are using the GUI, you must explicitly save the configuration information as noted in "Cluster Configuration Tools" on page 17:

File
> Save

You must manually copy the `/etc/cluster.xml` file to the other member in the cluster.

Step 14: Synchronize Configuration Changes Across the Cluster

Each member has an `/etc/cluster.xml` file that contains cluster configuration information. If you make a change to this file on one member, you must copy the file to the other member, such as by running `scp`.

After making configuration changes, you must verify that the configuration files across the cluster are in synchronization. To do this, you can verify that they have the same configuration file version number (`config_viewnumber`). For example:

```
# redhat-config-cluster-cmd  --cluster

cluster:
  name = test-cluster
  config_viewnumber = 42
```

Step 15: Start the Cluster Daemons

To automatically restart the SGI Cluster Manager daemons after a reboot, do the following in the CLI:

1. Enter the following command:

   ```
   # chkconfig clumanager on
   ```

2. Start local cluster daemons on each node in the cluster doing either of the following:

   - Enter the `service clumanager start` command (or its equivalent `/etc/init.d/clumanager start`)

In the GUI, click on the following in the cluster status window:

Cluster
> Start Local Cluster Daemons

For more information, see Chapter 5, "Administration" on page 35 and the Red Hat documentation.

Example Cluster Configuration

The following example uses `redhat-config-cluster-cmd` commands to create a two-member cluster with a service providing Samba shares and NFS service:

- `member1` is an Altix 350 system with no partitions that is connected to an L2 power controller
- `member2` is partition 3 of an Altix 3700 system that is connected to an L2 power controller
- The *tiebreaker* is the IP address of a network router that determines which member should have connectivity to the public network
- `service1` is the IP address will be used by clients to access the Samba share and NFS export point

Do the following:

1. Define shared state:

```
redhat-config-cluster-cmd --sharedstate --type=raw \
                          --rawprimary=/dev/shared_raw1 \
                          --rawshadow=/dev/shared_raw2
```

2. Create the cluster:

```
redhat-config-cluster-cmd --cluster --name "test-cluster"
```

3. Define the members:

```
redhat-config-cluster-cmd --add_member --name=member1 --watchdog=no

redhat-config-cluster-cmd --add_member --name=member2 --watchdog=no
```

4. Add power controller information for the members:

```
redhat-config-cluster-cmd --member=member1 --add_powercontroller --type=l2 \
                          --device=/dev/ttyIOC4/0 --partition=0

redhat-config-cluster-cmd --member=member2 --add_powercontroller --type=l2 \
                          --device=/dev/ttyIOC4/0 --partition=3
```

5. Change the heartbeat timeout to 20 seconds with heartbeat interval of 1 second, resulting in a failover speed of 20 seconds:

```
redhat-config-cluster-cmd --clumembd --interval=1000000 --tko_count=20
```

6. Set up a tiebreaker for the cluster:

```
redhat-config-cluster-cmd --cluquorumd --tiebreaker_ip=192.0.2.245
```

7. Create a failover domain with an ordered failover policy where the primary member is `member1` and the backup member is `member2`:

```
redhat-config-cluster-cmd --add_failoverdomain --name=domain1 \
                          --restricted=yes --ordered=yes

redhat-config-cluster-cmd --failoverdomain=domain1 --add_failoverdomainnode \
                          --name=member1

redhat-config-cluster-cmd --failoverdomain=domain1 --add_failoverdomainnode \
                          --name=member2
```

8. Create the service definition:

```
redhat-config-cluster-cmd --add_service --name=service1 --checkinterval=60  \
                          --servicetimeout=40 --monitorlevel="Check as client" \
                          --failoverdomain=domain1
```

9. Add a service IP address:

```
redhat-config-cluster-cmd --service=service1 --add_service_ipaddress \
                          --ipaddress=163.154.17.200 --netmask=255.255.255.0 \
                          --broadcast=163.154.17.255
```

10. Add the shared disk and filesystem information to `service1`:

```
redhat-config-cluster-cmd --service=service1 --add_device --name=/dev/shared1

redhat-config-cluster-cmd --service=service1 --device=/dev/shared1 --mount  \
                          --mountpoint=/mnt1 --fstype=xfs --options=rw,sync \
                          --forceunmount=yes
```

11. Add a Samba share name:

```
redhat-config-cluster-cmd --service=service1 --device=/dev/shared1 \
                          --sharename=share1
```

12. Define the NFS export point and NFS client information. The directory is exported to all clients with read-only access:

```
redhat-config-cluster-cmd --service=service1 --device=/dev/shared1 \
                          --add_nfsexport --name=/shared1/export_dir

redhat-config-cluster-cmd --service=service1 --device=/dev/shared1 \
                          --nfsexport=/shared1/export_dir --add_client \
                          --name=* --options=ro
```

13. Save the cluster configuration if you are using the GUI:

    File
    > Save

14. Synchronize the configuration changes. For example:

```
# scp /etc/cluster.xml root@member2:/etc/cluster.xml
root@member2's password:ENTER_ROOT_PASSWORD
cluster.xml                                   100% 3297    57.1MB/s   00:00
```

15. Start the SGI Cluster Manager daemons:

    a. Enter the following command:

    ```
    # chkconfig clumanager on
    ```

    b. Start local cluster daemons on each node in the cluster doing either of the following:

    ```
    service clumanager start
    ```

    ```
    /etc/init.d/clumanager start
    ```

For more information and additional examples, see the `redhat-config-cluster-cmd`(8) man page.

Administration

See Chapter 7, "Cluster Administration" in the *Red Hat Cluster Suite: Configuring and Managing a Cluster* manual.

Monitoring Status

To monitor status, use the following:

- The `redhat-config-cluster` GUI to monitor the status of the cluster and the services
- `clustat` to monitor the cluster status

Figure 5-1 shows an example of the GUI.

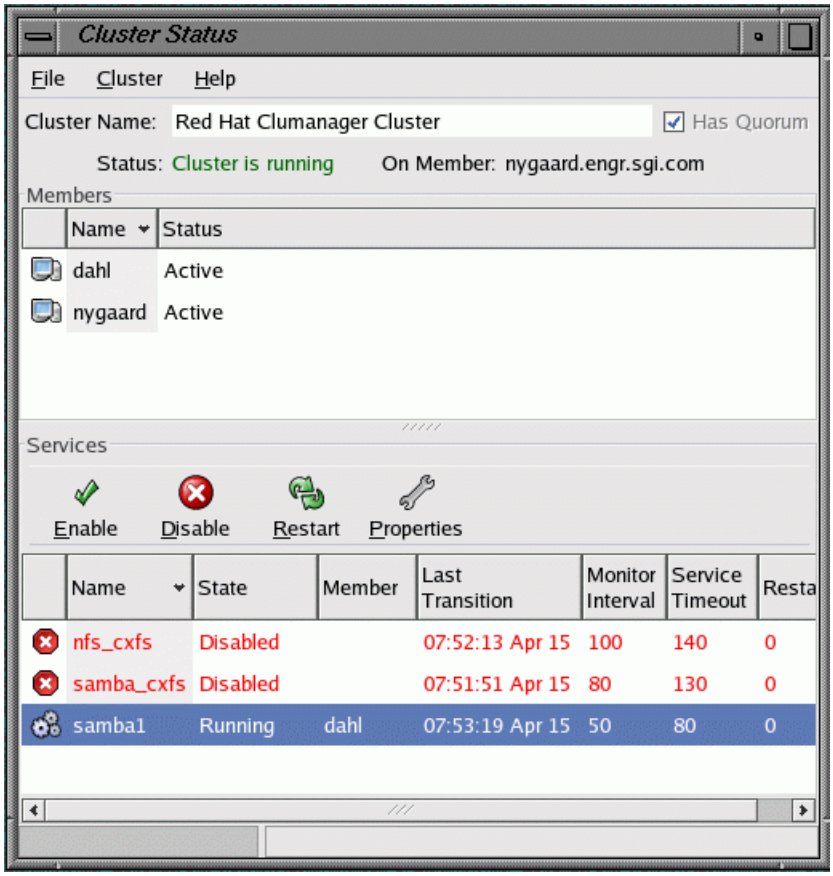

Figure 5-1 Status

Stopping Cluster Services

Use the following GUI selection in the **Cluster Status** window for all members or a specific member:

Cluster
> Stop Local Cluster Daemons

Use the following command:

```
/etc/init.d/clumanager stop
```

Service Administration

In the `redhat-config-cluster` GUI, use the **Cluster Status** window to enable or disable services. You can use drag and drop to relocate services.

You can also use the `clusvcadm` command to relocate, enable, or disable services.

Service States

Services can have one of the following states:

State	Description
Uninitialized	Transitioning when `clusvcmgrd` starts
Disabled	Not active
Pending	Transitioning to running or disabled
Running	Online
Failed	Needs attention. You must disable the service and then enable it.
Stopped	Disabled but will start when `clusvcmgrd` restarts

Message Logging

SGI Cluster Manager logs messages to `/var/log/messages` using the `syslog` facility `local4`. You can use `syslog.conf` to redirect messages to another location. To rotate logs, use `logrotate`(8).

SGI Cluster Manager uses the following message levels:

Level	Description
0	EMERG
1	ALERT
2	CRIT
3	ERROR
4	WARNING (default)
5	NOTICE
6	INFO
7	DEBUG

Creating a New Highly Available Application

All services in SGI Cluster Manager for Linux are managed by the `clusvcmgrd` daemon. The `clusvcmgrd` daemon does the following:

- Determines the cluster member where a service must run
- Processes service events
- Executes service scripts in a sequential manner

The `service` script starts, stops, or determines the status of given service. The `service` script takes the following parameters:

- An action, which can be one of `start`, `stop`, or `status`
- A service ID which identifies the service (the ID is automatically determined and is not user-configurable)

The `service` script runs application scripts in the following order:

- `start` order:

  `device` *(including local XVM volumes)*
  `filesystem` *(including CXFS)*
  `nfs`
  `ip address`
  `samba`
  user-defined application *(including DMF and TMF)*

- `stop` order:

  user-defined application *(including DMF and TMF)*
  `ip address`
  `nfs`
  `samba`
  `filesystem` *(including CXFS)*
  `device` *(including local XVM volumes)*

You cannot change the order in which the application scripts are run.

The status of applications in a service are checked in a sequential manner. If status of an application in the service fails, status of other applications are not checked.

Application-specific `start`, `stop`, and `status` scripts are present in the `/usr/lib/clumanager/services` directory. These scripts return `$FAIL` (value 1) on failure and `$SUCCESS` (value 0) on success.

To add a new application, you must write an application-specific set of scripts. The application-specific script (usually a `bash` shell script) takes an action (`start`, `stop`, or `status`) and service ID as parameter.

The newly written script is configured as a *user-defined script* parameter. You must add all devices and IP addresses that the application depends on to the service. NFS export points and Samba shares can also be part of the service.

The following command creates a service named `userapp` with the newly defined user script `new_application`:

```
redhat-config-cluster-cmd --add_service --name=userapp  \
                          --userscript=/usr/lib/clumanager/services/new_application   \
                          --checkinterval=40 servicetimeout=60
```

Figure 6-1 shows the GUI screen to create a service. To get to this window, click **New** with the focus on the service submenu in the configuration window.

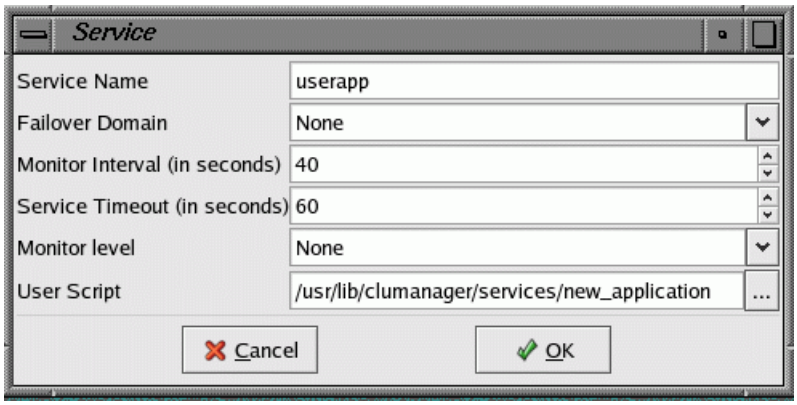

Figure 6-1 Creating a Service

You must copy the newly created script to the following location in all members in the cluster:

```
/usr/lib/clumanager/services/new_application
```

Samba Plug-In

The SGI Cluster Manager for Linux supports Samba 3.0 as shipped with SGI ProPack 3.

The Samba process ID (PID), locks, and password file are kept in the shared disk and in the log file on the local disk. The lock directory is not removed during failover.

The locations are as follows:

- PID directory: *mountpoint*`/.samba/`*sharename*`/pid`
- Log directory: `/var/log/samba`
- Lock directory: *mountpoint*`/.samba/`*sharename*`/locks`
- Password file: *mountpoint*`/.samba/`*sharename*`/private/smbpasswd`

For the order in which Samba is started/stopped, see Chapter 6, "Creating a New Highly Available Application" on page 39. For information about service monitoring levels, see "Step 8: Set the Service Monitoring Levels and Timeouts" on page 27.

CXFS Plug-In

Using the CXFS clustered filesystem product with SGI Cluster Manager for Linux requires the value-add SGI product on the *SGI Cluster Manager 3.0 for Linux - Storage Software Plug-ins* CD and CXFS 3.1 or later.

You should configure the CXFS cluster, nodes, and filesystems according to *CXFS Administration Guide for SGI InfiniteStorage*.

Note: To use CXFS relocation and failover the capability from one member to another, you must set the `relocation_ok` parameter to `1` (enable) on all potential CXFS metadata servers as follows:

```
# echo 1 > /proc/sys/fs/cxfs/cxfs_relocation_ok
```

For more information about relocation support, see *CXFS Administration Guide for SGI InfiniteStorage*.

To use SGI Cluster Manager for Linux with CXFS, all CXFS metadata servers must be on server types supported by SGI Cluster Manager for Linux; see "Hardware Requirements" on page 4.

In the GUI, click on **Add Child** with the focus on a particular service and choose **Add device** in the new window.

To include CXFS filesystems in the SGI Cluster Manager for Linux configuration, add filesystems as devices used by a service, as shown in Figure 8-1.

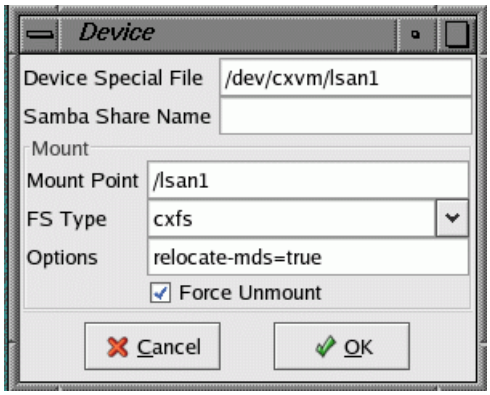

Figure 8-1 Adding a CXFS Filesystem as a Device

Enter the following:

- **Device Special File**: the block XVM device file
- **Mount Point**: the CXFS filesystem mount point
- **FS Type**: the filesystem type must be `cxfs`
- **Options**: one of the following:
  - `relocate-mds=true`, in which the metadata server for the CXFS filesystem is also failed over when service is failed over
  - `relocate-mds=false` (default)

Note: The **Force Unmount** item in the GUI or CLI is ignored for CXFS filesystems.

In the CLI, do the following:

```
# redhat-config-cluster-cmd --service=servname --add_device
                            --name=/dev/cxvm/volname

# redhat-config-cluster-cmd --service=servname --device=/dev/cxvm/volname \
                            --mount --mountpoint=mntpoint --fstype=cxfs
                            --options=relocate-mds=true|false
```

You can specify multiple CXFS filesystems by adding multiple devices to the service.

You should start CXFS cluster services and CXFS services before starting SGI Cluster Manager for Linux daemons. SGI Cluster Manager for Linux will wait for the CXFS filesystem to be mounted by CXFS before starting NFS, Samba, and other applications running on the CXFS filesystem. Therefore, service timeouts for all SGI Cluster Manager for Linux services that include CXFS filesystems should be carefully adjusted accordingly.

The members in the failover domain for the service that has CXFS filesystems should be same as the list of potential metadata servers for the CXFS filesystem. For example: machines `node1` and `node2` can be metadata servers for CXFS filesystem `/cxfs_san1`. The SGI Cluster Manager for Linux service `nfs1` that uses `/cxfs_san1` should have a failover domain of `node1` and `node2`.

For the order in which CXFS is started/stopped, see Chapter 6, "Creating a New Highly Available Application" on page 39.

Data Migration Facility (DMF) Plug-In

Using the Data Migration Facility (DMF) with SGI Cluster Manager for Linux requires the value-add SGI product on the *SGI Cluster Manager 3.0 for Linux - Storage Software Plug-ins* CD and DMF 3.0.1 or later.

You should configure DMF according to *DMF Administrator's Guide for SGI InfiniteStorage*.

Adding the DMF User Script to an Existing Service

The following command adds the DMF user script to an existing service. The script used is `/usr/lib/clumanager/services/svclib_dmf`.

```
redhat-config-cluster-cmd --service=service1   \
                          --userscript=/usr/lib/clumanager/services/svclib_dmf
```

DMF Administrative Filesystems and Directories

To run DMF, you must configure the parameters shown in Table 9-1. A *required* parameter must be defined by all users of DMF. An *optional* parameter is needed only by users of certain MSP types. DMF cannot start unless the required filesystems and directories defined by these parameters are first mounted and available on shared disks.

Table 9-1 DMF Administrative Filesystem and Directory Parameters

Parameter	Status	Description
`HOME_DIR`	Required	Specifies the DMF databases
`JOURNAL_DIR`	Required	Specifies the DMF database journals
`SPOOL_DIR`	Required	Specifies the DMF log files
`MOVE_FS`	Optional	Moves files between MSPs

Parameter	Status	Description
CACHE_DIR	Optional	Used by a library server as a cache for merging data from sparse tapes to new tapes
FTP_DIRECTORY	Optional	Used by an FTP MSP to store files
STORE_DIRECTORY	Optional	Used by a disk MSP to store files

In addition, the working directory used by the `dmaudit` command must also be available when DMF starts. To configure the directory, run the `dmaudit` command and select the `<workdir>` item in the `<config>` menu.

You can configure the DMF administrative filesystems as local XVM filesystems or as CXFS filesystems. SGI Cluster Manager ensures that the DMF plug–in script is called after the necessary filesystems are mounted.

If you use local XVM filesystems, you must define them as instructed in "Configuring DMF for Local XVM Filesystems" on page 50.

If you use CXFS filesystems, you just define them as instructed in "Configuring DMF for CXFS Filesystems " on page 50

Configuring DMF for Local XVM Filesystems

To configure the DMF administrative filesystems as local XVM filesystems, do the following:

1. Ensure that the DMF configuration is identical on all members.
2. Create the DMF administrative filesystems on shared disks as local XVM filesystems.
3. Configure the SGI Cluster Manager local XVM volumes using the local XVM plug-in.

Configuring DMF for CXFS Filesystems

DMF cannot start until the DMF administrative filesystems are available. If they are CXFS filesystems, CXFS must recover them before they are accessible.

To configure DMF filesystems as CXFS filesystems, do the following:

1. Ensure that the DMF configuration is identical on all members.
2. Create the DMF administrative filesystems as CXFS filesystems.
3. Configure the SGI Cluster Manager CXFS filesystems using the CXFS plug-in. For DMF-managed filesystems, configure `relocate-mds=true` (on) because DMF must run on the CXFS metadata server for that filesystem.

Start/Stop Order

For the order in which DMF is started/stopped, see Chapter 6, "Creating a New Highly Available Application" on page 39.

Ensuring that Only SGI Cluster Manager Starts DMF

After installing `clumanager-sgi-1.0.0`. you should perform the following actions on each member of the cluster in the failover domain. These commands ensure that DMF can only be started via SGI Cluster Manager:

```
[root@prod root]# touch /etc/dmf_failsafe
[root@prod root]# chkconfig dmf off
```

Using TMF with DMF

To use the Tape Management Facility (TMF) with DMF in a SGI Cluster Manager environment, you must configure the appropriate TMF device groups in the `/etc/tmf/sgicm_tmf.config` file according to the instructions in Chapter 10, "Tape Management Facility (TMF) Failover Script" on page 53.

If TMF is configured as a mount service in the `/etc/dmf/dmf.conf` file, the DMF plug-in will automatically call the `/usr/lib/clumanager/service/helper_tmf` TMF failover script and pass along the appropriate TMF device group names.

The service timeout value should be at least 100 seconds if DMF is being used with TMF-managed tape devices. The following command will set the service timeout to 100 seconds for the SGI Cluster Manager service `service1`:

```
redhat-config-cluster-cmd --service service1 --servicetimeout=100
```

Tape Management Facility (TMF) Failover Script

Using the Tape Management Facility (TMF) with SGI Cluster Manager for Linux requires the value-add SGI product on the *SGI Cluster Manager 3.0 for Linux - Storage Software Plug-ins* CD and TMF 1.4.1 or later. Only Storage Technology Corporation (STK) hardware controlled by the Automated Cartridge System Library Software (ACSLS) software is supported.

If your application requires tape support via TMF, then your user application script should call the `/usr/lib/clumanager/service/helper_tmf` TMF failover script, passing the appropriate parameters. See "Using the TMF Failover Script from the User Application Script" on page 59.

The DMF plug-in will automatically call the `helper_tmf` script if a Library Server Drive Group uses TMF as a mounting service.

The `helper_tmf` script lets you manage one or more *TMF device groups*, which are sets of tape devices defined in the `/etc/tmf/tmf.config` TMF configuration file. The following example is part of a `/etc/tmf/tmf.config` that defines a TMF device group named `EGLF`:

```
DEVICE_GROUP
                name = EGLF
        AUTOCONFIG
{
                DEVICE
                        NAME   = f9840f1 ,
                        device_group_name = EGLF ,
                        FILE    = /hw/tape/500104f000417a18/lun0/c4p1 ,
                        status = down ,
                        access = EXCLUSIVE ,
                        vendor_address = (1,0,0,2),
                        LOADER = l180
}
```

The `helper_tmf` script performs the following functions for the calling user service or `userapp` script:

- Starts TMF if it is not already running.
- Configures the associated loader up if it is not already up.

- Allows the monitoring of multiple TMF device groups and their associated tape devices.
- Monitors the number of tape devices that are available within each TMF device group. If the number of devices currently available is less than the minimum threshold level, a monitoring failure will occur.
- Releases previous reservations that are held by another member (if the tape device firmware supports this option).
- Lets you assign different TMF device groups to each instance of an SGI Cluster Manager service or `userapp` script.

For the order in which TMF is started/stopped, see Chapter 6, "Creating a New Highly Available Application" on page 39.

Configuring a TMF Device Group

The `helper_tmf` script lets you specify device groups to stop, start, and monitor. Each of these managed device groups must be defined in the following files:

- `/etc/tmf/sgicm_tmf.config` (SGI Cluster Manager configuration file for TMF)
- `/etc/tmf/tmf.config` (standard TMF configuration file)

The `resource` directive in the `/etc/tmf/sgicm_tmf.config` file specifies a TMF device group. This directive is required for each TMF device group that you plan to use within SGI Cluster Manager. See "The `resource` Directive " on page 56.

Optional Configuration Specifications

There are other optional configuration specifications associated with a TMF device group. These specifications provide information to the `helper_tmf` script that lets it communicate with the tape library. They also identify the tape devices within the library on which `helper_tmf` will force dismounts.

The `helper_tmf` script can force a dismount of tapes from devices within the library. There may be various reasons why you might want to do this when a failover occurs. In the case of DMF, you would want to ensure that any DMF tapes that were in use on a previous member are available to DMF on the new member after a failover. If these tapes were in tape devices assigned to the previous member, they must be

ejected and returned to the library so that they are again accessible to DMF on the new member. You may want the `helper_tmf` script to dismount only tape devices associated with a particular TMF device group or you may not want the `helper_tmf` script to dismount any tapes at all.

Some of the functions of the `helper_tmf` script are performed through TMF; the script issues commands to the TMF daemon to use these functions. However, the `helper_tmf` script forces a dismount of a tape from a device by issuing a command to the library software controlling the loader/library. The `helper_tmf` script communicates its request to the ACSLS software that controls the loader. The `helper_tmf` script uses an `expect` script that issues commands to login to the loader and issue a dismount request to a tape device.

The `/etc/tmf/sgicm_tmf.config` File

The `/etc/tmf/sgicm_tmf.config` file lets you configure other information required by the `helper_tmf` script. The `sgicm_tmf.config` file exists on all members in the cluster and should be edited as necessary on each member.

The contents of the `sgicm_tmf.config` file are dependent on the tape devices assigned to each member in the cluster. If all members in the failover domain are configured through TMF to use exactly the same tape devices, this file would be the same on each member in the failover domain.

Note: You must maintain the `sgicm_tmf.config` file on each member; a change on one member is unknown to the other members.

You can specify the following directives in the `sgicm_tmf.config` file:

- "The `resource` Directive " on page 56
- "The `loader` Directive " on page 56
- "The `remote_devices` Directive" on page 57

The `resource` Directive

The `resource` directive defines the TMF device groups that can be managed by the `helper_tmf` script:

`resource` *device-group devices-minimum devices-loaned email-addresses*

where:

device-group	The TMF device group that is to be monitored. This is a device group that is defined in `/etc/tmf/tmf.config`.
devices-minimum	The minimum number of devices of the specified *device-group* that you must have available to you on a member before you failover.
devices-loaned	Currently unused; should be set to `0`.
email-addresses	List of addresses to send email when the monitor script detects that tape devices in the *device-group* have become unavailable. Corrective action can then be taken to repair the tape devices before the *devices-minimum* threshold is crossed. This may be a comma- or white-space-separated list of names.

The `loader` Directive

The `loader` directive provides information about a TMF loader, which controls one or more tape devices that are members of TMF device groups being managed by SGI Cluster Manager. There may be multiple `loader` directives in the `sgicm_tmf.config` file.

The loader information is used by the `helper_tmf` script to force a dismount of tapes from tape devices that cannot be made available (that is, that have `tmstat` states other than `assn`, `free`, `conn`, or `idle`) so that those tapes can be used via other tape devices in the same device group. The information is also used to force a dismount of tapes from devices that are only connected to the other member, not to this member (as described in "The `remote_devices` Directive" on page 57).

If the file does not contain a `loader` directive, the `helper_tmf` script will make no attempt to force a dismount of tapes from any devices.

The directive has the following format:

`loader` *lname ltype lhost luser lpswd*

where:

lname	Name of the loader as defined in `/etc/tmf/tmf.config`
ltype	Type of the loader as defined in `/etc/tmf/tmf.config`, which must be `STKACS`
lhost	Server name of the loader as defined in `/etc/tmf/tmf.config`
luser	User name of the loader's administrator account, which must be `acssa`
lpswd	Password for the loader's administrator account

The TMF command `/usr/sbin/tmmls` shows the name of the loader and the server associated with it:

```
# tmmls
loader type status m  server   old  m_pnd  d_pnd  r_qd  comp  avg
 operator  OPERATOR      UP A  IRIX       0      0      0     0     0    0(sec)
 wolfy       STKACS    DOWN A  wolfcree   0      0      0     0     0    0(sec)
 panther     STKACS    DOWN A  stk9710    0      0      0     0     0    0(sec)
 l180        STKACS      UP A  stk9710    0      0      0     0     0    0(sec)
```

For example, suppose you want to have the `helper_tmf` script dismount tape devices that are in the `l180` loader/library listed above. That library has the `stk9710` server associated with it. The loader directive in the `sgicm_tmf.config` file would look like the following:

```
loader l180 STKACS stk9710 acssa acssapassword
```

If the initial attempt to configure the device up fails, the `helper_tmf` script would force a dismount for each tape device that is specified in the `tmf.config` file to be in the `l180` loader/library and in the TMF device group. If you do not want the script to dismount any tape devices associated with a particular TMF device group, you would not place a loader directive in the `sgicm_tmf.config` file.

The `remote_devices` Directive

The `remote_devices` directive provides information about one or more tape devices that are part of a TMF device group, but which are not visible on this member.

For example, suppose you have a library with four SCSI tape devices where two tape devices are connected to each of two cluster members. If member A should crash, member B must be able to force a dismount of any tapes in A's tape devices so that they can then be used from member B. Because the tape devices are not visible on member B, the `remote_devices` directive provides the information needed to force a dismount of unseen tape devices.

The directive has the following format:

`remote_devices` *device-group* *lname* *tape-device-ID* `...`

where:

device-group	Name of the TMF device group with which the *tape-device-IDs* are associated
lname	Name of the loader as defined in `/etc/tmf/tmf.config`. There must be a `loader` directive for *lname* elsewhere in this file, or the `remote_devices` directive will be ignored.
tape-device-ID	The vendor ID of the drive on which to force a dismount. This is the unique name by which the loader identifies the tape device. For `STKACS`, this will be a comma-separated four-digit string listing the ACS, LSM, drive panel, and drive (for example, `0,0,1,3`).

Note: No blanks should exist within the ID.

You can specify multiple vendor IDs in the same `remote_devices` directive as long as they all pertain to the same loader. If all the vendor IDs will not fit on a single line, add additional `remote_devices` directives for the same loader. For example, to enable the `helper_tmf` script to force a dismount of the remote tape devices `0,0,1,0` , `0,0,1,1` , `0,0,1,2` , and `0,0,1,3` in the `l180` loader/library for TMF device group `tmf_eglf`, the directive would be:

```
remote_devices tmf_eglfl  180  0,0,1,0  0,0,1,1  0,0,1,2  0,0,1,3
```

If multiple TMF device groups are defined, only the TMF device group named `tmf_eglf` will force a dismount of these tape devices.

Configuring Tapes and TMF

If tape devices that are managed by the `helper_tmf` script are configured on more than one member in the cluster, they should be configured consistently. The same tape driver (for example, `ts`) should be used on each member where the tape device is configured.

When configuring the `helper_tmf` script, you should be aware of several parameters in the `/etc/tmf/tmf.config` file. The `helper_tmf` script will try to start the loader associated with its device-group if it is not up. However, if the configuration file specifies `status=UP` for the loader, this step may not be necessary and the devices may become available sooner.

A tape device that is managed by the `helper_tmf` script will be configured in `/etc/tmf/tmf.config` on one or more members within the cluster. It should be configured with `status=down`.

If the tape devices being used do not support persistent reserve, then they should each be configured in `/etc/tmf/tmf.config` with `access=shared`. If the tape devices do support persistent reserve, it is recommended that you use this feature when using the `helper_tmf` script. To use persistent reserve, you should set `access=exclusive` in `/etc/tmf/tmf.config` for each tape device. The access option should be consistent across all members in the cluster where the tape devices are configured.

The `-g` option of the `tmconfig` command reassigns a device to a different device group name. The `helper_tmf` script does not support reassigning a device into a device group. That is because, in case of failover, the `helper_tmf` script on the member we have failed over to would not have any knowledge of this reassigned tape device . It would not be able to dismount tapes that are in the tape device . If you use `tmconfig -g` to move devices out of a device group, that will decrease the number of available tape devices that the monitor function of the `helper_tmf` script can detect. Also, in the case of failover or stop, the tape device will be configured down.

Using the TMF Failover Script from the User Application Script

In order to manage TMF device groups in an SGI Cluster Manager environment, the user application script must pass the appropriate parameters to the TMF failover script. This script called via:

`/usr/lib/clumanager/scripts/helper_tmf` *action* *device-groups*

where:

action	One of `start`, `stop`, or `status`
device-groups	one or more TMF device groups upon which the action should be taken

Note: It is more efficient to invoke the `helper_tmf` script once with several device-groups rather than invoking it several times with a single device-group.

For example, to start the 9840 device group:

```
/usr/lib/clumanager/services/helper_tmf start 9840
if [ $? -ne 0 ]; then
               logAndPrint $LOG_ERROR "start of 9840 device group failed"
               return 1;
fi
```

To stop the 9840 device group:

```
/usr/lib/clumanager/services/helper_tmf stop 9840
if [ $? -ne 0 ]; then
               logAndPrint $LOG_ERROR "unable to stop 9840 device group"
               return 1;
fi
```

To check the status of the 9840 device group:

```
/usr/lib/clumanager/services/helper_tmf status 9840
if [ $? -ne 0 ]; then
               logAndPrint $LOG_ERROR "device group 9840 not running"
               return 1;
fi
```

Service Timeout

The service timeout for the calling `userapp` or user script should be at least 100 seconds. The following command will set the service timeout to 100 seconds for the SGI Cluster Manager service `service1`:

```
redhat-config-cluster-cmd --service service1 --servicetimeout=100
```

Local XVM Plug-In

SGI Cluster Manager supports failover of XVM volumes in *local* mode. This support is available as part of the `clumanager-sgi` RPM in on the *SGI Cluster Manager 3.0 for Linux - Storage Software Plug-ins* CD.

Note: XVM in **cluster** mode is supported only with CXFS. See Chapter 8, "CXFS Plug-In" on page 45.

Local XVM devices are configured as a device for a service. You can specify multiple XVM devices for a service. For each local XVM volume device, specify the list of physical volumes that it contains, separating each element in the list by a comma (`,`) character.

Following is an example to fail over local XVM volume `m0`:

1. Install and configure XVM on both members in the cluster.
2. Find the physical volumes that are part of volume `m0`:

```
# xvm
xvm:local> show -topology -extend vol/m0
vol/m0                          0 online,open
    subvol/m0/data           497824768 online,open
        stripe/stripe0           497824768 online,tempname,open (unit size: 128)
            mirror/mirror8            35558944 online,tempname,open
                slice/dks5d1s0            35558944 online,open (dks5d1:/dev/rdsk/dks5d1vol)
                slice/dks4d1s0            35558944 online,open (dks4d1:/dev/rdsk/dks4d1vol)
            mirror/mirror4            35558944 online,tempname,open
                slice/dks11d1s0           35558944 online,open (dks11d1:/dev/rdsk/dks11d1vol)
                slice/dks7d1s0            35558944 online,open (dks7d1:/dev/rdsk/dks7d1vol)
```

   The list of physical volumes that belong to volume `m0` are `dks5d1`, `dks4d1`, `dks11d1`, and `dks7d1`.

3. Add the device to the service using the `redhat-config-cluster` GUI or the `redhat-config-cluster-cmd` command-line interface. The device name will be `/dev/lxvm/m0` and the physical volumes will be `dks5d1,dks4d1,dks11d1,dks7d1`.

For example, the following output from the CLI after the `device` item shows the information that has been added to service `nfs1`:

```
# redhat-config-cluster-cmd --service=nfs1 --device=
/dev/lxvm/m0

device:
  name = /dev/lxvm/m0
  sharename =   physvols = dks5d1,dks4d1,dks11d1,dks7d1

mount:
  mountpoint = /mnt5
  fstype = xfs
  options = rw
  forceunmount = yes

nfsexport:
  name = /mnt5

client:
  name = challenger.engr.sgi.com
  options = rw
```

Note: SGI Cluster Manager uses the `xvm` subcommands `give` and `steal` during failover for local XVM volumes. However, the list of physical volumes can be specified or modified only if the `clumanager-sgi` RPM is installed on the member.

Figure 11-1 shows an example in the GUI.

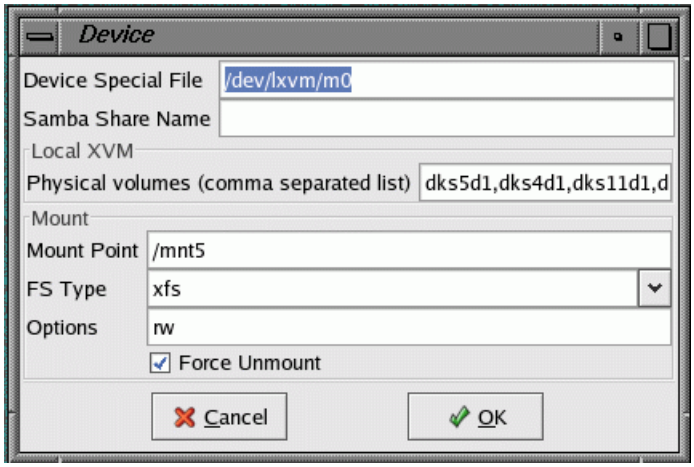

Figure 11-1 Adding an XVM Device

For the order in which local XVM is started/stopped, see Chapter 6, "Creating a New Highly Available Application" on page 39.

Troubleshooting

This chapter provides information about the following:

Best Practices

If you run into problems, do the following:

- Check the messages in `/var/log/messages`. See "Message Logging" on page 37.
- Use `shutil` to see if shared partitions are accessible.
- Use `clufence` to check the status of the reset cable
- Verify that the failover domain is defined correctly

Recovery from a `clulockd` Failure

If the `clulockd` daemon dies unexpectedly, it freezes all of the locks on the shared partition. `clulockd` will log a message such as the following in the logs:

```
Feb  6 17:25:14 3U:nygaard clulockd[6924]:  Signal 11 received; freezing
```

The `clusvcmgrd` daemon will not be able to monitor, start, or stop services . Logs on all members will have a message such as the following:

```
Feb  6 17:14:48 2U:dahl clusvcmgrd[3255]:  Couldn't connect to member #0: Connection timed out
Feb  6 17:14:48 3U:dahl clusvcmgrd[3255]:  Unable to obtain cluster lock: No locks available
```

To recover from this situation, do the following:

1. Stop cluster daemons on all members.
2. Reinitialize the shared state from one member in the cluster:

   ```
   shutil -i
   ```

3. Make sure that `/etc/cluster.xml` is same on all members.
4. Initialize the configuration on the shared partition from one member in the cluster:

   ```
   shutil -s /etc/cluster.xml
   ```

5. Verify that the configuration has been initialized correctly from one member in the cluster:

   ```
   shutil -p /cluster/config.xml
   ```

For more information, see the `shutil` man page.

Watchdog Errors

Software and hardware watchdog timers are not supported. If you enable software watchdog when configuring a member, you will see the following error messages when cluster daemons are starting:

```
Creating /dev/watchdog: execvp: No such file or directory
^[[FAILED]
Loading Watchdog Timer (softdog):  modprobe: Can't locate module softdog
^[[FAILED]
```

Shared Raw Partitions

This section discusses the following:

Verify Raw Devices are Character Special Devices

Use the following command to verify that the `rawprimary` and `rawshadow` devices are character special device (such as `rpart1`) :

`ls -lL` ***share_raw_device***

Using block special devices (such as `part1`) will lead to inconsistencies in cluster, member, and node states. The cluster appears to work but communication between members is affected. For an example, see "Verify Metadata Information is Consistent" on page 68.

See "Step 1: Define Shared State" on page 20.

Verify Accessibility

To see if shared partitions are accessible, enter the following:

```
shutil
```

Read the Configuration File

To read the configuration file from the shared partition, enter the following:

```
shutil -r -
```

You should use this command to compare the configuration files in the shared partitions and the local copy.

Verify Metadata Information is Consistent

To verify that the service metadata information is the same on all members, run the following command at the same time on each member:

```
shutil -m /service/0/status
```

For example, the following output from member `jackhammer` and member `jackhammer2`indicates a problem:

```
[root@jackhammer root]# shutil -m /service/0/status
Metadata information for /service/0/status

Data Length:   40 bytes
Data CRC:      0x2dae1205
Header CRC:    0x7c7185f1
Last modified: 12:34:58 Mar 31 2004

[root@jackhammer2 root]# shutil -m /service/0/status
Metadata information for /service/0/status

Data Length:   40 bytes
Data CRC:      0x80711487
Header CRC:    0x9ba9e2cf
Last modified: 12:34:51 Mar 31 2004
```

In this case, the service metadata information from both members is inconsistent (the `CRC` and the `Last modified` time stamps are different). The information must be identical from all the members. See "Verify Raw Devices are Character Special Devices" on page 67.

Write the Configuration File

To write the configuration file:

```
shutil -s /etc/cluster.xml
```

You should use this command if one of the following is true:

- The configuration file in the shared partitions is not consistent with the `/etc/cluster.xml` file
- The shared partition partition was cleared using the `shutil -i` command

Displaying Metadata Remotely

To display the metadata information from the shared partition, use the following command:

```
shutil -p /service/0/status
```

Last Resort: Clear Information

Caution: Do not run this command while the cluster is enabled.

To clear all cluster information:

```
shutil -i
```

State Inconsistencies

If you encounter state inconsistencies between members, verify that the shared raw devices are character special devices (such as `rpart1`) and not block special devices (such as `part1`). See "Verify Raw Devices are Character Special Devices" on page 67.

Serial cable or Reset issues

The `clufence` command will fail with a nonzero error code for the following reasons:

- The serial cable is not connected
- The cable is faulty
- The system controller is not responding

The messages shown in the following output are also logged to `/var/log/messages`:

```
# clufence -s jackhammer2
[12314] info: STONITH: Power controller l2 connected to peer's /dev/ttyIOC4/0 controls jackhammer
[12314] info: STONITH: Power controller l2 connected to peer's /dev/ttyIOC4/0 controls jackhammer2
[12314] err: STONITH: Device at /dev/ttyIOC4/0 controlling jackhammer2 FAILED status check:
Timed out
```

Error Messages

Following are common error messages.

```
Raw device file names must be defined.
```

An attempt was made to define the cluster before defining the shared state. You must define the shared state first. See "Step 1: Define Shared State" on page 20.

```
Raw device file names primary /dev/raw/raw1 and shadow /dev/raw/raw43 are not valid.

Shared storage initialization failed.
Fix shared storage and write configuration file to shared storage.
Continuing ...
```

The shared partitions are not accessible or not valid and a configuration change or query was made using the CLI.

```
[12314] err: STONITH: Device at /dev/ttyIOC4/0 controlling jackhammer2 FAILED status check:
Timed out
```

There is a problem with the serial cable or system controller. See "Serial cable or Reset issues" on page 69.

Reporting Problems to SGI

If you encounter problems, collect the following data from each member:

- Output from the following commands:

  ```
  hinv
  chkconfig --list
  shutil -r -
  rpm -qa
  uname -a
  ps -ef | grep clu
  clustat
  ls -l each_shared_raw_device
  ls -lL  each_shared_raw_device
  clufence -s other_members
  exportfs   (in NFS configurations)
  mount
  cxfs_dump   (in CXFS configurations)
  ```

- Contents of the following files:

  ```
  /etc/cluster.xml
  /usr/lib/clumanger/create_device_links
  /var/log/messages
  /etc/samba/smb.conf*  (in Samba configurations)
  ```

FailSafe and SGI Cluster Manager

Table A-1 summarizes the differences between IRIX FailSafe and SGI Cluster Manager for Linux for those readers who may be familiar with FailSafe.

Note: SGI Cluster Manager for Linux members and FailSafe nodes cannot be work together and cannot form a high-availability cluster.

Table A-1 Differences Between FailSafe and SGI Cluster Manager

Topic	FailSafe	SGI Cluster Manager
Operating system	IRIX	Red Hat Advanced Server for Linux 64-bit with SGI ProPack 3
Terminology	node resource	member application
Size of cluster	8 nodes	2 members
NFS lock failover	Supported	Not supported
Tiebreaker	A node that is participating the cluster membership. FailSafe tries to include the tiebreaker node in the membership in case of split-brain scenarios.	The IP address of machine or a router that **does not participate** in the cluster membership. Usually it is the IP address of a network router that connects the SGI Cluster Manager members to the external world (clients). In a split-brain scenario, only those members that can contact the tiebreaker IP address can form a cluster.
Rolling upgrade	Supported	Not supported

Topic	FailSafe	SGI Cluster Manager
Configuration information	Information is stored in the cluster database. The cluster database is replicated on all nodes automatically and kept in synchronization.	Information is stored in the `/etc/cluster.xml` configuration file and in shared storage. You must copy this file to all members, such as by using `scp`. After making configuration changes, you must verify that configuration files are in synchronization. See "Step 14: Synchronize Configuration Changes Across the Cluster" on page 30.
Making changes while the service is enabled	Device parameter, IP address parameters, and check interval can be changed.	Device parameter, IP address parameters, and check interval cannot be changed.
Script location for resources and resource types	`/var/cluster/ha/resource_types`	`/usr/lib/clumanager/services/service`
Heartbeat interval and timeout	You can specify cluster membership heartbeat interval and timeout (in milliseconds).	In the command line, you can specify the heartbeat interval (in microseconds) and the number of heartbeats that can be consecutively missed (`tko_count`). You can also specify the aggregate failover speed in the GUI.
Private network	Supports multiple networks for heartbeats and control messages and can failover from one network to another. Networks can be dedicated (private) or public networks.	Not supported. SGI Cluster Manager uses the public network (host names) for heartbeats and control messages.
Action scripts	Separate scripts named `start`, `stop`, `monitor`, `restart`, `exclusive`.	A `bash` script that contains `start`, `stop`, and `status` parameters (see Chapter 6, "Creating a New Highly Available Application" on page 39). The equivalent for `restart` in SGI Cluster Manager is to perform a `stop` and then a `start`; there is no no equivalent in SGI Cluster Manager for `exclusive`.

Topic	FailSafe	SGI Cluster Manager
Resource timeouts	Timeouts can be specified for each action (`start`, `stop`, `monitor`, `restart`, `exclusive`) and for each resource type independently.	Timeout can be specified for each service irrespective of the action or the number of resources it contains.
Resource dependencies	Resource and resource type dependencies are supported and can be modified by the user.	Applications have fixed dependencies. The start and stop order of applications cannot be modified.
Failover policies	The ordered and round-robin failover policies are predefined. User-defined failover policies are supported.	Only the predefined ordered policy is supported. No user-defined failover policies are supported.

Index

J

L

M

N

P

T

U

V

W

X