TruCluster Server

Cluster Administration

Part Number: AA-RHGYD-TE

June 2001

Product Version: TruCluster Server Version 5.1A

Operating System and Version: Tru64 UNIX Version 5.1A

This manual describes how to manage systems that run TruCluster Server software.

Compaq Computer Corporation
Houston, Texas

Contents

3 Managing the Cluster Alias Subsystem

4 Managing Cluster Membership

5 Managing Cluster Members

6 Managing Networks in a Cluster

7 Managing Network Services

8 Managing Highly Available Applications

9 Managing File Systems and Devices

10 Using Logical Storage Manager in a Cluster

11 Troubleshooting Clusters

Tables

About This Manual

This manual describes how to perform tasks related to the day-to-day management of a TruCluster™ Server system. In addition to discussing the management of quorum and votes, and the cluster file system, this manual describes how to manage the device request dispatcher and configure network services in a cluster. It also gives some suggestions for managing a cluster.

Audience

This manual is for the person who will configure and manage a TruCluster Server cluster. Instructions in this manual assume that you are an experienced UNIX administrator who can configure and maintain hardware, operating systems, and networks.

New and Changed Features

The following changes have been made to this manual since the Version 5.1 release:

- *Chapter 1* presents up-to-date information on the differences between managing a standalone Tru64 UNIX™ operating system software system and a TruCluster Server, including information on new `LSM` commands.
- *Chapter 2* describes the graphic user interfaces (GUI) and command-line tools for managing clusters, including new information about Compaq Insight Manager XE™.
- *Chapter 3* explains how you use cluster aliases to provide network applications with a single-system view of the cluster. This chapter includes new information about cluster alias support for Network File System (NFS) clients using non-default cluster aliases.
- *Chapter 5* describes how to configure, manage, and remove cluster members. This chapter includes new information about shutting down and starting cluster members and changing the cluster name or IP address.
- *Chapter 7* describes how to configure mail, printing, and other services in a cluster. This chapter includes new information about using context-dependent symbolic links (CDSLs) to mount NFS file systems, using AutoFS in a cluster, and forcibly unmounting file systems.

- *Chapter 9* describes how to manage the cluster file system and the device request dispatcher, how to add and remove storage devices, and how to load-balance disk servers. This chapter includes new information about direct access cached reads, user and group file system quota support, and read/write support for Memory File System (MFS) and UNIX File System (UFS) file systems.
- *Chapter 10* describes how to use the Logical Storage Manager (LSM) software in a cluster. This chapter includes new information about LSM support for mirrored root (/) and swap file systems.
- *Chapter 11* discusses how to investigate and resolve common TruCluster Server problems. This chapter includes an updated procedure for recovering the cluster root file system to a new disk.

Organization

This manual is organized as follows:

Chapter 1	Presents the differences between managing a standalone Compaq Tru64™ UNIX operating system software system and a TruCluster Server.
Chapter 2	Describes the graphic user interfaces (GUI) and command-line tools for managing clusters.
Chapter 3	Explains how you use cluster aliases to provide network applications with a single-system view of the cluster.
Chapter 4	Describes how to manage quorum and votes to maintain cluster availability.
Chapter 5	Describes how to configure, manage, and remove cluster members.
Chapter 6	Discusses how to configure and administer member and client networks in a cluster.
Chapter 7	Describes how to configure mail, printing, and other services in a cluster. Presents methods to provide highly available network services.
Chapter 8	Describes day-to-day tasks involved in managing highly available applications.
Chapter 9	Describes how to manage the cluster file system and the device request dispatcher, how to add and remove storage devices, and how to load-balance disk servers.
Chapter 10	Describes how to use the Logical Storage Manager (LSM) software in a cluster.
Chapter 11	Discusses how to investigate and resolve common TruCluster Server problems.

Appendix A	Lists events that are specific to TruCluster Server systems.
Appendix B	Lists the configuration variables that are provided with TruCluster Server.
Appendix C	Presents an example of a cluster-member delete log, `/cluster/admin/clu_delete_member.log`.

Related Documents

Consult the following TruCluster Server documentation for assistance in cluster configuration, installation, and administration tasks:

- *TruCluster Server Software Product Description (SPD)* — Presents the comprehensive description of the TruCluster Server Version 5.1A product.

  You can find the latest version of the SPD at the following URL: **`http://www.tru64unix.compaq.com/docs/pub_page/spds.html`**.
- *Cluster Technical Overview* — Presents concepts that are necessary for the effective management of a TruCluster Server system. Read this manual before you begin cluster installation.
- *Cluster Release Notes* — Documents new features, known restrictions, and other important information about the TruCluster Server software products.
- *Cluster Hardware Configuration* — Describes how to set up the processors that are to become cluster members, and how to configure cluster shared storage.
- *Cluster LAN Interconnect* — Describes how to use LAN hardware as the cluster interconnect.
- *Cluster Installation* — Describes how to install the TruCluster Server software on the systems that are to participate in the cluster.
- *Cluster Highly Available Applications* — Documents how to make applications highly available. Describes the application programming interfaces (APIs) provided by the distributed lock manager (DLM), cluster alias, and Memory Channel subsystems.

You can find the latest version of the TruCluster Server documentation at: **`http://www.tru64unix.compaq.com/docs/pub_page/cluster_list.html`**.

Because the administration of a TruCluster Server system is, by design, very similar to that of a Tru64 UNIX Version 5.1A system, the following Tru64 UNIX Version 5.1A operating system manuals provide useful information:

- Tru64 UNIX *Release Notes*

- Tru64 UNIX *System Administration*
- Tru64 UNIX *Network Administration: Connections*
- Tru64 UNIX *Network Administration: Services*
- Tru64 UNIX *Logical Storage Manager*
- Tru64 UNIX *AdvFS Administration*
- Tru64 UNIX *System Configuration and Tuning*
- Tru64 UNIX *Security*
- Tru64 UNIX *Programmer's Guide*
- Tru64 UNIX *Sharing Software on a Local Area Network*
- Tru64 UNIX *Advanced Printing Software Release Notes*.

For help with storage management, see the following manuals:

- POLYCENTER Advanced File System Utilities *Reference Manual*
- POLYCENTER Advanced File System Utilities *Release Notes*
- POLYCENTER Advanced File System Utilities *Installation Guide*

Icons on Tru64 UNIX Printed Manuals

The printed version of the Tru64 UNIX documentation uses letter icons on the spines of the manuals to help specific audiences quickly find the manuals that meet their needs. (You can order the printed documentation from Compaq.) The following list describes this convention:

G	Manuals for general users
S	Manuals for system and network administrators
P	Manuals for programmers
R	Manuals for reference page users

Some manuals in the documentation help meet the needs of several audiences. For example, the information in some system manuals is also used by programmers. Keep this in mind when searching for information on specific topics.

The *Documentation Overview* provides information on all of the manuals in the Tru64 UNIX documentation set.

Reader's Comments

Compaq welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on your system in the following location:

  `/usr/doc/readers_comment.txt`

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate Compaq technical support office. Information provided with the software media explains how to send problem reports to Compaq.

Conventions

This manual uses the following typographical conventions:

`#`	A number sign represents the superuser prompt.
`% cat`	Boldface type in interactive examples indicates typed user input.
`file`	Italic (slanted) type indicates variable values, placeholders, and function argument names.
. . .	A vertical ellipsis indicates that a portion of an example that would normally be present is not shown.
`cat(1)`	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, `cat(1)` indicates that you can find information on the `cat` command in Section 1 of the reference pages.

1

Managing Clusters Overview

Managing a TruCluster Server cluster is similar to managing a standalone Tru64 UNIX system. Of the more than 600 commands and utilities for system administration, fewer than 20 apply exclusively to clusters. You use most of those commands when creating a cluster, adding a new member to a cluster, or making an application highly available. If you know how to manage a Tru64 UNIX system, you already know most of what is needed to manage a TruCluster Server cluster.

This manual describes the relatively few situations where managing a cluster is different. For documentation about the other management procedures, see the Tru64 UNIX *System Administration* guide.

Before reading further, familiarize yourself with the material in the TruCluster Server *Cluster Technical Overview*. An understanding of the information in that manual is necessary to managing a cluster.

The chapter discusses the following topics:

- Commands and utilities for clusters (Section 1.1)
- Commands and features that are different in a cluster (Section 1.2)

In most cases, the fact that you are administering a cluster rather than a single system becomes apparent because of the occasional need to manage one of the following aspects of the TruCluster Server:

- Cluster creation and configuration, which includes creating the initial cluster member, adding and deleting members, and querying the cluster configuration.
- Cluster application availability (CAA), which you use to define and manage highly available applications.
- Cluster aliases, which provide a single-system view of the cluster to clients network.
- Cluster quorum and votes, which determine what constitutes a valid cluster and membership in that cluster, and thereby allows access to cluster resources.
- Device request dispatcher, which provides transparent, highly available access to all devices in the cluster.

- Cluster File System (CFS), which provides clusterwide coherent access to all file systems, including the root (/) file system.
- Memory Channel, which provides the private, clusterwide communications path interconnect between cluster members.

In addition to the previous items, there are some command-level exceptions when a cluster does not appear to the user like a single computer system. For example, when you execute the `wall` command, the message is sent only to users who are logged in on the cluster member where the command executes. To send a message to all users who are logged in on all cluster members, use the `wall -c` command.

1.1 Commands and Utilities for Clusters

Table 1–1 lists commands that are specific to managing TruCluster Server systems. These commands manipulate or query aspects of a cluster. You can find descriptions for these commands in the reference pages.

Table 1–1: Cluster Commands

Function	Command	Description
Create and configure cluster members	`clu_create`(8)	Creates an initial cluster member on a Tru64 UNIX system.
	`clu_add_member`(8)	Adds a member to a cluster.
	`clu_delete_member`(8)	Deletes a member from a cluster.
	`clu_check_config`(8)	Verifies that the TruCluster Server has been properly installed, and that the cluster is correctly configured.
	`clu_get_info`(8)	Displays information about a cluster and its members.
Define and manage highly available applications	`caad`(8)	Starts the CAA daemon.
	`caa_profile`(8)	Manages an application availability profile and performs basic syntax verification.
	`caa_register`(8)	Registers an application with CAA.
	`caa_relocate`(8)	Manually relocates a highly available application from one cluster member to another.
	`caa_start`(8)	Starts a highly available application registered with the CAA daemon.

Table 1–1: Cluster Commands (cont.)

Function	Command	Description
	caa_stat(1)	Provides status on applications registered with CAA.
	caa_stop(8)	Stops a highly available application.
	caa_unregister(8)	Unregisters a highly available application.
Manage cluster alias	cluamgr(8)	Creates and manages cluster aliases.
Manage quorum and votes	clu_quorum(8)	Configures or deletes a quorum disk, or adjusts quorum disk votes, member votes, or expected votes.
Manage context-dependent symbolc links (CDSLs)	mkcdsl(8)	Makes or checks CDSLs.
Manage device request dispatcher	drdmgr(8)	Gets or sets distributed device attributes.
Manage Cluster File System (CFS)	cfsmgr(8)	Manages a mounted file system in a cluster.
Query the status of Memory Channel	imcs(1)	Reports the status of the Memory Channel application programming interface (API) library, libimc.
	imc_init(1)	Initializes and configures the Memory Channel API library, libimc, on the current host.

1.2 Commands and Features That Are Different in a Cluster

The following tables list Tru64 UNIX commands and subsystems that have cluster-specific options, or that behave differently in a cluster than on a standalone Tru64 UNIX system.

In general, commands that manage processes are not cluster-aware and can be used only to manage the member on which they are executed.

Table 1–2 describes the differences in commands and utilities that manage files systems and storage.

In a standalone Tru64 UNIX system, the root file system (/) is root_domain#root. In a cluster, the root file system is always cluster_root#root. The boot partition for each cluster member is root*memberID*_domain#root.

For example, on the cluster member with member ID 6, the boot partition, /cluster/members/member6/boot_partition, is root6_domain#root.

Table 1–2: File Systems and Storage Differences

Command	Differences
addvol(8)	In a single system, you cannot use addvol to expand root_domain. However, in a cluster, you can use addvol to add volumes to the cluster_root domain. You can remove volumes from the cluster_root domain with the rmvol command. Logical Storage Manager (LSM) volumes cannot be used within the cluster_root domain. An attempt to use the addvol command to add an LSM volume to the cluster_root domain fails.
bttape(8)	The bttape utility is not supported in clusters. For more information about backing up and restoring files, see Section 9.8.
df(1)	The df command does not account for data in client caches. Data in client caches is synchronized to the server at least every 30 seconds. Until synchronization occurs, the physical file system is not aware of the cached data and does not allocate storage for it.
iostat(1)	The iostat command displays statistics for devices on a shared or private bus that are directly connected to the member on which the command executes. Statistics pertain to traffic that is generated to and from the local member.

Table 1–2: File Systems and Storage Differences (cont.)

Command	Differences
LSM voldisk(8) volencap(8) volreconfig(8) volstat(8) volmigrate(8) volunmigrate(8)	The `voldisk list` command can give different results on different members for disks that are not under LSM control (that is, `autoconfig` disks). The differences are typically limited to disabled disk groups. For example, one member might show a disabled disk group and another member might not display that disk group at all. In a cluster, the `volencap swap` command places the swap devices for an individual cluster member into an LSM volume. Run the command on each member whose swap devices you want to encapsulate. The `volreconfig` command is required only when you encapsulate members' swap devices. Run the command on each member whose swap devices you want to encapsulate. When encapsulating the `cluster_usr` domain with the `volencap` command, you must shut down the cluster to complete the encapsulation. The `volreconfig` command is called during the cluster reboot; you do not need to run it separately. The `volstat` command returns statistics only for the member on which it is executed. The `volmigrate` command modifies an Advanced File System (AdvFS) domain to use LSM volumes for its underlying storage. The `volunmigrate` command modifies any AdvFS domain to use physical disks instead of LSM volumes for its underlying storage. For more information on LSM in a cluster, see Chapter 10.
mount(8)	Network File System (NFS) loopback mounts are not supported. For more information, see Section 7.6.2.3. Other commands that run through `mountd`, like `umount` and `export`, receive a `Program unavailable` error when the commands are sent from external clients and do not use the default cluster alias or an alias listed in `/etc/exports.aliases`.

Table 1–2: File Systems and Storage Differences (cont.)

Command	Differences
Prestoserve presto(8) dxpresto(8X) prestosetup(8) prestoctl_svc(8)	Prestoserve is not supported in a cluster.
showfsets(8)	The showfsets command does not account for data in client caches. Data in client caches is synchronized to the server at least every 30 seconds. Until synchronization occurs, the physical file system is not aware of the cached data and does not allocate storage for it. Fileset quotas and storage limitations are enforced by ensuring that clients do not cache so much dirty data that they exceed quotas or the actual amount of physical storage.
UNIX File System (UFS) Memory File System (MFS)	A UFS file system is served for read-only access based on connectivity. Upon member failure, CFS selects a new server for the file system. Upon path failure, CFS uses an alternate device request dispatcher path to the storage. A cluster member can mount a UFS file system read/write. The file system is accessible only by that member. There is no remote access; there is no failover. MFS file system mounts, whether read-only or read/write, are accessible only by the member that mounts it. The server for an MFS file system or a read/write UFS file system is the member that initializes the mount.
verify(8)	You can use the verify command to learn the cluster root domain, but the -f and -d options cannot be used. For more information, see Section 9.11.1.

Table 1–3 describes the differences in commands and utilities that manage networking.

Table 1–3: Networking Differences

Command	Differences
Berkeley Internet Name Domain (BIND) `bindconfig`(8) `bindsetup`(8) `svcsetup`(8)	The `bindsetup` command was retired in Tru64 UNIX Version 5.0. Use the `sysman dns` command or the equivalent command, `bindconfig`, to configure BIND in a cluster. BIND client configuration is clusterwide. All cluster members have the same client configuration. Only one member of a cluster can be a BIND server. A BIND server is configured as a highly available service under CAA. The cluster alias acts as the server name. For more information, see Section 7.4.
Broadcast messages `wall`(1) `rwall`(1)	The `wall -c` command sends messages to all users on all members of the cluster. Without any options, the `wall` command sends messages to all users who are logged in to the member where the command is executed. Broadcast messages to the default cluster alias from `rwall` are sent to all users logged in on all cluster members. In a cluster, a `clu_wall` daemon runs on each cluster member to receive `wall -c` messages.
Dynamic Host Configuration Protocol (DHCP) `joinc`(8)	A cluster can be a DHCP server, but cluster members cannot be DHCP clients. Do not run `joinc` in a cluster. Cluster members must use static addressing. For more information, see Section 7.1.
`dsfmgr`(8)	When using the `-a class` option, specify `c` (cluster) as the `entry_type`. The output from the `-s` option indicates `c` (cluster) as the scope of the device. The `-o` and `-O` options, which create device special files in the old format, are not valid in a cluster.

Table 1–3: Networking Differences (cont.)

Command	Differences
Mail mailconfig(8) mailsetup(8) mailstats(8)	All members that are running mail must have the same mail configuration and, therefore, must have the same protocols enabled. All members must be either clients or servers. See Section 7.8 for details. The mailstats command returns mail statistics for the cluster member on which it was run. The mail statistics file, /usr/adm/sendmail/sendmail.st, is a member-specific file; each cluster member has its own version of the file.
Network File System (NFS) nfsconfig(8) rpc.lockd(8) rpc.statd(8)	Use sysman nfs or the nfsconfig command to configure NFS. Do not use the nfssetup command, it was retired in Tru64 UNIX Version 5.0. Cluster members can run client versions of lockd and statd. Only one cluster member runs an additional lockd and statd pair for the NFS server. The server lockd and statd are highly available and are under the control of CAA. For more information, see Section 7.6.
Network management netconfig(8) netsetup(8) gated(8) routed(8)	If, as we recommended, you configured networks during cluster configuration, gated was configured as the routing daemon. See the TruCluster Server *Cluster Installation* manual for more information. If you later run netconfig, you must select gated, not routed, as the routing daemon. The netsetup command has been retired. Do not use it.
Network Interface Failure Finder (NIFF) niffconfig(8) niffd(8)	In order for NIFF to monitor the network interfaces in the cluster, niffd, the NIFF daemon, must run on each cluster member. For more information, see Section 6.1.
Network Information Service (NIS) nissetup(8)	NIS runs as a highly available application. The default cluster alias name is used to identify the NIS master. For more information, see Section 7.2.

Table 1–3: Networking Differences (cont.)

Command	Differences
Network Time Protocol (NTP) `ntp`(1)	All cluster members require time synchronization. NTP meets this requirement. Each cluster member is automatically configured as an NTP peer of the other members. You do not need to do any special NTP configuration. For more information, see Section 7.5.
`routed`(8)	`routed` is not supported in TruCluster Server systems. The cluster alias requires `gated`. When you create the initial cluster member, `clu_create` configures `gated`. When you add a new cluster member, `clu_add_member` propagates the configuration to the new member. For more information about routers, see Section 6.2.

Table 1–4 describes the differences in printing management.

Table 1–4: Printing Differences

Command	Differences
`lprsetup`(8) `printconfig`(8)	A cluster-specific printer attribute, `on`, designates the cluster members that are serving the printer. The print configuration utilities, `lprsetup` and `printconfig`, provide an easy means for setting the `on` attribute. The file `/etc/printcap` is shared by all members in the cluster. For more information, see Section 7.3.
Advanced Printing Software	For information on installing and using Advanced Printing Software in a cluster, see the configuration notes chapter in the Tru64 UNIX *Advanced Printing Software Release Notes*.

Table 1–5 describes the differences in managing security. For information on enhanced security in a cluster, see the Tru64 UNIX *Security* manual.

Table 1–5: Security Differences

Command	Differences
auditd(8) auditconfig(8) audit_tool(8)	A cluster is a single security domain. To have root privileges on the cluster, you can log in as root on the cluster alias or on any one of the cluster members. Similarly, access control lists (ACLs) and user authorizations and privileges are clusterwide. With the exception of audit log files, security related files, directories, and databases are shared throughout the cluster. Audit log files are specific to each member — an audit daemon, auditd, runs on each member and each member has its own unique audit log files. If any single cluster member fails, auditing continues uninterrupted for the other cluster members. To generate an audit report for the entire cluster, you can pass the name of the audit log CDSL to the audit reduction tool, audit_tool. Specify the appropriate individual log names to generate an audit report for one or more members. If you want enhanced security, we strongly recommend that you configure enhanced security before cluster creation. A clusterwide shutdown and reboot are required to configure enhanced security after cluster creation.
rlogin(1) rsh(1) rcp(1)	An rlogin, rsh, or rcp request from the cluster uses the default cluster alias as the source address. Therefore, if a noncluster host must allow remote host access from any account in the cluster, its .rhosts file must include the cluster alias name (in one of the forms by which it is listed in the /etc/hosts file or one resolvable through NIS or the Domain Name System (DNS)). The same requirement holds for rlogin, rsh, or rcp to work between cluster members. For more information, see Section 5.3.

Table 1–6 describes the differences in commands and utilities for configuring and managing systems.

Table 1–6: General System Management Differences

Command	Differences
Dataless Management Services (DMS)	DMS is not supported in a TruCluster Server environment. A cluster can be neither a DMS client nor a server.
Event Manager (EVM) and event management	Events have a `cluster_event` attribute. When this attribute is set to `true`, the event, when it is posted, is posted to all members of the cluster. Events with `cluster_event` set to `false` are posted only to the member on which the event was generated. For a list of cluster events, see Appendix A.
`halt`(8) `reboot`(8) `init`(8) `shutdown`(8)	There is no clusterwide `halt` or `reboot`. The `halt` and `reboot` commands act only on the member on which the command is executed. `halt`, `reboot`, and `init` have been modified to leave file systems in a cluster mounted, because the file systems are automatically relocated to another cluster member. You can use `shutdown -c` to halt a cluster. The `shutdown -c` *time* command fails if any of the commands `clu_quorum`, `clu_add_member` or `clu_delete_member` is in progress. You can shut down a cluster to a halt, but you cannot reboot (`shutdown -r`) the entire cluster. To shut down a single cluster member, execute the `shutdown` command from that member. For more information, see `shutdown`(8).
`hwmgr`(8)	In a cluster, the `-member` option allows you to designate the host name of the cluster member that the `hwmgr` command acts upon. Use the `-cluster` option to specify that the command acts clusterwide. When neither the `-member` nor `-cluster` option is used, `hwmgr` acts on the system where it is executed.

Table 1–6: General System Management Differences (cont.)

Command	Differences
Process control ps(1)	A range of possible process identifiers (PIDs) is assigned to each cluster member to provide unique process IDs clusterwide. The ps command reports only on processes that are running on the member where the command executes.
kill(1)	If the passed parameter is greater than zero (0), the signal is sent to the process whose PID matches the passed parameter, no matter on which cluster member it is running. If the passed parameter is less than –1, the signal is sent to all processes (cluster-wide) whose process group ID matches the absolute value of the passed parameter. Even though the PID for init on a cluster member is not 1, kill 1 behaves as it would on a standalone system and sends the signal to all processes on the current cluster member, except for kernel idle and /sbin/init.
rcmgr(8)	The hierarchy of the /etc/rc.config* files allows an administrator to define configuration variables consistently over all systems within a local area network (LAN) and within a cluster. For more information, see Section 5.1.
sysman_clone(8) sysman -clone	Configuration cloning and replication is not supported in a cluster. Attempts to use the sysman -clone command in a cluster fail and return the following message: Error: Cloning in a cluster environment is not supported.
System accounting services and the associated commands fuser(8) mailstats(8) ps(1) uptime(1) vmstat(1) w(1) who(1)	These commands are not cluster-aware. Executing one of these commands returns information for only the cluster member on which the command executes. It does not return information for the entire cluster. See Section 5.13.

Table 1–7 describes features that TruCluster Server does not support.

Table 1–7: Features Not Supported

Feature	Comments
Archiving `bttape`(8)	The `bttape` utility is not supported in clusters. For more information about backing up and restoring files, see Section 9.8.
LSM `volrootmir`(8) `volunroot`(8)	The `volrootmir` and `volunroot` commands are not supported for clusters. For more information on LSM in a cluster, see Chapter 10.
`mount`(8)	NFS loopback mounts are not supported. For more information, see Section 7.6.2.3. Other commands that run through `mountd`, like `umount` and `export`, receive a `Program unavailable` error when the commands are sent from external clients and do not use the default cluster alias or an alias listed in `/etc/exports.aliases`.
Prestoserve `presto`(8) `dxpresto`(8X) `prestosetup`(8) `prestoctl_svc`(8)	Prestoserve is not supported in a cluster.
`routed`(8)	The `routed` daemon is not supported in TruCluster Server systems. The cluster alias requires `gated`. When you create the initial cluster member, `clu_create` configures `gated`. When you add a new cluster member, `clu_add_member` propagates the configuration to the new member. For more information about routers, see Section 6.2.
Dataless Management Services (DMS)	DMS is not supported in a TruCluster Server environment. A cluster can be neither a DMS client nor a server.

Table 1–7: Features Not Supported (cont.)

Feature	Comments
UNIX File System (UFS)	A cluster member can mount a UFS file system read/write. The file system is accessible only by that member. There is no remote access; there is no failover.
`sysman_clone`(8) sysman -clone	Configuration cloning and replication is not supported in a cluster. Attempts to use the `sysman -clone` command in a cluster fail and return the following message: `Error: Cloning in a cluster environment is not supported.`

2

Tools for Managing Clusters

This chapter describes the tools that you can use to manage TruCluster Server systems. The chapter discusses the following management tools and options:

- An overview of the management tools available for both single-system and cluster management (Section 2.1)
- Understanding available management tools and interfaces (Section 2.2)
- Understanding cluster configuration tools and related user interfaces (Section 2.3)
- An overview of SysMan (Section 2.4)
- Using SysMan Menu in a cluster (Section 2.5)
- Using SysMan Station in a cluster (Section 2.6)
- Using the SysMan Java applets in a cluster (Section 2.7)
- Using the SysMan Java PC applications in a cluster (Section 2.8)
- Using the SysMan command-line interface in a cluster (Section 2.9)
- Using Compaq Insight Manager in a cluster (Section 2.10)
- Using Tru64 UNIX configuration report (Section 2.11)

2.1 Introduction

Tru64 UNIX offers a wide array of management tools for both single-system and cluster management. Whenever possible, the cluster is managed as a single system.

We realize that many systems are used in heterogeneous environments, where the system manager might expect to manage TruCluster Server systems from a PC, from a Tru64 UNIX workstation, from a character-cell terminal, or even from a laptop PC via dialup lines.

In recognition of this fact, Tru64 UNIX and TruCluster Server provide tools with Web-based, graphical, and command-line interfaces to perform management tasks. In particular, SysMan offers command-line, character-cell terminal, Java, X Windows, and Web-based Java applet interfaces to system and cluster management.

SysMan is not a single application or interface. Rather, SysMan is a suite of applications for managing Tru64 UNIX and TruCluster Server systems. SysMan has three main components: SysMan Menu, SysMan Station, and the SysMan command-line interface. Each of these components is described in this chapter.

You can choose the tools and user interfaces that meet your needs. Perhaps you are most comfortable with the power and flexibility of the traditional Tru64 UNIX command line. Or, if cluster management from a PC is important to you, you can use the Java standalone graphical interface to SysMan to perform administrative tasks from a PC running Windows.

Because there are numerous cluster management tools and interfaces that you can use, this chapter begins with a description of the various options. The features and capabilities of each option are briefly described in the following sections, and are discussed fully in the Tru64 UNIX *System Administration* manual.

Some cluster operations do not have graphical interfaces and require that you use the command-line interface. These operations and commands are described in Section 2.3.

2.1.1 Cluster Tools Quick Start

If you are already familiar with the tools for managing clusters and want to start using them, see Table 2–1. This table presents only summary information; additional details are provided later in this chapter.

Table 2–1: Cluster Tools Quick Start

Tool	User Interface	How to Invoke
SysMan Menu	X Windows	`# /usr/sbin/sysman -menu [-display` *display*`]`
	Character cell	`# /usr/sbin/sysman -menu`
	Java applet	`http://`*cluster_member_name*`:2301/SYSMAN/index.html`
	PC application	`http://`*cluster_member_name*`:2301/SYSMAN/index.html`
SysMan Station	X Windows	`# /usr/sbin/sysman -station [`*hostname*`] [-display` *display*`]`
	Java applet	`http://`*cluster_member_name*`:2301/SYSMAN/index.html`
	PC application	`http://`*cluster_member_name*`:2301/SYSMAN/index.html`
SysMan -CLI	Command line	`# /usr/sbin/sysman -cli`

Table 2–1: Cluster Tools Quick Start (cont.)

Tool	User Interface	How to Invoke
Compaq Insight Manager	Web interface	`http://cluster_member_name:2301/`
Compaq Insight Manager XE	Web interface	`http://xe_server_name:280/`

2.1.2 A Word About Compaq Insight Manager Integration

The integration of SysMan and Compaq Insight Manager deserves special mention.

Compaq Insight Manager and Web-based system management via SysMan are tightly coupled. Compaq Insight Management agents and subagents provide device and system information for all managed subsystems. The Compaq Insight Manager Web-based Simple Network Management Protocol (SNMP) agents allow you to gather and display information about the state of the system. The SNMP agents provide read-only information and do not allow you to manage the system.

Compaq Insight Manager also provides the Web-based management (WBEM) framework for SysMan. Compaq Insight Manager is available on each Tru64 UNIX system from the following URL:

`http://cluster_member_name:2301/`

From this site you can run SysMan Menu or SysMan Station directly in a Web browser, or you can download a PC client kit to install these applications locally.

2.1.3 A Word About Compaq Insight Manager XE Integration

Compaq Insight Manager XE uses the Compaq Common Cluster Management Information Base (MIB) to discover and monitor TruCluster servers. This MIB is supported by the `/usr/sbin/clu_mibs` SNMP subagent, which comes with the cluster software and starts automatically.

`clu_mibs` is an Extensible SNMP subagent daemon for TruCluster Server systems that implements cluster MIB support. The daemon currently supports the Common Cluster MIB (`/usr/share/sysman/mibs/svrClu.mib`) and the TruCluster Server MIB (`/usr/share/sysman/mibs/truClu.mib`).

Through its Web interface, Compaq Insight Manager XE gathers and displays information about the state of the clusters. The SNMP agents provide read-only information and do not allow you to manage the system. Use other tools to perform management tasks.

2.2 Available Management Tools and Interfaces

This section describes which tools you can run from which platform and via which interface. The available management tools and interfaces are listed in Table 2–2.

In this table, NA means not available.

Table 2–2: Available Management Tools and Interfaces

Tool	X Windows	Character Cell	Tru64 UNIX Command Line	PC Application	Java Applet[a]	Web Browser on Any Platform
`sysman -menu`	Yes	Yes	NA	Yes	Yes	NA
`sysman -station`	Yes	NA	NA	Yes	Yes	NA
`sysman -cli`	NA	NA	Yes	NA	NA	NA
Compaq Insight Manager	NA	NA	NA	NA	NA	Yes
Compaq Insight Manager XE	NA	NA	NA	NA	NA	Yes

[a] See Section 2.7 for browser requirements.

The interfaces are consistent in operation no matter which user environment you use. For example, SysMan Menu is similar whether you invoke it via the character-cell terminal interface, as an X Windows application through the Common Desktop Environment (CDE), or through the Java interface. However, there are navigational differences between the interfaces. For example, the SysMan Menu character-cell interface does not contain graphical elements such as icons. In contrast, the X Windows interface is designed to run in a windowing environment, such as CDE, and contains clickable buttons, drop-down lists, and so forth.

The Compaq Insight Manager Web-based SNMP agents gather and display information about the state of the system. Use the other tools to perform management tasks.

2.3 Cluster Configuration Tools and Related User Interfaces

Not all TruCluster management tools have SysMan interfaces. Table 2–3 presents the tools for managing cluster-specific tasks and indicates which

tools are not available through SysMan Menu. In this table, NA means not available.

Table 2–3: Cluster Management Tools

Command	Available in SysMan Menu	Function
caa_profile(8) caa_register(8) caa_relocate(8) caa_start(8) caa_stat(1) caa_stop(8) caa_unregister(8)	sysman caa	Manages highly avaliable applications with cluster application availability (CAA).
cfsmgr(8)	sysman cfsmgr	Manages the cluster file system.
cluamgr(8)	sysman clu_aliases	Creates and manages cluster aliases.
clu_add_member(8)	NA	Adds a member to a cluster.
clu_create(8)	NA	Creates an initial cluster member on a Tru64 UNIX system.
clu_check_config(8)	NA	Verifies that the TruCluster Server has been properly installed, and that the cluster is correctly configured.
clu_delete_member(8)	NA	Deletes a member from a cluster.
clu_get_info(8)	sysman hw_cluhierarchy (approximate)	Gets information about a cluster and its members.
clu_quorum(8)	NA	Configures or deletes a quorum disk, or adjusts quorum disk votes, member votes, or expected votes.
drdmgr(8)	sysman drdmgr	Manages distributed devices.
imcs(1)	NA	Reports the status of the Memory Channel application programming interface (API) library, libimc.
imc_init(1)	NA	Initializes and configures the Memory Channel API library, libimc, on the current host.
mkcdsl(8)	NA	Makes or verifies CDSLs.

2.4 The Many Faces of SysMan

This section introduces the SysMan management options. For general information about SysMan, see `sysman_intro`(8) and `sysman`(8).

SysMan provides easy-to-use interfaces for common system management tasks, including managing the cluster file system, storage, and cluster aliases. The interface options to SysMan provide the following advantages:

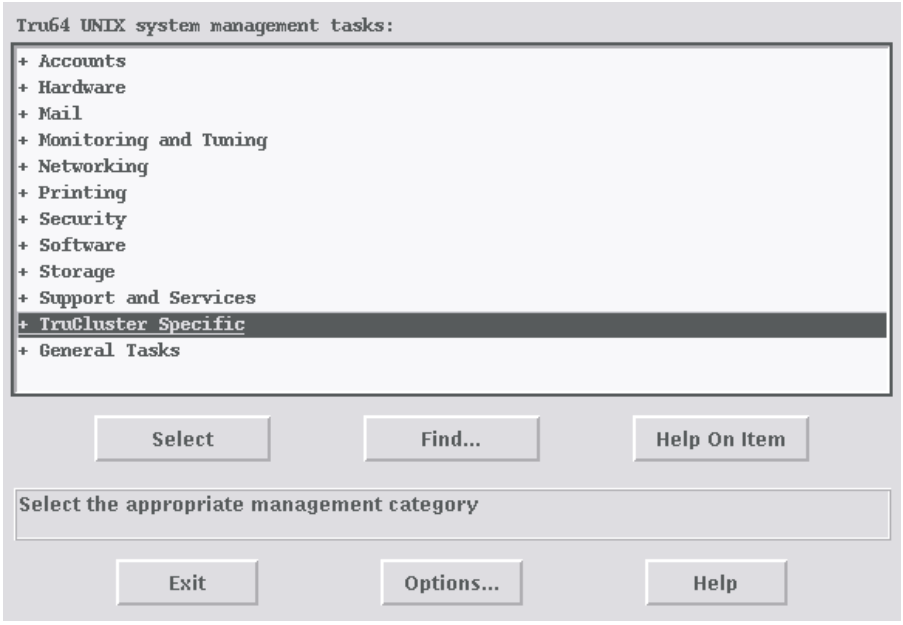
- A familiar interface that you access from the Tru64 UNIX and Microsoft Windows operating environments.
- Ease of management. There is no need to understand the command-line syntax or to manually edit configuration files.

SysMan has three main components: SysMan Menu, SysMan Station, and the SysMan command-line interface. The following sections describe these components.

2.4.1 Introduction to SysMan Menu

SysMan Menu integrates most available single-system and cluster administration utilities in a menu system, as shown in Figure 2–1.

Figure 2–1: The SysMan Menu Hierarchy

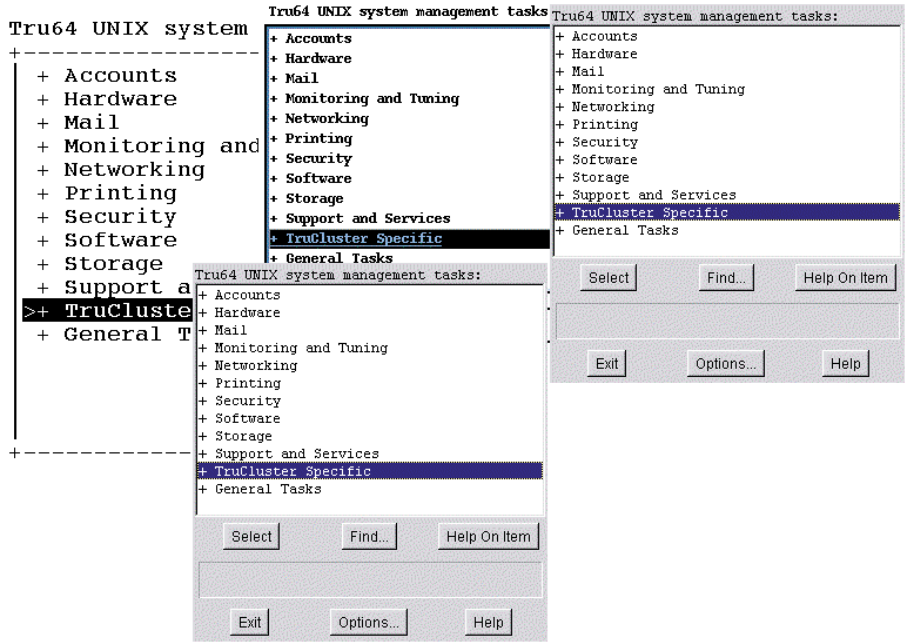


ZK-1702U-AI

SysMan Menu provides a menu of system management tasks in a tree-like hierarchy, with branches of management categories and leaves of actual tasks. Selecting a leaf invokes a task, which displays a dialog box for performing the task.

The SysMan Menu user interface is functionally equivalent no matter how SysMan Menu is invoked. For example, Figure 2–2 shows a composite of the character-cell, X Windows, Java, and Java applet user interfaces.

Figure 2–2: The SysMan Menu Interfaces



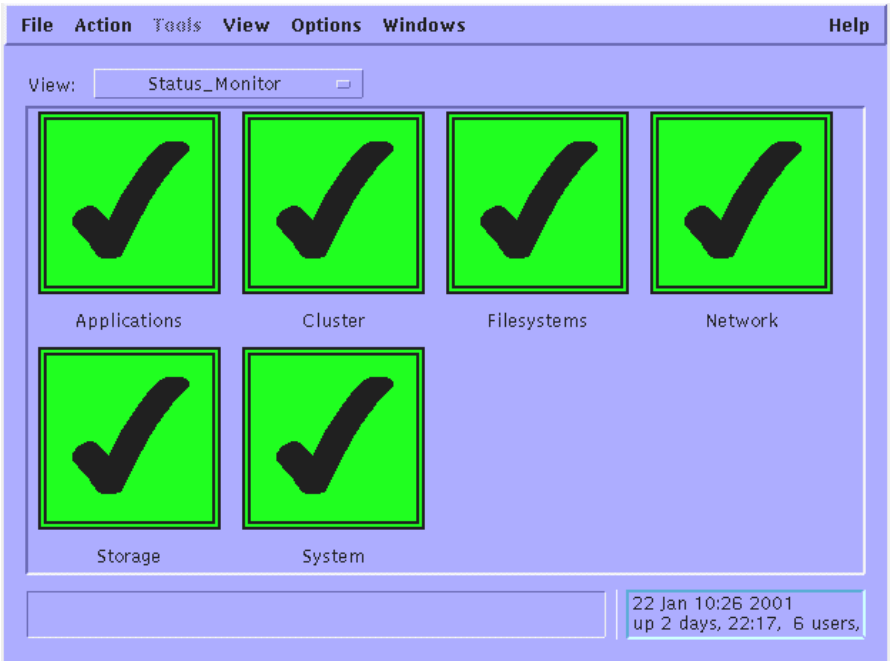
ZK-1695U-AI

2.4.2 Introduction to SysMan Station

SysMan Station is a graphical representation of a system (or cluster) that enables you to monitor system status down to the level of individual system components such as disks. You can also view and monitor logical groups, such as file systems or Advanced File System (AdvFS) domains, and create customized views. When viewing any system component, you can obtain detailed information on its properties or launch utilities that enable you to perform administrative tasks on the component. Unlike SysMan Menu, SysMan Station requires a graphics capability and cannot be run from the character-cell user environment.

Figure 2–3 shows an example of the SysMan Station graphical interface.

Figure 2–3: SysMan Station Graphical Interface



ZK-1703U-AI

As with SysMan Menu, the SysMan Station user interface is functionally equivalent no matter how SysMan Station is invoked.

2.4.3 Introduction to the SysMan Command Line

The `sysman -cli` command provides a generic command-line interface to SysMan functions. You can use the `sysman -cli` command to view or modify SysMan data. You can also use it to view dictionary-type information such as data descriptions, key information, and type information of the SysMan data, as described in `sysman_cli`(8). Use the `sysman -cli -list components` command to list all known components in the SysMan data hierarchy.

2.5 Using SysMan Menu in a Cluster

This section describes using SysMan Menu in a cluster. The section begins with a discussion of **focus** and how it affects SysMan Menu.

2.5.1 Getting in Focus

The range of effect of a given management operation is called its focus. In a TruCluster environment, there are four possibilities for the focus of a management operation:

- Clusterwide — The operation affects the entire cluster.
- Member-specific — The operation affects only the member that you specify. The operation requires a focus.
- Both — The operation can be clusterwide or member-specific. The operation requires a focus.
- None — The operation does not take focus and always operates on the current system.

For each management task, SysMan Menu recognizes which focus choices are appropriate. If the task supports both clusterwide and member-specific operations, SysMan Menu lets you select the cluster name or a specific member on which to operate. That is, if the cluster name and cluster members are available as a selection choice, the operation is `both`; if only the member names are available as a selection choice, the operation is `member-specific`.

Focus information for a given operation is displayed in the SysMan Menu title bar. For example, when you are managing local users on a cluster, which is a clusterwide operation, the title bar might appear similar to the following. In this example, `provolone` is a cluster member and `deli` is the cluster alias.

```
Manage Local Users on provolone.zk3.dec.com managing deli
```

2.5.2 Specifying a Focus on the Command Line

If an operation lets you specify a focus, the SysMan Menu `-focus` option provides a way to accomplish this from the command line.

Consider how specifying a focus on the command line affects the `shutdown` command. The `shutdown` command can be clusterwide or member-specific. If you start SysMan Menu from a cluster member with the following command, the cluster name is the initial focus of the `shutdown` option:

```
# sysman -menu
```

However, if you start SysMan Menu from a cluster member with the following command, the `amember` cluster member is the initial focus of the `shutdown` option:

```
# sysman -menu -focus amember
```

Whenever you begin a new task during a SysMan Menu session, the dialog box highlights your focus choice from the previous task. Therefore, if you have many management functions to perform on one cluster member, you need to select that member only once.

2.5.3 Invoking SysMan Menu

You can invoke SysMan Menu from a variety of interfaces, as explained in Table 2–4.

Table 2–4: Invoking SysMan Menu

User Interface	How to Invoke

Character-cell terminal	Start a terminal session (or open a terminal window) on a cluster member and enter the following command:
	`# /usr/sbin/sysman -menu`
	If an X Windows display is associated with this terminal window through the `DISPLAY` environment variable, directly on the SysMan Menu command line with the `-display` qualifier, or via some other mechanism, the X Windows interface to SysMan Menu is started instead. In this case, use the following command to force the use of the character-cell interface:
	`# /usr/sbin/sysman -menu -ui cui`

CDE (or other X Windows display)	SysMan Menu is available in X Windows windowing environments. To launch SysMan Menu, enter the following command: `# /usr/sbin/sysman -menu [-display` *`displayname`*`]` If you are using the CDE interface, you can launch SysMan Menu by clicking on the SysMan submenu icon on the root user's front panel and choosing SysMan Menu. If you click on the SysMan icon itself rather than on the submenu icon, SysMan Station is directly launched. You can also launch SysMan Menu from CDE by clicking on the Application Manager icon on the front panel and then clicking on the SysMan Menu icon in the System_Admin group.

Table 2–4: Invoking SysMan Menu (cont.)

User Interface	How to Invoke
Command line	SysMan Menu is not available from the command line. However, the SysMan command-line interface, `sysman -cli`, lets you execute SysMan routines from the command line or write programs to customize the input to SysMan interfaces. See `sysman_cli`(8) for details on options and flags. See Section 2.9 for more information.
Web-based Java Applets	See Section 2.7.1.
Standalone Java Application	See Section 2.8.1.

2.6 Using SysMan Station in a Cluster

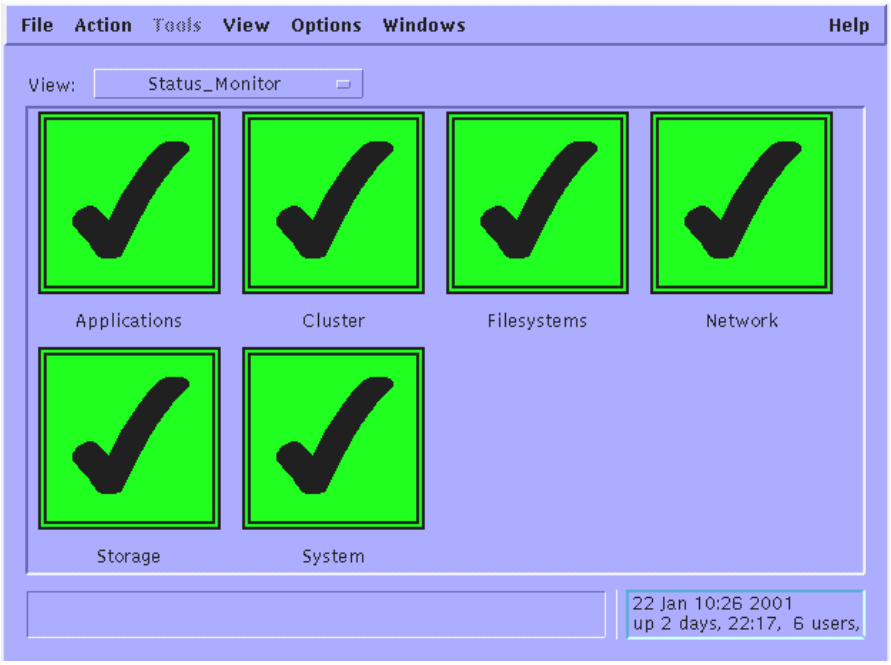
SysMan Station is a client/server application consisting of a daemon, `smsd`(8), and the SysMan Station graphical user interface. SysMan Station monitors and manages a single system or a cluster. You can also launch SysMan Menu or invoke applications directly from the SysMan Station. You can use SysMan Station to do the following tasks:

- Monitor the status of a system or cluster
- Display detailed information about a system or cluster
- Provide a single location for management activity
- Display events and track events that lead to a problem

You might find it convenient to launch SysMan Station and then leave it running on your desktop. In particular, if you are new to Tru64 UNIX system management, you can manage a cluster through SysMan Station without having to first learn the syntax of the Tru64 UNIX commands.

When you start SysMan Station from a cluster member, a monitor window is displayed (Figure 2–4).

Figure 2–4: SysMan Station Initial Cluster View



ZK-1703U-AI

The Monitor window displays the status of the following subsystems:

- Applications
- Cluster
- Filesystems
- Network
- Storage
- System

The status color and pattern indicates a failure or trouble condition, as follows:

- Healthy status is green with a check mark
- Trouble status is yellow with an exclamation point (!)
- Failure status is red with an x

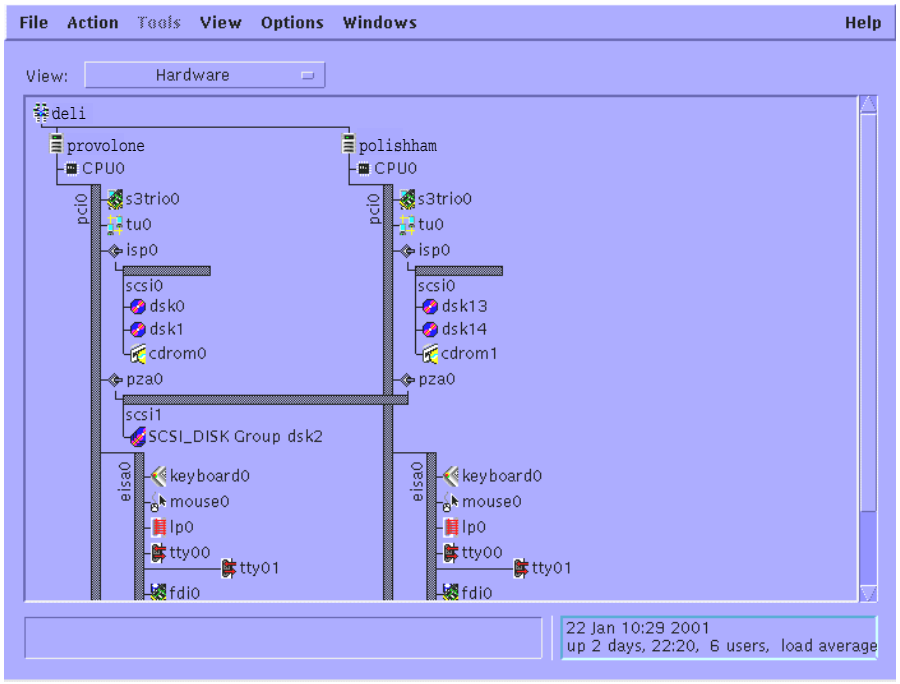
You can click on the status indicator, or the label beneath it, to view the events that are posted for a given subsystem. With the exception of events for which the `cluster_event` attribute is set to `true`, events are identified by the member on which the event was generated. For a list of cluster events, see Appendix A.

The left window pane shows the available views, including the following:

- AdvFS_Filesystems
- CAA_Applications_(active)
- CAA_Applications_(all)
- Hardware
- Mounted_Filesystems
- Physical_Filesystems

You can click on any of these views to open a new window containing that view. For example, if you click on the Hardware view, a view of the cluster hardware is displayed. An example of this view is shown in Figure 2–5.

Figure 2–5: A Sample SysMan Station Cluster Hardware View



ZK-1700U-AI

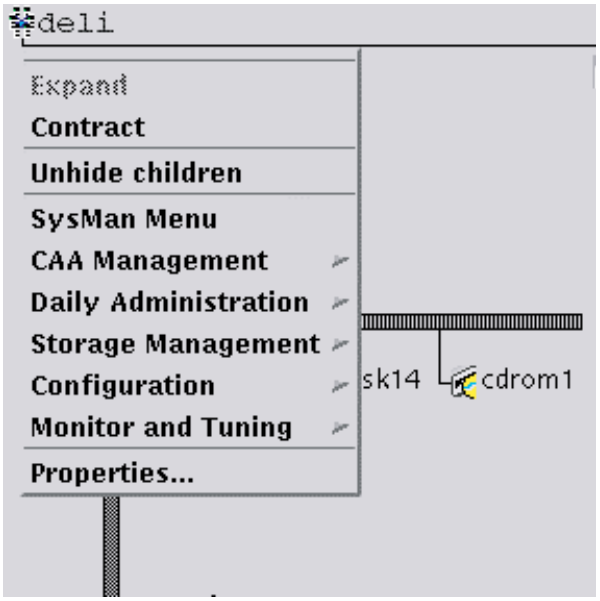
Objects in a view have actions that are based on their type; that is, objects such as the cluster or disk object have management actions associated with them, and the actions depend on the type of object. For example, a cluster object allows the following management actions:

- SysMan Menu
- Daily Administration
- Storage Management
- Configuration
- Monitor and Tuning
- CAA Management
- Properties

Some selections may not have any management tasks associated with them. For example, a graphics card allows you to view its properties and associated events, but you cannot otherwise manage it.

To see which actions are available for a given object, locate the cursor over the object and then click and hold the right mouse button, as shown in Figure 2–6.

Figure 2–6: Displaying Available Actions in SysMan Station



ZK-1704U-AI

2.6.1 Invoking SysMan Station

You can invoke SysMan Station from a variety of interfaces, as described in Table 2–5.

Table 2–5: Invoking SysMan Station

User Interface	How to Invoke
Character-cell terminal	SysMan Station is not available on a local or remote character cell terminal.
CDE (or other X Windows display)	SysMan Station is available in X Windows windowing environments. To launch SysMan Station, enter the following command: `# /usr/sbin/sysman -station [`*`hostname`*`] [-display `*`display`*`]` If you are using the CDE interface, you can launch SysMan Station by clicking on the SysMan icon on the root user's front panel. You can also click on the Application Manager icon on the front panel and then click on the SysMan Station icon in the System_Admin group.

Table 2–5: Invoking SysMan Station (cont.)

User Interface	How to Invoke
Command line	SysMan Station is not available from the command line. However, the SysMan command-line interface, `sysman -cli`, lets you execute SysMan routines from the command line or write programs to customize the input to SysMan interfaces. See `sysman_cli`(8) for details on options and flags. See Section 2.9 for more information.
Web-based Java Applets	See Section 2.7.1.
Standalone Java Application	SeeSection 2.8.1.

2.7 Using the SysMan Java Applets in a Cluster

Note

You can manage only the Tru64 UNIX system that is serving you the Web page. In a cluster, this means that you can manage only the cluster member that is serving you the Web page. Therefore, to manage a cluster, use the Java PC applications that are described in Section 2.8 instead.

Compaq Insight Manager provides the Web-based management (WBEM) framework for SysMan in the form of two Java applets.

There are two components to the applets: the applets themselves, which run inside a Web browser, and the SysMan Station daemon, `/usr/sbin/smsd`, which runs on the Tru64 UNIX system.

You use a browser to open the correct URL and launch one of the applets. The applets then communicate with the Tru64 UNIX system, partially via the Compaq Insight Manager `http` server on port 2301.

Browser Requirement

To run the Java applets from a Tru64 UNIX Version 5.1 or later system, you can use Netscape Navigator running on a Tru64 UNIX Version 5.1 or later system, as described in `http://`*`cluster_member_name`*`:2301/SYSMAN/plugin.html`. You can also use other browsers as described on this Web page.

To run the Java applets from a Tru64 UNIX Version 5.0A or earlier system, you must use Microsoft Internet Explorer V4.0

with Service Pack 1 or higher, running on a Windows PC. Other browsers are not known to work correctly.

On the Tru64 UNIX system, the Compaq Insight Manager agents (daemons) are configured by default when you install the operating system and are automatically started when the system boots.

The Compaq Insight Manager Web agent is initialized during the transition to run level 3 by the `/sbin/rc3.d/S50insightd` script. This script runs `/usr/share/sysman/bin/insightd` and prints a console boot-time message when the agent is successfully started. The SNMP subagents `/usr/sbin/os_mibs` and `/usr/sbin/cpq_mibs` are also invoked during the transition to run level 3 by the `/sbin/rc3.d/S49snmpd` script. To test that the system is properly configured, enter the following commands:

```
# ps agx | grep insight
# ps agx | grep cpq
# ps agx | grep os_mib
```

Or, alternately:

```
# ps agx | grep -E "insight|cpq|os_mibs"
```

If you do not want to have the Compaq Insight Manager Web Agent enabled by default, perhaps because you do not plan to use it, you can disable it through SysMan Menu or through the following `imconfig` command:

```
# /usr/sbin/sysman imconfig
```

If you disable the Compaq Insight Manager Web Agent, you will not be able to use the online help from the SysMan PC applications.

2.7.1 Invoking the SysMan Java Applets

For details on running the SysMan Java applets directly from a Web browser, go to the following location in a compatible Web browser:

```
http://cluster_member_name:2301/SYSMAN/index.html
```

If your browser is compatible, click on the link to SysMan Station or SysMan Menu to start the applet within the browser. It might take a few moments for the applet to start.

When the applet starts, it establishes a connection to the cluster member. Log in as root.

After you are familiar with running SysMan Menu and SysMan Station from a Web browser, you may find it more convenient to directly launch them from the following URLs:

- `http://`*`cluster_member_name`*`:2301/SYSMAN/suit_applet.html` to launch SysMan Menu
- `http://`*`cluster_member_name`* `:2301/SYSMAN/sms_applet.html` to launch SysMan Station

2.8 Using the SysMan Java PC Applications in a Cluster

SysMan Menu and SysMan Station are both available as Java standalone applications that run on a Windows PC. You install and run these applications just as you do any other PC application; they are not based on a Web browser.

Other than the fact that the applications run as Java run-time environment (JRE) applications, they look and function just like the character-cell and X Windows versions.

Unlike the SysMan Java applet, the standalone Java applications are not dependent on the Compaq Insight Manager daemons or `http` server. The `smsd` daemon, `smsd`(8), is responsible for gathering system management data from the host and presenting that information to the SysMan Station Java client.

2.8.1 Invoking the SysMan Java Applications on a PC

For details on downloading and running the standalone Java applications on a Windows PC, go to the following location in any Web browser:

`http://`*`cluster_member_name`*`:2301/SYSMAN/index.html#PC`

The section is titled, Managing Tru64 UNIX from a PC. Step 2 describes how to download, install, and run the SysMan Station Java standalone applications.

To run the Java applications on your PC, you must first install the Java run-time environment (JRE), a version of which is available at this URL. Both the `jre.exe` and `setup_sysman.exe` files are self-extracting files; you need only to click on them in Windows Explorer to run them.

2.8.2 SysMan Compatibility Issues

The SysMan Station standalone Java applications that are included with Tru64 UNIX Version 5.0A and later work with Tru64 UNIX Version 5.0A and later systems, but they do not work with Tru64 UNIX Version 5.0.

Conversely, the SysMan Station standalone Java applications that are included with Tru64 UNIX Version 5.0 work with Tru64 UNIX Version 5.0 systems, but they do not work with later versions of Tru64 UNIX.

If you are familiar with Windows file properties and shortcuts, you can install two (or more) versions of the SysMan Station Java application, and run them at the same time.

To do this, install either version first and accept the default directory. Then, install the other version but choose a different installation directory. For example, if you install Tru64 UNIX Version 5.0 second, you might use a directory named `\Program Files\Compaq\SysMan5.0`.

When the installation completes, the Start Menu entry now launches the second version that you installed. To be able to launch the first version that you installed, use Windows Explorer to find the file SysMan Station.lnk. Make a copy of this file and then change the copy's shortcut properties to point to the initial installation directory, typically `\Program\Files\Compaq\SysMan\`.

See the Windows Help shortcuts topic if you need help to complete this task.

2.9 Using the SysMan Command-Line Interface in a Cluster

The `sysman -cli` command provides a generic command-line interface to SysMan data. You can use the `sysman -cli` command to view or modify SysMan data. You can also use it to view dictionary-type information such as data descriptions, key information, and type information of the SysMan data, as described in `sysman_cli`(8).

Use the `-focus` option to specify the focus; that is, the range of effect of a given management task, which can be the cluster as a whole or a specific cluster member.

Use the `sysman -cli -list` *component* command to list all known components in the SysMan data hierarchy.

An example `sysman -cli` command is shown in Example 2–1. This command shows the attributes of the `clua` component for the cluster member named `amember`.

Example 2–1: Example `sysman` Output

```
# sysman -cli -focus amember -list attributes -comp clua
Component: clua
  Group: cluster-aliases
    Attribute(s):
      aliasname
      memberlist
  Group: clua-info
    Attribute(s):
      memberid
      aliasname
```

Example 2–1: Example `sysman` Output (cont.)

```
      membername
      selw
      selp
      rpri
      joined
      virtual
  Group: componentid
    Attribute(s):
      manufacturer
      product
      version
      serialnumber
      installation
      verify
  Group: digitalmanagementmodes
    Attribute(s):
      deferredcommit
      cdfgroups
```

2.10 Using Compaq Insight Manager in a Cluster

Compaq Insight Manager allows you to use any current Web browser to display a wide array of Tru64 UNIX configuration information. You can use a Web browser on your Tru64 UNIX system, or a Web browser that is running on a Windows PC; the choice is up to you.

As implemented for Tru64 UNIX, Compaq Insight Manager is a Web-based interface that uses a combination of a private `http` server (listening on port 2301) on the Tru64 UNIX system and Tru64 UNIX SNMP subagents to display configuration information for cluster members. That is, the SNMP subagents `/usr/sbin/os_mibs` and `/usr/sbin/cpq_mibs` can get, but not set, attributes.

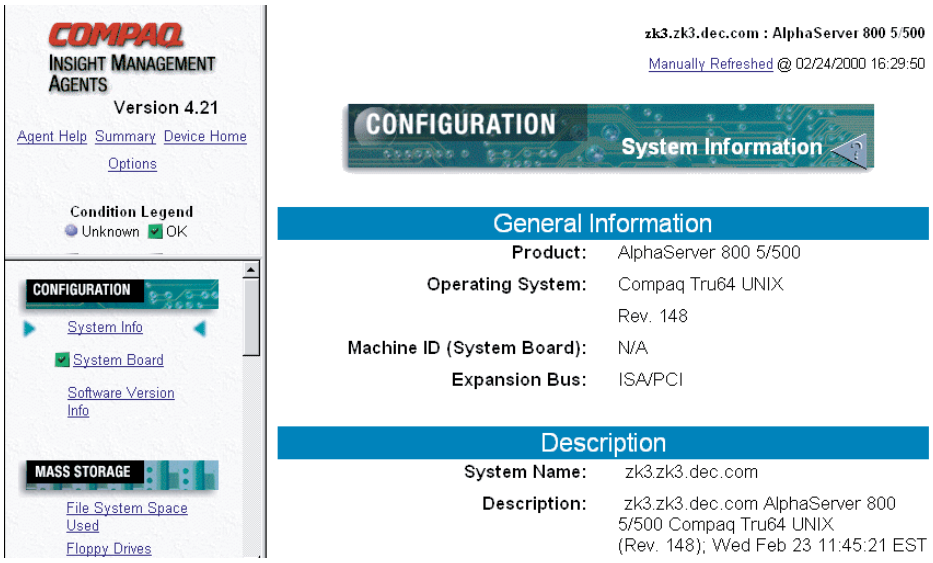
The Compaq Insight Manager Web agent is initialized during transition to run level 3, and the initialization script is located in `/sbin/rc3.d/S50insightd`. This script runs `/usr/share/sysman/bin/insightd` and prints a boot-time message at the console when the agent is successfully started. The SNMP subagents `/usr/sbin/os_mibs` and `/usr/sbin/cpq_mibs` are also invoked during transition to run level 3 and are invoked by the `/sbin/rc.3d/S49snmpd` script.

The Compaq Insight Management agents are not cluster-aware, but they provide useful device and status information about the cluster member you specify. In particular, you might find that the Compaq Insight Management agents allow less-experienced help and support staff to gather system and

device information, such as the capacity and serial number of a given disk device, without having to use the Tru64 UNIX command-line interface.

A sample Compaq Insight Manager display is shown in Figure 2–7.

Figure 2–7: The Compaq Insight Manager Display



ZK-1701U-AI

See `insight_manager(5)` for a description of the Compaq Insight Manager browser requirements. Compaq Insight Manager requires that Java, JavaScript, and cookies be enabled.

2.10.1 Invoking Compaq Insight Manager

To invoke Compaq Insight Manager, open the following URL on the cluster member that you want to manage and navigate to the Insight Management Agents section:

`http://`*`cluster_member_name`*`:2301`

The Navigation frame lists all the subcomponents for which data can be obtained and any associated data items. It provides status data on hardware, such as network interface (NIC) cards, and also data on general system status, such as CPU utilization. The content of this frame depends on what device data is available to Compaq Insight Manager. Typical categories include the following:

- Configuration
- Mass storage

- NIC
- Utilization
- Recovery

2.11 Using Tru64 UNIX Configuration Report

In addition to the features that are provided by Compaq Insight Manager, you can use your Web browser to run a system check on a cluster member. This system check runs the `sys_check` command for you, and requires the same privileges as `sys_check` run from the command line.

If you generate a new report, the browser launches the SysMan Menu Java applet for you. For this to work correctly, you need to use a compatible browser, as described in Section 2.7.

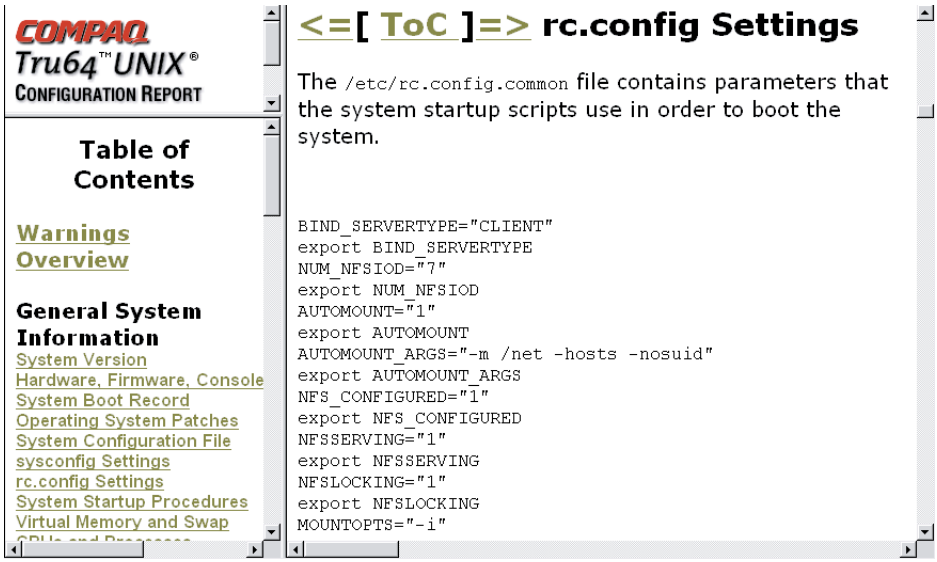
You can use any browser to view the resulting report.

To launch the Configuration Report, open the following URL on the cluster member that you want to manage:

`http://`*`cluster_member_name`*`:2301/WEBAGENT/syschk.TPL`

A sample Configuration Report display is shown in Figure 2–8.

Figure 2–8: Sample Configuration Report Display



ZK-1696U-AI

To generate a new system configuration report, click on Create New Report at the top of the Web page. This launches the SysMan Menu Java applet in

your browser, which allows you to specify the type of information that you want to see in the report before creating it.

To view this new report from the browser, make sure that you select the Export To Web option.

3

Managing the Cluster Alias Subsystem

As cluster administrator, you control the number of aliases, the membership of each alias, and the attributes specified by each member of an alias. For example, you can set the weighting selections that determine how client requests for `in_multi` services are distributed among members of an alias. You also control the alias-related attributes assigned to ports in the `/etc/clua_services` file.

This chapter discusses the following topics:

- A summary of cluster alias features (Section 3.1)
- The files that control alias configuration and behavior (Section 3.2)
- Planning for cluster aliases (Section 3.3)
- Preparing to create cluster aliases (Section 3.4)
- Specifying a cluster alias (Section 3.5)
- Modifying cluster alias and service attributes (Section 3.6)
- Leaving a cluster alias (Section 3.7)
- Monitoring cluster aliases (Section 3.8)
- Load balancing (Section 3.9)
- Extending clusterwide port space (Section 3.10)
- Enabling cluster alias vMAC support (Section 3.11)
- Routing configuration guidelines (Section 3.12)
- Cluster alias and NFS (Section 3.13)
- Cluster alias and cluster application availability (CAA) (Section 3.14)

You can use both the `cluamgr` command and the SysMan Menu to configure cluster aliases:

- The `cluamgr` command-line interface configures parameters for aliases on the cluster member where you run the command. The parameters take effect immediately; however, they do not survive a reboot unless you also add the command lines to the `clu_alias.config` file for that member.
- The SysMan Menu graphical user interface (GUI) configures static parameters for all cluster members. Static parameters are written to

the member's clu_alias.config file but do not take effect until the next boot.

3.1 Summary of Alias Features

The chapter on cluster alias in the TruCluster Server *Cluster Technical Overview* manual describes cluster alias concepts. Read that chapter before modifying any alias or service attributes.

The following list summarizes important facts about the cluster alias subsystem:

- A cluster can have multiple cluster aliases with different sets of members.
- There is one default cluster alias per cluster. The name of the default cluster alias is the name of the cluster.
- An alias is defined by an IP address, not by a Domain Name System (DNS) name. An alias IP address can reside in either a common subnet or a virtual subnet.
- A cluster member must specify an alias in order to advertise a route to it. A cluster member must join an alias to receive connection requests or packets addressed to that alias.
  - To specify the alias clua_ftp, enter the following command:

    ```
    # cluamgr -a alias=clua_ftp
    ```

    This makes an alias name known to the cluster member on which you run the command, and configures the alias with the default set of alias attributes. The cluster member will advertise a route for the alias, but is not a member of the alias.
  - To specify and join the alias clua_ftp, enter the following command:

    ```
    # cluamgr -a alias=clua_ftp,join
    ```

    This command makes an alias name known to the cluster member on which you run the command, configures the alias with the default set of alias attributes, and joins this alias. The cluster member is a member of the alias and can both advertise a route to and receive connection requests or packets addressed to the alias.
- Each cluster member manages its own set of aliases. Entering a cluamgr command on one member affects only that member.
- The /etc/clu_alias.config file is a context-dependent symbolic link (CDSL) pointing to member-specific cluster alias configuration files. Each member's file contains cluamgr command lines that:

– Specify and join the default cluster alias. The `clu_create` and `clu_add_member` commands add the following line to a new member's `clu_alias.config` file:

```
/usr/sbin/cluamgr -a selw=3,selp=1,join,alias=DEFAULTALIAS
```

(The cluster alias subsystem automatically associates the keyword `DEFAULTALIAS` with a cluster's default cluster alias.)

– Specify any other aliases that this member will either advertise a route for or join.

– Set options for aliases; for example, the selection weight and routing priority.

Because each cluster member reads its copy of `clu_alias.config` at boot time, alias definitions and membership survive reboots. Although you can manually edit the file, the preferred method is through the SysMan Menu. Because edits made by SysMan do not take effect until the next boot, use the `cluamgr` command to have the new values to take effect immediately.

- Members of aliases whose names are in the `/etc/exports.aliases` file will accept Network File System (NFS) requests addressed to those aliases. This lets you use aliases other than the default cluster alias as NFS servers.
- Because the mechanisms that cluster alias uses to advertise routes are incompatible with `ogated` and `routed` daemons, `gated` is the required routing daemon in a TruCluster Server cluster.

  When needed, the alias daemon `aliasd` adds host route entries to a cluster member's `/etc/gated.conf.member`*n* file. The alias daemon does not modify any member's `gated.conf` file.

Note

The `aliasd` daemon supports only the Routing Information Protocol (RIP).

- The ports that are used by services that are accessed through a cluster alias are defined as either `in_single` or `in_multi`. These definitions have nothing to do with whether the service can or cannot run on more than one cluster member at the same time. From the point of view of the cluster alias subsystem:

  – When a service is designated as `in_single`, only one alias member will receive connection requests or packets that are addressed to the service. If that member becomes unavailable, the cluster alias subsystem selects another member of the alias as the recipient for all requests and packets addressed to the service.

– When a service is designated as `in_multi`, the cluster alias subsystem routes connection requests and packets for that service to all eligible members of the alias.

By default, the cluster alias subsystem treats all service ports as `in_single`. In order for the cluster alias subsystem to treat a service as `in_multi`, the service must either be registered as an `in_multi` service in the `/etc/clua_services` file, through a call to the `clua_registerservice()` function, or through a call to the `clusvc_getcommport()` or `clusvc_getresvcommport()` functions.

- The following attributes identify each cluster alias:

  Clusterwide attributes:

  – IP address and mask: Identifies an alias

  Per-member attributes:

  – Router priority: Controls proxy Address Resolution Protocol (ARP) router selection for aliases on a common subnet.

  – Selection priority: Creates logical subsets of aliases within an alias. You can use selection priority to control which members of an alias normally service requests. As long as those members with the highest selection priority are up, members with a lower selection priority are not given any requests. You can think of selection priority as a way to establish a failover order for the members of an alias.

  – Selection weight: For `in_multi` services, provides static load balancing among members of an alias. It provides a simple, method for controlling which members of an alias get the most connections. The selection weight indicates the number of connections (on average) that a member is given before connections are given to the next alias member with the same selection priority.

3.2 Configuration Files

The following configuration files manage cluster aliases and services:

`/sbin/init.d/clu_alias`

The boot-time startup script for the cluster alias subsystem.

`/etc/clu_alias.config`

A CDSL pointing to a member-specific `clu_alias.config` file, which is called from the `/sbin/init.d/clu_alias` script. Each member's `clu_alias.config` file contains the `cluamgr` commands that are run at boot time to configure and join aliases, including the default cluster alias, for that member. (The `cluamgr` command does not modify or update this file; the SysMan utility edits this file.) Although you can

manually edit the file, the preferred method is through the SysMan Menu.

`/etc/clua_services`

Defines ports, protocols, and connection attributes for Internet services that use cluster aliases. The `cluamgr` command reads this file at boot time and calls `clua_registerservice()` to register each service that has one or more service attributes assigned to it.

If you modify the file, run `cluamgr -f` on each cluster member. For more information, see `clua_services`(4) and `cluamgr`(8).

`/etc/exports.aliases`

Contains the names of cluster aliases (one alias per line) whose members will accept NFS requests. By default, the default cluster alias is the only cluster alias that will accept NFS requests. Use the `/etc/exports.aliases` file to specify additional aliases as NFS servers.

`/etc/gated.conf.membern`

Each cluster member's cluster alias daemon, `aliasd`, creates a `/etc/gated.conf.membern` file for that member. The daemon starts `gated` using this file as `gated`'s configuration file rather than the member's `/cluster/members/{memb}/etc/gated.conf` file.

If you stop alias routing on a cluster member with `cluamgr -r stop`, the alias daemon restarts `gated` with that member's `gated.conf` as `gated`'s configuration file.

3.3 Planning for Cluster Aliases

Managing aliases can be divided into three broad categories:

- Planning the alias configuration for the cluster.
- Doing the general preparation work; for example, making sure that service entries for Internet services are in `/etc/clua_services` with the correct set of attributes.
- Managing aliases.

Consider the following things when planning the alias configuration for a cluster:

- What services will the cluster provide to clients (for example, mail hub, applications server, NFS server, and so on)?
- How many aliases are needed to support client requests effectively?

The default cluster alias might be all that you need. One approach is to use just the default cluster alias for a while, and then decide whether more aliases make sense for your configuration.

  - If your cluster is providing just the stock set of Internet services that are listed in `/etc/services`, the default cluster alias should be sufficient.
  - By default, when a cluster is configured as a Network File System (NFS) server, external clients must use the default cluster alias as the name of the NFS server when mounting file systems that are exported by the cluster. However, you can create additional cluster aliases and use them as NFS servers. This feature is described in Section 3.13 of this chapter, in the *Cluster Technical Overview*, and in `exports.aliases`(4).
  - If your cluster will run a third-party application that uses lots of system resources, you might want to use additional aliases to control access to, and load balancing for, that application.

    Section 3.9 provides an example that uses two cluster aliases to provide control and redundancy for a cluster that is used as a Web server.

- Which cluster members will belong to which aliases?

  If you create aliases that not all cluster members join, make sure that services that are accessed through those aliases are available on the members of the alias. For example, if you create an alias for use as an NFS server, make sure that its members are all directly connected to the storage containing the exported file systems. If a CAA-controlled application is accessed through an alias, make sure that the CAA placement policy does not start the service on a cluster member that is not a member of the alias.

- Which attributes will each member assign to each alias it specifies or joins?

  You can start by accepting the default set of attributes for an alias, `rpri=1,selw=1,selp=1`, and modify attributes later.

- What, if any, additional service attributes do you want to associate with the Internet service entries in `/etc/clua_services`? Do you want to add additional entries for services?

- Will alias addresses reside on an existing common subnet, on a virtual subnet, or on both?

  On a common subnet: Select alias addresses from existing subnets to which the cluster is connected.

Note

Because proxy Address Resolution Protocol (ARP) is used for common subnet cluster aliases, if an extended local area network (LAN) uses routers or switches that block proxy ARP, the alias will be invisible on nonlocal segments. Therefore, if you are using the common subnet configuration, do not configure routers or switches connecting potential clients of cluster aliases to block proxy ARP.

On a virtual subnet: The cluster alias software will automatically configure the host routes for aliases on a virtual subnet. If a cluster member adds the `virtual` attribute when specifying or joining a member, that member will also advertise a network route to the virtual subnet.

Note

A virtual subnet must not have any real systems in it.

The choice of subnet type depends mainly on whether the existing subnet (that is, the common subnet) has enough addresses available for cluster aliases. If addresses are not easily available on an existing subnet, consider creating a virtual subnet. A lesser consideration is that if a cluster is connected to multiple subnets, configuring a virtual subnet has the advantage of being uniformly reachable from all of the connected subnets. However, this advantage is more a matter of style than of substance. It does not make much practical difference which type of subnet you use for cluster alias addresses; do whatever makes the most sense at your site.

3.4 Preparing to Create Cluster Aliases

To prepare to create cluster aliases, follow these steps:

1. For services with fixed port assignments, examine the entries in `/etc/clua_services`. Add entries for any additional services.
2. For each alias, make sure that its IP address is associated with a host name in whatever hosts table your site uses; for example, `/etc/hosts`, Berkeley Internet Name Domain (BIND), or Network Information Service (NIS).

Note

If you modify a `.rhosts` file on a client to allow nonpassword-protected logins and remote shells from the

cluster, use the default cluster alias as the host name, not the host names of individual cluster members. Login requests originating from the cluster use the default cluster alias as the source address.

3. If any alias addresses are on virtual subnets, register the subnet with local routers. (Remember that a virtual subnet cannot have any real systems in it.)

3.5 Specifying and Joining a Cluster Alias

Before you can specify or join an alias, you must have a valid host name and IP address for the alias.

The `cluamgr` command is the command-line interface for specifying, joining, and managing aliases. When you specify an alias on a cluster member, that member is aware of the alias and can advertise a route to the alias. The simplest command that specifies an alias using the default values for all alias attributes is:

```
# cluamgr -a alias=alias
```

When you specify and join an alias on a cluster member, that member can advertise a route to the alias and receive connection requests or packets addressed to that alias. The simplest command that both specifies and joins an alias using the default values for all attributes is:

```
# cluamgr -a alias=alias,join
```

To specify and join a cluster alias, follow these steps:

1. Get a host name and IP address for the alias.
2. Using the SysMan Menu, add the alias. Specify alias attributes when you do not want to use the default values for the alias; for example, to change the value of `selp` or `selw`.

   SysMan Menu only writes the command lines to a member's `clu_alias.config` file. Putting the aliases in a member's `clu_alias.config` file means that the aliases will be started at the next boot, but it does not start them now.

   The following are sample `cluamgr` command lines from one cluster member's `clu_alias.config` file. All alias IP addresses are on a common subnet.

   ```
   /usr/sbin/cluamgr -a alias=DEFAULTALIAS,rpri=1,selw=3,selp=1,join
   /usr/sbin/cluamgr -a alias=clua_ftp,join,selw=1,selp=1,rpri=1,virtual=f
   /usr/sbin/cluamgr -a alias=printall,selw=1,selp=1,rpri=1,virtual=f
   ```

3. Manually run the appropriate `cluamgr` commands on those members to specify or join the aliases, and to restart alias routing. For example:

```
# cluamgr -a alias=clua_ftp,join,selw=1,selp=1,rpri=1
# cluamgr -a alias=printall,selw=1,selp=1,rpri=1
# cluamgr -r start
```

The previous example does not explicitly specify `virtual=f` for the two aliases because `f` is the default value for the `virtual` attribute. As mentioned earlier, to join an alias and accept the default values for the alias attributes, the following command will suffice:

```
# cluamgr -a alias=alias_name,join
```

The following example shows how to configure an alias on a virtual network; it is not much different from configuring an alias on a common subnet.

```
# cluamgr -a alias=virtestalias,join,virtual,mask=255.255.255.0
```

The cluster member specifies, joins, and will advertise a host route to alias `virtestalias` and a network route to the virtual network. The command explicitly defines the subnet mask that will be used when advertising a net route to this virtual subnet. If you do not specify a subnet mask, the alias daemon uses the network mask of the first interface through which the virtual subnet will be advertised.

If you do not want a cluster member to advertise a network route for a virtual subnet, you do not need to specify `virtual` or `virtual=t` for an alias in a virtual subnet. For example, the cluster member on which the following command is run will join the alias, but will not advertise a network route:

```
# cluamgr -a alias=virtestalias,join
```

See `cluamgr`(8) for detailed instructions on configuring an alias on a virtual subnet.

When configuring an alias whose address is in a virtual subnet, remember that the `aliasd` daemon does not keep track of the stanzas that it writes to a cluster member's `gated.conf.member`*n* configuration file for virtual subnet aliases. If more than one alias resides in the same virtual subnet, the `aliasd` daemon creates extra stanzas for the given subnet. This can cause `gated` to exit and write the following error message to the `daemon.log` file:

```
duplicate static route
```

To avoid this problem, modify `cluamgr` virtual subnet commands in `/etc/clu.alias.config` to set the virtual flag only once for each virtual subnet. For example, assume the following two virtual aliases are in the same virtual subnet:

```
/usr/sbin/cluamgr -a alias=virtualalias1,rpri=1,selw=3,selp=1,join,virtual=t
/usr/sbin/cluamgr -a alias=virtualalias2,rpri=1,selw=3,selp=1,join
```

Because there is no `virtual=t` argument for the `virtualalias2` alias, `aliasd` will not add a duplicate route stanza to this member's `gated.conf.member`*n* file.

3.6 Modifying Cluster Alias and Service Attributes

You can run the `cluamgr` command on any cluster member at any time to modify alias attributes. For example, to change the selection weight of the `clua_ftp` alias, enter the following command:

```
# cluamgr -a alias=clua_ftp,selw=2
```

To modify service attributes for a service in `/etc/clua_services`, follow these steps:

1. Modify the entry in `/etc/clua_services`.
2. On each cluster member, enter the following command to force `cluamgr` to reread the file:

   ```
   # cluamgr -f
   ```

Note

Reloading the `clua_services` file does not affect currently running services. After reloading the configuration file, you must stop and restart the service.

For example, the `telnet` service is started by `inetd` from `/etc/inetd.conf`. If you modify the service attributes for `telnet` in `clua_services`, you have to run `cluamgr -f`, and then stop and restart `inetd` in order for the changes to take effect. Otherwise the changes take effect at the next reboot.

3.7 Leaving a Cluster Alias

Enter the following command on each cluster member that you want to leave a cluster alias that it has joined:

```
# cluamgr -a alias=alias,leave
```

If configured to advertise a route to the alias, the member will still advertise a route to this alias but will not be a destination for any connections or packets that are addressed to this alias.

3.8 Monitoring Cluster Aliases

Use the `cluamgr -s all` command to learn the status of cluster aliases. For example:

```
# cluamgr -s all

Status of Cluster Alias: deli.zk3.dec.com

netmask: 0
aliasid: 1
flags: 7<ENABLED,DEFAULT,IP_V4>
connections rcvd from net: 72
connections forwarded: 14
connections rcvd within cluster: 52
data packets received from network: 4083
data packets forwarded within cluster: 2439
datagrams received from network: 28
datagrams forwarded within cluster: 0
datagrams received within cluster: 28
fragments received from network: 0
fragments forwarded within cluster: 0
fragments received within cluster: 0
Member Attributes:
memberid: 1, selw=3, selp=1, rpri=1 flags=11<JOINED,ENABLED>
memberid: 2, selw=2, selp=1, rpri=1 flags=11<JOINED,ENABLED>
```

Note

Running `netstat -i` does not display cluster aliases.

For aliases on a common subnet, you can run `arp -a` on each member to determine which member is routing for an alias. Look for the alias name and `permanent published`. For example:

```
# arp -a | grep permanent
deli (16.140.112.209) at 00-00-f8-24-a9-30 permanent published
```

3.9 Load Balancing

The concept of load balancing applies only to `in_multi` services. All packets and requests for a single-instance service go to only one member of the alias at a time.

The cluster alias subsystem does not monitor the performance of individual cluster members and perform automatic load balancing for `in_multi` services. You control the distribution of connection requests when you assign the selection priority and selection weight for each member of an alias. You can manually modify these values at any time.

You can use an alias's selection priority, `selp=`*n*, to create logical subsets within an alias. For example, assume that four cluster members have joined an alias:

- Members A and B have `selp=5`.
- Members C and D have `selp=4`.

As long as any `selp=5` member can respond to requests, no requests are directed to any `selp=4` member. Therefore, as long as members A and B are capable of serving requests, members C and D will not receive any packets or requests addressed to this alias. You can use selection priority to create a failover hierarchy among members of a cluster alias.

You can use an alias's selection weight, `selw=`*n*, to control the distribution of requests among members of an alias. The selection weight that a member attaches to an alias translates, on average, to the number of requests (per application) that are directed to this member before requests are directed to the next member of the alias with the same selection priority. For example, assume that four cluster members have joined a cluster alias:

- Members A and B have `selw=3`.
- Members C and D have `selw=2`.

Assuming that all selection priorities are the same, the round-robin algorithm will walk through the list of members, distributing `selw` requests to each member before moving to the next. Member A gets 3 requests, then member B gets 3 requests, then member C gets two requests, and so on.

When assigning selection weights to members of an alias, assign higher weights to members whose resources best match those of the application that is accessed through the alias.

An administrator with shell script experience can write a script to monitor the performance of cluster members and use this information as a basis for raising or lowering alias selection weights. In this case, performance is determined by whatever is relevant to the applications in question.

As an example, assume you have a four-member cluster that you want to configure as a Web site whose primary purpose is a file archive. Users will connect to the site and download large files. The cluster consists of four members that are connected to a common network. Within the cluster, members A and B share one set of disks while members C and D share another set of disks. The network interfaces for members A and B are tuned for bulk data transfer (for example, `ftp` transfers); the network interfaces for members C and D are tuned for short timeouts and low latency (connections from the Web).

You define two cluster aliases: `clua_ftp` and `clua_http`. All four cluster members join both aliases, but with different values.

A and B have the following lines in their `/etc/clu_alias.config` files:

```
/usr/sbin/cluamgr -a alias=clu_ftp,selw=1,selp=10,join
/usr/sbin/cluamgr -a alias=clu_http,selw=1,selp=5,join
```

C and D have the following lines in their `/etc/clu_alias.config` files:

```
/usr/sbin/cluamgr -a alias=clu_ftp,selw=1,selp=5,join
/usr/sbin/cluamgr -a alias=clu_http,selw=1,selp=10,join
```

The result is that as long as either A or B is up, they will handle all `ftp` requests; as long as either C or D is up, they will handle all `http` requests. However, because all four members belong to both aliases, if the two primary servers for either alias go down, the remaining alias members (assuming that quorum is maintained) will continue to service client requests.

3.10 Extending Clusterwide Port Space

The number of ephemeral (dynamic) ports that are available clusterwide for services is determined by the `inet` subsystem attributes `ipport_userreserved_min` (default: 1024) and `ipport_userreserved` (default: 5000).

Because port space is shared among all cluster members, clusters with more members might experience contention for available ports. If a cluster has more than two members, we recommend that you set the value of `ipport_userreserved` to its maximum allowable value (65535). (Setting `ipport_userreserved = 65535` has no adverse side effects.)

To set `ipport_userreserved` to its maximum value, follow these steps:

1. On one member of the cluster, add the following lines to the clusterwide `/etc/sysconfigtab.cluster` file to configure members to set `ipport_userreserved` to 65535 when they next reboot:

   ```
   inet:
    ipport_userreserved=65535
   ```

2. On each member of the cluster run the `sysconfig` command to modify the current value of `ipport_userreserved`:

   ```
   # sysconfig -r inet ipport_userreserved=65535
   ```

3.11 Enabling Cluster Alias vMAC Support

When a cluster alias IP address is configured in a common subnet, one cluster member in that subnet will, based on its router priority (`rpri`) value for that alias, act as the alias's proxy ARP master. This member will respond to local ARP requests addressed to the alias, and will broadcast a gratuitous ARP packet to inform other systems of the hardware (MAC) address that is associated with the alias's IP address. The other local systems then update their ARP tables to reflect this cluster-alias-to-MAC association.

However, this broadcast packet is a problem for systems that do not understand gratuitous ARP packets. They will not become aware of changes in the cluster alias-to-MAC association until the normal timeout interval for

their ARP tables has elapsed. A solution is to provide a virtual hardware address (vMAC address) for each cluster alias.

A virtual MAC address is a unique hardware address that can be automatically created for each alias IP address. An alias's vMAC address follows the cluster alias proxy ARP master from node to node as needed. Regardless of which cluster member is serving as the proxy ARP master for the alias, the alias's vMAC address does not change.

When vMAC support is enabled, if a cluster member becomes the proxy ARP master for a cluster alias, it creates a virtual MAC address for use with that cluster alias. A virtual MAC address consists of a prefix (the default is `AA:01`) followed by the IP address of the alias in hexadecimal format. For example, the default vMAC address for an alias whose IP address is 16.140.112.209 is AA:01:10:8C:70:D1:

```
Default vMAC prefix:       AA:01
Cluster Alias IP Address:  16.140.112.209
IP address in hex. format: 10.8C.70:D1
vMAC for this alias:       AA:01:10:8C:70:D1
```

When another cluster member becomes the proxy ARP master for this alias, the virtual MAC address moves with the alias so that a consistent MAC address is presented within the common subnet for each cluster alias.

When configuring vMAC support, configure all cluster members identically. For this reason, set vMAC configuration variables in `/etc/rc.config.common`.

By default, vMAC support is disabled. To enable vMAC support, use `rcmgr` to put the appropriate entry in `/etc/rc.config.common`:

```
# rcmgr -c set VMAC_ENABLED yes
```

Conversely, to disable vMAC support, enter:

```
# rcmgr -c set VMAC_ENABLED no
```

To change the default AA:01 vMAC prefix, enter:

```
# rcmgr -c set VMAC_PREFIX xx:xx
```

To manually enable or disable vMAC support on an individual cluster member, specify the `cluamgr` `vmac` or `novmac` routing option. For example, to enable vMAC support for a cluster member, enter:

```
# cluamgr -r vmac
```

To manually disable vMAC support for an individual cluster member, enter:

```
# cluamgr -r novmac
```

Because all cluster members should have the same vMAC settings, the recommended sequence when enabling vMAC support is as follows:

1. On any cluster member, enter:

   ```
   # rcmgr -c set VMAC_ENABLED yes
   ```

   This ensures that vMAC support is automatically enabled at boot time. However, because setting this variable only affects a member when it reboots, the currently running cluster does not have vMAC support enabled.

2. To manually enable vMAC support for the currently running cluster, enter the following command on each cluster member:

   ```
   # cluamgr -r vmac
   ```

   You do not have to add the `cluamgr -r vmac` command to each cluster member's `/etc/clu_alias.config` file. Running the `cluamgr -r vmac` command manually on each member enables vMAC support now; setting `VMAC_ENABLED` to `yes` in the shared `/etc/rc.config.common` file automatically enables vMAC support at boot time for all cluster members.

3.12 Routing Configuration Guidelines

Cluster alias operations require that all subnets that are connected to the cluster include a functioning router. This allows cluster alias connectivity to work without any manual routing configuration. For a connected subnet with no router, some manual routing configuration is required because the cluster alias daemons on cluster members cannot unambiguously determine and verify routes that act correctly for all possible routing topologies.

If you cannot configure a router in a subnet that is connected to the cluster (for example, one cluster member is connected to an isolated LAN containing only nonrouters), you must manually configure a network route to that subnet on each cluster member that is not connected to that subnet. For each member that *is* connected to a routerless subnet, add a network route to that subnet to that member's cluster interconnect interface.

Note

Multiple clusters on the same LAN can use the same virtual subnet.

This works because of host routes; any router on the LAN will see each cluster alias's individual host route, and will therefore direct packets to the correct cluster. Off of the LAN, advertisements to the virtual subnet will be propagated using the advertised network routes, and packets to cluster alias addresses in the

virtual subnet will find their way to a router on the LAN. In summary, you should not need to use a separate virtual subnet for each cluster as long as (1) host routes are being generated and (2) the clusters share the same LAN.

However, using the same virtual subnet for multiple clusters is more complicated when the clusters are multi-homed. For instance, if two clusters both connect to LAN 1 but are separately connected to LAN 2 and LAN 3, using the same virtual subnet for both clusters does not work for packets that are coming into LAN 2 and LAN 3. A homogeneous LAN connection is required.

3.13 Cluster Alias and NFS

When a cluster is configured as an NFS server, NFS client requests must be directed either to the default cluster alias or to an alias listed in `/etc/exports.aliases`. NFS mount requests directed at individual cluster members are rejected.

As shipped, the default cluster alias is the only alias that NFS clients can use. However, you can create additional cluster aliases. If you put the name of a cluster alias in the `/etc/exports.aliases` file, members of that alias accept NFS requests. This feature is useful when some members of a cluster are not directly connected to the storage that contains exported file systems. In this case, creating an alias with only directly connected systems as alias members can reduce the number of internal hops that are required to service an NFS request.

As described in the *Cluster Technical Overview*, you must make sure that the members of an alias serving NFS requests are directly connected to the storage containing the exported file systems. In addition, if any other cluster members are directly connected to this storage but are not members of the alias, you must make sure that these systems do not serve these exported file systems. Only members of the alias used to access these file systems should serve these file systems. One approach is to use `cfgmgr` to manually relocate these file systems to members of the alias. Another option is to create boot-time scripts that automatically learn which members are serving these file systems and, if needed, relocate them to members of the alias.

Before configuring additional aliases for use as NFS servers, read the sections in the *Cluster Technical Overview* that discuss how NFS and the cluster alias subsystem interact for NFS, TCP, and Internet User Datagram Protocol (UDP) traffic. Also read the `exports.aliases(4)` reference page and the comments at the beginning of the `/etc/exports.aliases` file.

3.14 Cluster Alias and Cluster Application Availability

This section provides a general discussion of the differences between the cluster alias subsystem and cluster application availability (CAA).

There is no obvious interaction between the two subsystems. They are independent of each other. CAA is an application-control tool that starts applications, monitors resources, and handles failover. Cluster alias is a routing tool that handles the routing of connection requests and packets addressed to cluster aliases. They provide complementary functions: CAA decides where an application will run; cluster alias decides how to get there, as described in the following:

- CAA is designed to work with applications that run on one cluster member at a time. CAA provides the ability to associate a group of required resources with an application, and make sure that those resources are available before starting the application. CAA also handles application failover, automatically restarting an application on another cluster member.
- Because cluster alias can distribute incoming requests and packets among multiple cluster members, it is most useful for applications that run on more than one cluster member. Cluster alias advertises routes to aliases, and sends requests and packets to members of aliases.

One potential cause for confusion is the term *single-instance* application. CAA uses this term to refer to an application that runs on only one cluster member at a time. However, for cluster alias, when an application is designated `in_single`, it means that the alias subsystem sends requests and packets to only one instance of the application, no matter how many members of the alias are listening on the port that is associated with the application. Whether the application is running on all cluster members or on one cluster member, the alias subsystem arbitrarily selects one alias member from those listening on the port and directs all requests to that member. If that member stops responding, the alias subsystem directs requests to one of the remaining members.

In the `/etc/clua_services` file, you can designate a service as either `in_single` or `in_multi`. In general, if a service is in `/etc/clua_services` and is under CAA control, designate it as an `in_single` service. However, even if the service is designated as `in_multi`, the service will operate properly for the following reasons:

- CAA makes sure that the application is running on only one cluster member at a time. Therefore, only one active listener is on the port.
- When a request or packet arrives, the alias subsystem will check all members of the alias, but will find that only one member is listening. The alias subsystem then directs all requests and packets to this member.

- If the member can no longer respond, the alias subsystem will not find any listeners, and will either drop packets or return errors until CAA starts the application on another cluster member. When the alias subsystem becomes aware that another member is listening, it will send all packets to the new port.

All cluster members are members of the default cluster alias. However, you can create a cluster alias whose members are a subset of the entire cluster. You can also restrict which cluster members CAA uses when starting or restarting an application (favored or restricted placement policy).

If you create an alias and tell users to access a CAA-controlled application through this alias, make sure that the CAA placement policy for the application matches the members of the alias. Otherwise, you can create a situation where the application is running on a cluster member that is not a member of the alias. The cluster alias subsystem cannot send packets to the cluster member that is running the application.

The following examples illustrate the interaction of cluster alias and service attributes with CAA.

For each alias, the cluster alias subsystem recognizes which cluster members have joined that alias. When a client request uses that alias as the target host name, the alias subsystem sends the request to one of its members based on the following criteria:

- If the requested service has an entry in `clua_services`, the values of the attributes set there. For example, `in_single` versus `in_multi`, or `in_nolocal` versus `in_noalias`. Assume that the example service is designated as `in_multi`.
- The selection priority (`selp`) that each member has assigned to the alias.
- The selection weight (`selw`) that each member has assigned to the alias. The alias subsystem uses `selp` and `selw` to determine which members of an alias are eligible to receive packets and connection requests.
- Is this eligible member listening on the port associated with the application?
- If so, forward the connection request or packet to the member.
- If not, look at the next member of the alias that meets the `selp` and `selw` requirements.

Assume the same scenario, but now the application is controlled by CAA. As an added complication, assume that someone has mistakenly designated the application as `in_multi` in `clua_services`.

- The cluster alias subsystem receives a connection request or packet.

- Of all eligible alias members, only one is listening (because CAA runs the application on only one cluster member).
- The cluster alias subsystem determines that it has only one place to send the connection request or packet, and sends it to the member where CAA is running the application (the `in_multi` is, in essence, ignored).

In yet another scenario, the application is not under CAA control and is running on several cluster members. All instances bind and listen on the same well-known port. However, the entry in `clua_services` is not designated `in_multi`; therefore, the cluster alias subsystem treats the port as `in_single`:

- The cluster alias subsystem receives a connection request or packet.
- The port is `in_single`.
- The cluster alias subsystem picks an eligible member of the alias to receive the connection request or packet.
- The cluster alias subsystem sends connection requests or packets only to this member until the member goes down or the application crashes, or for some reason there is no longer an active listener on that member.

And finally, a scenario that demonstrates how not to combine CAA and cluster alias:

- Cluster members A and B join a cluster alias.
- CAA controls an application that has a restricted host policy and can run on cluster members A and C.
- The application is running on node A. Node A fails. CAA relocates the application to node C.
- Users cannot access the application through the alias, even though the service is running on node C.

4

Managing Cluster Membership

Clustered systems share various data and system resources, such as access to disks and files. To achieve the coordination that is necessary to maintain resource integrity, the cluster must have clear criteria for membership and must disallow participation in the cluster by systems that do not meet those criteria.

This section provides the following information:

- An overview of the connection manager functions (Section 4.1)
- A discussion of quorum, votes, and cluster membership (Section 4.2)
- A discussion of how the connection manager calculates quorum (Section 4.3)
- An example using a three-member cluster (Section 4.4)
- When and how to use a quorum disk (Section 4.5)
- How to use the `clu_quorum` command to display cluster quorum information (Section 4.6)
- Examples that illustrate the results of various vote settings (Section 4.7)
- How to monitor the connection manager (Section 4.8)
- How to interpret connection manager panics (Section 4.9)
- How to troubleshoot unfortunate expected vote and node vote settings (Section 4.10)

4.1 Connection Manager

The **connection manager** is a distributed kernel component that monitors whether cluster members can communicate with each other, and enforces the rules of cluster membership. The connection manager:

- Forms a cluster, adds members to a cluster, and removes members from a cluster
- Tracks which members in a cluster are active
- Maintains a cluster membership list that is consistent on all cluster members

- Provides timely notification of membership changes using Event Manager (EVM) events
- Detects and handles possible cluster partitions

An instance of the connection manager runs on each cluster member. These instances maintain contact with each other, sharing information such as the cluster's membership list. The connection manager uses a three-phase commit protocol to ensure that all members have a consistent view of the cluster.

4.2 Quorum and Votes

The connection manager ensures data integrity in the face of communication failures by using a voting mechanism. It allows processing and I/O to occur in a cluster only when a majority of **votes** are present. When the majority of votes are present, the cluster is said to have **quorum**.

The mechanism by which the connection manager calculates quorum and allows systems to become and remain clusters members depends on a number of factors, including expected votes, current votes, node votes, and quorum disk votes. This section describes these concepts.

4.2.1 How a System Becomes a Cluster Member

The connection manager is the sole arbiter of cluster membership. A node that has been configured to become a cluster member, either through the `clu_create` or `clu_add_member` command, does not become a cluster member until it has rebooted with a clusterized kernel and is allowed to form or join a cluster by the connection manager. The difference between a cluster member and a node that is configured to become a cluster member is important in any discussion of quorum and votes.

After a node has formed or joined a cluster, the connection manager forever considers it to be a cluster member (until someone uses `clu_delete_member` to remove it from the cluster). In rare cases a disruption of communications in a cluster (such as that caused by broken or disconnected hardware) might cause an existing cluster to divide into two or more clusters. In such a case, which is known as a **cluster partition**, nodes may consider themselves to be members of one cluster or another. However, as discussed in Section 4.3, the connection manager at most allows only one of these clusters to function.

4.2.2 Expected Votes

Expected votes are the number of votes that the connection manager expects when all configured votes are available. In other words, expected votes should be the sum of all node votes (see Section 4.2.4) that are

configured in the cluster, plus the vote of the quorum disk, if one is configured (see Section 4.2.5). Each member brings its own notion of expected votes to the cluster; it is important that all members agree on the same number of expected votes.

The connection manager refers to the node expected votes settings of booting cluster members to establish its own internal clusterwide notion of expected votes, which is referred to as **cluster expected votes**. The connection manager uses its cluster expected votes value to determine the number of votes the cluster requires to maintain quorum, as explained in Section 4.3.

Use the `clu_quorum` or `clu_get_info -full` command to display the current value of cluster expected votes.

The `clu_create` and `clu_add_member` scripts automatically adjust each member's expected votes as a new voting member or quorum disk is configured in the cluster. The `clu_delete_member` command automatically lowers expected votes when a member is deleted. Similarly, the `clu_quorum` command adjusts each member's expected votes as a quorum disk is added or deleted or node votes are assigned to or removed from a member. These commands ensure that the member-specific expected votes value is the same on each cluster member and that it is the sum of all node votes and the quorum disk vote (if a quorum disk is configured).

A member's expected votes are initialized from the `cluster_expected_votes` kernel attribute in the `clubase` subsystem of its member-specific `etc/sysconfigtab` file. Use the `clu_quorum` command to display a member's expected votes.

To modify a member's expected votes, you must use the `clu_quorum -e` command. This ensures that all members have the same and correct expected votes settings. You cannot modify the `cluster_expected_votes` kernel attribute directly.

4.2.3 Current Votes

If expected votes are the number of configured votes in a cluster, **current votes** are the number of votes that are contributed by current members and any configured quorum disk that is on line. Current votes are the actual number of votes that are visible within the cluster.

4.2.4 Node Votes

Node votes are the fixed number of votes that a given member contributes towards quorum. Cluster members can have either 1 or 0 (zero) node votes. Each member with a vote is considered to be a **voting member** of the cluster. A member with 0 (zero) votes is considered to be a **nonvoting member**.

Note

Single-user mode does not affect the voting status of the member. A member contributing a vote before being shut down to single-user mode continues contributing the vote in single-user mode. In other words, the connection manager still considers a member that is shut down to single-user mode to be a cluster member.

Voting members can form a cluster. Nonvoting members can only join an existing cluster.

You typically assign votes to a member during cluster configuration; for example, while running `clu_create` to create the first cluster member or running `clu_add_member` to add new members. By default, `clu_create` gives the first member 1 vote. By default, the number of votes `clu_add_member` offers for new potential members is 0 (zero) if expected votes is 1, or 1 if expected votes is greater than 1. (`clu_create` and `clu_add_member` automatically increment expected votes when configuring a new vote in the cluster.) You can later adjust the number of node votes that is given to a cluster member by using the `clu_quorum -m` command.

A member's votes are initially determined by the `cluster_node_votes` kernel attribute in the `clubase` subsystem of its member-specific `etc/sysconfigtab` file. Use either the `clu_quorum` or `clu_get_info -full` command to display a member's node votes. See Section 4.6 for more information.

To modify a member's node votes, you must use the `clu_quorum` command. You cannot modify the `cluster_node_votes` kernel attribute directly.

4.2.5 Quorum Disk Votes

In certain cluster configurations, described in Section 4.5, you may enhance cluster availability by configuring a **quorum disk**. **Quorum disk votes** are the fixed number of votes that a quorum disk contributes towards quorum. A quorum disk can have either 1 or 0 (zero) votes.

You typically configure a quorum disk and assign it a vote while running `clu_create` to create the cluster. If you define a quorum disk at cluster creation, it is given one vote by default.

Quorum disk votes are initialized from the `cluster_qdisk_votes` kernel attribute in the `clubase` subsystem of each member's `etc/sysconfigtab` file. Use either the `clu_quorum` command or `clu_get_info` command to display quorum disk votes.

To modify the quorum disk votes, you must use the `clu_quorum` command. You cannot modify the `cluster_qdisk_votes` kernel attribute directly.

When configured, a quorum disk's vote plays a unique role in cluster formation because of the following rules that are enforced by the connection manager:

- A booting node cannot form a cluster unless it has quorum.
- Before the node can claim the quorum disk and its vote, it must be a cluster member.

In the situation where the booting node needs the quorum disk vote to achieve quorum, these rules create an impasse: the booting node would never be able to form a cluster.

The connection manager resolves this dilemma by allowing booting members to provisionally apply the quorum disk vote towards quorum. This allows a booting member to achieve quorum and form the cluster. After it has formed the cluster, it claims the quorum disk. At that point, the quorum disk's vote is no longer provisional; it is real.

4.3 Calculating Cluster Quorum

The quorum algorithm is the method by which the connection manager determines the circumstances under which a given member can participate in a cluster, safely access clusterwide resources, and perform useful work. The algorithm operates dynamically: that is, cluster events trigger its calculations, and the results of its calculations can change over the lifetime of a cluster.

The quorum algorithm operates as follows:

1. The connection manager selects a set of cluster members upon which it bases its calculations. This set includes all members with which it can communicate. For example, it does not include configured nodes that have not yet booted, members that are down, or members that it cannot reach due to a hardware failure (for example, a detached cluster interconnect cable or a bad Memory Channel adapter).
2. When a cluster is formed and each time a node boots and joins the cluster, the connection manager calculates a value for cluster expected votes using the largest of the following values:
   - Maximum member-specific expected votes value from the set of proposed members selected in step 1.
   - The sum of the node votes from the set of proposed members that were selected in step 1, plus the quorum disk vote if a quorum disk is configured.

- The previous cluster expected votes value.

Consider a three-member cluster with no quorum disk. All members are up and fully connected; each member has one vote and has its member-specific expected votes set to 3. The value of cluster expected votes is currently 3.

A fourth voting member is then added to the cluster. When the new member boots and joins the cluster, the connection manager calculates the new cluster expected votes as 4, which is the sum of node votes in the cluster.

Use the `clu_quorum` or `clu_get_info -full` command to display the current value of cluster expected votes.

3. Whenever the connection manager recalculates cluster expected votes (or resets cluster expected votes as the result of a `clu_quorum -e` command), it calculates a value for quorum votes.

   Quorum votes is a dynamically calculated clusterwide value, based on the value of cluster expected votes, that determines whether a given node can form, join, or continue to participate in a cluster. The connection manager computes the clusterwide quorum votes value using the following formula:

   ```
   quorum votes = round_down((cluster_expected_votes+2)/2)
   ```

   For example, consider the three-member cluster from the previous step. With cluster expected votes set to 3, quorum votes are calculated as round_down((3+2)/2), or 2. In the case where the fourth member was added successfully, quorum votes are calculated as 3 (round_down((4+2)/2)).

   Note

   Expected votes (and, hence, quorum votes) are based on cluster configuration, rather than on which nodes are up or down. When a member is shut down, or goes down for any other reason, the connection manager does not decrease the value of quorum votes. Only member deletion and the `clu_quorum -e` command can lower the quorum votes value of a running cluster.

4. Whenever a cluster member senses that the number of votes it can see has changed (a node has joined the cluster, an existing member has been deleted from the cluster, or a communications error is reported), it compares current votes to quorum votes.

   The action the member takes is based on the following conditions:

- If the value of current votes is greater than or equal to quorum votes, the member continues running or resumes (if it had been in a suspended state).
- If the value of current votes is less than quorum votes, the member suspends all process activity, all I/O operations to cluster-accessible storage, and all operations across networks external to the cluster until sufficient votes are added (that is, until enough members have joined the cluster or the communications problem is mended) to bring current votes to a value greater than or equal to quorum.

The comparision of current votes to quorum votes occurs on a member-by-member basis, although events may make it appear that quorum loss is a clusterwide event. When a cluster member loses quorum, all of its I/O is suspended and all network interfaces except the Memory Channel interfaces are turned off. No commands that must access a clusterwide resource work on that member. It may appear to be hung.

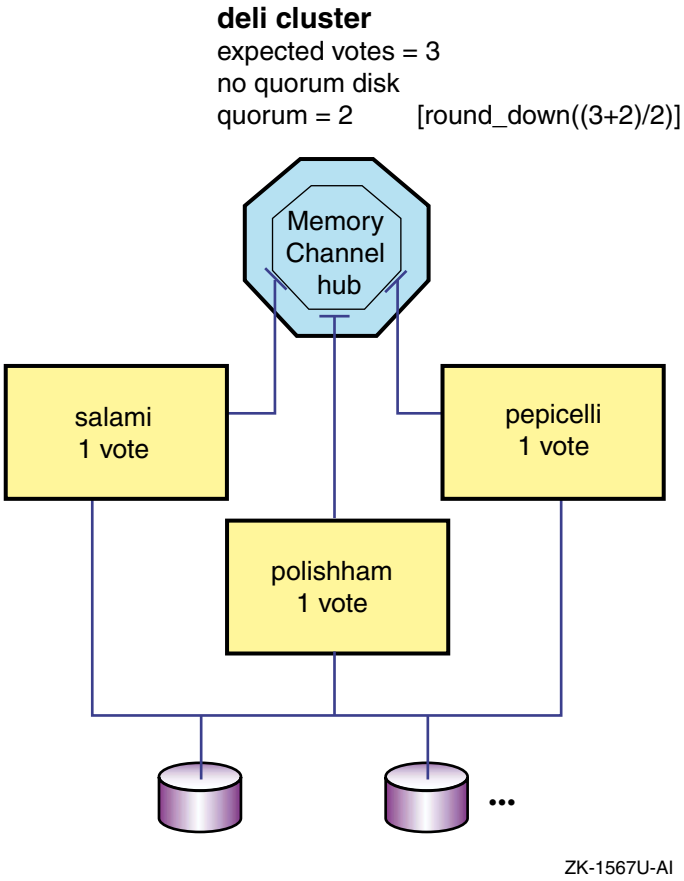
Depending upon how the member lost quorum, you may be able to remedy the situation by booting a member with enough votes for the member in quorum hang to achieve quorum. If all cluster members have lost quorum, your options are limited to booting a new member with enough votes for the members in quorum hang to achieve quorum, rebooting the entire cluster, or resorting to the procedures that are discussed in Section 4.10.

4.4 A Connection Manager Example

The connection manager forms a cluster when enough nodes with votes have booted for the cluster to have quorum, possibly after claiming the vote of a quorum disk.

Consider the three-member `deli` cluster in Figure 4–1. When all members are up and operational, each member contributes one node vote; cluster expected votes is 3, and quorum votes is calculated as 2. The `deli` cluster can survive the failure of any one member.

Figure 4–1: The Three-Member deli Cluster



When node `salami` was first booted, the console displayed the following messages:

```
CNX MGR: Node salami id 3 incarn 0xbde0f attempting to form or join cluster
deli
CNX MGR: insufficient votes to form cluster: have 1 need 2
CNX MGR: insufficient votes to form cluster: have  1 need 2
.
.
.
```

When node `polishham` was booted, its node vote plus `salami`'s node vote allowed them to achieve quorum (2) and proceed to form the cluster, as evidenced by the following `CNX MGR` messages:

```
.
.
.
CNX MGR: Cluster deli incarnation 0x1921b has been formed
Founding node id is 2 csid is 0x10001
CNX MGR: membership configuration index: 1 (1 additions, 0 removals)
```

```
CNX MGR: quorum (re)gained, (re)starting cluster operations.
CNX MGR: Node salami 3 incarn 0xbde0f csid 0x10002 has been added to the
cluster
CNX MGR: Node polishham 2 incarn 0x15141 csid 0x10001 has been added to the
cluster
```

The boot log of node `pepicelli` shows similar messages as `pepicelli` joins the existing cluster, although, instead of the cluster formation message, it displays:

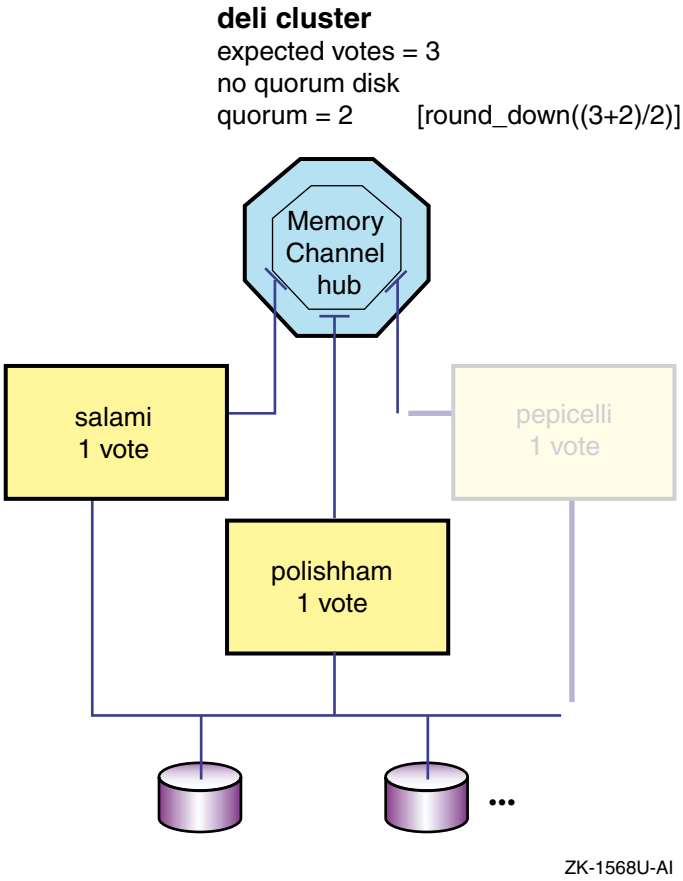
```
CNX MGR: Join operation complete
CNX MGR: membership configuration index: 2 (2 additions, 0 removals)
CNX MGR: Node pepicelli 1 incarn 0x26510f csid 0x10003 has been added to the
cluster
```

Of course, if `pepicelli` is booted at the same time as the other two nodes, it participates in the cluster formation and shows cluster formation messages like those nodes.

If `pepicelli` is then shut down, as shown in Figure 4–2, members `salami` and `polishham` each compare their notions of cluster current votes (2) against quorum votes (2). Because current votes equals quorum votes, they can proceed as a cluster and survive the shutdown of `pepicelli`. The following log messages describe this activity:

```
memory channel - removing node 2
rm_remove_node: removal took 0x0 ticks
ccomsub: Successfully reconfigured for member 2 down
ics_RM_membership_change: Node 3 in RM slot 2 has  gone down
CNX MGR: communication error detected for node 3
CNX MGR: delay 1 secs 0 usecs
        .
        .
        .
CNX MGR: Reconfig operation complete
CNX MGR: membership configuration index: 13 (2 additions, 1 removals)
CNX MGR: Node pepicelli 3 incarn 0x21d60 csid 0x10001 has been removed
         from the cluster
```

Figure 4–2: Three-Member deli Cluster Loses a Member



However, this cluster cannot survive the loss of yet another member. Shutting down member `polishham` results in the situation that is depicted in Figure 4–3 and discussed in Section 4.5. The `deli` cluster loses quorum and ceases operation with the following messages:

```
memory channel - removing node 4
rm_remove_node: removal took 0x0 ticks
ccomsub: Successfully reconfigured for member 4 down
ics_RM_membership_change: Node 2 in RM slot 4 has gone down
CNX MGR: communication error detected for node 2
CNX MGR: delay 1 secs 0 usecs
CNX MGR: quorum lost, suspending cluster operations.
        .
        .
        .
CNX MGR: Reconfig operation complete
CNX MGR: membership configuration index: 16 (8 additions, 8 removals)
CNX MGR: Node pepicelli 2 incarn 0x59fb4 csid 0x50001 has been removed
         from the cluster
```

4.5 Using a Quorum Disk

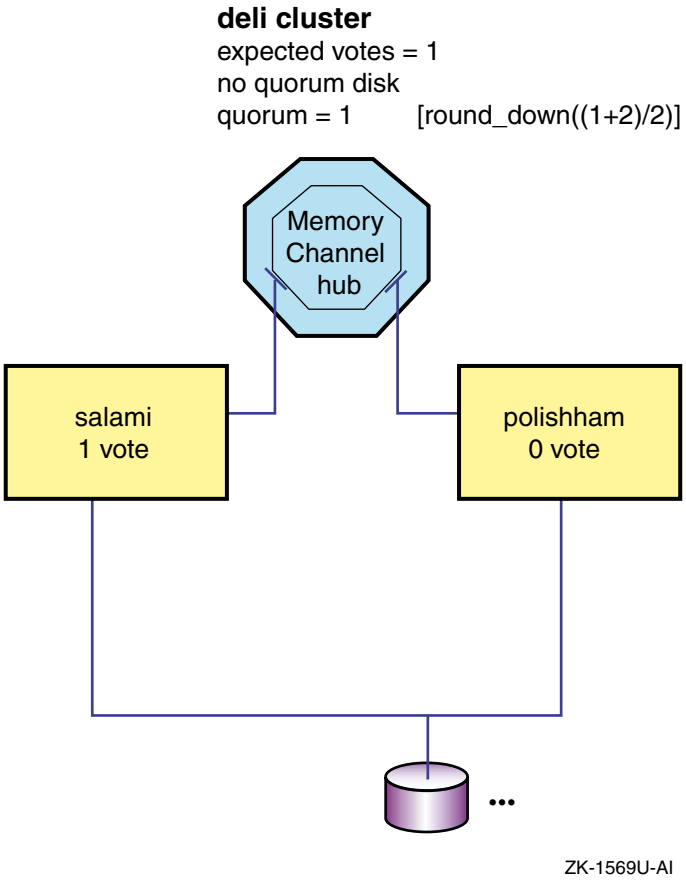
In a two-member cluster configuration, where each member has one member vote and expected votes has the value of 2, the loss of a single member will cause the cluster to lose quorum and all applications to be suspended. This type of configuration is not highly available.

A more realistic (but not substantially better) two-member configuration assigns one member 1 vote and the second member 0 (zero) votes. Expected votes are 1. This cluster can lose its second member (the one with no votes) and remain up. However, it cannot afford to lose the first member (the voting one).

To foster better availability in such a configuration, you can designate a disk on a shared bus as a quorum disk. The quorum disk acts as a virtual cluster member whose purpose is to add one vote to the total number of expected votes. When a quorum disk is configured in a two-member cluster, the cluster can survive the failure of either the quorum disk or one member and continue operating.

For example, consider the two-member `deli` cluster without a quorum disk shown in Figure 4–3.

Figure 4–3: Two-Member deli Cluster Without a Quorum Disk



One member contributes 1 node vote and the other contributes 0, so cluster expected votes is 1. The connection manager calculates quorum votes as follows:

```
quorum votes = round_down((cluster_expected_votes+2)/2) =
round_down((1+2)/2) = 1
```

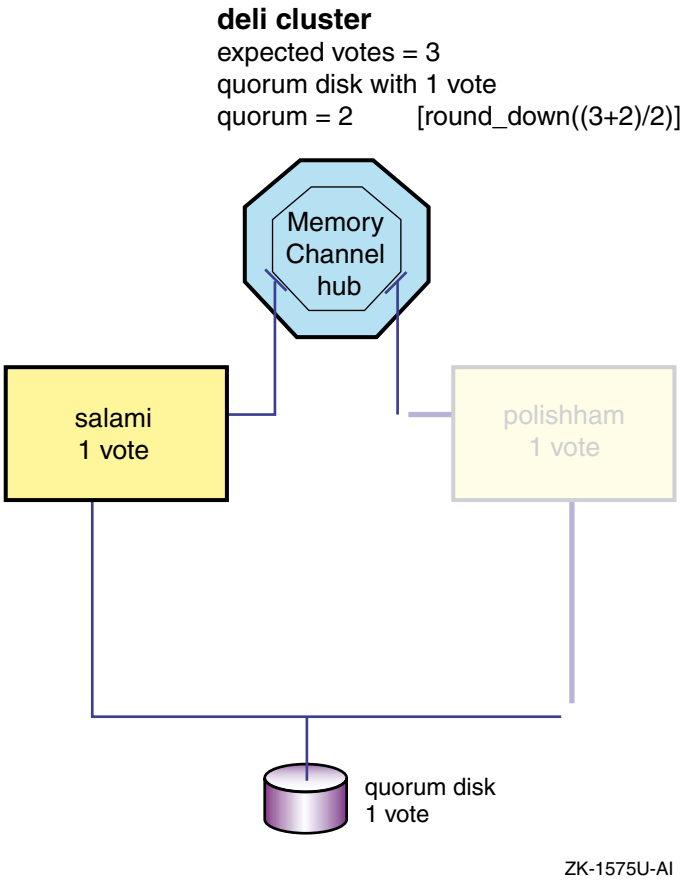
The failure or shutdown of member `salami` causes member `polishham` to lose quorum. Cluster operations are suspended.

However, if the cluster includes a quorum disk (adding one vote to the total of cluster expected votes), and member `polishham` is also given a vote, expected votes become 3 and quorum votes become 2:

```
quorum votes = round_down((cluster_expected_votes+2)/2) =
round_down((3+2)/2) = 2
```

Now, if either member or the quorum disk leaves the cluster, sufficient current votes remain to keep the cluster from losing quorum. The cluster in Figure 4–4 can continue operation.

Figure 4–4: Two-Member deli Cluster with Quorum Disk Survives Member Loss



The `clu_create` utility allows you to specify a quorum disk at cluster creation and assign it a vote. You can also use the `clu_quorum` utility to add a quorum disk at some other moment in the life of a cluster; for example, when the result of a `clu_delete_member` is a two-member cluster with compromised availability.

To configure a quorum disk, use the `clu_quorum -d add` command. For example, the following command defines `/dev/disk/dsk11` as a quorum disk with one vote:

```
# clu_quorum -d add dsk11 1
Collecting quorum data for Member(s): 1 2

Info: Disk available but has no label: dsk11
  Initializing cnx partition on quorum disk : dsk11h

Successful quorum disk creation.
# clu_quorum
```

```
Cluster Common Quorum Data
Quorum disk:   dsk11h
        .
        .
        .
```

The following restrictions apply to the use of a quorum disk:

- A cluster can have only one quorum disk.
- The quorum disk should be on a shared bus to which all cluster members are directly connected. If it is not, members that do not have a direct connection to the quorum disk may lose quorum before members that do have a direct connection to it.
- The quorum disk must not contain any data. The `clu_quorum` command will overwrite existing data when initializing the quorum disk. The integrity of data (or file system metadata) that is placed on the quorum disk from a running cluster is not guaranteed across member failures.

  This means that the member boot disks and the disk holding the clusterwide root (/) cannot be used as quorum disks.
- The quorum disk can be quite small. The cluster subsystems use only 1 MB of the disk.
- A quorum disk can have either 1 vote or no votes. In general, a quorum disk should always be assigned a vote. You might assign an existing quorum disk no votes in certain testing or transitory configurations, such as a one-member cluster (in which a voting quorum disk introduces a second point of failure).
- You cannot use the Logical Storage Manager (LSM) on the quorum disk.

Conceptually, a vote that is supplied by a quorum disk serves as a tie-breaker in cases where a cluster can partition with an even number of votes on either side of the partition. The tie-breaker vote allows one side to achieve quorum and continue cluster operations. In this regard, the quorum disk's vote is no different than a vote, for example, that is brought to a two-member cluster by a third voting member or brought to a four-member cluster by a fifth voting member. This is an important consideration when planning larger clusters containing many non-voting members that do not have direct connectivity to all shared storage.

Consider a cluster containing two large members that act as file servers. Because these members are directly connected to the important cluster file systems and application databases, they are considered critical to the operation to the cluster and are each assigned one vote. The other members of this cluster process client requests and direct them to the servers. Because they are not directly connected to shared storage, they are less critical to cluster operation and are assigned no votes. However, because this cluster

has only two votes, it cannot withstand the failure of a single file server member until we configure a tie-breaker vote.

In this case, what should provide the tie-breaker vote? Configuring a quorum disk with the vote is be a poor choice. The quorum disk in this configuration is directly connected to only the two file server members. The client processing members, as a result, cannot count its vote towards quorum. If the quorum disk or a single file server member fails, the client processing members lose quorum and stop shipping client requests to the servers. This effectively hampers the operation of the server members, even though they retain quorum. A better solution for providing a tie-breaker vote to this type of configuration is to assign a vote to one of the client processing members. The cluster as a whole can then survive the loss of a single vote and continue to operate.

If you attempt to add a quorum disk and that vote, when added, is be needed to sustain quorum, the `clu_quorum` command displays the following message:

```
Adding the quorum disk could cause a temporary loss
of quorum until the disk becomes trusted.
Do you want to continue with this operation? [yes]:
```

You can usually respond "yes" to this question. It usually takes about 20 seconds for the `clu_quorum` command to determine the trustworthiness of the quorum disk. For the quorum disk to become trusted, the member needs direct connectivity to it, must be able to read to and write from it, and must either claim ownership of it or be a member of the same cluster as a member that claims ownership.

If you attempt to adjust the votes of an existing quorum disk and the member does not consider that disk to be trusted (as indicated by a zero value in the `qdisk_trusted` attribute of the `cnx` subsystem), the `clu_quorum` command displays the following message:

```
The quorum disk does not currently appear to be trusted.
Adjusting the votes on the quorum disk could cause quorum loss.
Do you want to continue with this operation? [no]:
```

If the quorum disk is not currently trusted, it is unlikely to become trusted unless you do something that allows it to meet the preceding requirements. You should probably answer "no" to this question and investigate other ways of adding a vote to the cluster.

4.5.1 Replacing a Failed Quorum Disk

If a quorum disk fails during cluster operation and the cluster does not lose quorum, you can replace the disk by following these steps:

1. Make sure that the disk is disconnected from the cluster.

2. Use the `clu_quorum` command and note the running value of quorum disk votes.
3. Use the `clu_quorum -f -d remove` command to remove the quorum disk from the cluster.
4. Replace the disk. Enter the `hwmgr -scan scsi` command on each cluster member.

Note

You must run `hwmgr -scan scsi` on every cluster member.

Wait a few moments for all members to recognize the presence of the new disk.

5. Use the `hwmgr -view devices -cluster` command to determine the device special file name (that is, the `dsk` name) of the new disk. Its name will be different from that of the failed quorum disk. Optionally, you can use the `dsfmgr -n` command to rename the new device special file to the name of the failed disk.
6. Use the `clu_quorum -f -d add` command to configure the new disk as the quorum disk. The new disk should have the same number of votes as noted in step 2.

If a quorum disk fails during cluster operation and the cluster loses quorum and suspends operations, you must use the procedure in Section 4.10.1 to halt one cluster member and reboot it interactively to restore quorum to the cluster. You can then perform the previous steps.

4.6 Using the clu_quorum Command to Display Cluster Vote Information

When specified without options (or with `-f` and/or `-v`), the `clu_quorum` command displays information about the current quorum disk, member node votes, and expected votes configuration of the cluster. This information includes:

- Cluster common quorum data. This includes the device name of any configured quorum disk, plus quorum information from the clusterwide `/etc/sysconfigtab.cluster`.
- Member-specific quorum data from each member's running kernel and `/etc/sysconfigtab` file, plus an indication of whether the member is `UP` or `DOWN`. By default, no quorum data is returned for a member with `DOWN` status. However, as long as the `DOWN` member's boot partition is accessible to the member running the `clu_quorum` command, you can use the `-f` option to display the `DOWN` member's file quorum data values.

See clu_quorum(8) for a description of the individual items the clu_quorum displays.

4.7 Cluster Vote Assignment Examples

Table 4–1 presents how various settings of the cluster_expected_votes and cluster_node_votes attributes on cluster members affect the cluster's ability to form. It also points out which setting combinations can be disastrous and highlights those that foster the best cluster availability. The table represents two-, three-, and four-member cluster configurations.

In this table:

- "Node Expected Votes" indicates the on-disk setting of the cluster_expected_votes attribute in the clubase stanza of a member's /etc/sysconfigtab file.
- "M1," "M2," "M3", and "M4" indicate the votes that are assigned to cluster members.
- "Qdisk" represents the votes that are assigned to the quorum disk (if configured).
- The notation "---" indicates that a given node has not been configured in the cluster.

Table 4–1: Effects of Various Member cluster_expected_votes Settings and Vote Assignments in a Two- to Four-Member Cluster

Node Expected Votes	M1	M2	M3	M4	Qdisk	Result
1	1	0	---	---	0	Cluster can form only when M1 is present. Cluster can survive the failure of M2 but not M1. This is a common configuration in a two-member cluster when a quorum disk is not used. Try adding a vote to M2 and a quorum disk to this configuration.
2	1	1	---	---	0	Cluster can form only when both members are present. Cluster cannot survive a failure of either member. As discussed in Section 4.4, this is a less available configuration than the previous one. Try a quorum disk in this configuration. See Section 4.5.

Table 4–1: Effects of Various Member cluster_expected_votes Settings and Vote Assignments in a Two- to Four-Member Cluster (cont.)

Node Expected Votes	M1	M2	M3	M4	Qdisk	Result
3	1	1	---	---	1	With the quorum disk configured and given 1 vote, the cluster can survive the failure of either member or the quorum disk. This is the recommended two-member configuration.
1	1	0	0	---	0	Cluster can survive failures of members M2 and M3 but not a failure of M1.
2	1	1	0	---	0	Cluster requires both M1 and M2 to be up. It can survive a failure of M3.
3	1	1	1	---	0	Cluster can survive the failure of any one member. This is the recommended three-member cluster configuration.
4	1	1	1	---	1	Because 3 votes are required for quorum, the presence of a voting quorum disk does not make this configuration any more highly available than the previous one. In fact, if the quorum disk were to fail (an unlikely event), the cluster would not survive a member failure.[a]
4	1	1	1	1	0	Cluster can survive failure of any one member. Try a quorum disk in this configuration. See Section 4.5.
5	1	1	1	1	1	Cluster can survive failure of any two members or of any member and the quorum disk. This is the recommended four-member configuration.

[a] One possible course of action in this situation is to give each member a vote and configure a quorum disk with 1 vote. Expected votes is 4, and quorum is 3. If the quorum disk fails, remove the quorum disk vote by using the `clu_quorum -d adjust 0` command. Or, if a member fails, remove the member vote by using the `clu_quorum -m` *`failed-cluster-member`* `0` command. The resulting cluster (expected votes is 3; quorum is 2) would survive another failure.

4.8 Monitoring the Connection Manager

The connection manager provides several kinds of output for administrators. It posts Event Manager (EVM) events for four types of events:

- Node joining cluster

- Node removed from cluster
- Quorum disk becoming unavailable (due to error, removal, and so on)
- Quorum disk becoming available again

Each of these events also results in console message output.

The connection manager displays various informational messages on the console during member boots and cluster transactions.

A cluster transaction is the mechanism for modifying some clusterwide state on all cluster members atomically; either all members adopt the new value or none do. The most common transactions are membership transactions, such as when the cluster is formed, members join, or members leave. Certain maintenance tasks also result in cluster transactions, such as the addition or removal of a quorum disk, the modification of the clusterwide expected votes value, or the modification of a member's vote.

Cluster transactions are global (clusterwide) occurrences. Console messages are also printed on the console of an individual member in response to certain local events, such as when the connection manager notices a change in connectivity on a given node (to another node or to the quorum disk), or when it gains or loses quorum.

4.9 Connection Manager Panics

The connection manager continuously monitors cluster members. In the rare case of a cluster partition, in which an existing cluster divides into two or more clusters, nodes may consider themselves to be members of one cluster or another. As discussed in Section 4.3, the connection manager at most allows only one of these clusters to function.

To preserve data integrity if a cluster partitions, the connection manager will cause a member to panic. The panic string indicates the conditions under which the partition was discovered. These panics are not due to connection manager problems but are reactions to bad situations, where drastic action is appropriate to ensure data integrity. You cannot repair a partition without rebooting one or more members to have them rejoin the cluster.

The connection manager reacts to the following situations by panicking a cluster member:

- Quorum disk that is attached to two different clusters:

```
CNX QDISK: configuration error. Qdisk in use by cluster of different name.
CNX QDISK: configuration error. Qdisk written by cluster of different name.
```

- Quorum disk ownership that is being contested by different clusters after a cluster partition. The member that discovers this condition

decides either to continue trying to claim the quorum disk or to yield to the other cluster by panicking:

```
CNX QDISK: Yielding to foreign owner with quorum.
CNX QDISK: Yielding to foreign owner with provisional quorum.
CNX QDISK: Yielding to foreign owner without quorum.
```

- Connection manager on a node that is already a cluster member discovers a node that is a member of a different cluster (may be a different incarnation of the same cluster). Depending on quorum status, the discovering node either directs the other node to panic, or panics itself.

```
CNX MGR: restart requested to resynchronize with cluster with quorum.
CNX MGR: restart requested to resynchronize with cluster
```

- Panicking node has discovered a cluster and will try to reboot and join:

```
CNX MGR: rcnx_status: restart requested to resynchronize with cluster
         with quorum.
CNX MGR: rcnx_status: restart requested to resynchronize with cluster
```

- A node is removed from the cluster during a reconfiguration because of communication problems:

```
CNX MGR: this node removed from cluster
```

4.10 Troubleshooting Unfortunate Expected Vote and Node Vote Settings

As long as a cluster maintains quorum, you can use the `clu_quorum` command to adjust node votes, expected votes, and quorum disk votes across the cluster. Using the `-f` option to the command, you can force changes on members that are currently down.

However, if a cluster member loses quorum, all I/O is suspended and all network interfaces except the Memory Channel interfaces are turned off. No commands that must access cluster shared resources work, including the `clu_quorum` command. Either a member with enough votes rejoins the cluster and quorum is regained, or you must halt and reboot a cluster member.

Sometimes you may need to adjust the vote configuration of a cluster that is hung in quorum loss or for a cluster that has insufficient votes to form. The following scenarios describe some cluster problems and the mechanisms you can use to resolve them.

4.10.1 Joining a Cluster After a Cluster Member or Quorum Disk Fails and Cluster Loses Quorum

Consider a cluster that has lost one or more members (or a quorum disk) due to hardware problems — problems that prevent these members from being rebooted. Without these members, the cluster has lost quorum, and its

surviving members' expected votes or node votes settings are not realistic for the downsized cluster. Having lost quorum, the cluster hangs.

You can resolve this type of quorum loss situation without shutting the entire cluster down. The procedure involves halting a single cluster member and rebooting it in such a way that it can join the cluster and restore quorum. After you have booted this member, you must use the `clu_quorum` command to fix the original problem.

Note

If only a single cluster member survives the member or quorum disk failures, use the procedure in Section 4.10.2 for booting a cluster member with sufficient votes to form a cluster.

To restore quorum for a cluster that has lost quorum due to one or more member or quorum disk failures, follow these steps:

1. Halt one cluster member by using its Halt button.
2. Reboot the halted cluster member interactively. When the boot procedure requests you to enter the name of the kernel from which to boot, specify both the kernel name and a value of 0 (zero) for the `cluster_adjust_expected_votes` `clubase` attribute. A value of 0 (zero) causes the connection manager to set expected votes to the total number of member and quorum disk votes that are currently available in the cluster.

   Note

   Because the `cluster_adjust_expected_votes` transaction is performed only after the booting node joins the cluster, this method is effective only for those cases where an existing cluster is hung in quorum loss. If the cluster cannot form because expected votes is too high, the `cluster_adjust_expected_votes` transaction cannot run and the booting member will hang. In this case, you must use one of the methods in Section 4.10.2 to boot the member and form a cluster.

   For example:

```
>>> boot -fl "ia"
 (boot dkb200.2.0.7.0 -flags ia)
 block 0 of dkb200.2.0.7.0 is a valid boot block
 reading 18 blocks from dkb200.2.0.7.0
 bootstrap code read in
 base = 200000, image_start = 0, image_bytes = 2400
 initializing HWRPB at 2000
```

```
initializing page table at fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
```

.
.
.

```
Enter kernel_name [option_1 ... option_n]
Press Return to boot default kernel
'vmunix':vmunix clubase:cluster_adjust_expected_votes=0 Return
```

When you resume the boot, the member can join the cluster and the connection manager communicates the new operative expected votes value to the other cluster members so that they regain quorum.

Caution

The `cluster_adjust_expected_votes` setting modifies only the operative expected votes setting in the currently active cluster, and is used only as long as the entire cluster remains up. It does not modify the values that are stored in the `/etc/sysconfigtab` file. Unless you now explicitly reconfigure node votes, expected votes, and the quorum disk configuration in the cluster, a subsequent cluster reboot may result in booting members not being able to attain quorum and form a cluster. For this reason, you must proceed to fix node votes and expected votes values on this member and other cluster members, as necessary.

3. Consulting Table 4–2, use the appropriate `clu_quorum` commands to temporarily fix the configuration of votes in the cluster until the broken hardware is repaired or replaced. In general, as soon as the cluster is up and stable, you may use the `clu_quorum` command to fix the original problem. For example, you might:

   - Lower the node votes on the members who are having hardware problems:

     ```
     # clu_quorum -f -m member-ID lower_node_votes_value
     ```

     This command may return an error if it cannot access the member's boot disk (for example, if the boot disk is on a member private bus). If the command fails for this reason, use the `clu_quorum -f -e` command to adjust expected votes appropriately.

   - Lower the expected votes on all members to compensate for the members who can no longer vote due to loss of hardware and whose votes you cannot remove:

     ```
     # clu_quorum -f -e lower_expected_votes_value
     ```

If a `clu_quorum -f` command cannot access a down member's `/etc/sysconfigtab` file, it fails with an appropriate message. This usually happens when the down member's boot disk is on a bus private to that member. To resolve quorum problems involving such a member, boot that member interactively, setting `cluster_expected_votes` to a value that allows the member to join the cluster. When it joins, use the `clu_quorum` command to correct vote settings as suggested in this section.

See Table 4–2 for examples on how to restore quorum to a four-member cluster with a quorum disk and a five-member cluster without one. In the table, the abbreviation NC indicates that the member or quorum disk is not configured in the cluster.

Table 4–2: Examples of Resolving Quorum Loss in a Cluster with Failed Members or Quorum Disk

M1	M2	M3	M4	M5	Qdisk	Procedure
Up, 1 vote	Up, 1 vote	Failed, 1 vote	Failed, 1 vote	NC	Failed	1. Boot M1 or M2 interactively with `clubase:adjust_expected_votes=0`. 2. Remove the node votes from M3 and M4 by using `clu_quorum -f -m` commands. 3. Delete the quorum disk by using the `clu_quorum -f -d remove` command. 4. Repair or replace the broken hardware. The most immediate need of the two-member cluster, if it is to survive a failure, is a voting quorum disk. Use the `clu_quorum -f -d add` command to add a new quorum disk. To have the quorum disk recognized throughout the cluster, you must run the `hwmgr -scan scsi` command on every cluster member. If you cannot add a quorum disk, use the `clu_quorum -f -m` command to remove a vote from M1 or M2. If the broken members will be unavailable for a considerable time, use the `clu_delete_member` command to remove them from the cluster.
Up, 1 vote	Up, 1 vote	Failed, 1 vote	Failed, 1 vote	Failed, 1 vote	NC	1. Boot M1 or M2 interactively with `clubase:adjust_expected_votes=0`. 2. Remove the node votes from M3, M4, and M5 by using `clu_quorum -f -m` commands. 3. Repair or replace the broken hardware. The most immediate need of the two-member cluster, if it is to survive a failure, is a voting quorum disk. Use the `clu_quorum -f -d add` command to add a new quorum disk. To have the quorum disk recognized throughout the cluster, you must run the `hwmgr -scan scsi` command on every cluster member. If the broken members will be unavailable for a considerable time, use the `clu_delete_member` command to remove them from the cluster.

4.10.2 Forming a Cluster When Members Do Not Have Enough Votes to Boot and Form a Cluster

Consider a cluster that cannot form. When you attempt to boot all members, each hangs, waiting for a cluster to form. All together they lack sufficient votes to achieve quorum. A small cluster that experiences multiple hardware failures can also devolve to a configuration in which the last surviving voting member has lost quorum.

The following procedure effectively allows you to form the cluster by booting a single cluster member with sufficient votes to form the cluster. You then can adjust node votes and boot the remaining members into the cluster.

1. Halt each cluster member.
2. Consult Table 4–3 to determine the kernel attributes that must be adjusted at boot time to resolve your cluster's specific quorum loss situation.
3. Boot one voting cluster member interactively. When the boot procedure requests you to enter the name of the kernel from which to boot, specify both the kernel name and the recommended kernel attribute setting. For instance, for a two-member cluster (with two node votes and a quorum disk) that has experienced both a member failure and a quorum disk failure, enter **clubase:cluster_expected_votes=1 clubase:cluster_qdisk_votes=0**.

   For example:

```
>>> boot -fl "ia"
 (boot dkb200.2.0.7.0 -flags ia)
 block 0 of dkb200.2.0.7.0 is a valid boot block
 reading 18 blocks from dkb200.2.0.7.0
 bootstrap code read in
 base = 200000, image_start = 0, image_bytes = 2400
 initializing HWRPB at 2000
 initializing page table at fff0000
 initializing machine state
 setting affinity to the primary CPU
 jumping to bootstrap code

 .
 .
 .

 Enter kernel_name [option_1 ... option_n]
 Press Return to boot default kernel
 'vmunix':vmunix
 clubase:cluster_expected_votes=1 clubase:cluster_qdisk_votes=0 Return
```

   When you resume the boot, the member can form a cluster.
4. While referring to Table 4–3, use the appropriate `clu_quorum` commands to fix the configuration of votes in the cluster temporarily until the broken hardware is repaired or replaced. If an unavailable quorum disk contributed to the problem, make sure that the disk is

available and has a vote. Replace the quorum disk if necessary (see Section 4.5.1). Otherwise, other members may not be able to boot.

5. Reboot remaining members.

See Table 4–3 for examples on how to repair a quorum deficient cluster by booting a cluster member with sufficient votes to form the cluster. In the table, the abbreviation NC indicates that the member or quorum disk is not configured in the cluster.

Table 4–3: Examples of Repairing a Quorum Deficient Cluster by Booting a Member with Sufficient Votes to Form the Cluster

M1	M2	M3	Qdisk	Procedure
Up, 1 vote	Up, 0 votes	NC	Failed, 1 vote	1. Boot M2 interactively with `clubase:cluster_node_votes=1`. 2. Use the `clu_quorum -f -d remove` command to delete the quorum disk. 3. Replace the broken quorum disk using the `clu_quorum -f -d add` command. This will result in a two-member cluster with two node votes and a quorum disk vote (a configuration that can tolerate the failure of the disk or any one member). If you cannot replace the quorum disk, use the `clu_quorum -f -m` command to remove one member's vote. This will result in a configuration that can survive the failure of the nonvoting member.

Table 4–3: Examples of Repairing a Quorum Deficient Cluster by Booting a Member with Sufficient Votes to Form the Cluster (cont.)

M1	M2	M3	Qdisk	Procedure
Up, 1 vote	Failed, 1 vote	NC	Failed, 1 vote	1. Boot M1 interactively with `clubase:cluster_expected_votes=1` and `clubase:cluster_qdisk_votes=0`. 2. Use the `clu_quorum -f -d remove` command to delete the quorum disk. 3. Use the `clu_quorum -f -m 2 0` to remove M2's vote. 4. Repair or replace the broken hardware. If you cannot immediately obtain a second voting member with a voting quorum disk, adding a second member with no votes may be a reasonable interim solution. This will result in a configuration that can survive the failure of the nonvoting member.
Up, 1 vote	Failed, 1 vote	Failed, 1 vote	NC	1. Boot M1 interactively with `clubase:cluster_expected_votes=1`. 2. Use the appropriate `clu_quorum -f -m` commands to remove M2 and M3's votes. 3. Repair or replace the broken hardware. If you cannot immediately obtain a second voting member with a voting quorum disk, adding a second member with no votes may be a reasonable interim solution. This will result in a configuration that can survive the failure of the nonvoting member.

5

Managing Cluster Members

This chapter discusses the following topics:

- Managing configuration variables (Section 5.1)
- Managing kernel attributes (Section 5.2)
- Managing remote access to the cluster (Section 5.3)
- Shutting down the cluster (Section 5.4)
- Shutting down and starting one cluster member (Section 5.5)
- Shutting down a cluster member to single-user mode (Section 5.6)
- Deleting a cluster member (Section 5.7)
- Removing a member and restoring it as a standalone system (Section 5.8)
- Changing the cluster name or IP address (Section 5.9)
- Changing the member name, IP address, or cluster interconnect address (Section 5.10)
- Managing software licenses (Section 5.11)
- Installing and deleting layered applications (Section 5.12)
- Managing accounting services (Section 5.13)

For information on the following topics that are related to managing cluster members, see the TruCluster Server *Cluster Installation* manual:

- Adding new members to a cluster
- Reinstalling cluster members
- Software licensing issues

For information about configuring and managing your Tru64 UNIX and TruCluster Server systems for availability and serviceability, see *Managing Online Addition and Removal*. This manual provides users with guidelines for configuring and managing any system for higher availability, with an emphasis on those capable of Online Addition and Replacement (OLAR) management of system components.

Note

As described in *Managing Online Addition and Removal*, the `/etc/olar.config` file is used to define system-specific policies and the `/etc/olar.config.common` file is used to define cluster-wide policies. Any settings in a system's `/etc/olar.config` file override clusterwide policies in the `/etc/olar.config.common` file for that system only.

5.1 Managing Configuration Variables

The hierarchy of the `/etc/rc.config*` files lets you define configuration variables consistently over all systems within a local area network (LAN) and within a cluster. Table 5–1 presents the uses of the configuration files.

Table 5–1: /etc/rc.config* Files

File	Scope
`/etc/rc.config`	Member-specific variables. `/etc/rc.config` is a context-dependent symbolic link (CDSL). Each cluster member has a unique version of the file. Configuration variables in `/etc/rc.config` override those in `/etc/rc.config.common` and `/etc/rc.config.site`.
`/etc/rc.config.common`	Clusterwide variables. These configuration variables apply to all members. Configuration variables in `/etc/rc.config.common` override those in `/etc/rc.config.site`, but are overridden by those in `/etc/rc.config`.

`/etc/rc.config.site file`	Sitewide variables, which are the same for all machines on the LAN.
	Values in this file are overridden by any corresponding values in `/etc/rc.config.common` or `/etc/rc.config`.
	By default, there is no `/etc/rc.config.site`. If you want to set sitewide variables, you have to create the file and copy it to `/etc/rc.config.site` on every participating system.
	You must then edit `/etc/rc.config` on each participating system and add the following code just before the line that executes `/etc/rc.config.common`:
	`# Read in the cluster sitewide attributes`
	`# before overriding them with the`
	`# clusterwide and member-specific values.`
	`#`
	`./etc/rc.config.site`
	For more information, see `rcmgr`(8).

The rcmgr command accesses these variables in a standard search order (first /etc/rc.config, then /etc/rc.config.common, and finally etc/rc.config.site) until it finds or sets the specified configuration variable.

Use the -h option to get or set the run-time configuration variables for a specific member. The command then acts on /etc/rc.config, the member-specific CDSL configuration file.

To make the command act clusterwide, use the -c option. The command then acts on /etc/rc.config.common, which is the clusterwide configuration file.

If you specify neither -h nor -c, then the member-specific values in /etc/rc.config are used.

For information about member-specific configuration variables, see Appendix B.

5.2 Managing Kernel Attributes

Each member of a cluster runs its own kernel and therefore has its own /etc/sysconfigtab file. This file contains static member-specific attribute settings. Although a clusterwide /etc/sysconfigtab.cluster file exists, its purpose is different from that of /etc/rc.config.common, and it is reserved to utilities that are shipped in the TruCluster Server product.

This section presents a partial list of those kernel attributes that are provided by each TruCluster Server subsystem.

Use the following command to display the current settings of these attributes for a given subsystem:

```
#  sysconfig -q subsystem-name attribute-list
```

To get a list and the status of all the subsystems, use the following command:

```
# sysconfig -s
```

In addition to the cluster-related kernel attributes presented here, two kernel attributes are set during cluster installation. Table 5–2 lists these kernel attributes. You can increase the values for these attributes, but do not decrease them.

Table 5–2: Kernel Attributes Not to Decrease

Attribute	Value (Do Not Decrease)
vm_page_free_min	30
vm_page_free_reserved	20

Table 5–3 lists the subsystem names that are associated with each TruCluster Server component.

Table 5–3: Configurable TruCluster Server Subsystems

Subsystem Name	Component	For More Information
cfs	Cluster File System (CFS)	sys_attrs_cfs(5)
clua	Cluster alias	sys_attrs_clua(5)
clubase	Cluster base	sys_attrs_clubase(5)
cms	Cluster mount service	sys_attrs_cms(5)
cnx	Connection manager	sys_attrs_cnx(5)
dlm	Distributed lock manager	sys_attrs_dlm(5)
drd	Device request dispatcher	sys_attrs_drd(5)
hwcc	Hardware components cluster	sys_attrs_hwcc(5)
icsnet	Internode communications service's network service	sys_attrs_icsnet(5)
ics_hl	Internode communications service (ICS) high level	sys_attrs_ics_hl(5)
mcs	Memory Channel application programming interface (API)	sys_attrs_mcs(5)
rm	Memory Channel	sys_attrs_rm(5)
token	CFS token subsystem	sys_attrs_token(5)

To tune the performance of a kernel subsystem, use one of the following methods to set one or more attributes in the /etc/sysconfigtab file:

- Add or edit a *subsystem name* stanza entry in the /etc/sysconfigtab file to change an attribute's value and have the new value take effect at the next system boot.
- Use the following command to change the value of an attribute that can be reset so that its new value takes effect immediately at run time:

```
# sysconfig -r subsystem-name  attribute-list
```

To allow the change to be preserved over the next system boot, you must also edit the /etc/sysconfigtab file. For example, to change the value of the drd-print-info attribute to 1, enter the following command:

```
# sysconfig -r drd drd-print-info=1
drd-print-info: reconfigured
```

You can also use the configuration manager framework, as described in the Tru64 UNIX *System Administration* manual, to change attributes and otherwise administer a cluster kernel subsystem on another host. To do this, set up the host names in the /etc/cfgmgr.auth file on the remote client system and then specify the -h option to the /sbin/sysconfig command, as in the following example:

```
# sysconfig -h fcbra13 -r drd drd-do-local-io=0

drd-do-local-io: reconfigured
```

5.3 Managing Remote Access Within and From the Cluster

An rlogin, rsh, or rcp command from the cluster uses the default cluster alias as the source address. Therefore, if a noncluster host must allow remote host access from any account in the cluster, the .rhosts file on the noncluster member must include the cluster alias name in one of the forms by which it is listed in the /etc/hosts file or one resolvable through Network Information Service (NIS) or Domain Name System (DNS).

The same requirement holds for rlogin, rsh, or rcp to work between cluster members. At cluster creation, the clu_create utility prompts for all required host names and puts them in the correct locations in the proper format. The clu_add_member does the same when a new member is added to the cluster. You do not need to edit /.rhosts to enable /bin/rsh commands from a cluster member to the cluster alias or between individual members. Do not change the generated name entries in /etc/hosts and /.rhosts.

If the /etc/hosts and /.rhosts files are configured incorrectly, many applications will not function properly. For example, the Advanced File System (AdvFS) rmvol and addvol commands use rsh when the member where the commands are executed is not the server of the domain. These commands fail if /etc/hosts or /.rhosts is configured incorrectly.

The following error indicates that the /etc/hosts or /.rhosts file has been configured incorrectly:

```
rsh cluster-alias date
Permission denied.
```

5.4 Shutting Down the Cluster

To halt all members of a cluster, use the `-c` option to the `shutdown` command. For example, to shut down the cluster in 5 minutes, enter the following command:

```
# shutdown -c +5 Cluster going down in 5 minutes
```

For information on shutting down a single cluster member, see Section 5.5.

During the shutdown grace period, which is the time between when the cluster `shutdown` command is entered and when actual shutdown occurs, the `clu_add_member` command is disabled and new members cannot be added to the cluster.

To cancel a cluster shutdown during the grace period, kill the processes that are associated with the `shutdown` command as follows:

1. Get the process identifiers (PIDs) that are associated with the `shutdown` command. For example:

   ```
   # ps ax | grep -v grep | grep shutdown
    14680 ttyp5    I <    0:00.01 /usr/sbin/shutdown
   +20 going down
   ```

   Depending on how far along `shutdown` is in the grace period, `ps` might show either `/usr/sbin/shutdown` or `/usr/sbin/clu_shutdown`.

2. Terminate all shutdown processes by specifying their PIDs in a `kill` command from any member. For example:

   ```
   # kill 14680
   ```

If you kill the shutdown processes during the grace period, the shutdown is canceled.

The `shutdown -c` command fails if a `clu_quorum`, `clu_add_member`, `clu_delete_member`, or `clu_upgrade` is in progress.

There is no clusterwide reboot. The `shutdown -r` command, the `reboot` command, and the `halt` command act only on the member on which they are executed. The `halt`, `reboot`, and `init` commands have been modified to leave file systems in a cluster mounted, so the cluster continues functioning when one of its members is halted or rebooted, as long as it retains quorum.

For more information, see `shutdown`(8).

5.5 Shutting Down and Starting One Cluster Member

When booting a member, you must boot from the boot disk that was created by the `clu_add_member` command. You cannot boot from a copy of the boot disk.

Shutting down a single cluster member is more complex than shutting down a standalone server. If you halt a cluster member whose vote is required for quorum (referred to as a critical voting member), the cluster will lose quorum and hang. As a result, you will be unable to enter commands from any cluster member until you reboot the halted member. Therefore, before you shut down a cluster member, you must first determine whether that member's vote is required for quorum.

5.5.1 Identifying a Critical Voting Member

A cluster that contains a critical voting member is either operating in a degraded mode (for example, one or more voting members or a quorum disk is down) or was not configured for availability to begin with (for example, it is a two-member configuration with each member assigned a vote). Removing a critical voting member from a cluster causes the cluster to hang and compromise availability. Before halting or deleting a cluster member, ensure that it is not supplying a critical vote.

To determine whether a member is a critical voting member, follow these steps:

1. If possible, make sure that all voting cluster members are up.
2. Enter the `clu_quorum` command and note the running values of current votes, quorum votes, and the node votes of the member in question.
3. Subtract the member's node votes from the current votes. If the result is less than the quorum votes, the member is a critical voting member and you cannot shut it down without causing the cluster to lose quorum and hang.

5.5.2 Preparing to Halt or Delete a Critical Voting Member

Before halting or deleting a critical voting member, ensure that its votes are no longer critical to the cluster retaining quorum. The best way to do this involves restoring node votes or a quorum disk vote to the cluster without increasing expected votes. Some ways to accomplish this are:

- Booting a voting member that is currently down.
- Removing the vote of a down member (using the `clu_quorum -f -m` command) and configuring a quorum disk with a vote (using the `clu_quorum -f -d add` command). This has the effect of not increasing expected votes or changing the value of quorum votes, but brings an additional current vote to the cluster.

If the cluster has an even number of votes, adding a new voting member or configuring a quorum disk can also make a critical voting member

noncritical. In these cases, expected votes is incremented, but quorum votes remains the same.

5.5.3 Halting a Noncritical Member

A noncritical member, one with no vote or whose vote is not required to maintain quorum, can be shut down, halted, or rebooted like a standalone system.

Execute the `shutdown` command on the member to be shut down. To halt a member, enter the following command:

```
# shutdown -h time
```

To reboot a member, enter the following command:

```
# shutdown -r time
```

For information on identifying critical voting members, see Section 5.5.1.

5.5.4 Shutting Down a Hosting Member

The cluster application availability (CAA) profile for an application allows you to specify an ordered list of members, separated by white space, that can host the application resource. The hosting members list is used in conjunction with the application resource's failover policy (favored or restricted), as discussed in `caa(4)`.

If the cluster member that you are shutting down is the only hosting member for one or more applications with a restricted placement policy, you need to specify another hosting member or the application cannot run while the member is down. You can add an additional hosting member, or replace the existing hosting member with another.

To do this, perform these steps:

1. Verify the current hosting members and placement policy.

   ```
   # caa_profile -print resource-name
   ```

2. If the cluster member that you are shutting down is the only hosting member, you can add an additional hosting member to the hosting members list, or replace the existing member.

   ```
   # caa_profile -update resource-name -h hosting-member another-hosting-member
   # caa_profile -update resource-name -h hosting-member
   ```

3. Update the CAA registry entry with the latest resource profile.

   ```
   # caa_register -u resource-name
   ```

4. Relocate the application to the other member.

```
# caa_relocate resource-name -c member-name
```

5.6 Shutting Down a Cluster Member to Single-User Mode

If you need to shut down a cluster member to single-user mode, you must first halt the member and then boot it to single user-mode. Shutting down the member in this manner assures that the member provides the minimal set of services to the cluster and that the running cluster has a minimal reliance on the member running in single-user mode. In particular, halting the member satisfies services that require the cluster member to have a status of DOWN before completing a service failover. If you do not first halt the cluster member, the services do not fail over as expected.

To take a cluster member to single-user mode, use the `shutdown -h` command to halt the member, and then boot the member to single-user mode. When the system reaches single-user mode, run the `init s`, `bcheckrc`, and `lmf reset` commands. For example:

Note

Before halting a cluster member, make sure that the cluster can maintain quorum without the member's vote.

```
# /sbin/shutdown -h now

>>> boot -fl s

# /sbin/init s
# /sbin/bcheckrc
# /usr/sbin/lmf reset
```

A cluster member that is shut down to single-user mode (that is, not shut down to a halt and then booted to single-user mode as recommended) continues to have a status of UP. Shutting down a cluster member to single-user mode in this manner does not affect the voting status of the member: a member contributing a vote before being shut down to single-user mode continues contributing the vote in single-user mode.

5.7 Deleting a Cluster Member

The `clu_delete_member` command permanently removes a member from the cluster.

Caution

If you are reinstalling TruCluster Server, see the TruCluster Server *Cluster Installation* manual. Do not delete a member from

an existing cluster and then create a new single-member cluster from the member that you just deleted. If the new cluster has the same name as the old cluster, the newly installed system might join the old cluster. This can cause data corruption.

The `clu_delete_member` command has the following syntax:

/usr/sbin/clu_delete_member [-f] [-m *memberid*]

If you do not supply a member ID, the command prompts you for the member ID of the member to delete.

The `clu_delete_member` command does the following:

- Mounts the member's boot partition and deletes all files in the boot partition. The system can no longer boot from this disk.

  Caution

  The `clu_delete_member` command will delete a member, even when the member's boot disk is inaccessible. This lets you delete a member whose boot disk has failed.

  If the command cannot access the disk, you must make sure that no cluster member can inadvertently boot from that disk. Remove the disk from the cluster, reformat it, or use the `disklabel` command to make it a nonbootable disk.

- If the member has votes, adjusts the value of cluster expected votes throughout the cluster.
- Deletes all member-specific directories and files in the clusterwide file systems.

  Note

  The `clu_delete_member` command deletes member-specific files from the `/cluster`, `/usr/cluster`, and `/var/cluster` directories. However, an application or an administrator can create member-specific files in other directories, such as `/usr/local`. You must manually remove those files after running `clu_delete_member`. Otherwise, if you add a new member and reuse the same member ID, the new member will have access to these (outdated and perhaps erroneous) files.

- Removes the deleted member's host name for its Memory Channel interface from the `/.rhosts` and `/etc/hosts.equiv` files.

- Writes a log file of the deletion to `/cluster/admin/clu_delete_member.log`. Appendix C contains a sample `clu_delete_member` log file.

To delete a member from the cluster, follow these steps:

1. Determine whether or not the member is a critical voting member of the cluster. If the member supplies a critical vote to the cluster, halting it will cause the cluster to lose quorum and suspend operations. Before halting the member, use the procedure in Section 5.5 to determine whether it is safe to do so.
2. Halt the member to be deleted.
3. If possible, make sure that all voting cluster members are up.
4. Use the `clu_delete_member` command from another member to remove the member from the cluster. For example, to delete a halted member whose member ID is 3, enter the following command:

   ```
   # clu_delete_member -m 3
   ```

5. When you run `clu_delete_member` and the boot disk for the member is inaccessible, the command displays a message to that effect.

   If the member being deleted is a voting member, after the member is deleted you must manually lower by one vote the expected votes for the cluster. Do this with the following command:

   ```
   # clu_quorum -e expected-votes
   ```

Note

This step applies only when the member boot disk cannot be accessed by `clu_delete_member` and the member that is being deleted is a voting member.

For an example of the `/cluster/admin/clu_delete_member.log` that results when a member is deleted, see Appendix C.

5.8 Removing a Cluster Member and Restoring It as a Standalone System

To restore a cluster member as a standalone system, follow these steps:

1. Halt and delete the member by following the procedures in Section 5.5 and Section 5.7.
2. Physically disconnect the halted member from the cluster, disconnecting the Memory Channel and storage.

3. On the halted member, select a disk that is local to the member and install Tru64 UNIX. See the Tru64 UNIX *Installation Guide* for information on installing system software.

For information about moving clusterized Logical Storage Manager (LSM) volumes to a noncluster system, see Section 10.5.

5.9 Changing the Cluster Name or IP Address

Changing the name of a cluster requires a shutdown and reboot of the entire cluster. Changing the IP address of a cluster requires that you shut down and reboot each member individually.

To change the cluster name, follow these steps carefully. Any mistake can prevent the cluster from booting.

1. Create a file with the new `cluster_name` attribute for the `clubase` subsystem stanza entry. For example, to change the cluster name to `deli`, add the following `clubase` subsystem stanza entry:

```
clubase:
 cluster_name=deli
```

Notes

Ensure that you include a line-feed at the end of each line in the file that you create. If you do not, when the `sysconfigtab` file is modified, you will have two attributes on the same line. This may prevent your system from booting.

If you create the file in the cluster root directory, you can use it on every system in the cluster without a need to copy the file.

2. On each cluster member, use the `sysconfigdb -m -f` *file* `clubase` command to merge the new `clubase` subsystem attributes from the file that you created with the `clubase` subsystem attributes in the `/etc/sysconfig` file.

   For example, assume that the file `cluster-name-change` contains the information shown in the example in step 1. To use the file `cluster-name-change` to change the cluster name from `poach` to `deli`, use the following command:

```
# sysconfigdb -m -f cluster-name-change clubase
Warning: duplicate attribute in clubase:
was cluster_name = poach, now cluster_name = deli
```

Caution

Do not use the `sysconfigdb -u` command with a file with only one or two attributes to be changed. The `-u` flag causes the subsystem entry in the input file to replace a subsystem entry (for instance `clubase`). If you specify only the `cluster_name` attribute for the `clubase` subsystem, the new `clubase` subsystem will contain only the `cluster_name` attribute and none of the other required attributes.

3. Change the cluster name in each of the following files:
   - `/etc/hosts`
   - `/etc/hosts.equiv`

   There is only one copy of these files in a cluster.
4. Add the new cluster name to the `/.rhosts` file (which is common to all cluster members).

   Leave the current cluster name in the file. The current name is needed for the `shutdown -c` command in the next step to function.

   Change any client `.rhosts` file as appropriate.
5. Shut down the entire cluster with the `shutdown -c` command and reboot each system in the cluster.
6. Remove the previous cluster name from the `/.rhosts` file.
7. To verify that the cluster name has changed, run the `/usr/sbin/clu_get_info` command:

```
# /usr/sbin/clu_get_info
Cluster information for cluster deli
.
.
.
```

5.9.1 Changing the Cluster IP Address

To change the cluster IP address, follow these steps:

1. Edit the `/etc/hosts` file, and change the IP address for the cluster.
2. One at a time (to keep quorum), shut down and reboot each cluster member system.

To verify that the cluster IP address has changed, run the `/usr/sbin/ping` command from a system that is not in the cluster to ensure that the cluster provides the echo response when you use the cluster address:

```
# /usr/sbin/ping -c 3 16.160.160.160
PING 16.160.160.160 (16.160.160.160): 56 data bytes
64 bytes from 16.160.160.160: icmp_seq=0 ttl=64 time=26 ms
64 bytes from 16.160.160.160: icmp_seq=1 ttl=64 time=0 ms
64 bytes from 16.160.160.160: icmp_seq=2 ttl=64 time=0 ms

----16.160.160.160 PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/9/26 ms
```

5.10 Changing the Member Name, IP Address, or Cluster Interconnect Address

To change the member name, member IP address, or member cluster interconnect address, you must remove the member from the cluster and then add it back in with the desired member name or address. Do this as follows:

1. Halt the member. See Section 5.5 for information on shutting down a single cluster member.
2. On an active member of the cluster, delete the member that you just shut down. Do this by running the `clu_delete_member` command:

   ```
   # clu_delete_member -m memberid
   ```

   To learn the member ID of the member to be deleted, use the `clu_get_info` command.

   See Section 5.7 for details on using `clu_delete_member`.
3. Use the `clu_add_member` command to add the system back into the cluster, specifying the desired member name, member IP address, and cluster interconnect address.

   For details on adding a member to the cluster, see the TruCluster Server *Cluster Installation* manual.

5.11 Managing Software Licenses

When you add a new member to a cluster, you must register application licenses on that member for those applications that may run on that member.

For information about adding new cluster members and Tru64 UNIX licenses, see the chapter on adding members in the TruCluster *Cluster Installation* manual.

5.12 Installing and Deleting Layered Applications

The procedure to install or delete an application is usually the same for both a cluster and a standalone system. Applications can be installed once in a cluster. However, some applications require additional steps.

- Installing an application

  If an application has member-specific configuration requirements, you might need to log on to each member where the application will run and configure the application. For more information, see the configuration documentation for the application.

- Deleting an application

  Before using `setld` to delete an application, make sure that the application is not running. This may require you to stop the application on several members. For example, for multi-instance application, stopping the application may involve killing daemons running on multiple cluster members.

  For applications that are managed by CAA, use the following command to find out the status of the highly available applications:

  ```
  # caa_stat
  ```

  If the application to be deleted is running (`STATE=ONLINE`), stop it and remove it from the CAA registry with the following commands:

  ```
  # caa_stop application_name
  # caa_unregister application_name
  ```

  After the application is stopped, delete it with the `setld` command. Follow any application-specific directions in the documentation for the application. If the application is installed on a member that is not currently available, the application is automatically removed from the unavailable member when that member rejoins the cluster.

5.13 Managing Accounting Services

The system accounting services are not cluster-aware. The services rely on files and databases that are member-specific. Because of this, to use accounting services in a cluster, you must set up and administer the services on a member-by-member basis.

The `/usr/sbin/acct` directory is a CDSL. The accounting services files in `/usr/sbin/acct` are specific to each cluster member.

To set up accounting services on a cluster, use the following modifications to the directions in the chapter on administering system accounting services in the Tru64 UNIX *System Administration* manual:

1. You must enable accounting services on each cluster member where you want accounting to run. To enable accounting on all cluster members, enter the following command on each member:

   ```
   # rcmgr -c set ACCOUNTING YES
   ```

   If you want to enable accounting on only certain members, use the `-h` option to the `rcmgr` command. For example, to enable accounting on members 2, 3, and 6, enter the following commands:

   ```
   # rcmgr -h 2 set ACCOUNTING YES
   # rcmgr -h 3 set ACCOUNTING YES
   # rcmgr -h 6 set ACCOUNTING YES
   ```

2. You must start accounting on each member. Log in to each member where you want to start accounting, and enter the following command:

   ```
   # /usr/sbin/acct/startup
   ```

   To stop accounting on a member, you must log in to that member and run the command `/usr/sbin/acct/shutacct`.

The directory `/usr/spool/cron` is a CDSL; the files in this directory are member-specific, and you can use them to tailor accounting on a per-member basis. To do so, log in to each member where accounting is to run. Use the `crontab` command to modify the `crontab` files as desired. For more information, see the chapter on administering the system accounting services in the Tru64 UNIX *System Administration* manual.

The file `/usr/sbin/acct/holidays` is a CDSL. Because of this, you set accounting service holidays on a per-member basis.

For more information on accounting services, see `acct`(8).

6

Managing Networks in a Cluster

This chapter discusses the following topics:

- Providing failover for network interfaces (Section 6.1)
- Running IP routers (Section 6.2)
- Configuring the network (Section 6.3)

See the Tru64 UNIX *Network Administration: Connections* and *Network Administration: Services* manuals for information about managing networks on single systems.

6.1 Providing Failover for Network Interfaces

The Redundant Array of Independent Network Adapters (NetRAIN) interface provides protection against certain kinds of network connectivity failures. NetRAIN integrates multiple network interfaces on the same LAN segment into a single virtual interface called a NetRAIN set. One network interface in the set is active while the others remain idle. If the active interface fails, one of the idle set members comes on line with the same IP address.

The Network Interface Failure Finder (NIFF) is an additional feature that monitors the status of its network interfaces and reports indications of network failures. You can use NIFF to generate events when network devices, including a composite NetRAIN device, fail. You can monitor these events and take appropriate actions when a failure occurs.

To configure NIFF in a cluster, you must enable the NIFF daemon, `niffd`, on each cluster member. For example, to enable startup of `niffd` and have NIFF monitor the `tu` network interface, log in to each cluster member (or use the `-h` to `rcmgr`) and enter the following commands:

```
# rcmgr set NIFFD "YES"
# rcmgr set NIFFC_FLAGS "-a tu0"
```

For information about providing failover for applications that depend on network resources, see the TruCluster Server *Cluster Highly Available Applications* manual.

For more information about NIFF and NetRAIN, see the Tru64 UNIX *Network Administration: Services* and *Network Administration: Connections*

manuals, niffd(8), niff(7), and nr(7). For more information on the rcmgr command, see rcmgr(8).

6.2 Running IP Routers

Cluster members can be IP routers, and you can configure more than one member as an IP router. However, the only supported way to do this requires that you use the TruCluster Server gated configuration. You can customize the gated configuration to run a specialized routing environment. For example, you can run a routing protocol such as Open Shortest Path First (OSPF).

To run a customized gated configuration on a cluster member, log on to that member and follow these steps:

1. If gated is running, stop it with the following command:

   ```
   # /sbin/init.d/gateway stop
   ```

2. Enter the following command:

   ```
   #  cluamgr -r start,nogated
   ```

3. Modify gated.conf (or the name that you are using for the configuration file). Use the version of /etc/gated.conf.member*n* that was created by the cluamgr -r nogated,start command as the basis for edits to a customized gated configuration file. You will need to correctly merge the cluster alias information from the /etc/gated.conf.member*n* file into your customized configuration file.

4. Start gated with the following command:

   ```
   # /sbin/init.d/gateway start
   ```

The cluamgr -r start,nogated command does the following tasks:

- Creates a member-specific version of gated.conf with a different name.
- Does *not* start the gated daemon.
- Generates a console warning message that indicates alias route failover will not work if gated is not running, and references the newly created gated file.
- Issues an Event Manager (EVM) warning message.

The option to customize the gated configuration is provided solely to allow a knowledgeable system manager to modify the standard TruCluster Server version of gated.conf so that it adds support needed for that member's routing operations. After the modification, gated is run to allow the member to operate as a customized router.

For more information, see cluamgr(8)

Notes

The cluamgr option nogated is not a means to allow the use of routed.

Only gated is supported.

We strongly recommend that cluster members use routing only for cluster alias support, and that the job of general-purpose IP routing within the network be handled by general-purpose routers that are tuned for that function.

6.3 Configuring the Network

Typically, you configure the network when you install the Tru64 UNIX Version 5.1A software. If you later need to alter the network configuration, the following information might be useful. Use the sysman net_wizard command or the equivalent command, netconfig to configure the following:

- Network interface cards
- Static routes (/etc/routes)
- Routing services (gated, IP router)
- Hosts file (/etc/hosts)
- Hosts equivalency file (/etc/hosts.equiv)
- Remote who services (rwhod)
- Dynamic Host Configuration Protocol (DHCP) server (joind)
- Networks file (/etc/networks)

You can run the nfsconfig command without any focus. In this case, the configurations that are performed are considered to be clusterwide, and all configurations are placed in the /etc/rc.config.common file.

If you specify a focus member, either on the command line or through the Sysman Menu, the configurations are performed for the specified member. All configurations are placed in the member-specific /etc/rc.config file.

The following configuration tasks require a focus member:

- Network interfaces
- Gateway routing daemon (`gated`)
- Static routes (`/etc/routes`)
- Remote `who` daemon (`rwhod`)
- Internet Protocol (IP) router

Starting and stopping network services also requires member focus.

The preceding tasks require focus on a specific member because they are member-specific functions. A restart or stop of network services clusterwide would be disruptive; therefore, these tasks are performed on one member at a time.

The following configuration tasks must be run clusterwide:

- DHCP server daemon
- Hosts (`/etc/hosts`)
- Hosts equivalencies (`/etc/hosts.equiv`)
- Networks (`/etc/networks`)

For information about configuring DHCP, see Section 7.1.

7

Managing Network Services

The TruCluster Server *Cluster Installation* manual describes how to initially configure services. We strongly suggest you configure services before the cluster is created. If you wait until after cluster creation to set up services, the process can be more complicated.

This chapter describes the procedures to set up network services after cluster creation. The chapter discusses the following topics:

- Configuring DHCP (Section 7.1)
- Configuring NIS (Section 7.2)
- Configuring printing (Section 7.3)
- Configuring DNS/BIND (Section 7.4)
- Managing time synchronization (Section 7.5)
- Managing NFS (Section 7.6)
- Managing `inetd` configuration (Section 7.7)
- Managing mail (Section 7.8)
- Configuring a cluster for RIS (Section 7.9)
- Displaying X Windows Applications Remotely (Section 7.10)

7.1 Configuring DHCP

A cluster can be a highly available Dynamic Host Configuration Protocol (DHCP) server. It cannot be a DHCP client. A cluster must use static addressing. On a cluster, DHCP runs as a single-instance application with cluster application availability (CAA) providing failover. At any one time, only one member of the cluster is the DHCP server. If failover occurs, the new DHCP server uses the same common database that was used by the previous server.

The DHCP server attempts to match its host name and IP address with the configuration in the DHCP database. If you configure the database with the host name and IP address of a cluster member, problems can result. If the member goes down, DHCP automatically fails over to another member, but

the host name and IP address of this new DHCP server does not match the entry in the database. To avoid this and other problems, follow these steps:

1. Familiarize yourself with the DHCP server configuration process that is described in the chapter on DHCP in the Tru64 UNIX *Network Administration: Connections* manual.
2. On the cluster member that you want to act as the initial DHCP server, run `/usr/bin/X11/xjoin` and configure DHCP.
3. Select `Server/Security`.
4. Under `Server/Security Parameters`, set the `Canonical Name` entry to the default cluster alias.
5. From the pulldown menu that currently shows `Server/Security Parameters`, select `IP Ranges`.
6. Set the `DHCP Server` entry to the IP address of the default cluster alias.

   There can be multiple entries for the DHCP Server IP address in the DHCP database. You might find it more convenient to use the `jdbdump` command to generate a text file representation of the DHCP database. Then use a text editor to change all the occurrences of the original DHCP server IP address to the cluster alias IP address. Finally, use `jdbmod` to repopulate the DHCP database from the file you edited. For example:

   ```
   # jdbdump > dhcp_db.txt
   # vi dhcp_db.txt
   ```

   Edit `dhcp_db.txt` and change the owner IP address to the IP address of the default cluster alias.

   Update the database with your changes by entering the following command:

   ```
   # jdbmod -e dhcp_db.txt
   ```
7. When you finish with `xjoin`, make DHCP a highly available application. DHCP already has an action script and a resource profile, and it is already registered with the CAA daemon. To start DHCP with CAA, enter the following command:

   ```
   # caa_start dhcp
   ```

For information about highly available applications and CAA, see the TruCluster Server *Cluster Highly Available Applications* manual.

7.2 Configuring NIS

To provide high availability, the Network Information Service (NIS) daemons `ypxfrd` and `rpc.yppasswdd` run on every cluster member.

As described in Section 3.1, the ports that are used by services that are accessed through a cluster alias are defined as either `in_single` or `in_multi`. (These definitions have nothing to do with whether the service can or cannot run on more than one cluster member at the same time.)

`ypxfrd` runs as an `in_multi` service, which means that the cluster alias subsystem routes connection requests and packets for that service to all eligible members of the alias.

`rpc.yppasswdd` runs as an `in_single` service, which means that only one alias member receives connection requests or packets that are addressed to the service. If that member becomes unavailable, the cluster alias subsystem selects another member of the alias as the recipient for all requests and packets addressed to the service.

NIS parameters are stored in `/etc/rc.config.common`. The database files are in the `/var/yp/src` directory. Both `rc.config.common` and the databases are shared by all cluster members. The cluster is a slave, a master, or a client. The functions of slave, master, and client cannot be mixed among individual cluster members.

If you configured NIS at the time of cluster creation, then as far as NIS is concerned, you need do nothing when adding or removing cluster members.

To configure NIS after the cluster is running, follow these steps:

1. Run the `nissetup` command and configure NIS according to the instructions in the chapter on NIS in the Tru64 UNIX *Network Administration: Services* manual.

   You have to supply the host names that NIS binds to. Include the cluster alias in your list of host names.

2. On each cluster member, enter the following commands:

   ```
   # /sbin/init.d/nis stop
   # /sbin/init.d/nis start
   ```

7.2.1 Configuring an NIS Master in a Cluster with Enhanced Security

You can configure an NIS master to provide extended user profiles and to use the protected password database. For information about NIS and enhanced security features, see the Tru64 UNIX *Security* manual. For details on configuring NIS with enhanced security, see the appendix on enhanced security in a cluster in the same manual.

7.3 Configuring Printing

With a few exceptions, printer setup on a cluster is the same as printer setup on a standalone Tru64 UNIX system. See the Tru64 UNIX *System*

Administration manual for general information about managing the printer system.

In a cluster, a member can submit a print job to any printer anywhere in the cluster. A printer daemon, `lpd`, runs on each cluster member. This parent daemon serves both local `lpr` requests and incoming remote job requests.

The parent printer daemon that runs on each node uses `/var/spool/lpd`, which is a context-dependent symbolic link (CDSL) to `/cluster/members/{memb}/spool/lpd`. Do not use `/var/spool/lpd` for any other purpose.

Each printer that is local to the cluster has its own spooling directory, which is located by convention under `/usr/spool`. The spooling directory must not be a CDSL.

A new printer characteristic, `:on`, has been introduced to support printing in clusters. To configure a printer, run either `printconfig` or `lprsetup` on any cluster member.

If a printer is a local device that is connected to a member via a COM port (`/dev/tty01`) or a parallel port (`/dev/lp0`), then set `:on` to the name of the member where the printer is connected. For example, `:on=memberA`

The printer is connected to the member `memberA`.

When configuring a network printer that is connected via TCP/IP, you have two choices for values for the `:on` characteristic:

- `:on=localhost`

  Specify `localhost` when you want every member of the cluster to serve the printer. When a print job is submitted, the first member that responds handles all printing until the queue is empty. For local jobs, the first member to respond is the member on which the first job is submitted. For incoming remote jobs, the jobs are served based on the cluster alias.

- `:on=`*`member1,member2,...,memberN`*

  List specific cluster members when you want all printing to be handled by a single cluster member. The first member in the `:on` list handles all printing. If that member becomes unavailable, then the next member in the list takes over, and so on.

Using Advanced Printing Software

For information on installing and using Advanced Printing Software in a cluster, see the configuration notes chapter in the Tru64 UNIX *Advanced Printing Software Release Notes*.

7.4 Configuring DNS/BIND

Configuring a cluster as a Berkeley Internet Name Domain (BIND) server is similar to configuring an individual Tru64 UNIX system as a BIND server. In a cluster, the `named` daemon runs on a single cluster member, and that system is the actual BIND server. The cluster alias handles queries, so that it appears the entire cluster is the server. Failover is provided by CAA. If the serving member becomes unavailable, CAA starts the `named` daemon on another member.

If the cluster is configured as a BIND client, then the entire cluster is configured as a client. No cluster member can be a BIND client if the cluster is configured as a BIND server.

Whether you configure BIND at the time of cluster creation or after the cluster is running, the process is the same.

To configure a cluster as either a BIND server or client, use the command `bindconfig` or `sysman dns`.

If you are configuring the cluster as a client, then it does not matter on which member you run the command. If you are configuring a BIND server, then you determine which member becomes the server by running the command on that member. Note that the `sysman -focus` option does not work for configuring BIND. You must log in to the system you want to act as the BIND server and then run `sysman dns` or `bindconfig`.

The `/etc/resolv.conf` and `/etc/svc.conf` files are clusterwide files.

For details on configuring BIND, see the chapter on the Domain Name System (DNS) in the Tru64 UNIX *Network Administration: Services* manual.

7.5 Managing Time Synchronization

All cluster members need time synchronization. The Network Time Protocol (NTP) meets this requirement. Because of this, the `clu_create` command configures NTP on the initial cluster member at the time of cluster creation, and NTP is automatically configured on each member as it is added to the cluster. All members are configured as NTP peers.

If your site chooses not to use NTP, make sure that whatever time service you use meets the granularity specifications that are defined in RFC 1035 Network Time Protocol (Version 3) Specification, Implementation and Analysis.

Because the system times of cluster members should not vary by more than a few seconds, we do not recommend using the `timed` daemon to synchronize the time.

7.5.1 Configuring NTP

The *Cluster Installation* manual recommends that you configure NTP on the Tru64 UNIX system before you install the cluster software that makes the system the initial cluster member. If you did not do this, `clu_create` and `clu_add_member` configured NTP automatically on each cluster member. In this configuration, the NTP server for each member is `localhost`. Members are set up as NTP peers of each other, and use the IP address of their cluster interconnect interfaces.

The `localhost` entry is used only when the member is the only node running. The peer entries act to keep all cluster members synchronized so that the time offset is in microseconds across the cluster. Do not change these initial server and peer entries even if you later change the NTP configuration and add external servers.

To change the NTP configuration after the cluster is running, you must run either `ntpconfig` or `sysman ntp` on each cluster member. These commands always act on a single cluster member. You can either log in to each member or you can use the `-focus` option to `sysman` in order to designate the member on which you want to configure NTP. Starting and stopping the NTP daemon, `xntpd`, is potentially disruptive to the operation of the cluster, and should be performed on only one member at a time.

When you use `sysman` to learn the status of the NTP daemon, you can get the status for either the entire cluster or a single member.

7.5.2 All Members Should Use the Same External NTP Servers

You can add an external NTP server to just one member of the cluster. However, this creates a single point of failure. To avoid this, add the same set of external servers to all cluster members.

We strongly recommend that the list of external NTP servers be the same on all members. If you configure differing lists of external servers from member to member, you must ensure that the servers are all at the same stratum level and that the time differential between them is very small.

7.5.2.1 Time Drift

If you notice a time drift among cluster members, you need to resynchronize members with each other. To do this you must log on to each member of the cluster and enter the `ntp -s -f` command and specify the cluster interconnect name of a member other than the one where you are logged on. By default a cluster interconnect name is the short form of the hostname with `-mc0` appended. For example, if `provolone` is a cluster member, and you are logged on to a member other than `provolone`, enter the following command:

```
# ntp -s -f provolone-mc0
```

You then log on to the other cluster members and repeat this command, in each case using a cluster interconnect name other than the one of the system where you are logged on.

7.6 Managing NFS

A cluster can provide highly available Network File System (NFS) service. When a cluster acts as an NFS server, client systems that are external to the cluster see it as a single system with the cluster alias as its name. When a cluster acts as an NFS client, an NFS file system that is external to the cluster that is mounted by one cluster member is accessible to all cluster members. File accesses are funneled through the mounting member to the external NFS server. The external NFS server sees the cluster as a set of independent nodes and is not aware that the cluster members are sharing the file system.

7.6.1 Configuring NFS

To configure NFS, use the `nfsconfig` or `sysman nfs` command.

Note

Do not use the `nfssetup` command in a cluster. It is not cluster-aware and will incorrectly configure NFS.

One or more cluster members can run NFS daemons and the `mount` daemons, as well as client versions of `lockd` and `statd`.

With `nfsconfig` or `sysman nfs`, you can:

- Start, restart, or stop NFS daemons clusterwide or on an individual member.
- Configure or unconfigure server daemons clusterwide or on an individual member.
- Configure or unconfigure client daemons clusterwide or on an individual member.
- View the configuration status of NFS clusterwide or on an individual member.
- View the status of NFS daemons clusterwide or on an individual member.

To configure NFS on a specific member, use the `-focus` option to `sysman`.

When you configure NFS without any focus, the configuration applies to the entire cluster and is saved in `/etc/rc.config.common`. If a focus is

specified, then the configuration applies to only the specified cluster member and is saved in the CDSL file /etc/rc.config for that member.

Local NFS configurations override the clusterwide configuration. For example, if you configure member mutt as not being an NFS server, then mutt is not affected when you configure the entire cluster as a server; mutt continues not to be a server.

For a more interesting example, suppose you have a three-member cluster with members alpha, beta, and gamma. Suppose you configure 8 TCP server threads clusterwide. If you then set focus on member alpha and configure 10 TCP server threads, the ps command will show 10 TCP server threads on alpha, but only 8 on members beta and gamma. If you then set focus clusterwide and set the value from 8 TCP server threads to 12, alpha still has 10 TCP server threads, but beta and gamma now each have 12 TCP server threads.

If a member runs nfsd it must also run mountd, and vice versa. This is automatically taken care of when you configure NFS with nfsconfig or sysman nfs.

If locking is enabled on a cluster member, then the rpc.lockd and rpc.statd daemons are started on the member. If locking is configured clusterwide, then the lockd and statd run clusterwide (rpc.lockd -c and rpc.statd -c), and the daemons are highly available and are managed by CAA. The server uses the default cluster alias or an alias that is specified in /etc/exports.aliases as its address.

When a cluster acts as an NFS server, client systems that are external to the cluster see it as a single system with the cluster alias as its name. Client systems that mount directories with CDSLs in them see only those paths that are on the cluster member that is running the clusterwide statd and lockd pair.

You can start and stop services either on a specific member or on the entire cluster. Typically, you should not need to manage the clusterwide lockd and statd pair. However, if you do need to stop the daemons, enter the following command:

```
# caa_stop cluster_lockd
```

To start the daemons, enter the following command:

```
# caa_start cluster_lockd
```

To relocate the server lockd and statd pair to a different member, enter the caa_relocate command as follows:

```
# caa_relocate cluster_lockd
```

For more information about starting and stopping highly available applications, see Chapter 8.

7.6.2 Considerations for Using NFS in a Cluster

This section describes the differences between using NFS in a cluster and in a standalone system.

7.6.2.1 Clients Must Use a Cluster Alias

When a cluster acts as an NFS server, clients must use the default cluster alias, or an alias that is listed in `/etc/exports.aliases`, to specify the host when mounting file systems served by the cluster. If a node that is external to the cluster attempts to mount a file system from the cluster and the node does not use the default cluster alias, or an alias that is listed in `/etc/exports.aliases`, a "connection refused" error is returned to the external node.

Other commands that run through `mountd`, like `umount` and `export`, receive a "Program unavailable" error when the commands are sent from external clients and do not use the default cluster alias or an alias listed in `/etc/exports.aliases`.

Before configuring additional aliases for use as NFS servers, read the sections in the *Cluster Technical Overview* that discuss how NFS and the cluster alias subsystem interact for NFS, TCP, and User Datagram Protocol (UDP) traffic. Also read the `exports.aliases`(4) reference page and the comments at the beginning of the `/etc/exports.aliases` file.

7.6.2.2 Using CDSLs to Mount NFS File Systems

When a cluster acts as an NFS client, an NFS file system that is mounted by one cluster member is accessible to all cluster members: the Cluster File System (CFS) funnels file accesses through the mounting member to the external NFS server. That is, the cluster member performing the mount becomes the CFS server for the NFS file system and is the node that communicates with the external NFS server. By maintaining cache coherency across cluster members, CFS guarantees that all members at all times have the same view of the NFS file system.

However, in the event that the mounting member becomes unavailable, there is no failover. Access to the NFS file system is lost until another cluster member mounts the NFS file system.

There are several ways to address this possible loss of file system availability. You might find that using AutoFS to provide automatic failover of NFS file systems is the most robust solution because it allows for both availability

and cache coherency across cluster members. Using AutoFS in a cluster environment is described in Section 7.6.2.5.

As an alternative to using AutoFS, you can use the `mkcdsl -a` command to convert a mount point into a CDSL. This will copy an existing directory to a member-specific area on all members. You then use the CDSL as the mount point for the NFS file system. In this scenario, there is still only one NFS server for the file system, but each cluster member is an NFS client. Cluster members are not dependent on one cluster member functioning as the CFS server of the NFS file system. If one cluster member becomes unavailable, access to the NFS file system by the other cluster members is not affected. However, cache coherency across cluster members is not provided by CFS: the cluster members rely on NFS to maintain the cache coherency using the usual NFS methods, which do not provide single-system semantics.

If relying on NFS to provide the file system integrity is acceptable in your environment, perform the following steps to use a CDSL as the mount point:

1. Create the mount point if one does not already exist.

   ```
   # mkdir /mountpoint
   ```

2. Use the `mkcdsl -a` command to convert the directory into a CDSL. This will copy an existing directory to a member-specific area on all members.

   ```
   #mkcdsl -a /mountpoint
   ```

3. Mount the NFS file system on each cluster member, using the same NFS server.

   ```
   # mount server:/filesystem  /mountpoint
   ```

We recommend adding the mount information to the `/etc/fstab` file so that the mount is performed automatically on each cluster member.

7.6.2.3 Loopback Mounts Not Supported

NFS loopback mounts do not work in a cluster. Attempts to NFS-mount a file system that is served by the cluster onto a directory on the cluster fail and return the message, `Operation not supported`.

7.6.2.4 Do Not Mount Non-NFS File Systems on NFS-Mounted Paths

CFS does not permit non-NFS file systems to be mounted on NFS-mounted paths. This limitation prevents problems with availability of the physical file system in the event that the serving cluster member goes down.

7.6.2.5 Using AutoFS in a Cluster

If you want automatic mounting of NFS file systems, use AutoFS. AutoFS provides automatic failover of the automounting service by means of CAA.

One member acts as the CFS server for automounted file systems, and runs the one active copy of the AutoFS daemon, autofsd. If this member fails, CAA starts autofsd on another member.

For instructions on configuring AutoFS, see the section on automatically mounting a remote file system in the Tru64 UNIX *Network Administration: Services* manual. After you have configured AutoFS, you must start the daemon as follows:

```
# caa_start autofs
```

In TruCluster Server Version 5.1A, the value of the SCRIPT_TIMEOUT attribute has been increased to 3600 to reduce the possibility of the autofs timing out. You can increase this value, but we recommend that you do not decrease it.

In previous versions of TruCluster Server, depending on the number of file systems being imported, the speeds of datalinks, and the distribution of imported file systems among servers, you might see a CAA message like the following:

```
# CAAD[564686]: RTD #0: Action Script \
/var/cluster/caa/script/autofs.scr(start) timed out! (timeout=180)
```

In this situation, you need to increase the value of the SCRIPT_TIMEOUT attribute in the CAA profile for autofs to a value greater than 180. You can do this by editing /var/cluster/caa/profile/autofs.cap, or you can use the caa_profile -update autofs command to update the profile.

For example, to increase SCRIPT_TIMEOUT to 3600 seconds, enter the following command:

```
# caa_profile -update autofs -o st=3600
```

For more information about CAA profiles and using the caa_profile command, see caa_profile(8).

If you use AutoFS, keep in mind the following:

- On a cluster that imports a large number of file systems from a single NFS server, or imports from a server over an especially slow datalink, you might need to increase the value of the mount_timeout kernel attribute in the autofs subsystem. The default value for mount_timeout is 30 seconds. You can use the sysconfig command to change the attribute while the member is running. For example, to change the timeout value to 50 seconds, use the following command:

  ```
  # sysconfig -r autofs mount_timeout=50
  ```

- When the autofsd daemon starts or when autofsmount runs to process maps for automounted file systems, AutoFS makes sure that

all cluster members are running the same version of the TruCluster Server software.

7.6.2.6 Forcibly Unmounting File Systems

If AutoFS on a cluster member is stopped or becomes unavailable (for example, if the CAA autofs resource is stopped), intercept points and file systems auto-mounted by AutoFS continue to be available. However, in the case where AutoFS is stopped on a cluster member on which there are busy file systems, and then started on another member, there is a likely problem in which AutoFS intercept points continue to recognize the original cluster member as the server. This occurs because the AutoFS intercept points are busy when the file systems that are mounted under them are busy, and these intercept points still claim the original cluster member as the server. These intercept points do not allow new auto-mounts.

7.6.2.6.1 Determining Whether a Forced Unmount is Required

There are two situations under which you might encounter this problem:

- You detect an obvious problem accessing an auto-mounted file system.
- You move the CAA autofs resource.

In the case where you detect an obvious problem accessing an auto-mounted file system, ensure that the auto-mounted file system is being served as expected. To do this, perform the following steps:

1. Use the caa_stat autofs command to see where CAA indicates the autofs resource is running.
2. Use the ps command to verify that the autofsd daemon is running on the member on which CAA expects it to run:

   ```
   # ps agx | grep autofsd
   ```

   If it is not running, run it and see whether this fixes the problem.
3. Determine the auto-mount map entry that is associated with the inaccessible file system. One way to do this is to search the /etc/auto.*x* files for the entry.
4. Use the cfsmgr -e command to determine whether the mount point exists and is being served by the expected member.

   If the server is not what CAA expects, the problem exists.

In the case where you move the CAA resource to another member, use the mount -e command to identify AutoFS intercept points and the cfsmgr -e command to show the servers for all mount points. Verify that all AutoFS intercept points and auto-mounted file systems have been unmounted on the member on which AutoFS was stopped.

When you use the `mount -e` command, search the output for `autofs` references similar to the following:

```
# mount -e | grep autofs
/etc/auto.direct on /mnt/mytmp type autofs (rw, nogrpid, direct)
```

When you use the `cfsmgr -e` command, search the output for map file entries similar to the following. The `Server Status` field does not indicate whether the file system is actually being served; look in the `Server Name` field for the name of the member on which `AutoFS` was stopped.

```
# cfsmgr -e
Domain or filesystem name = /etc/auto.direct
Mounted On = /mnt/mytmp
Server Name = provolone
Server Status : OK
```

7.6.2.6.2 Correcting the Problem

If you can wait until the busy file systems in question become inactive, do so. Then, run the `autofsmount -U` command on the former `AutoFS` server node to unmount them. Although this approach takes more time, it is a less intrusive solution.

If waiting until the busy file systems in question become inactive is not possible, use the `cfsmgr -K` *`directory`* command on the former `AutoFS` server node to forcibly unmount all `AutoFS` intercept points and auto-mounted file systems served by that node, even if they are busy.

Note

The `cfsmgr -K` command makes a best effort to unmount all `AutoFS` intercept points and auto-mounted file systems served by the node. However, the `cfsmgr -K` command may not succeed in all cases. For example, the `cfsmgr -K` command does not work if an NFS operation is stalled due to a down NFS server or an inability to communicate with the NFS server.

The `cfsmgr -K` command results in applications receiving I/O errors for open files in affected file systems. An application with its current working directory in an affected file system will no longer be able to navigate the file system namespace using relative names.

Perform the following steps to relocate the `autofs` CAA resource and forcibly unmount the `AutoFS` intercept points and auto-mounted file systems:

1. Bring the system to a quiescent state if possible to minimize disruption to users and applications.
2. Stop the `autofs` CAA resource by entering the following command:

```
# caa_stop autofs
```

CAA considers the `autofs` resource to be stopped even if some auto-mounted file systems are still busy.

3. Enter the following command to verify that all `AutoFS` intercept points and auto-mounted file systems have been unmounted. Search the output for autofs references.

```
# mount -e
```

4. In the event that they have not all been unmounted, enter the following command to forcibly unmount the `AutoFS` intercepts and auto-mounted file systems:

```
# cfsmgr -K directory
```

5. Specify the directory on which an `AutoFS` intercept point or auto-mounted file system is mounted. You need enter only one mounted-on directory to remove all of the intercepts and auto-mounted file systems served by the same node.
6. Enter the following command to start the `autofs` resource:

```
# caa_start autofs -c cluster_member_to_be_server
```

7.7 Managing inetd Configuration

Configuration data for the Internet server daemon (`inetd`) is kept in the following two files:

- `/etc/inetd.conf`

  Shared clusterwide by all members. Use `/etc/inetd.conf` for services that should run identically on every member.

- `/etc/inetd.conf.local`

  The `/etc/inetd.conf.local` file holds configuration data specific to each cluster member. Use it to configure per-member network services.

To disable a clusterwide service on a local member, edit `/etc/inetd.conf.local` for that member, and enter `disable` in the `ServerPath` field for the service to be disabled. For example, if `finger` is enabled clusterwide in `inetd.conf` and you want to disable it on a member, add a line like the following to that member's `inetd.conf.local` file:

```
finger  stream  tcp     nowait  root   disable       fingerd
```

When `/etc/inetd.conf.local` is not present on a member, the configuration in `/etc/inetd.conf` is used. When `inetd.conf.local` is present, its entries take precedence over those in `inetd.conf`.

7.8 Managing Mail

TruCluster Server supports the following mail protocols:

- Simple Mail Transfer Protocol (SMTP)
- DECnet Phase IV
- DECnet Phase V
- Message Transport System (MTS)
- Unix-to-Unix Copy Program (UUCP)
- X.25

In a cluster, all members must have the same mail configuration. If DECnet, SMTP, or any other protocol is configured on one cluster member, it must be configured on all members, and it must have the same configuration on each member. You can configure the cluster as a mail server, client, or as a standalone configuration, but the configuration must be clusterwide. For example, you cannot configure one member as a client and another member as a server.

Of the supported protocols, only SMTP is cluster-aware, so only SMTP can make use of the cluster alias. SMTP handles e-mail sent to the cluster alias, and labels outgoing mail with the cluster alias as the return address.

When configured, an instance of `sendmail` runs on each cluster member. Every member can handle messages waiting for processing because the mail queue file is shared. Every member can handle mail delivered locally because each user's maildrop is shared among all members.

The other mail protocols, DECnet Phase IV, DECnet Phase V, Message Transport System (MTS), UUCP, and X.25, can run in a cluster environment, but they act as though each cluster member is a standalone system. Incoming e-mail using one of these protocols must be addressed to an individual cluster member, not to the cluster alias. Outgoing e-mail using one of these protocols has as its return address the cluster member where the message originated.

Configuring DECnet Phase IV, DECnet Phase V, Message Transport System (MTS), UUCP, or X.25 in a cluster is like configuring it in a standalone system. It must be configured on each cluster member, and any hardware that is required by the protocol must be installed on each cluster member.

The following sections describe managing mail in more detail.

7.8.1 Configuring Mail

Configure mail with either the `mailsetup` or `mailconfig` command. Whichever command you choose, you have to use it for future mail

configuration on the cluster, because each command understands only its own configuration format.

7.8.1.1 Mail Files

The following mail files are all common files shared clusterwide:

- `/usr/adm/sendmail/sendmail.cf`
- `/usr/adm/sendmail/aliases`
- `/var/spool/mqueue`
- `/usr/spool/mail/*`

The following mail files are member-specific:

- `/usr/adm/sendmail/sendmail.st`
- `/var/adm/sendmail/protocols.map`

Files in `/var/adm/sendmail` that have *hostname* as part of the file name use the default cluster alias in place of *hostname*. For example, if the cluster alias is `accounting`, `/var/adm/sendmail` contains files named `accounting.m4` and `Makefile.cf.accounting`.

Because the mail statistics file, `/usr/adm/sendmail/sendmail.st`, is member-specific, mail statistics are unique to each cluster member. The `mailstat` command returns statistics only for the member on which the command executed.

When mail protocols other than SMTP are configured, the member-specific `/var/adm/sendmail/protocols.map` file stores member-specific information about the protocols in use. In addition to a list of protocols, `protocols.map` lists DECnet Phase IV and DECnet Phase V aliases, when those protocols are configured.

7.8.1.2 The Cw Macro (System Nicknames List)

Whether you configure mail with `mailsetup` or `mailconfig`, the configuration process automatically adds the names of all cluster members and the cluster alias to the `Cw` macro (nicknames list) in the `sendmail.cf` file. The nicknames list must contain these names. If, during mail configuration, you accidentally delete the cluster alias or a member name from the nicknames list, the configuration program will add it back in.

During configuration you are given the opportunity to specify additional nicknames for the cluster. However, if you do a quick setup in `mailsetup`, you are not prompted to update the nickname list. The cluster members and the cluster alias are still automatically added to the `Cw` macro.

7.8.1.3 Configuring Mail at Cluster Creation

We recommend that you configure mail on your Tru64 UNIX system before you run the `clu_create` command. If you run only SMTP, then you do not need to perform further mail configuration when you add new members to the cluster. The `clu_add_member` command takes care of correctly configuring mail on new members as they are added.

If you configure DECnet Phase IV, DECnet Phase V, MTS, UUCP, or X.25, then each time that you add a new cluster member, you must run `mailsetup` or `mailconfig` and configure the protocol on the new member.

7.8.1.4 Configuring Mail After the Cluster Is Running

All members must have the same mail configuration. If you want to run only SMTP, then you need configure mail only once, and you can run `mailsetup` or `mailconfig` from any cluster member.

If you want to run a protocol other than SMTP, you must manually run `mailsetup` or `mailconfig` on every member and configure the protocols. Each member must also have any hardware required by the protocol. The protocols must be configured for every cluster member, and the configuration of each protocol must be the same on every member.

The `mailsetup` and `mailconfig` commands cannot be focused on individual cluster members. In the case of SMTP, the commands configure mail for the entire cluster. For other mail protocols, the commands configure the protocol only for the cluster member on which the command runs.

If you try to run `mailsetup` with the `-focus` option, you get the following error message:

```
Mail can only be configured for the entire cluster.
```

Whenever you add a new member to the cluster, and you are running any mail protocol other than SMTP, you must run `mailconfig` or `mailsetup` and configure the protocol on the new member. If you run only SMTP, then no mail configuration is required when a member is added.

Deleting members from the cluster requires no reconfiguration of mail, regardless of the protocols that you are running.

7.8.2 Distributing Mail Load Among Cluster Members

Mail handled by SMTP can be load balanced by means of the cluster alias selection priority (`selp`) and selection weight (`selw`), which load balance network connections among cluster members as follows:

- The cluster member with the highest selection priority receives all connection requests.

The selection priority can be any integer from 1 through 100. The default value is 1.

- Selection weight determines the distribution of connections among members with the same selection priority. A member receives, on average, the number of connection requests equal to the selection weight, after which requests are routed to the next member with the same selection priority.

  The selection weight can be any integer from 0 through 100. A member with a selection weight of 0 receives no incoming connection requests, but can send out requests.

By default, all cluster members have the same selection priority (`selp=1`) and selection weight (`selw=1`), as determined by the `/etc/clu_alias.config` file on each member. (The `clu_create` command uses a default selection weight of 3, but if you create an alias the default selection weight is 1.) When all members share the same selection priority and the same selection weight, then connection requests are distributed equally among the members. In the case of the default system configuration, each member in turn handles one incoming connection.

If you want all incoming mail (and all other connections) to be handled by a subset of cluster members, set the selection priority for those cluster members to a common value that is higher than the selection priority of the remaining members.

You can also create a mail alias that includes only those cluster members that you want to handle mail, or create a mail alias with all members and use the selection priority to determine the order in which members of the alias receive new connection requests.

Set the selection weight or selection priority for a member by running the `cluamgr` command on that member. If your cluster members have the default values for `selp` and `selw`, and you want all incoming mail (and *all* other connections) to be handled by a single cluster member, log in to that member and assign it a `selp` value greater than the default. For example, enter the following command:

```
# cluamgr -a alias=DEFAULTALIAS,selp=50
```

Suppose you have an eight-member cluster and you want two of the members, `alpha` and `beta`, to handle all incoming connections, with the load split 40/60 between `alpha` and `beta`, respectively. Log in to `alpha` and enter the following command:

```
# cluamgr -a alias=DEFAULTALIAS,selp=50,selw=2
```

Then log in to `beta` and enter the following command:

```
# cluamgr -a alias=DEFAULTALIAS,selp=50,selw=3
```

Assuming that the other members have the default `selp` of 1, `beta` and `alpha` will handle all connection requests. `beta` will take three connections, then `alpha` will take two, then `beta` will take the next three, and so on.

Note

Setting `selp` and `selw` in this manner affects all connections through the cluster alias, not just the mail traffic.

For more information on balancing connection requests, see Section 3.9 and `cluamgr`(8).

7.9 Configuring a Cluster for RIS

To create a Remote Installation Services (RIS) server in a cluster, perform the following procedure in addition to the procedure that is described in the Tru64 UNIX *Sharing Software on a Local Area Network* manual:

- Modify `/etc/bootptab` so that the NFS mount point is set to the default cluster alias.
- Set the `tftp` server address to the default cluster alias:

  `sa=`*`default_cluster_alias`*

For information about `/etc/bootptab`, see `bootptab`(4).

Note

Depending on your network configuration, you may need to supply a unique, arbitrary hardware address when registering the alias with the RIS server.

To use a cluster as an RIS client, you must do the following:

1. Register the cluster member from which you will be using the `setld` command with the RIS server. Do this by registering the member name and the hardware address of that member.
2. Register the default cluster alias.

   If you are registering for an operating system kit, you will be prompted to enter a hardware address. The cluster alias does not have a physical interface associated with its host name. Instead, use any physical address that does not already appear in either `/etc/bootptab` or `/usr/var/adm/ris/clients/risdb`.

If your cluster uses the cluster alias virtual MAC (vMAC) feature, register that virtual hardware address with the RIS server as the default cluster alias's hardware address. If your cluster does not use the vMAC feature, you can still generate a virtual address by using the algorithm that is described in the virtual MAC (vMAC) section, Section 3.11.

A virtual MAC address consists of a prefix (the default is AA:01) followed by the IP address of the alias in hexadecimal format. For example, the default vMAC address for the default cluster alias `deli` whose IP address is `16.140.112.209` is `AA:01;10:8C:70:D1`. The address is derived in the following manner:

```
Default vMAC prefix:       AA:01
Cluster Alias IP Address:  16.140.112.209
IP address in hex. format: 10.8C.70:D1
vMAC for this alias:       AA:01:10:8C:70:D1
```

Therefore, when registering this default cluster alias as a RIS client, the host name is `deli` and the hardware address is `AA:01:10:8C:70:D1`.

If you do not register both the default cluster alias and the member, the `setld` command will return a message such as one of the following:

```
# setld -l ris-server:setld: Error contacting server
ris-server: Permission denied.
setld: cannot initialize ris-server:
```

```
# setld -l ris-server:setld: ris-server: not in server database
setld: cannot load control information
```

7.10 Displaying X Window Applications Remotely

You can configure the cluster so that a user on a system outside the cluster can run X applications on the cluster and display them on the user's system using the cluster alias.

The following example shows the use of `out_alias` as a way to apply single-system semantics to X applications that are displayed from cluster members.

In `/etc/clua_services`, the `out_alias` attribute is set for the X server port (6000). A user on a system outside the cluster wants to run an X application on a cluster member and display back to the user's system. Because the `out_alias` attribute is set on port 6000 in the cluster, the user must specify the name of the default cluster alias when running the `xhost` command to allow X clients access to the user's local system. For example, for a cluster named `deli`, the user runs the following command on the local system:

```
# xhost +deli
```

This use of `out_alias` allows any X application from any cluster member to be displayed on that user's system. A cluster administrator who wants users to allow access on a per-member basis can either comment out the `Xserver` line in `/etc/clua_services`, or remove the `out_alias` attribute from that line (and then run `cluamgr -f` on each cluster member to make the change take effect).

For more information on cluster aliases, see Chapter 3.

8

Managing Highly Available Applications

This chapter describes the management tasks that are associated with highly available applications and the cluster application availability (CAA) subsystem. The following sections discuss these and other topics:

- Learning the status of a resource (Section 8.1)
- Relocating applications (Section 8.2)
- Starting and stopping application resources (Section 8.3)
- Registering and unregistering application resources (Section 8.4)
- Managing network, tape, and media changer resources (Section 8.5)
- Using SysMan to manage CAA (Section 8.6)
- Understanding CAA considerations for startup and shutdown (Section 8.7)
- Managing `caad`, the CAA daemon (Section 8.8)
- Using EVM to view CAA events (Section 8.9)
- Troubleshooting with events (Section 8.10)
- Troubleshooting with command-line messages (Section 8.11)

For detailed information on setting up applications with CAA, see the TruCluster Server *Cluster Highly Available Applications* manual. For a general discussion of CAA, see the TruCluster Server *Cluster Technical Overview*.

After an application has been made highly available and is running under the management of the CAA subsystem, it requires little intervention from you. However, the following situations can arise where you might want to actively manage a highly available application:

- The planned shutdown or reboot of a cluster member.

  You might want to learn which highly available applications are running on the member to be shut down by using `caa_stat`. Optionally, you might want to manually relocate one or more of those applications by using `caa_relocate`.

- Load balancing.

As the loads on various cluster members change, you might want to manually relocate applications to members with lighter loads by using `caa_stat` and `caa_relocate`.

- A new application resource profile has been created.

  If the resource has not already been registered and started, you need to do this with `caa_register` and `caa_start`.

- The resource profile for an application has been updated.

  For the updates to become effective, you must update the resource using `caa_register -u`.

- An existing application resource is being retired.

  You will want to stop and unregister the resource by using `caa_stop` and `caa_unregister`.

When you work with application resources, the actual names of the applications that are associated with a resource are not necessarily the same as the resource name. The name of an application resource is the same as the root name of its resource profile. For example, the resource profile for the `cluster_lockd` resource is `/var/cluster/caa/profile/cluster_lockd.cap`. The applications that are associated with the `cluster_lockd` resource are `rpc.lockd` and `rpc.statd`.

Because a resource and its associated application can have different names, there are cases where it is futile to look for a resource name in a list of processes running on the cluster. When managing an application with CAA, you must use its resource name.

8.1 Learning the Status of a Resource

Registered resources have an associated state. A resource can be in one of the following three states:

- `ONLINE`

  In the case of an application resource, `ONLINE` means that the application that is associated with the resource is running normally.

  In the case of a network, tape, or media changer resource, `ONLINE` means that the device that is associated with the resource is available and functioning correctly.

- `OFFLINE`

  The resource is not running. It may be an application resource that was registered but never started with `caa_start`, or at some earlier time it was successfully stopped with `caa_stop`. If the resource is a network, tape, or media changer resource, the device that is associated

with the resource is not functioning correctly. This state also happens when a resource has failed more times than the FAILURE_THRESHOLD value in its profile.

- UNKNOWN

  CAA cannot determine whether the application is running or not due to an unsuccessful execution of the stop entry point of the resource action script. This state applies only to application resources. Look at the stop entry point of the resource action script for why it is failing (returning a value other than 0).

CAA will always try to match the state of an application resource to its target state. The target state is set to ONLINE when you use caa_start, and set to OFFLINE when you use caa_stop. If the target state is not equal to the state of the application resource, then CAA is either in the middle of starting or stopping the application, or the application has failed to run or start successfully. If the target state for a nonapplication resource is ever OFFLINE, the resource has failed too many times within the failure threshold. See Section 8.5 for more information.

From the information given in the Target and State fields, you can ascertain information about the resource. Descriptions of what combinations of the two fields can mean for the different types of resources are listed in Table 8–1 (application), Table 8–2 (network), and Table 8–3 (tape, media changer). If a resource has any combination of State and Target other than both ONLINE, all resources that require that resource have a state of OFFLINE.

Table 8–1: Target and State Combinations for Application Resources

Target	State	Description
ONLINE	ONLINE	Application has started successfully
ONLINE	OFFLINE	Start command has been issued but execution of action script start entry point not yet complete.
		Application stopped because of failure of required resource.
		Application has active placement on and is being relocated due to the starting or addition of a new cluster member.
		Application being relocated due to explicit relocation or failure of cluster member.
		No suitable member to start the application is available.
OFFLINE	ONLINE	Stop command has been issued, but execution of action script stop entry point not yet complete.
OFFLINE	OFFLINE	Application has not been started yet.

Table 8–1: Target and State Combinations for Application Resources (cont.)

Target	State	Description
		Application stopped because Failure Threshold has been reached.
		Application has been successfully stopped.
ONLINE	UNKNOWN	Action script stop entry point has returned failure.
OFFLINE	UNKNOWN	A command to stop the application was issued on an application in state UNKNOWN. Action script stop entry point still returns failure. To set application state to OFFLINE use `caa_stop -f`.

Table 8–2: Target and State Combinations for Network Resources

Target	State	Description
ONLINE	ONLINE	Network is functioning correctly.
ONLINE	OFFLINE	There is no direct connectivity to the network from the cluster member.
OFFLINE	ONLINE	Network card is considered failed and no longer monitored by CAA because Failure Threshold has been reached.
OFFLINE	OFFLINE	Network is not directly accessible to machine.
		Network card is considered failed and no longer monitored by CAA because Failure Threshold has been reached.

Table 8–3: Target and State Combinations for Tape and Media Changer Resources

Target	State	Description
ONLINE	ONLINE	Tape or media changer has a direct connection to the machine and is functioning correctly.
ONLINE	OFFLINE	Tape device or media changer associated with resource has sent out an Event Manager (EVM) event that it is no longer working correctly. Resource is considered failed.
OFFLINE	ONLINE	Tape device or media changer is considered failed and no longer monitored by CAA because Failure Threshold has been reached.
OFFLINE	OFFLINE	Tape device or media changer does not have a direct connection to the cluster member.

8.1.1 Learning the State of a Resource

To learn the state of a resource, enter the caa_stat command as follows:

```
# caa_stat resource_name
```

The command returns the following values:

- NAME

  The name of the resource, as specified in the NAME field of the resource profile.

- TYPE

  The type of resource: application, tape, changer, or network.

- TARGET

  For an application resource, describes the state, ONLINE or OFFLINE, in which CAA attempts to place the application. For all other resource types, the target should always be ONLINE unless the device that is associated with the resource has had its failure count exceed the failure threshold. If this occurs, the TARGET will be OFFLINE.

- STATE

  For an application resource, whether the resource is ONLINE or OFFLINE; and if the resource is on line, the name of the cluster member where it is currently running. The state for an application can also be UNKNOWN if an action script stop entry point returned failure. The application resource cannot be acted upon until it successfully stops. For all other resource types, the ONLINE or OFFLINE state is shown for each cluster member.

For example:

```
# caa_stat clock
NAME=clock
TYPE=application
TARGET=ONLINE
STATE=ONLINE on provolone
```

To use a script to learn whether an resource is on line, use the `-r` option for the `caa_stat` command as follows:

```
# caa_stat  resource_name -r ; echo $?
```

A value of 0 (zero) is returned if the resource is in the `ONLINE` state.

With the `-g` option for the `caa_stat` command, you can use a script to learn whether an application resource is registered as follows:

```
# caa_stat  resource_name -g ; echo $?
```

A value of 0 (zero) is returned if the resource is registered.

8.1.2 Learning Status of All Resources on One Cluster Member

The `caa_stat -c` *`cluster_member`* command returns the status of all resources on *`cluster_member`*. For example:

```
# caa_stat -c polishham
NAME=dhcp
TYPE=application
TARGET=ONLINE
STATE=ONLINE on polishham

NAME=named
TYPE=application
TARGET=ONLINE
STATE=ONLINE on polishham

NAME=xclock
TYPE=application
TARGET=ONLINE
STATE=ONLINE on polishham
```

This command is useful when you need to shut down a cluster member and want to learn which applications are candidates for failover or manual relocation.

8.1.3 Learning Status of All Resources on All Cluster Members

The `caa_stat` command returns the status of all resources on all cluster members. For example:

```
# caa_stat
NAME=dhcp
TYPE=application
TARGET=ONLINE
STATE=ONLINE on polishham

NAME=xclock
TYPE=application
TARGET=ONLINE
STATE=ONLINE on provolone

NAME=named
TYPE=application
TARGET=OFFLINE
STATE=OFFLINE

NAME=ln0
TYPE=network
TARGET=ONLINE on provolone
TARGET=ONLINE on polishham
TARGET=ONLINE on peppicelli
STATE=OFFLINE on provolone
STATE=ONLINE on polishham
STATE=ONLINE on peppicelli
```

When you use the `-t` option, the information is displayed in tabular form. For example:

```
# caa_stat  -t
Name           Type           Target    State     Host
------------------------------------------------------------
cluster_lockd  application    ONLINE    ONLINE    provolone
dhcp           application    OFFLINE   OFFLINE
named          application    OFFLINE   OFFLINE
ln0            network        ONLINE    ONLINE    provolone
ln0            network        ONLINE    OFFLINE   polishham
```

8.1.4 Getting Number of Failures and Restarts and Target States

The `caa_stat -v` command returns the status, including number of failures and restarts, of all resources on all cluster members. For example:

```
# caa_stat -v
NAME=cluster_lockd
TYPE=application
RESTART_COUNT=0
RESTART_ATTEMPTS=30
FAILURE_COUNT=0
FAILURE_THRESHOLD=0
TARGET=ONLINE
STATE=ONLINE on provolone
```

```
NAME=dhcp
TYPE=application
RESTART_COUNT=0
RESTART_ATTEMPTS=1
FAILURE_COUNT=1
FAILURE_THRESHOLD=3
TARGET=ONLINE
STATE=OFFLINE

NAME=ln0
TYPE=network
FAILURE_THRESHOLD=5
FAILURE_COUNT=1 on provolone
FAILURE_COUNT=0 on polishham
TARGET=ONLINE on provolone
TARGET=OFFLINE on polishham
STATE=ONLINE on provolone
STATE=OFFLINE on polishham
```

When you use the `-t` option, the information is displayed in tabular form. For example:

```
# caa_stat -v -t
Name           Type           R/RA   F/FT   Target    State     Host
----------------------------------------------------------------------
cluster_lockd  application    0/30   0/0    ONLINE    ONLINE    provolone
dhcp           application    0/1    0/0    OFFLINE   OFFLINE
named          application    0/1    0/0    OFFLINE   OFFLINE
ln0            network               0/5    ONLINE    ONLINE    provolone
ln0            network               1/5    ONLINE    OFFLINE   polishham
```

This information can be useful for finding resources that frequently fail or have been restarted many times.

8.2 Relocating Applications

There are times when you may want to relocate applications from one cluster to another. You may want to:

- Relocate all applications on a cluster member (Section 8.2.1)
- Relocate a single application to another cluster member (Section 8.2.2)
- Relocate dependent applications to another cluster member (Section 8.2.3)

You use the `caa_relocate` command to relocate applications. Whenever you relocate applications, the system returns messages tracking the relocation. For example:

```
Attempting to stop 'cluster_lockd' on member 'provolone'
Stop of 'cluster_lockd' on member 'provolone' succeeded.
```

```
Attempting to start 'cluster_lockd' on member 'pepicelli'
Start of 'cluster_lockd' on member 'pepicelli' succeeded.
```

The following sections discuss relocating applications in more detail.

8.2.1 Manual Relocation of All Applications on a Cluster Member

When you shut down a cluster member, CAA automatically relocates all applications under its control running on that member, according to the placement policy for each application. However, you might want to manually relocate the applications before shutdown of a cluster member for the following reasons:

- If you plan to shut down multiple members, use manual relocation to avoid situations where an application would automatically relocate to a member that you plan to shut down soon.
- If a cluster member is experiencing problems or even failing, manual relocation can minimize performance hits to application resources that are running on that member.
- If you want to do maintenance on a cluster member and want to minimize disruption to the work environment.

To relocate all applications from `member1` to `member2`, enter the following command:

```
# caa_relocate -s member1 -c member2
```

To relocate all applications on `member1` according to each application's placement policy, enter the following command:

```
# caa_relocate -s member1
```

Use the `caa_stat` command to verify that all application resources were successfully relocated.

8.2.2 Manual Relocation of a Single Application

You may want to relocate a single application to a specific cluster member for one of the following reasons:

- The cluster member that is currently running the application is overloaded and another member has a low load.
- You are about to shut down the cluster member, and you want the application to run on a specific member that may not be chosen by the placement policy.

To relocate a single application to `member2`, enter the following command:

```
# caa_relocate resource_name -c member2
```

Use the caa_stat command to verify that the application resource was successfully relocated.

8.2.3 Manual Relocation of Dependent Applications

You may want to relocate a group of applications that depend on each other. An application resource that has at least one other application resource listed in the REQUIRED_RESOURCE field of its profile depends on these applications. If you want to relocate an application with dependencies on other application resources, you must force the relocation by using the -f option with the caa_relocate command.

Forcing a relocation makes CAA relocate resources that the specified resource depends on, as well as all ONLINE application resources that depend on the resource specified. The dependencies may be indirect: one resource may depend on another through one or more intermediate resources.

To relocate a single application resource and its dependent application resources to member2, enter the following command:

```
# caa_relocate resource_name -f -c member2
```

Use the caa_stat command to verify that the application resources were successfully relocated.

8.3 Starting and Stopping Application Resources

The following section describes how to start and stop CAA application resources.

Note

Always use caa_start and caa_stop or the SysMan equivalents to start and stop applications that CAA manages. Never start or stop the applications manually after they are registered with CAA.

8.3.1 Starting Application Resources

To start an application resource, use the caa_start command followed by the name of the application resource to be started. To stop an application resource, use the caa_stop command followed by the name of the application resource to be stopped. A resource must be registered using caa_register before it can be started.

Immediately after the caa_start command is executed, the target is set to ONLINE. CAA always attempts to match the state to equal the target, so the

CAA subsystem starts the application. Any application required resources have their target states set to ONLINE as well and the CAA subsystem attempts to start them.

To start a resource named clock on the cluster member determined by the resource's placement policy, enter the following command:

```
# /usr/sbin/caa_start clock
```

An example of the output of the previous command follows:

```
Attempting to start 'clock' on member 'polishham'
Start of 'clock' on member 'polishham' succeeded.
```

The command will wait up to the SCRIPT_TIMEOUT value to receive notification of success or failure from the action script each time the action script is called.

To start clock on a specific cluster member, assuming that the placement policy allows it, enter the following command:

```
# /usr/sbin/caa_start clock -c member_name
```

If the specified member is not available, the resource will not start.

If required resources are not available and cannot be started on the specified member, caa_start fails. You will instead see a response that the application resource could not be started because of dependencies.

To force a specific application resource and all its required application resources to start or relocate to the same cluster member, enter the following command:

```
#/usr/sbin/caa_start -f clock
```

See caa_start(8) for more information.

8.3.2 Stopping Application Resources

To stop highly available applications, use the caa_stop command. As noted earlier, never use the kill command or other methods to stop a resource that is under the control of the CAA subsystem.

Immediately after the caa_stop command is executed, the target is set to OFFLINE. CAA always attempts to match the state to equal the target, so the CAA subsystem stops the application.

The command in the following example stops the clock resource:

```
#/usr/sbin/caa_stop clock
```

If other application resources have dependencies on the application resource that is specified, the previous command will not stop the application. You will instead see a response that the application resource could not be stopped because of dependencies. To force the application to stop the specified resource and all the other resources that depend on it, enter the following command:

```
#/usr/sbin/caa_stop -f clock
```

See caa_stop(8) for more information.

8.3.3 No Multiple Instances of an Application Resource

If multiple start and/or stop operations on the same application resource are initiated simultaneously, either on separate members or on a single member, it is uncertain which operation will prevail. However, multiple start operations do not result in multiple instances of an application resource.

8.3.4 Using caa_stop to Reset UNKNOWN State

If an application resource state is set to UNKNOWN, first try to run caa_stop. If it does not reset the resource to OFFLINE, use the caa_stop -f command. The command will ignore any errors returned by the stop script, set the resource to OFFLINE, and set all applications that depend on the application resource to OFFLINE as well.

Before you attempt to restart the application resource, look at the stop entry point of the action to be sure that it successfully stops the application and returns 0. Also make sure that it returns 0 if the application is not currently running.

8.4 Registering and Unregistering Resources

A resource must be registered with the CAA subsystem before CAA can manage that resource. This task needs to be performed only once for each resource.

Before a resource can be registered, a valid resource profile for the resource must exist in the /var/cluster/caa/profile directory. The TruCluster Server *Cluster Highly Available Applications* manual describes the process for creating resource profiles.

To learn which resources are registered on the cluster, enter the following caa_stat command:

```
# /usr/sbin/caa_stat
```

8.4.1 Registering Resources

Use the caa_register command to register an application resource as follows:

```
# caa_register resource_name
```

For example, to register an application resource named dtcalc, enter the following command:

```
# /usr/sbin/caa_stat dtcalc
```

If an application resource has resource dependencies defined in the REQUIRED_RESOURCES attribute of the profile, all resources listed for this attribute must be registered first.

For more information, see caa_register(8).

8.4.2 Unregistering Resources

You might want to unregister a resource to remove it from being monitored by the CAA subsystem. To unregister an application resource, you must first stop it, which changes the state of the resource to OFFLINE. See Section 8.3.2 for instructions on how to stop an application.

To unregister a resource, use the caa_unregister command. For example, to unregister the resource dtcalc, enter the following command:

```
# /usr/sbin/caa_unregister dtcalc
```

For more information, see caa_unregister(8).

For information on registering or unregistering a resource with the SysMan Menu, see the SysMan online help.

8.4.3 Updating Registration

You may need to update the registration of an application resource if you have modified its profile. For a detailed discussion of resource profiles see the *Cluster Highly Available Applications* manual.

To update the registration of a resource, use the caa_register -u command. For example, to update the resource dtcalc, enter the following command:

```
# /usr/sbin/caa_register -u dtcalc
```

Note

The caa_register -u command and the SysMan Menu allow you to update the REQUIRED_RESOURCES field in the profile of an ONLINE resource with the name of a resource that is OFFLINE.

This can cause the system to be out of synch with the profiles if you update the REQUIRED_RESOURCES field with an application that is OFFLINE. If you do this, you must manually start the required resource or stop the updated resource.

Similarly, a change to the HOSTING_MEMBERS list value of the profile only affects future relocations and starts. If you update the HOSTING_MEMBERS list in the profile of an ONLINE application resource with a restricted placement policy, make sure that the application is running on one of the cluster members in that list. If the application is not running on one of the allowed members, run the caa_relocate on the application after running the caa_register -u command.

8.5 Network, Tape, and Media Changer Resources

Only application resources can be stopped using caa_stop. However, nonapplication resources can be restarted using caa_start if they have had more failures than the resource failure threshold within the failure interval. Starting a nonapplication resource resets its TARGET value to ONLINE. This causes any applications that are dependent on this resource to start as well.

Network, tape, and media changer resources may fail repeatedly due to hardware problems. If this happens, do not allow CAA on the failing cluster member to use the device and, if possible, relocate or stop application resources. Exceeding the failure threshold within the failure interval causes the resource for the device to be disabled. If a resource is disabled, the TARGET state for the resource on a particular cluster member is set equal to OFFLINE, as shown with caa_stat *resource_name*. For example:

```
# /usr/sbin/caa_stat network1

NAME=network1
TYPE=network
TARGET=OFFLINE on provolone
TARGET=ONLINE on polishham
STATE=ONLINE on provolone
STATE=ONLINE on polishham
```

If a network, tape, or changer resource has the TARGET state set to OFFLINE because the failure count exceeds the failure threshold within the failure interval, the STATE for all resources that depend on that resource become OFFLINE though their TARGET remains ONLINE. These dependent applications will relocate to another machine where the resource is ONLINE. If no cluster member is available with this resource ONLINE, the applications remain OFFLINE until both the STATE and TARGET are ONLINE for the resource on the current member.

You can reset the TARGET state for a nonapplication resource to ONLINE by using the caa_start (for all members) or caa_start -c *cluster_member* command (for a particular member). The failure count is reset to zero (0) when this is done.

If the TARGET value is set to OFFLINE by a failure count that exceeds the failure threshold, the resource is treated as if it were OFFLINE by CAA, even though the STATE value may be ONLINE.

Note

If a tape or media changer resource is reconnected to a cluster after removal of the device while the cluster is running or a physical failure occurs, the cluster does not automatically detect the reconnection of the device. You must run the drdmgr -a DRD_CHECK_PATH *device_name* command.

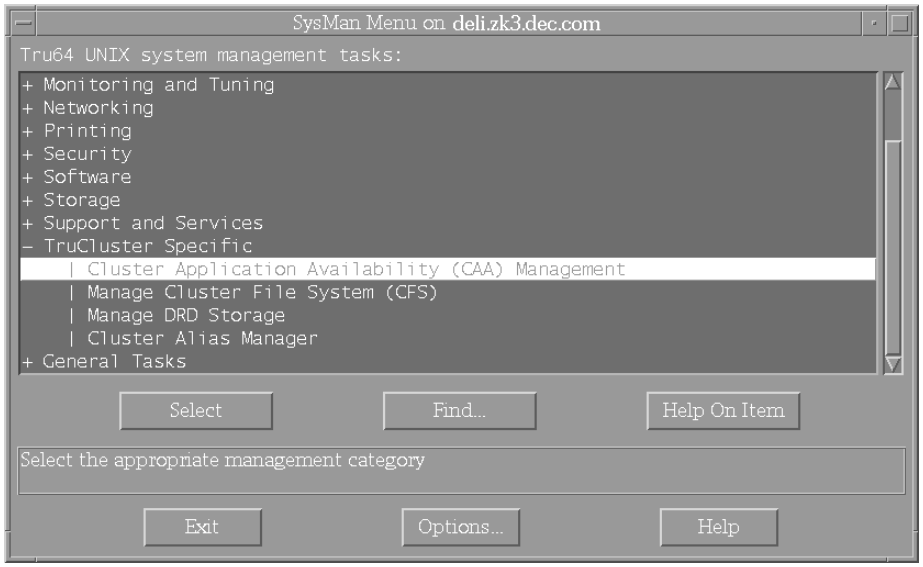
8.6 Using SysMan to Manage CAA

This section describes how to use the SysMan suite of tools to manage CAA. For a general discussion of invoking SysMan and using it in a cluster, see Chapter 2.

8.6.1 Managing CAA with SysMan Menu

The Cluster Application Availability (CAA) Management branch of the SysMan Menu is located under the TruCluster Specific heading as shown in Figure 8–1. You can open the CAA Management dialog box by either selecting Cluster Application Availability (CAA) Management on the menu and clicking on the Select button, or by double-clicking on the text.

Figure 8–1: CAA Branch of SysMan Menu



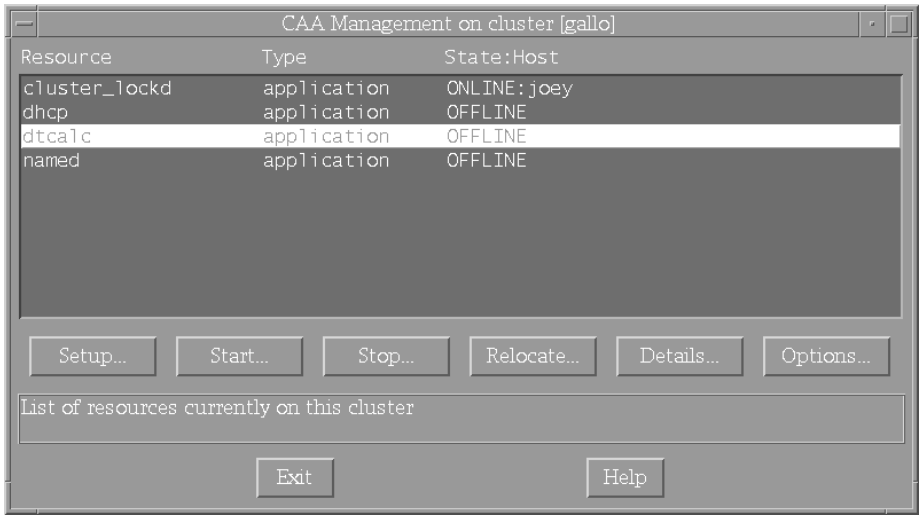
ZK-1756U-AI

8.6.1.1 CAA Management Dialog Box

The CAA Management dialog box (Figure 8–2) allows you to start, stop, and relocate applications. If you start or relocate an application, a dialog box prompts you to decide placement for the application.

You can also open the Setup dialog box to create, modify, register, and unregister resources.

Figure 8–2: CAA Management Dialog Box



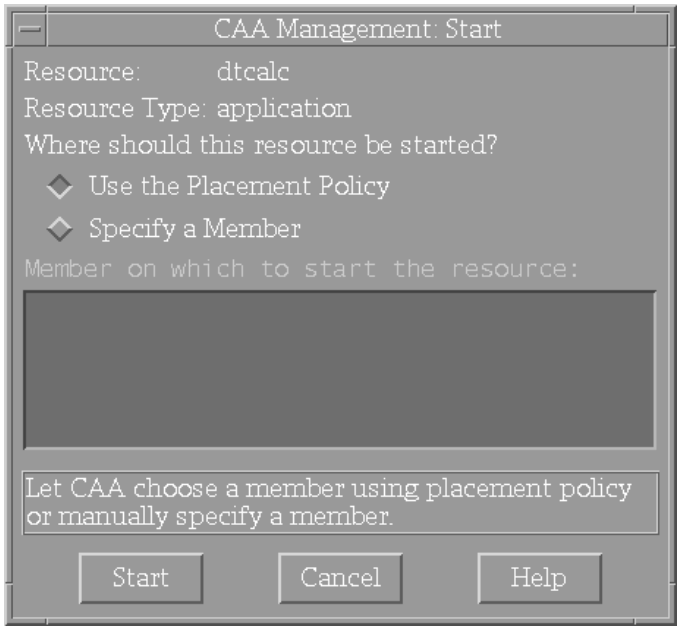
ZK-1755U-AI

8.6.1.2 Start Dialog Box

The Start dialog box (Figure 8–3) allows you to choose whether you want the application resource to be placed according to its placement policy or explicitly on another member.

You can place an application on a member explicitly only if it is allowed by the hosting member list. If the placement policy is `restricted`, and you try to place the application on a member that is not included in the hosting members list, the start attempt will fail.

Figure 8–3: Start Dialog Box

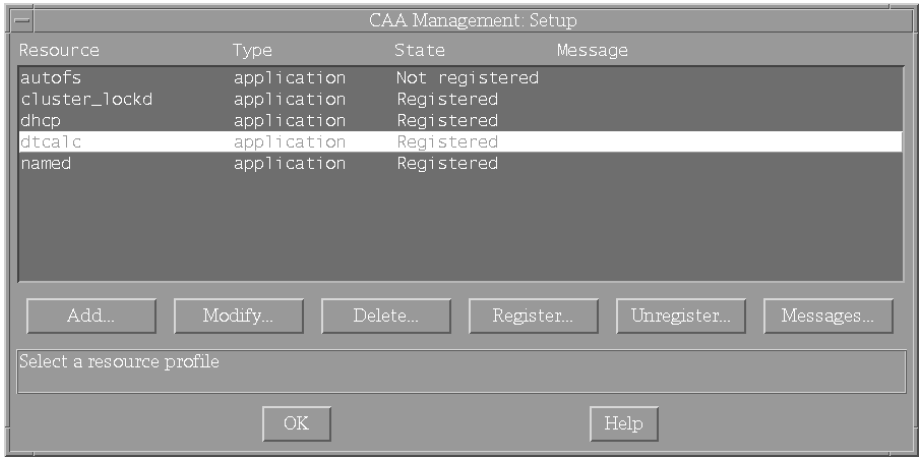


ZK-1757U-AI

8.6.1.3 Setup Dialog Box

To add, modify, register, and unregister profiles of any type, use the Setup dialog box, as shown in Figure 8–4. This dialog box can be reached from the Setup... button on the CAA Management dialog box. For details on setting up resources with SysMan Menu, see the online help.

Figure 8–4: Setup Dialog Box



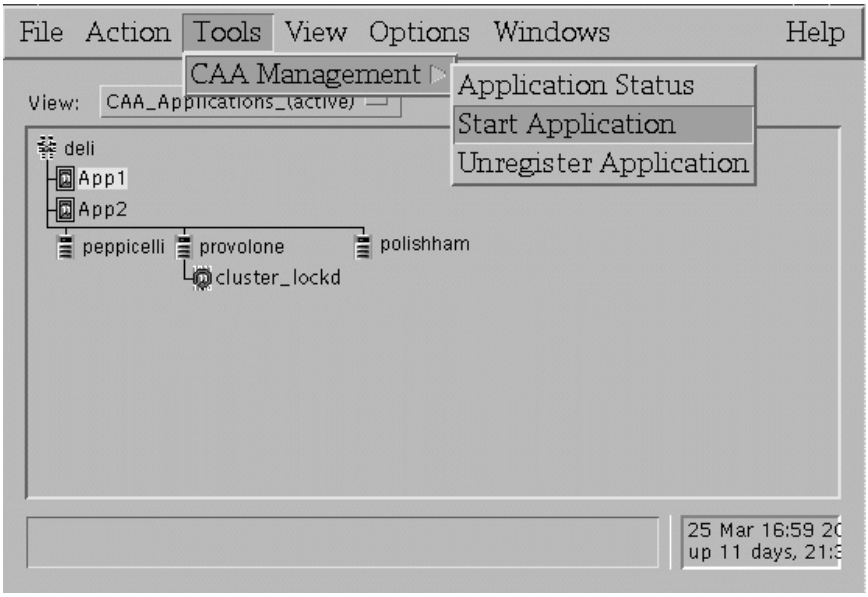
ZK-1758U-AI

8.6.2 Managing CAA with SysMan Station

The SysMan Station can be used to manage CAA resources. Figure 8–5 shows theSysMan Station CAA_Applications_(active) View. Figure 8–6 shows theSysMan Station CAA_Applications_(all) View. Select one of these views using the View menu at the top of the window. Selecting a cluster icon or cluster member icon makes the whole SysMan Menu available under the Tools menu, including CAA-specific tasks.

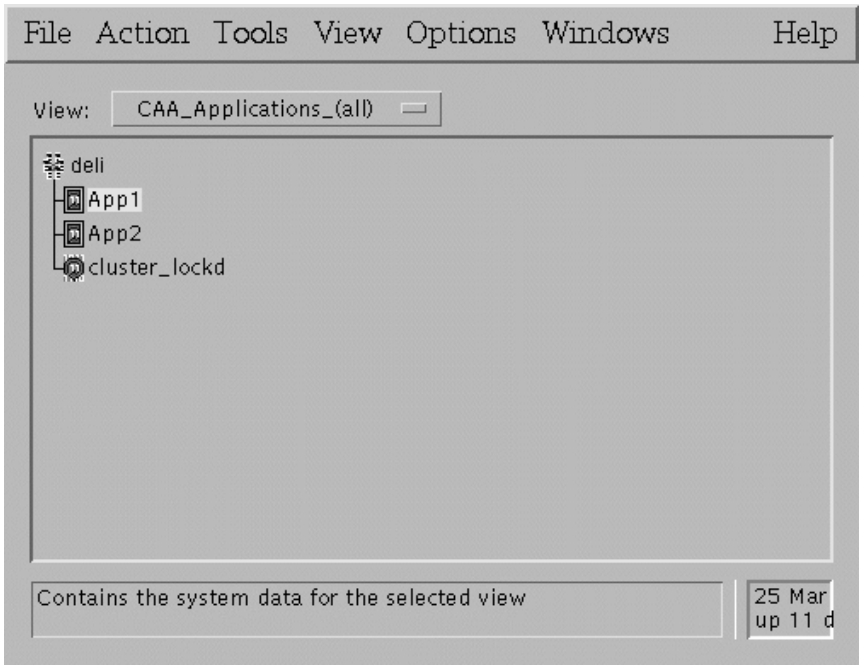
The icons for the application resources represent the resource state. In these two figures App1 and App2 are currently offline and cluster_lockd is online.

Figure 8–5: SysMan Station CAA_Applications_(active) View



ZK-1753U-AI

Figure 8–6: SysMan Station CAA_Applications_(all) View



ZK-1752U-AI

8.6.2.1 Starting an Application with SysMan Station

To start applications in either the CAA_Applications_(active) view (Figure 8–5) or the CAA_Applications_(all) View (Figure 8–6), select the application name under the cluster icon, click the right mouse button or click on the Tools Menu and select CAA Management ⇒ Start Application.

8.6.2.2 Resource Setup with SysMan Station

To set up resources using SysMan Station, select either the cluster icon or a cluster member icon. Click the right mouse button or click on the Tools menu, and select `CAA Management ⇒ CAA Setup`. See Figure 8–7. The rest of the steps are the same as for SysMan Menu and are described in detail in the Tasks section of the online help.

Figure 8–7: SysMan Station CAA Setup Screen

ZK-1754U-AI

8.7 CAA Considerations for Startup and Shutdown

The CAA daemon needs to read the information for every resource from the database. Because of this, if there are a large number of resources registered, your cluster members might take a long time to boot.

CAA may display the following message during a member boot:

```
Cannot communicate with the CAA daemon.
```

This message may or may not be preceded by the message:

```
Error: could not start up CAA Applications
Cannot communicate with the CAA daemon.
```

These messages indicate that you did not register the TruCluster Server license. When the member finishes booting, enter the following command:

```
# lmf list
```

If the TCS-UA license is not active, register it as described in the *Cluster Installation* guide and start the CAA daemon (caad) as follows:

```
#/usr/sbin/caad
```

When you shut down a cluster, CAA notes for each application resource whether it is `ONLINE` or `OFFLINE`. On restart of the cluster, applications that were `ONLINE` are restarted. Applications that were `OFFLINE` are not restarted. Applications that were marked as `UNKNOWN` are considered to be stopped. If an application was stopped because of an issue that the cluster reboot resolves, use the `caa_start` command to start the application.

If you want to choose placement of applications before shutting down a cluster member, determine the state of resources and relocate any applications from the member to be shut down to another member. Reasons for relocating applications are listed in Section 8.2.

Applications that are currently running when the cluster is shut down will be restarted when the cluster is reformed. Any applications that have AUTO_START set to 1 will also start when the cluster is reformed.

8.8 Managing caad

You should not have to manage the CAA daemon (caad). The CAA daemon is started at boot time and stopped at shutdown on every cluster member. However, if there are problems with the daemon, you may need to intervene.

If one of the commands caa_stat, caa_start, caa_stop, or caa_relocate responds with "Cannot communicate with the CAA daemon!", the caad daemon is probably not running. To determine whether the daemon is running, see Section 8.8.1.

8.8.1 Determining Status of the Local CAA Daemon

To determine the status of the CAA daemon, enter the following command:

```
# ps ax | grep -v grep | grep caad
```

If caad is running, output similar to the following is displayed:

```
545317 ??       S        0:00.38 caad
```

If nothing is displayed, caad is not running.

You can determine the status of other caad daemons by logging in to the other cluster members and running the ps ax |grep -v grep | grep caad command.

If the caad daemon is not running, CAA is no longer managing the application resources that were started on that machine. You cannot use caa_stop to stop the applications. After the daemon is restarted as described in Section 8.8.2, the resources on that machine should be fully manageable by CAA.

8.8.2 Restarting the CAA Daemon

If the caad daemon dies on one cluster member, all application resources continue to run, but you can no longer manage them with the CAA subsystem. You can restart the daemon by entering the /usr/sbin/caad command.

Do not use the startup script /sbin/init.d/clu_caa to restart the CAA daemon. Use this script only to start caad when a cluster member is booting up.

8.8.3 Monitoring CAA Daemon Messages

You can view information about changes to the state of resources by looking at events that are posted to EVM by the CAA daemon. For details on EVM messages, see Section 8.9.

8.9 Using EVM to View CAA Events

CAA posts events to Event Manager (EVM). These may be useful in troubleshooting errors that occur in the CAA subsystem.

Note

Some CAA actions are logged via syslog to /var/cluster/members/*{member}*/adm/ syslog.dated/*[date]*/daemon.log. When trying to identify problems, it may be useful to look in both the daemon.log and EVM for information. EVM has the advantage of being a single source of information for the whole cluster while daemon.log information is specific to each member. Some information is available only in the daemon.log files.

You can access EVM events either by using the SysMan Station or the EVM commands at the command line. For detailed information on how to use SysMan Station, see the Tru64 UNIX *System Administration* manual. See the online help for information on how to perform specific tasks.

Many events that CAA generates are defined in the EVM configuration file, /usr/share/evm/templates/clu/caa/caa.evt. These events all have a name in the form of sys.unix.clu.caa.*.

CAA also creates some events that have the name sys.unix.syslog.daemon. Events posted by other daemons are also posted with this name, so there will be more than just CAA events listed.

For detailed information on how to get information from the EVM Event Management System, see EVM(5), evmget(1), or evmshow(1).

8.9.1 Viewing CAA Events

To view events related to CAA that have been sent to EVM, enter the following command:

```
# evmget -f "[name *.caa.*]" | evmshow
CAA cluster_lockd was registered
```

```
CAA cluster_lockd is transitioning from state ONLINE to state OFFLINE
CAA resource sbtest action script /var/cluster/caa/script/foo.scr (start): success
CAA Test2002_Scale6 was registered
CAA Test2002_Scale6 was unregistered
```

To get more verbose event detail from EVM, use the `-d` option as follows:

```
# evmget -f '[name *.caa.*]' | evmshow -d | more
============================ EVM Log event ============================
EVM event name: sys.unix.clu.caa.app.registered

    This event is posted by the Cluster Application Availability
    subsystem (CAA) when a new application has been registered.

=======================================================================

Formatted Message:
    CAA a was registered

Event Data Items:
    Event Name        : sys.unix.clu.caa.app.registered
    Cluster Event     : True
    Priority          : 300
    PID               : 1109815
    PPID              : 1103504
    Event Id          : 4578
    Member Id         : 2
    Timestamp         : 18-Apr-2001 16:56:17
    Cluster IP address: 16.69.225.123
    Host Name         : provolone.zk4.dec.com
    Cluster Name      : deli
    User Name         : root
    Format            : CAA $application was registered
    Reference         : cat:evmexp_caa.cat

Variable Items:
    application (STRING) = "a"

=======================================================================
```

The template script `/var/cluster/caa/template/template.scr` has been updated to create scripts that post events to EVM when CAA attempts to start, stop, or check applications. Any action scripts that were newly created with `caa_profile` or SysMan will now post events to EVM. To view only these events, enter the following command

```
# evmget -f "[name sys.unix.clu.caa.action_script]" | evmshow -t "@timestamp  @@"
```

CAA events can also be viewed by using SysMan Station. Click on the Status Light or Label Box for Applications in the SysMan Station Monitor Window.

To view other events that are logged by the `caad` daemon, as well as other daemons, enter the following command:

```
# evmget -f "[name sys.unix.syslog.daemon]" | \
evmshow -t "@timestamp  @@"
```

8.9.2 Monitoring CAA Events

To monitor CAA events with time stamps on the console, enter the following command:

```
# evmwatch -f "[name *.caa.*]" | evmshow "@timestamp  @@"
```

As events that are related to CAA are posted to EVM, they are displayed on the terminal where this command is executed. An example of the messages is as follows:

```
CAA cluster_lockd was registered
CAA cluster_lockd is transitioning from state ONLINE to state OFFLINE
CAA Test2002_Scale6 was registered
CAA Test2002_Scale6 was unregistered
CAA xclock is transitioning from state ONLINE to state OFFLINE
CAA xclock had an error, and is no longer running
CAA cluster_lockd is transitioning from state ONLINE to state OFFLINE
CAA cluster_lockd started on member polishham
```

To monitor other events that are logged by the CAA daemon using the `syslog` facility, enter the following command:

```
# evmwatch -f "[name sys.unix.syslog.daemon]" | evmshow | grep CAA
```

8.10 Troubleshooting with Events

The error messages in this section may be displayed when showing events from the CAA daemon by entering the following command:

```
# evmget -f "[name sys.unix.syslog.daemon]" | evmshow | grep CAA
```

Action Script Has Timed Out

```
CAAD[564686]: RTD #0: Action Script \
/var/cluster/caa/script/[script_name].scr(start) timed out! (timeout=60)
```

First determine that the action script correctly starts the application by running `/var/cluster/caa/script/[script_name].scr start`. If the action script runs correctly and successfully returns with no errors, but it takes longer to execute than the `SCRIPT_TIMEOUT` value, increase the `SCRIPT_TIMEOUT` value. If an application that is executed in the script takes a long time to finish, you may want to background the task in the script by adding an ampersand (`&`) to the line in the script that starts the application. This will however cause the command to always return a status of 0 and CAA will have no way of detecting a command that failed to start for some trivial reason, such as a misspelled command path.

Action Script Stop Entry Point Not Returning 0

```
CAAD[524894]: 'foo' on member 'provolone' has experienced an unrecoverable failure.
```

This message occurs when a stop entry point returns a value other than 0. The resource is put into the `UNKNOWN` state. The application must be stopped by correcting the stop action script to return 0 and running `caa_stop` or

caa_stop -f. In either case, fix the stop action script to return 0 before you attempt to restart the application resource.

Network Failure

```
CAAD[524764]: 'tu0' has gone offline on member 'skiing'
```

A message like this for network resource tu0 indicates that the network has gone down. Make sure that the network card is connected correctly. Replace the card, if necessary.

Lock Preventing Start of CAA Daemon

```
CAAD[526369]: CAAD exiting; Another caad may be running, could not obtain \
lock file /var/cluster/caa/locks/.lock-provolone.dec.com
```

A message similar to this is displayed when attempting to start a second caad. Determine whether caad is running as described in Section 8.8.1. If there is no daemon running, remove the lock file that is listed in the message and restart caad as described in Section 8.8.2.

8.11 Troubleshooting a Command-Line Message

A message like the following indicates that CAA cannot find the profile for a resource that you attempted to register:

```
Cannot access the resource
profile file_name
```

For example, if there is no profile for clock, an attempt to register clock fails as follows:

```
# caa_register clock
Cannot access the resource profile '/var/cluster/caa/profile/clock.cap'.
```

The resource profile is either not in the right location or does not exist. You must make sure that the profile exists in the location that is cited in the message.

9

Managing File Systems and Devices

This chapter contains information specific to managing storage devices in a TruCluster Server system. The chapter discusses the following subjects:

- Working with CDSLs (Section 9.1)
- Managing devices (Section 9.2)
- Managing the Cluster File System (Section 9.3)
- Managing the device request dispatcher (Section 9.4)
- Managing AdvFS in a cluster (Section 9.5)
- Creating new file systems (Section 9.6)
- Managing CDFS file systems (Section 9.7)
- Backing up and restoring files (Section 9.8)
- Managing swap space (Section 9.9)
- Fixing problems with boot parameters (Section 9.10)
- Using the `verify` command in a cluster (Section 9.11)

You can find other information on device management in the Tru64 UNIX Version 5.1A documentation that is listed in Table 9–1.

Table 9–1: Sources of Information of Storage Device Management

Topic	Tru64 UNIX Manual
Administering devices	*System Administration* manual
Administering file systems	*System Administration* manual
Administering the archiving services	*System Administration* manual
Managing AdvFS	*AdvFS Administration* manual

For information about Logical Storage Manager (LSM) and clusters, see Chapter 10.

9.1 Working with CDSLs

A context-dependent symbolic link (CDSL) is a link that contains a variable that identifies a cluster member. This variable is resolved at run time into a target.

A CDSL is structured as follows:

```
/etc/rc.config -> ../cluster/members/{memb}/etc/rc.config
```

When resolving a CDSL pathname, the kernel replaces the string {memb} with the string member*n*, where *n* is the member ID of the current member. For example, on a cluster member whose member ID is 2, the pathname /cluster/members/*{memb}*/etc/rc.config resolves to /cluster/members/member2/etc/rc.config.

CDSLs provide a way for a single file name to point to one of several files. Clusters use this to allow member-specific files that can be addressed throughout the cluster by a single file name. System data and configuration files tend to be CDSLs. They are found in the root (/), /usr, and /var directories.

9.1.1 Making CDSLs

The mkcdsl command provides a simple tool for creating and populating CDSLs. For example, to make a new CDSL for the file /usr/accounts/usage-history, enter the following command:

```
# mkcdsl /usr/accounts/usage-history
```

When you list the results, you see the following output:

```
# ls -l /usr/accounts/usage-history

... /usr/accounts/usage-history -> cluster/members/{memb}/accounts/usage-history
```

The CDSL usage-history is created in /usr/accounts. No files are created in any member's /usr/cluster/members/*{memb}* directory.

To move a file into a CDSL, enter the following command:

```
# mkcdsl -c targetname
```

To replace an existing file when using the copy (-c) option, you must also use the force (-f) option.

The -c option copies the source file to the member-specific area on the cluster member where the mkcdsl command executes and then replaces the source file with a CDSL. To copy a source file to the member-specific area on all cluster members and then replace the source file with a CDSL, use the -a option to the command as follows:

```
# mkcdsl -a filename
```

Remove a CDSL with the rm command, as you would any symbolic link.

The file /var/adm/cdsl_admin.inv stores a record of the cluster's CDSLs. When you use mkcdsl to add CDSLs, the command updates /var/adm/cdsl_admin.inv. If you use the ln -s command to create CDSLs, /var/adm/cdsl_admin.inv is not updated.

To update /var/adm/cdsl_admin.inv, enter the following:

```
# mkcdsl -i targetname
```

Update the inventory when you remove a CDSL, or if you use the ln -s command to create a CDSL.

For more information, see mkcdsl(8).

9.1.2 Maintaining CDSLs

The following tools can help you maintain CDSLs:

- clu_check_config(8)
- cdslinvchk(8)
- mkcdsl(8) (with the -i option)

The following example shows the output (and the pointer to a log file containing the errors) when clu_check_config finds a bad or missing CDSL:

```
# clu_check_config -s check_cdsl_config
Starting Cluster Configuration Check...
check_cdsl_config : Checking installed CDSLs
check_cdsl_config : CDSLs configuration errors : See /var/adm/cdsl_check_list
clu_check_config : detected one or more configuration errors
```

As a general rule, before you move a file, make sure that the destination is not a CDSL. If by mistake you do overwrite a CDSL on the appropriate cluster member, use the mkcdsl -c *filename* command to copy the file and re-create the CDSL.

9.1.3 Kernel Builds and CDSLs

When you build a kernel in a cluster, use the mv command to move the new kernel from /sys/*HOSTNAME*/vmunix to /cluster/members/member*n*/boot_partition/vmunix. If you move the kernel to /vmunix, you will overwrite the /vmunix CDSL. The result will be that the next time that cluster member boots, it will use the old vmunix in /sys/*HOSTNAME*/vmunix.

9.1.4 Exporting and Mounting CDSLs

CDSLs are intended for use when files of the same name must necessarily have different contents on different cluster members. Because of this, CDSLs are not intended for export.

Mounting CDSLs through the cluster alias is problematic, because the file contents differ depending on which cluster system gets the mount request.

However, nothing prevents CDSLs from being exported. If the entire directory is a CDSL, then the node that gets the mount request provides a file handle corresponding to the directory for that node. If a CDSL is contained within an exported clusterwide directory, then the Network File System (NFS) server that gets the request will do the expansion. As with normal symbolic links, the client cannot read the file or directory unless that area is also mounted on the client.

9.2 Managing Devices

Device management in a cluster is similar to that in a standalone system, with the following exceptions:

- The `dsfmgr` command for managing device special files takes special options for clusters.
- Because of the mix of shared and private buses in a cluster, device topology can be more complex.
- You can control which cluster members act as servers for the devices in the cluster, and which members act as access nodes.

The rest of this section describes these differences.

9.2.1 Managing the Device Special File

When using `dsfmgr`, the device special file management utility, in a cluster, keep the following in mind:

- The `-a` option requires that you use `c` (cluster) as the `entry_type`.
- The `-o` and `-O` options, which create device special files in the old format, are not valid in a cluster.
- In the output from the `-s` option, the `class scope` column in the first table uses a `c` (cluster) to indicate the scope of the device.

For more information, see `dsfmgr`(8). For information on devices, device naming, and device management, see the chapter on hardware management in the Tru64 UNIX *System Administration* manual.

9.2.2 Determining Device Locations

The Tru64 UNIX `hwmgr` command can list all hardware devices in the cluster, including those on private buses, and correlate bus-target-LUN names with `/dev/disks/dsk*` names. For example:

```
# hwmgr -view devices -cluster
HWID: Device Name         Mfg     Model            Hostname   Location
------------------------------------------------------------------------------
  3: kevm                                         pepicelli
 28: /dev/disk/floppy0c          3.5in floppy     pepicelli  fdi0-unit-0
```

```
 40: /dev/disk/dsk0c     DEC     RZ28M    (C) DEC pepicelli  bus-0-targ-0-lun-0
 41: /dev/disk/dsk1c     DEC     RZ28L-AS (C) DEC pepicelli  bus-0-targ-1-lun-0
 42: /dev/disk/dsk2c     DEC     RZ28     (C) DEC pepicelli  bus-0-targ-2-lun-0
 43: /dev/disk/cdrom0c   DEC     RRD46    (C) DEC  pepicelli bus-0-targ-6-lun-0
 44: /dev/disk/dsk3c     DEC     RZ28M    (C) DEC pepicelli  bus-1-targ-1-lun-0
 44: /dev/disk/dsk3c     DEC     RZ28M    (C) DEC polishham  bus-1-targ-1-lun-0
 44: /dev/disk/dsk3c     DEC     RZ28M    (C) DEC provolone  bus-1-targ-1-lun-0
 45: /dev/disk/dsk4c     DEC     RZ28L-AS (C) DEC pepicelli  bus-1-targ-2-lun-0
 45: /dev/disk/dsk4c     DEC     RZ28L-AS (C) DEC polishham  bus-1-targ-2-lun-0
 45: /dev/disk/dsk4c     DEC     RZ28L-AS (C) DEC provolone  bus-1-targ-2-lun-0
 46: /dev/disk/dsk5c     DEC     RZ29B    (C) DEC pepicelli  bus-1-targ-3-lun-0
 46: /dev/disk/dsk5c     DEC     RZ29B    (C) DEC polishham  bus-1-targ-3-lun-0
 46: /dev/disk/dsk5c     DEC     RZ29B    (C) DEC provolone  bus-1-targ-3-lun-0
 47: /dev/disk/dsk6c     DEC     RZ28D    (C) DEC pepicelli  bus-1-targ-4-lun-0
 47: /dev/disk/dsk6c     DEC     RZ28D    (C) DEC polishham  bus-1-targ-4-lun-0
 47: /dev/disk/dsk6c     DEC     RZ28D    (C) DEC provolone  bus-1-targ-4-lun-0
 48: /dev/disk/dsk7c     DEC     RZ28L-AS (C) DEC pepicelli  bus-1-targ-5-lun-0
 48: /dev/disk/dsk7c     DEC     RZ28L-AS (C) DEC polishham  bus-1-targ-5-lun-0
 48: /dev/disk/dsk7c     DEC     RZ28L-AS (C) DEC provolone  bus-1-targ-5-lun-0
 49: /dev/disk/dsk8c     DEC     RZ1CF-CF (C) DEC pepicelli  bus-1-targ-8-lun-0
 49: /dev/disk/dsk8c     DEC     RZ1CF-CF (C) DEC polishham  bus-1-targ-8-lun-0
 49: /dev/disk/dsk8c     DEC     RZ1CF-CF (C) DEC provolone  bus-1-targ-8-lun-0
 50: /dev/disk/dsk9c     DEC     RZ1CB-CS (C) DEC pepicelli  bus-1-targ-9-lun-0
 50: /dev/disk/dsk9c     DEC     RZ1CB-CS (C) DEC polishham  bus-1-targ-9-lun-0
 50: /dev/disk/dsk9c     DEC     RZ1CB-CS (C) DEC provolone  bus-1-targ-9-lun-0
 51: /dev/disk/dsk10c    DEC     RZ1CF-CF (C) DEC pepicelli  bus-1-targ-10-lun-0
 51: /dev/disk/dsk10c    DEC     RZ1CF-CF (C) DEC polishham  bus-1-targ-10-lun-0
 51: /dev/disk/dsk10c    DEC     RZ1CF-CF (C) DEC provolone  bus-1-targ-10-lun-0
 52: /dev/disk/dsk11c    DEC     RZ1CF-CF (C) DEC pepicelli  bus-1-targ-11-lun-0
 52: /dev/disk/dsk11c    DEC     RZ1CF-CF (C) DEC polishham  bus-1-targ-11-lun-0
 52: /dev/disk/dsk11c    DEC     RZ1CF-CF (C) DEC provolone  bus-1-targ-11-lun-0
 53: /dev/disk/dsk12c    DEC     RZ1CF-CF (C) DEC pepicelli  bus-1-targ-12-lun-0
 53: /dev/disk/dsk12c    DEC     RZ1CF-CF (C) DEC polishham  bus-1-targ-12-lun-0
 53: /dev/disk/dsk12c    DEC     RZ1CF-CF (C) DEC provolone  bus-1-targ-12-lun-0
 54: /dev/disk/dsk13c    DEC     RZ1CF-CF (C) DEC pepicelli  bus-1-targ-13-lun-0
 54: /dev/disk/dsk13c    DEC     RZ1CF-CF (C) DEC polishham  bus-1-targ-13-lun-0
 54: /dev/disk/dsk13c    DEC     RZ1CF-CF (C) DEC provolone  bus-1-targ-13-lun-0
 59: kevm                                     polishham
 88: /dev/disk/floppy1c          3.5in floppy    polishham  fdi0-unit-0
 94: /dev/disk/dsk14c    DEC     RZ26L    (C) DEC polishham  bus-0-targ-0-lun-0
 95: /dev/disk/cdrom1c   DEC     RRD46   (C) DEC  polishham  bus-0-targ-4-lun-0
 96: /dev/disk/dsk15c    DEC     RZ1DF-CB (C) DEC polishham  bus-0-targ-8-lun-0
 99: /dev/kevm                                    provolone
127: /dev/disk/floppy2c          3.5in floppy    provolone  fdi0-unit-0
134: /dev/disk/dsk16c    DEC     RZ1DF-CB (C) DEC provolone  bus-0-targ-0-lun-0
135: /dev/disk/dsk17c    DEC     RZ1DF-CB (C) DEC provolone  bus-0-targ-1-lun-0
136: /dev/disk/cdrom2c   DEC     RRD47   (C) DEC  provolone  bus-0-targ-4-lun-0
```

The `drdmgr` *devicename* command reports which members serve the device. Disks with multiple servers are on a shared SCSI bus. With very few exceptions, disks that have only one server are private to that server. For details on the exceptions, see Section 9.4.1.

To learn the hardware configuration of a cluster member, enter the following command:

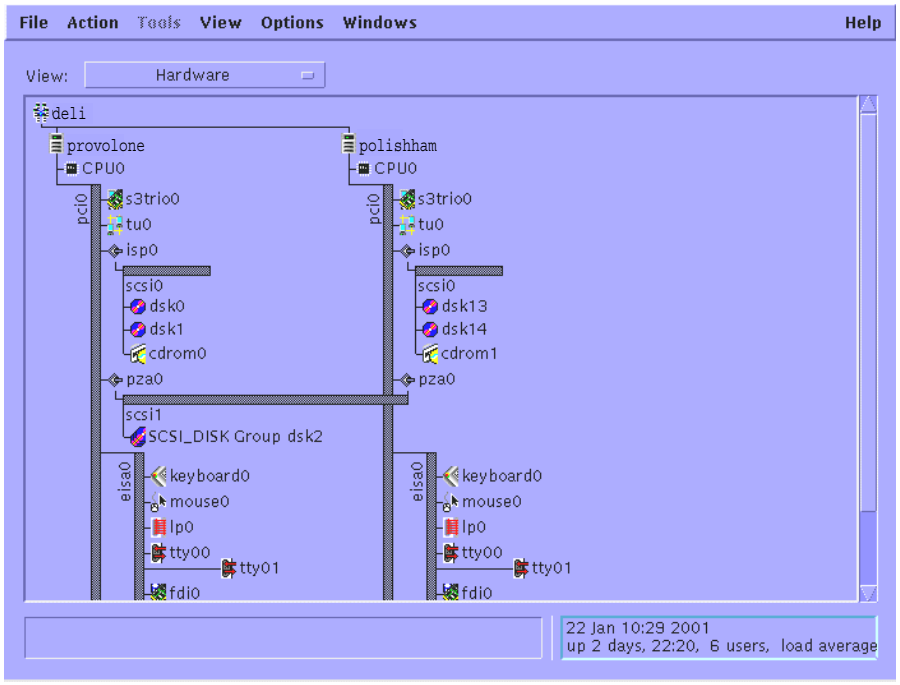
```
# hwmgr -view hierarchy -member membername
```

If the member is on a shared bus, the command reports devices on the shared bus. The command does not report on devices private to other members.

To get a graphical display of the cluster hardware configuration, including active members, buses, both shared and private storage devices, and their connections, use the `sms` command to invoke the graphical interface for the SysMan Station, and then select Hardware from the View menu.

Figure 9–1 shows the SysMan Station representation of a two-member cluster.

Figure 9–1: SysMan Station Display of Hardware Configuration



ZK-1700U-AI

9.2.3 Adding a Disk to the Cluster

For information on physically installing SCSI hardware devices, see the TruCluster Server *Cluster Hardware Configuration* manual. After the new disk has been installed, follow these steps:

1. So that all members recognize the new disk, run the following command on each member:

   ```
   # hwmgr -scan comp -cat scsi_bus
   ```

Note

You must run the `hwmgr -scan comp -cat scsi_bus` command on every cluster member that needs access to the disk.

Wait a minute or so for all members to register the presence of the new disk.

2. If the disk that you are adding is an RZ26, RZ28, RZ29, or RZ1CB-CA model, run the following command on each cluster member:

   ```
   # /usr/sbin/clu_disk_install
   ```

   If the cluster has a large number of storage devices, this command can take several minutes to complete.

3. To learn the name of the new disk, enter the following command:

   ```
   # hwmgr -view devices -cluster
   ```

   You can also run the SysMan Station command and select Hardware from the Views menu to learn the new disk name.

For information about creating file systems on the disk, see Section 9.6.

9.2.4 Managing Third-party Storage

When a cluster member loses quorum, all of its I/O is suspended, and the remaining members erect I/O barriers against nodes that have been removed from the cluster. This I/O barrier operation inhibits non-cluster members from performing I/O with shared storage devices.

The method that is used to create the I/O barrier depends on the types of storage devices that the cluster members share. In certain cases, a Task Management function called a **Target_Reset** is sent to stop all I/O to and from the former member. This Task Management function is used in either of the following situations:

- The shared SCSI device does not support the SCSI Persistent Reserve command set and uses the Fibre Channel interconnect.
- The shared SCSI device does not support the SCSI Persistent Reserve command set, uses the SCSI Parallel interconnect, is a multiported device, and does not propagate the SCSI `Target_Reset` signal.

In either of these situations, there is a delay between the `Target_Reset` and the clearing of all I/O pending between the device and the former member. The length of this interval depends on the device and the cluster configuration. During this interval, some I/O with the former member might

still occur. This I/O, sent after the Target_Reset, completes in a normal way without interference from other nodes.

During an interval configurable with the drd_target_reset_wait kernel attribute, the device request dispatcher suspends all new I/O to the shared device. This period allows time to clear those devices of the pending I/O that originated with the former member and were sent to the device after it received the Target_Reset. After this interval passes, the I/O barrier is complete.

The default value for drd_target_reset_wait is 30 seconds, which should be sufficient. However, if you have doubts because of third-party devices in your cluster, contact the device manufacturer and ask for the specifications on how long it takes their device to clear I/O after the receipt of a Target_Reset.

You can set drd_target_reset_wait at boot time and run time.

For more information about quorum loss and system partitioning, see the chapter on the connection manager in the TruCluster Server *Cluster Technical Overview*.

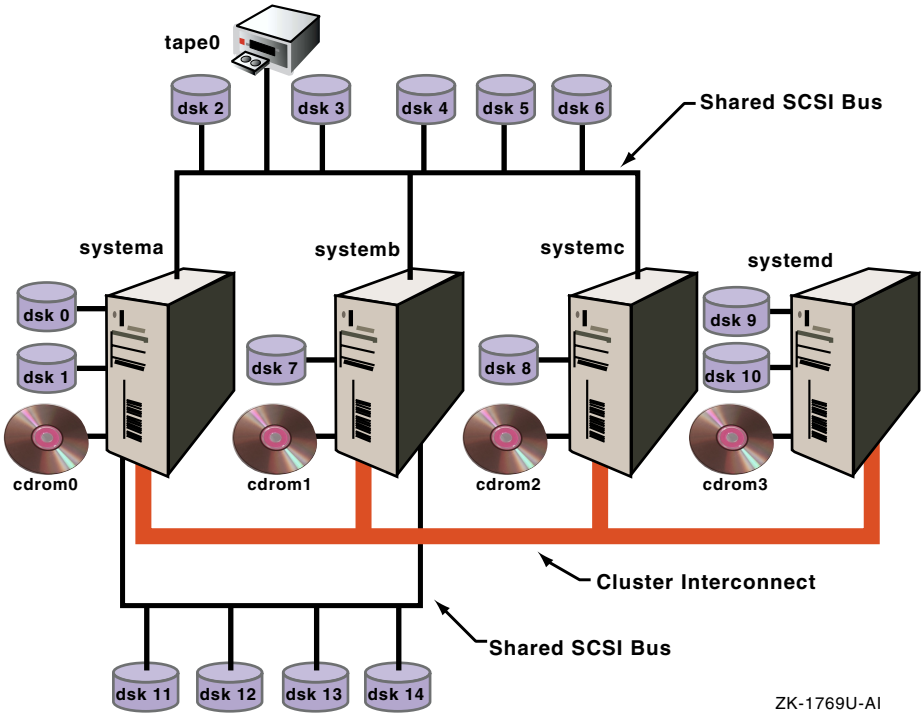
9.2.5 Tape Devices

You can access a tape device in the cluster from any member, regardless of whether it is located on that member's private bus, on a shared bus, or on another member's private bus.

Certain operations, such as mcutil, can be performed only on a device that is directly connected to the member where the operation is performed. For this reason, it is advantageous to place a tape device on a shared bus, where multiple members have direct access to the device.

Performance considerations also argue for placing a tape device on a shared bus. Backing up storage connected to a system on a shared bus with a tape drive is faster than having to go over the cluster interconnect. For example, in Figure 9–2, the backup of dsk9 and dsk10 to the tape drive requires the data to go over the cluster interconnect. For the backup of any other disk, including the semi-private disks dsk11, dsk12, dsk13, and dsk14, the data transfer rate will be faster.

Figure 9–2: Cluster with Semi-private Storage



If the tape device is located on the shared bus, applications that access the device must be written to react appropriately to certain events on the shared SCSI bus, such as bus and device resets. Bus and device resets (such as those that result from cluster membership transitions) cause any tape device on the shared SCSI bus to rewind.

A `read()` or `write()` by a tape server application causes an `errno` to be returned. You must explicitly set up the tape server application to retrieve error information that is returned from its I/O call to reposition the tape. When a `read()` or `write()` operation fails, use `ioctl()` with the `MTIOCGET` command option to return a structure that contains the error information that is needed by the application to reposition the tape. For a description of the structure, see `/usr/include/sys/mtio.h`.

The commonly used utilities `tar`, `cpio`, `dump`, and `vdump` are not designed in this way, so they may unexpectedly terminate when used on a tape device that resides on a shared bus in a cluster. Currently, the only advantage to situating a tape device on a shared bus in this release is that multiple systems are physically connected to it, and any one of those systems can access it.

9.2.6 Formatting Floppy Disks in a Cluster

TruCluster Server Version 5.1A includes support for read/write UNIX File System (UFS) file systems, as described in Section 9.3.4, and you can use TruCluster Server Version 5.1A to format a floppy disk.

Versions of TruCluster Server prior to Version 5.1A do not support read/write UFS file systems. Because prior versions of TruCluster Server do not support read/write UFS file systems and AdvFS metadata overwhelms the capacity of a floppy disk, the typical methods to format a floppy cannot be used in a cluster.

If you must format a floppy disk in a cluster with a version of TruCluster Server prior to Version 5.1A, use the `mtools` or `dxmtools` tool sets. For more information, see `mtools`(1) and `dxmtools`(1).

9.2.7 CD-ROM and DVD-ROM

CD-ROM drives and DVD-ROM drives are always served devices. This type of drive must be connected to a local bus; it cannot be connected to a shared bus.

For information about managing a CD-ROM File System (CDFS) in a cluster, see Section 9.7.

9.3 Managing the Cluster File System

The Cluster File System (CFS) provides transparent access to files that are located anywhere on the cluster. Users and applications enjoy a single-system image for file access. Access is the same regardless of the cluster member where the access request originates, and where in the cluster the disk containing the file is connected. CFS follows a server/client model, with each file system served by a cluster member. Any cluster member can serve file systems on devices anywhere in the cluster. If the member serving a file system becomes unavailable, the CFS server automatically fails over to an available cluster member.

The primary tool for managing the cluster file system is the `cfsmgr` command. A number of examples of using the command appear in this section. For more information about the `cfsmgr` command, see `cfsmgr`(8).

To gather statistics about the CFS file system, use the `cfsstat` command or the `cfsmgr -statistics` command. An example of using `cfsstat` to get information about direct I/O appears in Section 9.3.3.5. For more information on the command, see `cfsstat`(8).

For file systems on devices on the shared bus, I/O performance depends on the load on the bus and the load on the member serving the file system.

To simplify load balancing, CFS allows you to easily relocate the server to a different member. Access to file systems on devices that are private to a member is faster when the file systems are served by that member.

Use the `cfsmgr` command to learn which files systems are served by which member. For example, to learn the server of the clusterwide root file system (`/`), enter the following command:

```
# cfsmgr /

 Domain or filesystem name = /
 Server Name = systemb
 Server Status : OK
```

To move the CFS server to a different member, enter the following `cfsmgr` command to change the value of the `SERVER` attribute:

```
# cfsmgr -a server=systema /
# cfsmgr /

 Domain or filesystem name = /
 Server Name = systema
 Server Status : OK
```

Although you can relocate the CFS server of the clusterwide root, you cannot relocate the member root domain to a different member. A member always serves its own member root domain, `root`*`memberID`*`_domain#root`.

When a cluster member boots, that member serves any file systems on the devices that are on buses that are private to the member. However, when you manually mount a file system or mount it via the `fstab` file, the server is chosen based on connectivity to the device from available servers. This can result in a file system being served by a member that is not local to it. In this case, you might see a performance improvement if you manually relocate the CFS server to the local member.

9.3.1 When File Systems Cannot Fail Over

In most instances, CFS provides seamless failover for the file systems in the cluster. If the cluster member serving a file system becomes unavailable, CFS fails over the server to an available member. However, in the following situations, no path to the file system exists and the file system cannot fail over:

- The file system's storage is on a private bus that is connected directly to a member and that member becomes unavailable.
- The storage is on a shared bus and all the members on the shared bus become unavailable.

In either case, the `cfsmgr` command returns the following status for the file system (or domain):

```
Server Status : Not Served
```

Attempts to access the file system return the following message:

filename `I/O error`

When a cluster member that is connected to the storage becomes available, the file system becomes served again and accesses to the file system begin to work. Other than making the member available, you do not need to take any action.

9.3.2 Direct Access Cached Reads

TruCluster Server implements **direct access cached reads**, which is a performance enhancement for AdvFS file systems. Direct access cached reads allow CFS to read directly from storage simultaneously on behalf of multiple cluster members.

If the cluster member that issues the read is directly connected to the storage that makes up the file system, direct access cached reads access the storage directly and do not go through the cluster interconnect to the CFS server.

If a CFS client is not directly connected to the storage that makes up a file system (for example, if the storage is private to a cluster member), that client will still issue read requests directly to the devices, but the device request dispatcher layer sends the read request across the cluster interconnect to the device.

Direct access cached reads are consistent with the existing CFS served file-system model, and the CFS server continues to perform metadata and log updates for the read operation.

Direct access cached reads are implemented only for AdvFS file systems. In addition, direct access cached reads are performed only for files that are at least 64K in size. The served I/O method is more efficient when processing smaller files.

Direct access cached reads are enabled by default and are not user-settable or tunable. However, if an application uses direct I/O, as described in Section 9.3.3.5, that choice is given priority and direct access cached reads are not performed for that application.

Use the `cfsstat directio` command to display direct I/O statistics. The `direct i/o reads` field includes direct access cached read statistics. See Section 9.3.3.5.3 for a description of these fields.

```
# cfsstat directio
Concurrent Directio Stats:
     941 direct i/o reads
       0 direct i/o writes
       0 aio raw reads
       0 aio raw writes
       0 unaligned block reads
      29 fragment reads
      73 zero-fill (hole) reads
       0 file-extending writes
       0 unaligned block  writes
       0 hole writes
       0 fragment writes
       0 truncates
```

9.3.3 Optimizing CFS Performance

You can tune CFS performance by doing the following:

- Balancing the CFS load (Section 9.3.3.1)
- Automatically distributing CFS server load (Section 9.3.3.2)
- Adjusting I/O transfer size (Section 9.3.3.3)
- Changing the number of read-ahead and write-behind threads (Section 9.3.3.4)
- Taking advantage of direct I/O (Section 9.3.3.5)
- Adjusting CFS memory usage (Section 9.3.3.6)
- Using memory mapped files (Section 9.3.3.7)
- Avoid full file systems (Section 9.3.3.8)
- Other strategies (Section 9.3.3.9)

9.3.3.1 CFS Load Balancing

When a cluster boots, the TruCluster Server software ensures that each file system is directly connected to the member that serves it. This means that file systems on a device connected to a member's local bus are served by that member. A file system on a device on a shared SCSI bus is served by one of the members that is directly connected to that SCSI bus.

In the case of AdvFS, the first fileset that is assigned to a CFS server determines that all other filesets in that domain will have that same cluster member as their CFS server.

When a cluster boots, typically the first member up that is connected to a shared SCSI bus is the first member to see devices on the shared bus.

This member then becomes the CFS server for all the file systems on all the devices on that shared bus. Because of this, most file systems are probably served by a single member. This situation can have negative consequences for performance. It is important to monitor file system activity on the cluster and load balance the CFS servers as necessary.

Use the `cfsmgr` command to determine good candidates for relocating the CFS servers. The `cfsmgr` command displays statistics on file system usage on a per-member basis. For example, suppose you want to determine whether to relocate the server for `/accounts` to improve performance. First, confirm the current CFS server of `/accounts` as follows:

```
# cfsmgr /accounts

 Domain or filesystem name = /accounts
 Server Name = systemb
 Server Status : OK
```

Then, get the CFS statistics for the current server and the candidate servers by entering the following commands:

```
# cfsmgr -h systemb -a statistics /accounts

 Counters for the filesystem /accounts:
        read_ops = 4149
        write_ops = 7572
        lookup_ops = 82563
        getattr_ops = 408165
        readlink_ops = 18221
        access_ops = 62178
        other_ops = 123112

 Server Status : OK
# cfsmgr -h systema -a statistics /accounts

 Counters for the filesystem /accounts:
        read_ops = 26836
        write_ops = 3773
        lookup_ops = 701764
        getattr_ops = 561806
        readlink_ops = 28712
        access_ops = 81173
        other_ops = 146263

 Server Status : OK
# cfsmgr -h systemc -a statistics /accounts

 Counters for the filesystem /accounts:
        read_ops = 18746
        write_ops = 13553
        lookup_ops = 475015
```

```
        getattr_ops = 280905
        readlink_ops = 24306
        access_ops = 84283
        other_ops =  103671

 Server Status : OK
# cfsmgr -h systemd -a statistics /accounts

 Counters for the filesystem /accounts:
        read_ops = 98468
        write_ops = 63773
        lookup_ops = 994437
        getattr_ops = 785618
        readlink_ops = 44324
        access_ops = 101821
        other_ops = 212331

 Server Status : OK
```

In this example, most of the read and write activity for /accounts is from member systemd, not from the member that is currently serving it, systemb. Assuming that systemd is physically connected to the storage for /accounts, systemd is a good choice as the CFS server for /accounts.

Determine whether systemd and the storage for /accounts are physically connected as follows:

1. Find out where /accounts is mounted. You can either look in /etc/fstab or use the mount command. If there are a large number of mounted file systems, you might want to use grep as follows:

   ```
   # mount | grep accounts
   accounts_dmn#accounts on /accounts type advfs (rw)
   ```

2. Look at the directory /etc/fdmns/accounts_dmn to learn the device where the AdvFS domain accounts_dmn is mounted as follows:

   ```
   # ls /etc/fdmns/accounts_dmn
   dsk6c
   ```

3. Enter the drdmgr command to learn the servers of dsk6 as follows:

   ```
   # drdmgr -a server dsk6
                      Device Name: dsk6
                      Device Type: Direct Access IO Disk
                    Device Status: OK
                Number of Servers: 4
                      Server Name: membera
                     Server State: Server
                      Server Name: memberb
                     Server State: Server
                      Server Name: memberc
                     Server State: Server
   ```

```
 Server Name: memberd
Server State: Server
```

Because dsk6 has multiple servers, it is on a shared bus. Because systemd is one of the servers, there is a physical connection.

4. Relocate the CFS server of /accounts to systemd as follows:

```
# cfsmgr -a server=systemd /accounts
```

Even in cases where the CFS statistics do not show an inordinate load imbalance, we recommend that you distribute the CFS servers among the available members that are connected to the shared bus. Doing so can improve overall cluster performance.

9.3.3.2 Automatically Distributing CFS Server Load

To automatically have a particular cluster member act as the CFS server for a file system or domain, you can place a script in /sbin/init.d that calls the cfsmgr command to relocate the server for the file system or domain to the desired cluster member.

For example, if you want cluster member alpha to serve the domain accounting, place the following cfsmgr command in a startup script:

```
# cfsmgr -a server=alpha -d accounting
```

Have the script look for successful relocation and retry the operation if it fails. The cfsmgr command returns a nonzero value on failure; however, it is not sufficient for the script to keep trying on a bad exit value. The relocation might have failed because a failover or relocation is already in progress.

On failure of the relocation, have the script search for one of the following messages:

```
Server Status : Failover/Relocation in Progress

Server Status : Cluster is busy, try later
```

If either of these messages occurs, have the script retry the relocation. On any other error, have the script print an appropriate message and exit.

9.3.3.3 Tuning the Block Transfer Size

During client-side reads and writes, CFS passes data in a predetermined block size. Generally, the larger the block size, the better the I/O performance.

There are two ways to control the CFS I/O blocksize:

- cfsiosize kernel attribute

The cfsiosize kernel attribute sets the CFS I/O blocksize for all file systems served by the cluster member where the attribute is set. If a file system relocates to another cluster member, due to either a failover or a planned relocation, the CFS transfer size stays the same. Changing the cfsiosize kernel attribute on a member after it is booted affects only file systems that are mounted after the change.

To change the default size for CFS I/O blocks clusterwide, set the cfsiosize kernel attribute on each cluster member.

You can set cfsiosize at boot time and at run time. The value must be between 8192 bytes (8K) and 131072 bytes (128K), inclusive.

To change the transfer size of a mounted file system, use cfsmgr FSBSIZE attribute, which is described next.

- FSBSIZE CFS attribute

  The FSBSIZE CFS attribute sets the I/O blocksize on a per-filesystem basis. To set FSBSIZE, use the cfsmgr command. The attribute can be set only for mounted file systems. You cannot set FSBSIZE on an AdvFS domain (the cfsmgr -d option).

  When you set FSBSIZE, the value is automatically rounded to the nearest page. For example:

```
# cfsmgr -a fsbsize=80000 /var

fsbsize for filesystem set to /var: 81920
```

  For more information, see cfsmgr(8).

Although a large block size generally yields better performance, there are special cases where doing CFS I/O in smaller block sizes can be advantageous. If reads and writes for a file system are small and random, then a large CFS I/O block size does not improve performance and the extra processing is wasted.

For example, if the I/O for a file system is 8K or less and totally random, then a value of 8 for FSBSIZE is appropriate for that file system.

The default value for FSBSIZE is determined by the value of the cfsiosize kernel attribute. To learn the current value of cfsiosize, use the sysconfig command. For example:

```
# sysconfig -q cfs cfsiosize
cfs:
cfsiosize = 65536
```

A file system where all the I/O is small in size but multiple threads are reading or writing the file system sequentially is not a candidate for a small value for FSBSIZE. Only when the I/O to a file system is both small and

random does it make sense to set FSBSIZE for that file system to a small value.

9.3.3.4 Changing the Number of Read-Ahead and Write-Behind Threads

When CFS detects sequential accesses to a file, it employs read-ahead threads to read the next I/O block size worth of data. CFS also employs write-behind threads to buffer the next block of data in anticipation that it too will be written to disk. Use the cfs_async_biod_threads kernel attribute to set the number of I/O threads that perform asynchronous read ahead and write behind. Read-ahead and write-behind threads apply only to reads and writes originating on CFS clients.

The default size for cfs_async_biod_threads is 32. In an environment where at one time you have more than 32 large files sequentially accessed, increasing cfs_async_biod_threads can improve CFS performance, particularly if the applications using the files can benefit from lower latencies.

The number of read-ahead and write-behind threads is tunable from 0 through 128. When not in use, the threads consume few system resources.

9.3.3.5 Taking Advantage of Direct I/O

When an application opens an AdvFS file with the O_DIRECTIO flag in the open system call, data I/O is direct to the storage; the system software does no data caching for the file at the file-system level. In a cluster, this arrangement supports concurrent direct I/O on the file from any member in the cluster. That is, regardless of which member originates the I/O request, I/O to a file does not go through the cluster interconnect to the CFS server. Database applications frequently use direct I/O in conjunction with raw asynchronous I/O (which is also supported in a cluster) to improve I/O performance.

The best performance on a file that is opened for direct I/O is achieved under the following conditions:

- A read from an existing location of the file
- A write to an existing location of the file
- When the size of the data being read or written is a multiple of the disk sector size, 512 bytes

The following conditions can result in less than optimal direct I/O performance:

- Operations that cause a metadata change to a file. These operations go across the cluster interconnect to the CFS server of the file system when

the application that is doing the direct I/O runs on a member other than the CFS server of the file system. Such operations include the following:

  - Any modification that fills a sparse hole in the file
  - Any modification that appends to the file
  - Any modification that truncates the file
  - Any read or write on a file that is less than 8K and consists solely of a fragment or any read/write to the fragment portion at the end of a larger file

- Any unaligned block read or write that is not to an existing location of the file. If a request does not begin or end on a block boundary, multiple I/Os are performed.
- When a file is open for direct I/O, any AdvFS migrate operation (such as `migrate`, `rmvol`, `defragment`, or `balance`) on the domain will block until the I/O that is in progress completes on all members. Conversely, direct I/O will block until any AdvFS migrate operation completes.

An application that uses direct I/O is responsible for managing its own caching. When performing multithreaded direct I/O on a single cluster member or multiple members, the application must also provide synchronization to ensure that, at any instant, only one thread is writing a sector while others are reading or writing.

For a discussion of direct I/O programming issues, see the chapter on optimizing techniques in the Tru64 UNIX *Programmer's Guide*.

9.3.3.5.1 Differences Between Cluster and Standalone AdvFS Direct I/O

The following list presents direct I/O behavior in a cluster that differs from that in a standalone system:

- Performing any migrate operation on a file that is already opened for direct I/O blocks until the I/O that is in progress completes on all members. Subsequent I/O will block until the migrate operation completes.
- AdvFS in a standalone system provides a guarantee at the sector level that, if multiple threads attempt to write to the same sector in a file, one will complete first and then the other. This guarantee is not provided in a cluster.

9.3.3.5.2 Cloning a Fileset With Files Open in Direct I/O Mode

As described in Section 9.3.3.5, when an application opens a file with the `O_DIRECTIO` flag in the `open` system call, I/O to the file does not go through the cluster interconnect to the CFS server. However, if you clone a fileset

that has files open in Direct I/O mode, the I/O does not follow this model and might cause considerable performance degradation. (Read performance is not impacted by the cloning.)

The `clonefset` utility, which is described in the `clonefset(8)` reference page, creates a read-only copy, called a **clone fileset**, of an AdvFS fileset. A clone fileset is a read-only snapshot of fileset data structures (metadata). That is, when you clone a fileset, the utility copies only the structure of the original fileset, not its data. If you then modify files in the original fileset, every write to the fileset causes a synchronous copy-on-write of the original data to the clone if the original data has not already been copied. In this way, the clone fileset contents remain the same as when you first created it.

If the fileset has files open in Direct I/O mode, when you modify a file AdvFS copies the original data to the clone storage. AdvFS does not send this copy operation over the cluster interconnect. However, CFS does send the write operation for the changed data in the fileset over the interconnect to the CFS server unless the application using Direct I/O mode happens to be running on the CFS server. Sending the write operation over the cluster interconnect negates the advantages of opening the file in Direct I/O mode.

To retain the benefits of Direct I/O mode, remove the clone as soon as the backup operation is complete so that writes are again written directly to storage and are not sent over the cluster interconnect.

9.3.3.5.3 Gathering Statistics on Direct I/O

If the performance gain for an application that uses direct I/O is less than you expected, you can use the `cfsstat` command to examine per-node global direct I/O statistics.

Use `cfsstat` to look at the global direct I/O statistics without the application running. Then execute the application and examine the statistics again to determine whether the paths that do not optimize direct I/O behavior were being executed.

The following example shows how to use the `cfsstat` command to get direct I/O statistics:

```
# cfsstat directio
Concurrent Directio Stats:
     160 direct i/o reads
     160 direct i/o writes
       0 aio raw reads
       0 aio raw writes
       0 unaligned block reads
       0 fragment reads
       0 zero-fill (hole) reads
     160 file-extending writes
```

```
0 unaligned block  writes
0 hole writes
0 fragment writes
0 truncates
```

The individual statistics have the following meanings:

- `direct i/o reads`

  The number of normal direct I/O read requests. These read requests were processed on the member that issued the request and were not sent to the AdvFS layer on the CFS server.

- `direct i/o writes`

  The number of normal direct I/O write requests processed. These write requests were processed on the member that issued the request and were not sent to the AdvFS layer on the CFS server.

- `aio raw reads`

  The number of normal direct I/O asynchronous read requests. These read requests were processed on the member that issued the request and were not sent to the AdvFS layer on the CFS server.

- `aio raw writes`

  The number of normal direct I/O asynchronous write requests. These read requests were processed on the member that issued the request and were not sent to the AdvFS layer on the CFS server.

- `unaligned block reads`

  The number of reads that were not a multiple of a disk sector size (currently 512 bytes). This count will be incremented for requests that do not start at a sector boundary or do not end on a sector boundary. An unaligned block read operation results in a read for the sector and a copyout of the user data requested from the proper location of the sector.

  If the I/O request encompasses an existing location of the file and does not encompass a fragment, this operation does not get sent to the CFS server.

- `fragment reads`

  The number of read requests that needed to be sent to the CFS server because the request was for a portion of the file that contains a fragment.

  A file that is less than 140K might contain a fragment at the end that is not a multiple of 8K. Also small files less than 8K in size may consist solely of a fragment.

  To ensure that a file of less than 8K does not consist of a fragment, always open the file only for direct I/O. Otherwise, on the close of a normal open, a fragment will be created for the file.

- `zero-fill (hole) reads`

  The number of reads that occurred to sparse areas of the files that were opened by direct I/O. This request is not sent to the CFS server.

- `file-extending writes`

  The number of write requests that were sent to the CFS server because they appended data to the file.

- `unaligned block writes`

  The number of writes that were not a multiple of a disk sector size (currently 512 bytes). This count will be incremented for requests that do not start at a sector boundary or do not end on a sector boundary. An unaligned block write operation results in a read for the sector, a copyin of the user data that is destined for a portion of the block, and a subsequent write of the merged data. These operations do not get sent to the CFS server.

  If the I/O request encompasses an existing location of the file and does not encompass a fragment, this operation does not get sent to the CFS server.

- `hole writes`

  The number of write requests to an area that encompasses a sparse hole in the file that needed to be sent to AdvFS on the CFS server.

- `fragment writes`

  The number of write requests that needed to be sent to the CFS server because the request was for a portion of the file that contains a fragment.

  A file that is less than 140K might contain a fragment at the end that is not a multiple of 8K. Also small files less than 8K in size may consist solely of a fragment.

  To ensure that a file of less than 8K does not consist of a fragment, always open the file only for direct I/O. Otherwise, on the close of a normal open, a fragment will be created for the file.

- `truncates`

  The number of truncate requests for direct I/O opened files. This request does get sent to the CFS server.

9.3.3.6 Adjusting CFS Memory Usage

In situations where one cluster member is the CFS server for a large number of file systems, the client members may cache a great many vnodes from the served file systems. For each cached vnode on a client, even vnodes that are not actively used, the CFS server must allocate 800 bytes of system memory for the CFS token structure that is needed to track the file at the CFS layer.

In addition to this, the CFS token structures typically require corresponding AdvFS access structures and vnodes, resulting in a near-doubling of the amount of memory that is used.

By default, each client can use up to 4 percent of memory to cache vnodes. When multiple clients fill up their caches with vnodes from a CFS server, system memory on the server can become overtaxed, causing it to hang.

The svrcfstok_max_percent kernel attribute is designed to prevent such system hangs. The attribute sets an upper limit on the amount of memory that is allocated by the CFS server to track vnode caching on clients. The default value is 25 percent. The memory is used only if the server load requires it. It is not allocated up front.

After the svrcfstok_max_percent limit is reached on the server, an application accessing files that are served by the member gets an EMFILE error. Applications that use perror() to check errno will return the message too many open files to the standard error stream, stderr, the controlling tty or log file used by the applications. Although you see EMFILE error messages, no cached data is lost.

If applications start getting EMFILE errors, follow these steps:

1. Determine whether the CFS client is out of vnodes, as follows:

   a. Get the current value of the max_vnodes kernel attribute:

      ```
      # sysconfig -q vfs max_vnodes
      ```

   b. Use dbx to get the values of total_vnodes and free_vnodes:

      ```
      # dbx -k /vmunix /dev/mem
      dbx version 5.0
      Type 'help' for help.
      (dbx)pd total_vnodes
      total_vnodes_value
      ```

      Get the value for max_vnodes:

      ```
      (dbx)pd max_vnodes
      max_vnodes_value
      ```

      If total_vnodes equals max_vnodes and free_vnodes equals 0, then that member is out of vnodes. In this case, you can increase the value of the max_vnodes kernel attribute. You can use the sysconfig command to change max_vnodes on a running member. For example, to set the maximum number of vnodes to 20000, enter the following:

      ```
      # sysconfig -r vfs max_vnodes=20000
      ```

2. If the CFS client is not out of vnodes, then determine whether the CFS server has used all the memory that is available for token structures (svrcfstok_max_percent), as follows:

   a. Log on to the CFS server.

   b. Start the dbx debugger and get the current value for svrtok_active_svrcfstok:

   ```
   # dbx -k /vmunix /dev/mem
   dbx version 5.0
   Type 'help' for help.
   (dbx)pd svrtok_active_svrcfstok
   active_svrcfstok_value
   ```

   c. Get the value for cfs_max_svrcfstok:

   ```
   (dbx)pd cfs_max_svrcfstok
   max_svrcfstok_value
   ```

   If svrtok_active_svrcfstok is equal to or greater than cfs_max_svrcfstok, then the CFS server has used all the memory that is available for token structures.

   In this case, the best solution to make the file systems usable again is to relocate some of the file systems to other cluster members. If that is not possible, then the following solutions are acceptable:

   - Increase the value of cfs_max_svrcfstok.

     You cannot change cfs_max_svrcfstok with the sysconfig command. However, you can use the dbx assign command to change the value of cfs_max_svrcfstok in the running kernel. For example, to set the maximum number of CFS server token structures to 80000, enter the following command:

     ```
     (dbx)assign cfs_max_svrcfstok=80000
     ```

     Values you assign with the dbx assign command are lost when the system is rebooted.

   - Increase the amount of memory that is available for token structures on the CFS server.

     This option is undesirable on systems with small amounts of memory.

     To increase svrcfstok_max_percent, log on to the server and run the dxkerneltuner command. On the main window, select the cfs kernel subsystem. On the cfs window, enter an appropriate value for svrcfstok_max_percent. This change will not take effect until the cluster member is rebooted.

Typically, when a CFS server reaches the `svrcfstok_max_percent` limit, relocate some of the CFS file systems so that the burden of serving the file systems is shared among cluster members. You can use startup scripts to run the `cfsmgr` and automatically relocate file systems around the cluster at member startup.

Setting `svrcfstok_max_percent` below the default is recommended only on smaller memory systems that run out of memory because 25 percent default value is too high.

9.3.3.7 Using Memory Mapped Files

Using memory mapping to share a file across the cluster for anything other than read-only access can negatively affect performance. CFS I/O to a file does not perform well if multiple members are simultaneously modifying the data. This situation forces premature cache flushes to ensure that all nodes have the same view of the data at all times.

9.3.3.8 Avoid Full File Systems

If free space in a file system is less than 50 MB or less than 10 percent of the file system's size, whichever is smaller, then write performance to the file system from CFS clients suffers. This is because all writes to nearly full file systems are sent immediately to the server to guarantee correct ENOSPC semantics.

9.3.3.9 Other Strategies

The following measures can improve CFS performance:

- Ensure that the cluster members have sufficient system memory.
- In general, sharing a file for read/write access across cluster members may negatively affect performance because of all of the cache invalidations. CFS I/O to a file does not perform well if multiple members are simultaneously modifying the data. This situation forces premature cache flushes to ensure that all nodes have the same view of the data at all times.
- If a distributed application does reads and writes on separate members, try locating the CFS servers for the application to the member performing writes. Writes are more sensitive to remote I/O than reads.
- If multiple applications access different sets of data in a single AdvFS domain, consider splitting the data into multiple domains. This arrangement allows you to spread the load to more than a single CFS server. It also presents the opportunity to colocate each application with the CFS server for that application's data without loading everything on a single member.

9.3.4 MFS and UFS File Systems Supported

TruCluster Server Version 5.1A includes read/write support for Memory File System (MFS) and UNIX File System (UFS) file systems.

When you mount a UFS file system in a cluster for read/write access, or when you mount an MFS file system in a cluster for read-only or read/write access, the `mount` command `server_only` argument is used by default. These file systems are treated as partitioned file systems, as described in Section 9.3.5. That is, the file system is accessible for both read-only and read/write access only by the member that mounts it. Other cluster members cannot read from, or write to, the MFS or UFS file system. There is no remote access; there is no failover.

If you want to mount a UFS file system for read-only access by all cluster members, you must explicitly mount it read-only.

9.3.5 Partitioning File Systems

CFS makes all files accessible to all cluster members. Each cluster member has the same access to a file, whether the file is stored on a device that is connected to all cluster members or on a device that is private to a single member. However, CFS does make it possible to mount an AdvFS file system so that it is accessible to only a single cluster member. This is referred to as **file system partitioning**.

The Available Server Environment (ASE), which is an earlier version of the TruCluster Server product, offered functionality like that of file system partitioning. File partitioning is provided in TruCluster Server as of Version 5.1 to ease migration from ASE. File system partitioning in TruCluster Server is not intended as a general purpose method for restricting file system access to a single member.

To mount a partitioned file system, log on to the member that you want to give exclusive access to the file system. Run the `mount` command with the `server_only` option. This mounts the file system on the member where you execute the `mount` command and gives that member exclusive access to the file system. Although only the mounting member has access to the file system, all members, cluster-wide, can see the file system mount.

The `server_only` option can be applied only to AdvFS, MFS, and UFS file systems.

Partitioned file systems are subject to the following limitations:

- No file systems can be mounted under a partitioned file system

  You cannot mount a file system, partitioned or otherwise, under a partitioned file system.

- No failover via CFS

  If the cluster member serving a partitioned file system fails, the file system is unmounted. You must remount the file system on another cluster member.

  You can work around this by putting the application that uses the partitioned file system under the control of CAA. Because the application must run on the member where the partitioned file system is mounted, if the member fails, both the file system and application fail. An application that is under control of CAA will fail over to a running cluster member. You can write the application's CAA action script to mount the partitioned file system on the new member.

- NFS export

  The best way to export a partitioned file system is to create a single node cluster alias for the node serving the partitioned file system and include that alias in the `/etc/exports.aliases` file. See Section 3.13 for additional information on how to best utilize the `/etc/exports.aliases` file.

  If you use the default cluster alias to NFS-mount file systems that the cluster serves, some NFS requests will be directed to a member that does not have access to the file system and will fail.

  Another way to export a partitioned file system is to assign the member that serves the partitioned file system the highest cluster-alias selection priority (`selp`) in the cluster. If you do this, the member will serve all NFS connection requests. However, the member will also have to handle all network traffic of any type that is directed to the cluster. This is not likely to be acceptable in most environments.

  For more information about distributing connection requests, see Section 3.9.

- No mixing partitioned and conventional filesets in the same domain

  The `server_only` option applies to all file systems in a domain. The type of the first fileset mounted determines the type for all filesets in the domain:

  – If a fileset is mounted without the `server_only` option, then attempts to mount another fileset in the domain `server_only` will fail.

  – If a fileset in a domain is mounted `server_only`, then all subsequent fileset mounts in that domain must be `server_only`.

- No manual relocation

  To move a partitioned file system to a different CFS server, you must unmount the file system and then remount it on the target member.

At the same time, you will need to move applications that use the file system.

- No mount updates with `server_only` option

  After you mount a file system normally, you cannot use the `mount -u` command with the `server_only` option on the file system. For example, if `file_system` has already been mounted without use of the `server_only` flag, the following command fails:

  ```
  # mount -u -o server_only file_system
  ```

9.3.6 Block Devices and Cache Coherency

A single block device can have multiple aliases. In this situation, multiple block device special files in the file system namespace will contain the same `dev_t`. These aliases can potentially be located across multiple domains or file systems in the namespace.

On a standalone system, cache coherency is guaranteed among all opens of the common underlying block device regardless of which alias was used on the `open()` call for the device. In a cluster, however, cache coherency can be obtained only among all block device file aliases that reside on the same domain or file system.

For example, if cluster member `mutt` serves a domain with a block device file and member `jeff` serves a domain with another block device file with the same `dev_t`, then cache coherency is not provided if I/O is performed simultaneously through these two aliases.

9.4 Managing the Device Request Dispatcher

The device request dispatcher subsystem makes physical disk and tape storage transparently available to all cluster members, regardless of where the storage is physically located in the cluster. When an application requests access to a file, CFS passes the request to AdvFS, which then passes it to the device request dispatcher. In the file system hierarchy, the device request dispatcher sits right above the device drivers.

The primary tool for managing the device request dispatcher is the `drdmgr` command. A number of examples of using the command appear in this section. For more information, see `drdmgr`(8).

9.4.1 Direct-Access I/O and Single-Server Devices

The device request dispatcher follows a client/server model; members serve devices, such as disks, tapes, and CD-ROM drives.

Devices in a cluster are either **direct-access I/O devices** or **single-server devices**. A direct-access I/O device supports simultaneous access from multiple cluster members. A single-server device supports access from only a single member.

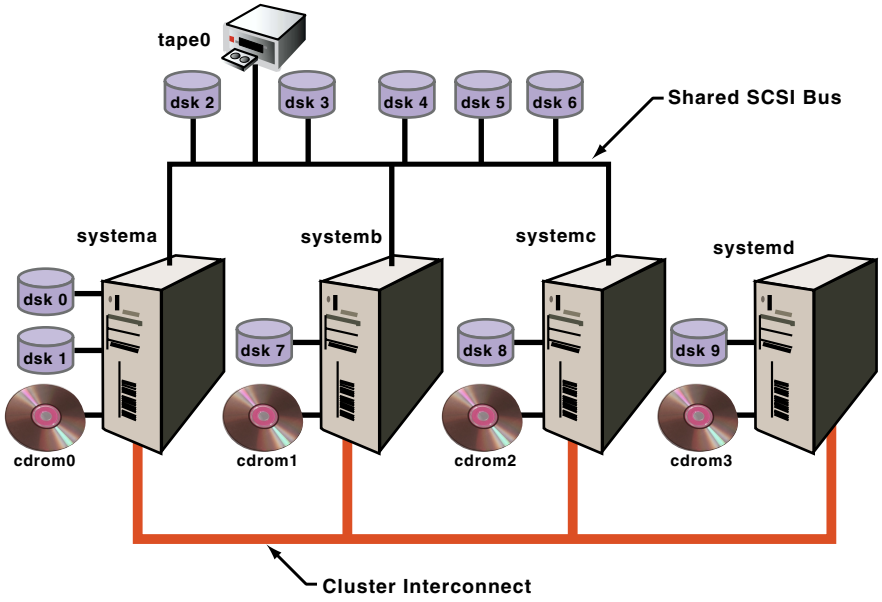
Direct-access I/O devices on a shared bus are served by all cluster members on that bus. A single-server device, whether on a shared bus or directly connected to a cluster member, is served by a single member. All other members access the served device through the serving member. Note that direct-access I/O devices are part of the device request dispatcher subsystem, and have nothing to do with direct I/O (opening a file with the `O_DIRECTIO` flag to the `open` system call), which is handled by CFS. See Section 9.3.3.5 for information about direct I/O and CFS.

Typically, disks on a shared bus are direct-access I/O devices, but in certain circumstances, some disks on a shared bus can be single-server. The exceptions occur when you add an RZ26, RZ28, RZ29, or RZ1CB-CA disk to an established cluster. Initially, such devices are single-server devices. See Section 9.4.1.1 for more information. Tape devices are always single-server devices.

Although single-server disks on a shared bus are supported, they are significantly slower when used as member boot disks or swap files, or for the retrieval of core dumps. We recommend that you use direct-access I/O disks in these situations.

Figure 9–3 shows a four-node cluster with five disks and a tape drive on the shared bus. Note that `SystemD` is not on the shared bus. Its access to cluster storage is routed through the Memory Channel cluster interconnect.

Figure 9–3: Four Node Cluster



Disks on the shared bus are served by all the cluster members on the bus. You can confirm this by looking for the device request dispatcher server of `dsk3` as follows:

```
# drdmgr -a server dsk3
                  Device Name: dsk3
                  Device Type: Direct Access IO Disk
                Device Status: OK
             Number of Servers: 3
                  Server Name: systema
                 Server State: Server
                  Server Name: systemb
                 Server State: Server
                  Server Name: systemc
                 Server State: Server
```

From the `View` line in the preceding output, you can see that the `drdmgr` command was executed on `systemc`.

Because `dsk3` is a direct-access I/O device on the shared bus, all three systems on the bus serve it. This means that, when any member on the shared bus accesses the disk, the access is directly from the member to the device.

Disks on private buses are served by the system that they are local to. For example, the server of `dsk7` is `systemb`:

```
# drdmgr -a server dsk7
                  Device Name: dsk7
                  Device Type: Direct Access IO Disk
                Device Status: OK
             Number of Servers: 1
                  Server Name: systemb
                 Server State: Server
```

Tape drives are always single-server. Because `tape0` is on a shared bus, any member on that bus can act as its server. When the cluster is started, the first member up that has access to the tape drive becomes the server for the tape drive.

The numbering of disks indicates that when the cluster booted, `systema` came up first. It detected its private disks first and labeled them, then it detected the disks on the shared bus and labeled them. Because `systema` came up first, it is also the server for `tape0`. To confirm this, enter the following command:

```
# drdmgr -a server tape0
                  Device Name: tape0
                  Device Type: Served Tape
                Device Status: OK
             Number of Servers: 1
                  Server Name: systema
                 Server State: Server
```

To change `tape0`'s server to `systemc`, enter the `drdmgr` command as follows:

```
# drdmgr -a server=systemc /dev/tape/tape0
```

For any single-server device, the serving member is also the access node. The following command confirms this:

```
# drdmgr -a accessnode tape0
                  Device Name: tape0
              Access Node Name: systemc
```

Unlike the device request dispatcher `SERVER` attribute, which for a given device is the same on all cluster members, the value of the `ACCESSNODE` attribute is specific to a cluster member.

Any system on a shared bus is always its own access node for the direct-access I/O devices on the same shared bus.

Because `systemd` is not on the shared bus, for each direct-access I/O device on the shared bus you can specify the access node to be used by `systemd` when it accesses the device. The access node must be one of the members on the shared bus.

The result of the following command is that `systemc` handles all device request dispatcher activity between `systemd` and `dsk3`:

```
# drdmgr -h systemd -a accessnode=systemc dsk3
```

9.4.1.1 Devices Supporting Direct-Access I/O

RAID-fronted disks are direct-access I/O capable. The following are Redundant Array of Independent Disks (RAID) controllers:

- HSZ40
- HSZ50
- HSZ70
- HSZ80
- HSG60
- HSG80

Any RZ26, RZ28, RZ29, and RZ1CB-CA disks already installed in a system at the time the system becomes a cluster member, either through the `clu_create` or `clu_add_member` command, are automatically enabled as direct-access I/O disks. To later add one of these disks as a direct-access I/O disk, you must use the procedure in Section 9.2.3.

9.4.1.2 Replacing RZ26, RZ28, RZ29, or RZ1CB-CA as Direct-Access I/O Disks

If you replace an RZ26, RZ28, RZ29, or RZ1CB-CA direct-access I/O disk with a disk of the same type (for example, replace an RZ28-VA with another RZ28-VA), follow these steps to make the new disk a direct-access I/O disk:

1. Physically install the disk in the bus.
2. On each cluster member, enter the `hwmgr` command to scan for the new disk as follows:

   ```
   # hwmgr -scan comp -cat scsi_bus
   ```

   Allow a minute or two for the scans to complete.
3. If you want the new disk to have the same device name as the disk it replaced, use the `hwmgr -redirect scsi` command. For details, see `hwmgr`(8) and the section on replacing a failed SCSI device in the Tru64 UNIX *System Administration* manual.
4. On each cluster member, enter the `clu_disk_install` command:

   ```
   # clu_disk_install
   ```

Note

If the cluster has a large number of storage devices, the `clu_disk_install` command can take several minutes to complete.

9.4.1.3 HSZ Hardware Supported on Shared Buses

For a list of hardware that is supported on shared buses, see the TruCluster Server Version 5.1A *Software Product Description*.

If you try to use an HSZ40A or an HSZ that does not have the proper firmware revision on a shared bus, the cluster might hang when there are multiple simultaneous attempts to access the HSZ.

9.5 Managing AdvFS in a Cluster

For the most part, the Advanced File System (AdvFS) on a cluster is like that on a standalone system. However, there are some cluster-specific considerations, which are described in this section:

- Integrating AdvFS files from a newly added member (Section 9.5.1)
- Using the `addvol` and `rmvol` commands (Section 9.5.4)
- Using user and group file system quotas (Section 9.5.5)
- Understanding storage connectivity and AdvFS volumes (Section 9.5.6)

9.5.1 Integrating AdvFS Files from a Newly Added Member

Suppose that you add a new member to the cluster and that new member has AdvFS volumes and filesets from when it ran as a standalone system. To integrate these volumes and filesets into the cluster, you need to do the following:

1. Modify the `/etc/fstab` file listing the `domains#filesets` that you want to integrate into the cluster.
2. Make the new domains known to the cluster, either by manually entering the domain information into `/etc/fdmns` or by running the `advscan` command.

For information on the `advscan` command, see `advscan`(8). For examples of reconstructing `/etc/fdmns`, see the section on restoring an AdvFS file system in the Tru64 UNIX *AdvFS Administration* manual.

9.5.2 Create Only One Fileset in Cluster Root Domain

The root domain, `cluster_root`, must contain only a single fileset. If you create more than one fileset in `cluster_root` (you are not prevented from doing so), it can lead to a panic if the `cluster_root` domain needs to fail over.

As an example of when this situation might occur, consider cloned filesets. As described in `advfs(4)`, a clone fileset is a read-only copy of an existing fileset, which you can mount as you do other filesets. If you create a clone of the clusterwide root (/) and mount it, the cloned fileset is added to the `cluster_root` domain. If the `cluster_root` domain has to fail over while the cloned fileset is mounted, the cluster will panic.

Note

If you make backups of the clusterwide root from a cloned fileset, minimize the amount of time during which the clone is mounted. Mount the cloned fileset, perform the backup, and unmount the clone as quickly as possible.

9.5.3 Do Not Add a Volume to a Member's Root Domain

You cannot use the `addvol` command to add volumes to a member's root domain (`root`*`memberID`*`_domain#root`). Instead, you must delete the member from the cluster, use `diskconfig` or SysMan to configure the disk appropriately, and then add the member back into the cluster. For the configuration requirements for a member boot disk, see the *Cluster Installation* manual.

9.5.4 Using the addvol and rmvol Commands in a Cluster

You can manage AdvFS domains from any cluster member, regardless of whether the domains are mounted on the local member or a remote member. However, when you use the `addvol` or `rmvol` command from a member that is not the CFS server for the domain you are managing, the commands use `rsh` to execute remotely on the member that is the CFS server for the domain. This has the following consequences:

- If `addvol` or `rmvol` is entered from a member that is not the server of the domain, and if member that is serving the domain fails, the command can hang on the system where it was executed until TCP times out, which can take as long as an hour.

If this situation occurs, you can kill the command and its associated rsh processes and repeat the command as follows:

1. Get the process identifiers (PIDs) with the ps command and pipe the output through more, searching for addvol or rmvol, whichever is appropriate. For example:

```
# ps -el | more +/addvol
80808001 I  + 0 16253977 16253835  0.0  44   0 451700 424K wait     pts/0
 0:00.09 addvol
80808001 I  + 0 16253980 16253977  0.0  44   0 1e6200 224K event    pts/0
 0:00.02 rsh
  808001 I  + 0 16253981 16253980  0.0  44   0 a82200  56K tty      pts/0
 0:00.00 rsh
```

2. Use the process IDs (in this example, PIDs 16253977, 16253980, and 16253981) and parent process IDs (PPIDs 16253977 and 16253980) to confirm the association between the addvol or rmvol and the rsh processes. Note that two rsh processes are associated with the addvol process. All three processes must be killed.
3. Kill the appropriate processes. In this example:

```
# kill -9 16253977 16253980 16253981
```

4. Reenter the addvol or rmvol command. In the case of addvol, you must use the -F option because the hung addvol command might have already changed the disk label type to AdvFS.

Alternately, before using either the addvol or rmvol command on a domain, you can do the following:

1. Use the cfsmgr command to learn the name of the CFS server of the domain:

   ```
   # cfsmgr -d domain_name
   ```

   Or, enter only the command cfsmgr and get a list of the servers of all CFS domains.
2. Log in to the serving member.
3. Use the addvol or rmvol command.

- If the CFS server for the volume fails over to another member in the middle of an addvol or rmvol operation, you may need to reenter the command because the new server undoes any partial operation. The command does not return a message indicating that the server failed, and the operation must be repeated.

  We recommend that you enter a showfdmn command for the target domain of an addvol or rmvol command after the command returns.

The rmvol and addvol commands use rsh when the member where the commands are executed is not the server of the domain. For rsh to function,

the default cluster alias must appear in the /.rhosts file. The entry for the cluster alias in /.rhosts can take the form of the fully-qualified hostname or the unqualified hostname. Although the plus sign (+) can appear in place of the hostname, allowing all hosts access, this is not recommended for security reasons.

The clu_create command automatically places the cluster alias in /.rhosts, so rsh should work without your intervention. If the rmvol or addvol command fails because of rsh failure, the following message is returned:

```
rsh failure, check that the /.rhosts file allows cluster alias access.
```

9.5.5 User and Group File System Quotas Are Supported

TruCluster Server Version 5.1A includes quota support that allows you to limit both the number of files and the total amount of disk space that are allocated in an AdvFS filesystem on behalf of a given user or group.

Quota support in a TruCluster Server environment is similar to quota support in the Tru64 UNIX base system, with the following exceptions:

- Hard limits are not absolute because the Cluster File System (CFS) makes certain assumptions about how and when cached data is written.
- Soft limits and grace periods are supported, but there is no guarantee that a user will get a message when the soft limit is exceeded from a client node, or that such a message will arrive in a timely manner.
- The quota commands are effective clusterwide. However, you must edit the /sys/conf/*NAME* system configuration file on each cluster member to configure the system to include the quota subsystem. If you do not perform this step on a cluster member, quotas are enabled on that member but you cannot enter quota commands from that member.
- TruCluster Server supports quotas only for AdvFS filesystems.
- Users and groups are managed clusterwide. Therefore, user and group quotas are also managed clusterwide.

This section describes information that is unique to managing disk quotas in a TruCluster Server environment. For general information about managing quotas, see the Tru64 UNIX *System Administration* guide.

9.5.5.1 Quota Hard Limits

In a Tru64 UNIX system, a hard limit places an absolute upper boundary on the number of files or amount of disk space that a given user or group can allocate on a given filesystem. When a hard limit is reached, disk space allocations or file creations are not allowed. System calls that would cause the hard limit to be exceeded fail with a quota violation.

In a TruCluster Server environment, hard limits for the number of files are enforced as they are in a standalone Tru64 UNIX system.

However, hard limits on the total amount of disk space are not as rigidly enforced. For performance reasons, CFS allows client nodes to cache a configurable amount of data for a given user or group without any communication with the member serving that data. After the data is cached on behalf of a given write operation and the write operation returns to the caller, CFS guarantees that, barring a failure of the client node, the cached data will eventually be written to disk at the server.

Writing the cached data takes precedence over strictly enforcing the disk quota. If and when a quota violation occurs, the data in the cache is written to disk regardless of the violation. Subsequent writes by this group or user are not cached until the quota violation is corrected.

Because additional data is not written to the cache while quota violations are being generated, the hard limit is never exceeded by more than the sum of `quota_excess_blocks` on all cluster members. The actual disk space quota for a user or group is therefore determined by the hard limit plus the sum `quota_excess_blocks` on all cluster members.

The amount of data that a given user or group is allowed to cache is determined by the `quota_excess_blocks` value, which is located in the member-specific `etc/sysconfigtab` file. The `quota_excess_blocks` value is expressed in units of 1024-byte blocks and the default value of 1024 represents 1 MB of disk space. The value of `quota_excess_blocks` does not have to be the same on all cluster members. You might use a larger `quota_excess_blocks` value on cluster members on which you expect most of the data to be generated, and accept the default value for `quota_excess_blocks` on other cluster members.

9.5.5.2 Setting the `quota_excess_blocks` Value

The value for `quota_excess_blocks` is maintained in the `/etc/sysconfigtab` file in the `cfs` stanza.

Avoid making manual changes to this file. Instead, use the `sysconfigdb` command to make changes. This utility automatically makes any changes available to the kernel and preserves the structure of the file so that future upgrades merge in correctly.

Performance for a given user or group can be affected by `quota_excess_blocks`. If this value is set too low, CFS cannot use the cache efficiently. Setting `quota_excess_blocks` to less than 64K will have a severe performance impact. Conversely, setting `quota_excess_blocks` too high increases the actual amount of disk space that a user or group can consume.

We recommend accepting the quota_excess_blocks default of 1 MB, or increasing it as much as is considered practical given its effect of raising the potential upper limit on disk block usage. When determining how to set this value, consider that the worst-case upper boundary is determined as follows:

```
(admin specified hard limit) +
  (sum of "quota_excess_blocks" on each client node)
```

CFS makes a significant effort to minimize the amount by which the hard quota limit is exceeded, and it is very unlikely that you would reach the worst-case upper boundary.

9.5.6 Storage Connectivity and AdvFS Volumes

All volumes in an AdvFS domain must have the same connectivity if failover capability is desired. Volumes have the same connectivity when either one of the following conditions is true:

- All volumes in the AdvFS domain are on the same shared SCSI bus.
- Volumes in the AdvFS domain are on different shared SCSI buses, but all of those buses are connected to the same cluster members.

The drdmgr and hwmgr commands can give you information about which systems serve which disks. To get a graphical display of the cluster hardware configuration, including active members, buses, storage devices, and their connections, use the sms command to invoke the graphical interface for the SysMan Station, and then select Hardware from the Views menu.

9.6 Considerations When Creating New File Systems

Most aspects of creating new file systems are the same in a cluster and a standalone environment. The Tru64 UNIX *AdvFS Administration* manual presents an extensive description of how to create AdvFS file systems in a standalone environment.

For information about adding disks to the cluster, see Section 9.2.3.

The following are important cluster-specific considerations for creating new file systems:

- To ensure the highest availability, all disks that are used for volumes in an AdvFS domain should have the same connectivity.

  We recommend that all LSM volumes that are placed into an AdvFS domain share the same connectivity. See Section 10.2 for more on LSM volumes and connectivity.

- When you determine whether a disk is in use, make sure it is not used as any of the following:
  - The cluster quorum disk

Do not use any of the partitions on a quorum disk for data.

– The clusterwide root file system, the clusterwide `/var` file system, or the clusterwide `/usr` file system

– A member's boot disk

Do not put any data on a member's boot disk. See Section 11.1.4 for a description of the member boot disk and how to configure one.

- There is a single `/etc/fstab` file for all members of a cluster.

9.6.1 Verifying Disk Connectivity

To ensure the highest availability, make sure that all disks that are used for volumes in an AdvFS domain have the same connectivity.

Disks have the same connectivity when either one of the following conditions is true:

- All disks that are used for volumes in the AdvFS domain are on the same shared SCSI bus.
- Disks that are used for volumes in the AdvFS domain are on different shared SCSI buses, but all of those buses are connected to the same cluster members.

The easiest way to verify disk connectivity is to use the `sms` command to invoke the graphical interface for the SysMan Station, and then select Hardware from the Views menu.

For example, in Figure 9–1, the SCSI bus that is connected to the `pza0`s is shared by all three cluster members. All disks on that base have the same connectivity.

You can also use the `hwmgr` command to view all the devices on the cluster and then pick out those disks that show up multiple times because they are connected to several members. For example:

```
# hwmgr -view devices -cluster
HWID: Device Name          Mfg     Model            Hostname   Location
------------------------------------------------------------------------------
  3: kevm                                          pepicelli
 28: /dev/disk/floppy0c           3.5in floppy     pepicelli  fdi0-unit-0
 40: /dev/disk/dsk0c     DEC     RZ28M     (C) DEC pepicelli  bus-0-targ-0-lun-0
 41: /dev/disk/dsk1c     DEC     RZ28L-AS  (C) DEC pepicelli  bus-0-targ-1-lun-0
 42: /dev/disk/dsk2c     DEC     RZ28      (C) DEC pepicelli  bus-0-targ-2-lun-0
 43: /dev/disk/cdrom0c   DEC     RRD46     (C) DEC pepicelli  bus-0-targ-6-lun-0
 44: /dev/disk/dsk13c    DEC     RZ28M     (C) DEC pepicelli  bus-1-targ-1-lun-0
 44: /dev/disk/dsk13c    DEC     RZ28M     (C) DEC polishham  bus-1-targ-1-lun-0
 44: /dev/disk/dsk13c    DEC     RZ28M     (C) DEC provolone  bus-1-targ-1-lun-0
 45: /dev/disk/dsk14c    DEC     RZ28L-AS  (C) DEC pepicelli  bus-1-targ-2-lun-0
 45: /dev/disk/dsk14c    DEC     RZ28L-AS  (C) DEC polishham  bus-1-targ-2-lun-0
 45: /dev/disk/dsk14c    DEC     RZ28L-AS  (C) DEC provolone  bus-1-targ-2-lun-0
 46: /dev/disk/dsk15c    DEC     RZ29B     (C) DEC pepicelli  bus-1-targ-3-lun-0
 46: /dev/disk/dsk15c    DEC     RZ29B     (C) DEC polishham  bus-1-targ-3-lun-0
```

```
46: /dev/disk/dsk15c    DEC     RZ29B    (C) DEC provolone  bus-1-targ-3-lun-0
       .
       .
       .
```

In this partial output, dsk0, dsk1, and dsk2 are private disks that are connected to pepicelli's local bus. None of these are appropriate for a file system that needs failover capability, and they are not good choices for Logical Storage Manager (LSM) volumes.

dsk13 (HWID 44), dsk14 (HWID 45), and dsk15 (HWID 46) are connected to pepicelli, polishham, and provolone. These three disks all have the same connectivity.

9.6.2 Looking for Available Disks

When you want to determine whether disks are already in use, look for the quorum disk, disks containing the clusterwide file systems, and member boot disks and swap areas.

9.6.2.1 Looking for the Location of the Quorum Disk

You can learn the location of the quorum disk by using the clu_quorum command. In the following example, the partial output for the command shows that dsk10 is the cluster quorum disk:

```
# clu_quorum
 Cluster Quorum Data for: deli as of Wed Apr 25 09:27:36 EDT 2001

Cluster Common Quorum Data
Quorum disk:   dsk10h
        .
        .
        .
```

You can also use the disklabel command to look for a quorum disk. All partitions in a quorum disk should be unused, except for the h partition, which has fstype cnx.

9.6.2.2 Looking for the Location of Member Boot Disks and Clusterwide AdvFS File Systems

To learn the locations of member boot disks and clusterwide AdvFS file systems, look for the file domain entries in the /etc/fdmns directory. You can use the ls command for this. For example:

```
# ls /etc/fdmns/*
/etc/fdmns/cluster_root:
dsk3c

/etc/fdmns/cluster_usr:
```

```
dsk5c

/etc/fdmns/cluster_var:
dsk6c

/etc/fdmns/projects1_data:
dsk9c

/etc/fdmns/projects2_data:
dsk11c

/etc/fdmns/projects_tools:
dsk12c

/etc/fdmns/root1_domain:
dsk4a

/etc/fdmns/root2_domain:
dsk8a

/etc/fdmns/root3_domain:
dsk2a

/etc/fdmns/root_domain:
dsk0a

/etc/fdmns/usr_domain:
dsk0g
```

This output from the `ls` command indicates the following:

- Disk `dsk3` is used by the clusterwide root file system (`/`). You cannot use this disk.
- Disk `dsk5` is used by the clusterwide `/usr` file system. You cannot use this disk.
- Disk `dsk6` is used by the clusterwide `/var` file system. You cannot use this disk.
- Disks `dsk4`, `dsk8`, and `dsk2` are member boot disks. You cannot use these disks.

  You can also use the `disklabel` command to identify member boot disks. They have three partitions: the `a` partition has `fstype AdvFS`, the `b` partition has `fstype swap`, and the `h` partition has `fstype cnx`.
- Disks `dsk9`, `dsk11`, and `dsk12` appear to be used for data and tools.
- Disk `dsk0` is the boot disk for the noncluster, base Tru64 UNIX operating system.

Keep this disk unchanged in case you need to boot the noncluster kernel to make repairs.

9.6.2.3 Looking for Member Swap Areas

A member's primary swap area is always the `b` partition of the member boot disk. (For information about member boot disks, see Section 11.1.4.) However, a member might have additional swap areas. If a member is down, be careful not to use the member's swap area. To learn whether a disk has swap areas on it, use the `disklabel -r` command. Look in the `fstype` column in the output for partitions with `fstype swap`.

In the following example, partition `b` on `dsk11` is a swap partition:

```
# disklabel -r dsk11
        .
        .
        .
8 partitions:
#          size     offset    fstype   [fsize bsize cpg] # NOTE: values not exact
 a:     262144          0     AdvFS                     # (Cyl.    0 - 165*)
 b:     401408     262144      swap                     # (Cyl.  165*- 418*)
 c:    4110480          0    unused        0     0      # (Cyl.    0 - 2594)
 d:    1148976     663552    unused        0     0      # (Cyl.  418*- 1144*)
 e:    1148976    1812528    unused        0     0      # (Cyl. 1144*- 1869*)
 f:    1148976    2961504    unused        0     0      # (Cyl. 1869*- 2594)
 g:    1433600     663552     AdvFS                     # (Cyl.  418*- 1323*)
 h:    2013328    2097152     AdvFS                     # (Cyl. 1323*- 2594)
```

9.6.3 Editing /etc/fstab

You can use the SysMan Station graphical user interface (GUI) to create and configure an AdvFS volume. However, if you choose to use the command line, when it comes time to edit `/etc/fstab`, you need do it only once, and you can do it on any cluster member. The `/etc/fstab` file is not a CDSL. A single file is used by all cluster members.

9.7 Managing CDFS File Systems

In a cluster, a CD-ROM drive is always a served device. The drive must be connected to a local bus; it cannot be connected to a shared bus. The following are restrictions on managing a CD-ROM File System (CDFS) in a cluster:

- The `cddevsuppl` command is not supported in a cluster.
- The following commands work only when executed from the cluster member that is the CFS server of the CDFS file system:
  - `cddrec`(1)
  - `cdptrec`(1)
  - `cdsuf`(1)

– cdvd(1)

– cdxar(1)

– cdmntsuppl(8)

Regardless of which member mounts the CD-ROM, the member that is connected to the drive is the CFS server for the CDFS file system.

To manage a CDFS file system, follow these steps:

1. Enter the cfsmgr command to learn which member currently serves the CDFS:

   ```
   # cfsmgr
   ```

2. Log in on the serving member.
3. Use the appropriate commands to perform the management tasks.

For information about using library functions that manipulate the CDFS, see the TruCluster Server *Cluster Highly Available Applications* manual.

9.8 Backing Up and Restoring Files

Back up and restore for user data in a cluster is similar to that in a standalone system. You back up and restore CDSLs like any other symbolic links. To back up all the targets of CDSLs, back up the /cluster/members area.

Make sure that all restore software that you plan to use is available on the Tru64 UNIX disk of the system that was the initial cluster member. Treat this disk as the emergency repair disk for the cluster. If the cluster loses the root domain, cluster_root, you can boot the initial cluster member from the Tru64 UNIX disk and restore cluster_root.

The bttape utility is not supported in clusters.

9.8.1 Suggestions for Files to Back Up

You should regularly back up data files and the following file systems:

- The clusterwide root file system

  Use the same backup/restore methods that you use for user data.

- The clusterwide /usr file system

  Use the same backup/restore methods that you use for user data.

- The clusterwide /var file system

  Use the same backup/restore methods that you use for user data.

If, before installing TruCluster Server, you were using AdvFS and had /var located in /usr, the installation process moved /var into a separate fileset under usr_domain.

Because of this move, you must back up /var as a separate file system from /usr.

- Member boot disks

  There are special considerations for backing up and restoring member boot disks. See Section 11.1.4.

9.9 Managing Swap Space

Do not put swap entries in /etc/fstab. In Tru64 UNIX Version 5.0 the list of swap devices was moved from the /etc/fstab file to the /etc/sysconfigtab file. Additionally, you no longer use the /sbin/swapdefault file to indicate the swap allocation; use the /etc/sysconfigtab file for this purpose as well. The swap devices and swap allocation mode are automatically placed in the /etc/sysconfigtab file during installation of the base operating system. For more information, see the Tru64 UNIX *System Administration* manual and swapon(8).

Put each member's swap information in that member's sysconfigtab file. Do not put any swap information in the clusterwide /etc/fstab file.

Swap information in sysconfigtab is identified by the swapdevice attribute. The format for swap information is as follows:

swapdevice=*disk_partition*,*disk_partition*,...

For example:

```
swapdevice=/dev/disk/dsk1b,/dev/disk/dsk3b
```

Specifying swap entries in /etc/fstab does not work in a cluster because /etc/fstab is not member-specific; it is a clusterwide file. If swap were specified in /etc/fstab, the first member to boot and form a cluster would read and mount all the file systems in /etc/fstab. The other members would never see that swap space.

The file /etc/sysconfigtab is a context-dependent symbolic link (CDSL), so that each member can find and mount its specific swap partitions. The installation script automatically configures one swap device for each member, and puts a swapdevice= entry in that member's sysconfigtab file.

If you want to add additional swap space, specify the new partition with swapon, and then put an entry in sysconfigtab so the partition is available following a reboot. For example, to configure dsk3b for use as a secondary swap device for a member already using dsk1b for swap, enter the following command:

```
swapon -s /dev/disk/dsk3b
```

Then, edit that member's /etc/sysconfigtab and add /dev/disk/dsk3b. The final entry in /etc/sysconfigtab will look like the following:

```
swapdevice=/dev/disk/dsk1b,/dev/disk/dsk3b
```

9.9.1 Locating Swap Device for Improved Performance

Locating a member's swap space on a device on a shared bus results in additional I/O traffic on the bus. To avoid this, you can place swap on a disk on the member's local bus.

The only downside to locating swap local to the member is the unlikely case where the member loses its path to the swap disk, as can happen when an adapter fails. In this situation, the member will fail. When the swap disk is on a shared bus, the member can still use its swap partition as long as at least one member still has a path to the disk.

9.10 Fixing Problems with Boot Parameters

If a cluster member fails to boot due to parameter problems in the member's root domain (root*N*_domain), you can mount that domain on a running member and make the needed changes to the parameters. However, before booting the down member, you must unmount the newly updated member root domain from the running cluster member.

Failure to do so can cause a crash and result in the display of the following message:

```
cfs_mountroot:  CFS server already exists for node boot
partition.
```

For more information, see Section 11.1.9.

9.11 Using the verify Utility in a Cluster

The verify utility examines the on-disk metadata structures of AdvFS file systems. Before using the utility, you must unmount all filesets in the file domain to be verified.

If you are running the verify utility and the cluster member on which it is running fails, extraneous mounts may be left. This can happen because the verify utility creates temporary mounts of the filesets that are in the domain that is being verified. On a single system these mounts go away if the system fails while running the utility, but, in a cluster, the mounts fail over to another cluster member. The fact that these mounts fail over also prevents you from mounting the filesets until you remove the spurious mounts.

When verify runs, it creates a directory for each fileset in the domain and then mounts each fileset on the corresponding directory. A directory is named as follows: /etc/fdmns/*domain*/*set*_verify_*XXXXXX*, where *XXXXXX* is a unique ID.

For example, if the domain name is dom2 and the filesets in dom2 are fset1, fset2, and fset3, enter the following command:

```
# ls -l /etc/fdmns/dom2
total 24
lrwxr-xr-x   1 root     system        15 Dec 31 13:55 dsk3a -> /dev/disk/dsk3a
lrwxr-x---   1 root     system        15 Dec 31 13:55 dsk3d -> /dev/disk/dsk3d
drwxr-xr-x   3 root     system      8192 Jan  7 10:36 fset1_verify_aacTxa
drwxr-xr-x   4 root     system      8192 Jan  7 10:36 fset2_verify_aacTxa
drwxr-xr-x   3 root     system      8192 Jan  7 10:36 fset3_verify_aacTxa
```

To clean up the failed-over mounts, follow these steps:

1. Unmount all the filesets in /etc/fdmns:

   ```
   # umount /etc/fdmns/*/*_verify_*
   ```

2. Delete all failed over mounts with the following command:

   ```
   # rm -rf /etc/fdmns/*/*_verify_*
   ```

3. Remount the filesets as you would after a normal completion of the verify utility.

For more information about verify, see verify(8).

9.11.1 Using the verify Utility on Cluster Root

The verify utility has been modified to allow it to run on active domains. Use the -a option to examine the cluster root file system, cluster_root.

You must execute the verify -a utility on the member that is serving the domain that you are examining. Use the cfsmgr command to determine which member serves the domain.

When verify runs with the -a option, it only examines the domain. No fixes can be done on the active domain. The -f and -d options cannot be used with the -a option.

10

Using Logical Storage Manager in a Cluster

This chapter presents configuration and usage information that is specific to Logical Storage Manager (LSM) in a TruCluster Server environment. The chapter discusses the following subjects:

- Understanding differences between managing LSM in clusters and standalone systems (Section 10.1)
- Understanding storage connectivity and LSM volumes (Section 10.2)
- Configuring LSM for a cluster (Section 10.3)
- Adding cluster members with LSM legacy volumes (Section 10.4)
- Moving LSM disk groups between standalone systems and clusters (Section 10.5)
- Configuring dirty-region log sizes (Section 10.6)
- Encapsulating the `/usr` file system, the cluster root domain, other Advanced File System (AdvFS) domains, or an individual member's swap devices into LSM volumes (Section 10.7)

For complete documentation on LSM, see the Tru64 UNIX *Logical Storage Manager* manual. Information on installing LSM software can be found in that manual and the Tru64 UNIX *Installation Guide*.

Using LSM in a cluster is like using LSM on a single system. The same LSM software subsets are used for both clusters and standalone configurations.

In a cluster, LSM provides the following features:

- High availability

  LSM operations continue despite the loss of cluster members, as long as the cluster itself continues operation and a physical path to the storage is available.

- Performance:
  - For I/O within the cluster environment, LSM volumes incur no additional LSM I/O overhead.

LSM follows a fully symmetric, shared I/O model, where all members share a common LSM configuration and each member has private dirty-region logging.

  - Disk groups can be used simultaneously by all cluster members.
  - There is one shared `rootdg` disk group.
  - Any member can handle all LSM I/O directly, and does not have to pass it to another cluster member for handling.

- Ease of management

  The LSM configuration can be managed from any member.

10.1 Differences Between Managing LSM in Clusters and in Standalone Systems

The following restrictions apply to LSM in a cluster:

- LSM volumes cannot be used for the boot partitions of individual members.
- LSM cannot be used to mirror a quorum disk or any partitions on that disk.
- LSM Redundant Array of Independent Disks (RAID) 5 volumes are not supported in clusters.
- To place the `cluster_root` domain under LSM control, you must use the `volmigrate` command. To place the `cluster_usr` or `cluster_var` domains under LSM control, use either the `volmigrate` command or the `volencap` command.
- There are small but important differences in the process of configuring LSM. See Section 10.3.
- The size requirements for log subdisks in a cluster differ from those in a standalone system. See Section 10.6.

The following LSM behavior in a cluster varies from the single-system image model:

- Statistics that are returned by the `volstat` command apply only to the member on which the command executes.
- The `voldisk list` command can give different results on different members for disks that are not part of LSM (that is, `autoconfig` disks).

  The differences are typically limited to disabled disk groups. For example, one member might show a disabled disk group and on another member that same disk group might not show at all.

10.2 Storage Connectivity and LSM Volumes

When adding disks to an LSM disk group on a cluster, note the following points:

- Make sure that all storage in an LSM volume has the same connectivity. LSM volumes have the same connectivity when either one of the following conditions is true:
  - All disks in an LSM disk group are on the same shared SCSI bus.
  - Disks in an LSM disk group are on different shared SCSI buses, but all of those buses are connected to the same cluster members.
- Storage availability increases as more members have direct access to all disks in a disk group.

  Availability is highest when all disks in a disk group are on a shared bus that is directly connected to all cluster members.
- Private disk groups (a disk group whose volumes are all connected to the private bus of a single cluster member) are supported, but if that member becomes unavailable, then the cluster loses access to the disk group.

  Because of this, a private disk group is suitable only when the member that the disk group is physically connected to is also the only member that needs access to the disk group.
- Avoid configuring a disk group with volumes that are distributed among the private buses of multiple members. Such disk groups are not recommended, because no single member has direct access to all volumes in the group.

The `drdmgr` and `hwmgr` commands can give you information about which systems serve which disks. To get a graphical display of the cluster hardware configuration, including active members, buses, storage devices, and their connections, use the `sms` command to invoke the graphical interface for the SysMan Station, and then select Hardware from the Views menu.

10.3 Configuring LSM for a Cluster

The procedure for configuring LSM for a cluster depends on the state of the system where LSM is to be configured. There are three possibilities:

- Tru64 UNIX has been installed on the system, but the system has not yet been initiated as a cluster (you have not run the `clu_create` command). See Section 10.3.1.
- The initial cluster member has been created (you have run the `clu_create` command), but it is still a single-member cluster (you have not yet run the `clu_add_member` command). See Section 10.3.2.

- The cluster has been created and members have been added, forming a multimember cluster. See Section 10.3.3.

10.3.1 Configuring LSM Before Cluster Creation

If you configured LSM on the Tru64 UNIX system before cluster creation, the existing LSM configuration is propagated to the cluster when the cluster is created.

All LSM volumes from the system are available on the new cluster, including volumes created by encapsulating the single system's boot partitions and swap space. However, the original system-related volumes are not used after the system is converted to a cluster, and the domains are not available until explicitly mounted. If you halt the cluster and boot the base operating system again, these domains are still available.

When you add new members to the cluster, they are automatically configured to run LSM. You do not need to do any further LSM configuration.

For information about installing LSM software and configuring LSM on a Tru64 UNIX system before cluster creation, see the Tru64 UNIX *Logical Storage Manager* manual.

10.3.2 Configuring LSM After Cluster Creation and Before Members Have Been Added

The procedure for configuring LSM after cluster creation and before any members have been added is the same as the procedure for configuring LSM on a standalone Tru64 UNIX system. See the chapter on setting up the LSM software in the Tru64 UNIX *Logical Storage Manager* manual.

After LSM is configured on the initial cluster member, LSM is automatically configured on the new members as they are added to the cluster. You do not need to do any further LSM configuration.

10.3.3 Configuring LSM in a Multimember Cluster

To configure LSM on an established multimember cluster, follow these steps after installing the LSM software:

1. Enter the following command on one cluster member. It does not matter which member.

   ```
   # volsetup
   ```

   You are queried to list disk names or partitions to be added to the `rootdg` disk group. For more information, see `volsetup`(8).

2. Synchronize LSM throughout the cluster by running the following command on each of the other cluster members:

   ```
   # volsetup -s
   ```

   Note

   Do not run `volsetup -s` on the cluster member where you first configured LSM.

   If a new member is later added to the cluster, do not run the `volsetup -s` command on the new member. The `clu_add_member` command automatically synchronizes LSM on the new member.

10.4 Adding Cluster Members with LSM Legacy Volumes

If you have a standalone system with LSM volumes, you can make that system a cluster member and incorporate its LSM volumes in the established cluster.

If you want the cluster to be able to use the disks in rootdg, you must rename rootdg as you import it on the cluster. Any volumes in the old rootdg disk group that were not used by the standalone system's root file system, `/usr`, `/var`, or swap partitions will be usable on the cluster.

To prepare a standalone system to become a cluster member:

1. Before halting the standalone system in preparation to connecting it to the cluster, deport each disk group (except rootdg) with the `voldg` command:

   ```
   # voldg deport diskgroup
   ```

2. Display the disks in the rootdg disk group:

   ```
   # voldisk -g rootdg list
   ```

   Information similar to the following is displayed:

   ```
   DEVICE       TYPE      DISK         GROUP        STATUS
   dsk1         sliced    dsk1         rootdg       online
   dsk2         sliced    dsk2         rootdg       online
   dsk3         sliced    dsk3         rootdg       online
   dsk4         sliced    dsk4         rootdg       online
   dsk5         sliced    dsk5         rootdg       online
   dsk6         sliced    dsk6         rootdg       online
   dsk7         sliced    dsk7         rootdg       online
   dsk8         sliced    dsk8         rootdg       online
   dsk9         sliced    dsk9         rootdg       online
   .
   .
   .
   ```

3. Use the name of any disk in rootdg to identify the disk group ID (DGID) of rootdg:

```
# voldisk list dsk1
```

Information similar to the following is displayed:

```
Device:    dsk1
devicetag: dsk1
type:      sliced
hostid:    lsmtemp.xyz.abc.com
disk:      name=dsk1 id=962390779.1466.lsmtemp.xyz.abc.com
group:     name=rootdg id=959293418.1026.lsmtemp.xyz.abc.com
flags:     online ready autoimport imported
pubpaths:  block=/dev/disk/dsk1g char=/dev/rdisk/dsk1g
privpaths: block=/dev/disk/dsk1h char=/dev/rdisk/dsk1h
version:   2.1
iosize:    min=512 (bytes) max=2048 (blocks)
public:    slice=6 offset=16 len=8375968
private:   slice=7 offset=0 len=4096
update:    time=973707788 seqno=0.35
headers:   0 248
configs:   count=1 len=2993
logs:      count=1 len=453
Defined regions:
 config   priv     17-   247[   231]: copy=01 offset=000000 enabled
 config   priv    249-  3010[  2762]: copy=01 offset=000231 enabled
 log      priv   3011-  3463[   453]: copy=01 offset=000000 enabled
```

The DGID appears in the line that begins with `group:`. In the previous output, the DGID of rootdg is `959293418.1026.lsmtemp.xyz.abc.com`

4. Add the system to the cluster.
5. After the former standalone system has been added to the cluster, import the disk groups as described in Section 10.5.

10.5 Moving LSM Disk Groups Between Standalone and Cluster Environments

In a cluster, the LSM configuration database of each disk group is marked to indicate that it is being used in a cluster environment. A nonautoimported disk group that is designated for use in a cluster environment cannot be used in a standalone environment. Similarly, a nonautoimported disk group designated for use in a standalone environment cannot be used in a cluster.

Note

To move an LSM disk group between standalone and cluster environments, one of the following must be true:

- The disk group was deported with the `voldg` command.
- The system or cluster using the disk group was cleanly shut down.

A disk group that requires recovery, for example, due to a system or cluster crash, cannot be moved between standalone and cluster

environments. The disk group must first be recovered in the environment where it was last used.

10.5.1 Importing Tru64 UNIX Version 5.1A Standalone Disk Groups

Manually importing a Tru64 UNIX Version 5.1A disk group from a standalone system into a cluster environment requires that you explicitly change the designation of that disk group for use in a cluster. If you move the disk group to a cluster and then boot the cluster, the disk group is imported and its configuration database is converted automatically.

To manually import a disk group from a standalone system and make the volumes available to the cluster, follow these steps:

1. Import the disk group and mark it for use in a cluster by doing one of the following:
   - For all disk groups other than the standalone system's rootdg disk group, enter:

     ```
     # voldg -o shared import diskgroup
     ```

   - To import and convert the standalone system's rootdg disk group, assign it a new name and import it using its DGID as identified in Section 10.4:

     ```
     # voldg -o shared -n newname import id=rootdg_dgid
     ```

2. Start the volumes in the disk group:

   ```
   # volrecover -g diskgroup
   ```

You can also move a disk group from a cluster back to a standalone system. This requires marking the disk group for single system use.

To move a disk group from a cluster environment to a standalone environment:

1. Physically move the disk group to the standalone system.
2. Import the disk group, marking it for single system use:

   ```
   # voldg -o private import diskgroup
   ```

3. Start the volumes in the disk group:

   ```
   # volrecover -g diskgroup
   ```

10.5.2 Importing Tru64 UNIX Version 4.0 Standalone Disk Groups

To import a Tru64 UNIX Version 4.0 disk group from a standalone system to a TruCluster Server environment, you must do the following:

1. Determine the device name, media name, and LSM disk type of each disk.

   You need to identify each of the old `rz` disks or partitions and their corresponding `dsk` names. The old `rz` names identify the device names either directly or by the error entries that are returned by the `voldisk list` command.

2. Convert the disk group and mark it for use in a cluster.
3. Convert the legacy device special file names.

The rest of this section describes each step in more detail.

10.5.2.1 Determining the Device Name, Media Name, and LSM Disk Types

Determine the access names, media names, and LSM disk types with the following command:

```
# voldisk list
```

Use the `voldisk list` command to determine the old device names that are in an error state, as well as to see what new names were autodiscovered. In the case of multiple `nopriv` disks, `voldisk list` returns nothing to differentiate the disks. In this case, you might need to use the `hwmgr` command to determine physical addresses. Then you have to map these physical addresses to previously recorded information for access names, device names, and media names.

10.5.2.2 Converting the Disk Group for Cluster Use

Note

After a disk group is converted, it cannot be used again on a Tru64 UNIX Version 4.0 system.

To convert the disk group and mark it for cluster use, enter the following command:

```
# voldg -o convert_old -o shared import diskgroup
```

10.5.2.3 Converting Legacy Device Special Files

LSM automatic configuration locates all disks and imports them. LSM marks devices with legacy device names as `failed`.

To convert the legacy device special file names, follow these steps:

1. Remove each device that is marked `failed` with the following command. Substitute the actual legacy device name for `rz`*`NN`*:

   # **voldisk rm rz*NN***

2. Reattach each disk of type `nopriv` to associate a media name with a device name. For each disk of type `nopriv`, enter the following command, using the information that you collected in step 1:

   # **voldg -k adddisk *media_name=device_name***

10.6 Dirty-Region Log Sizes for Clusters

LSM uses log subdisks to store the dirty-region logs of volumes that have Dirty Region Logging (DRL) enabled. By default, the `volassist` command configures a log subdisk large enough so that the associated mirrored volume can be used in either a cluster or a standalone system.

For performance reasons, standalone systems might be configured with values other than the default. If a standalone system has log subdisks that are configured for optimum performance, and that system is to become part of a cluster, the log subdisks must be reconfigured with 65 or more blocks.

To reconfigure the log subdisk, use the `volplex` command to delete the old DRL, and then use `volassist` to create a new log. You can do this while the volume is active; that is, while users are performing I/O to the volume.

In the following example, the `volprint` command gets the name of the current log for `vol1`. Then the `volplex` command dissociates and removes the old log. Finally, the `volassist` command creates a new log subdisk for `vol1`. By default, the `volassist` command creates a log subdisk of a size that is appropriate to a cluster environment.

```
#  volprint vol1 | grep LOGONLY
pl vol1-03    vol1     ENABLED  LOGONLY    -    ACTIVE    -     -
# volplex -o rm dis vol1-03
# volassist addlog vol1
```

Note

In a cluster, LSM DRL sizes must be at least 65 blocks for the DRL to be used with a mirrored volume.

If the DRL size for a mirrored volume is less than 65 blocks, DRL is disabled. However, the mirrored volume can still be used.

Table 10–1 lists some suggested DRL sizes for small, medium, and large storage configurations in a cluster. The `volassist addlog` command creates a DRL of the appropriate size.

Table 10–1: Sizes of DRL Log Subdisks

Volume Size (GB)	DRL Size (blocks)
<= 1	65
2	130
3	130
4	195
60	2015
61	2015
62	2080
63	2080
1021	33215
1022	33280
1023	33280
1024	33345

10.7 Placing Cluster Domains into LSM Volumes

You can place cluster domains or cluster members' swap devices into LSM volumes. This allows you to use the features of LSM, such as mirroring, on those volumes. The following methods are available on a cluster to place domains and members' swap devices under LSM control:

- Use the `volencap` and `volreconfig` commands to encapsulate a domain or swap device into an LSM volume.

  The `volencap` command creates LSM volumes on the same disks or disk partitions that the domain or swap space is currently using. Use the `volencap` command to encapsulate the `/usr` file system, any AdvFS domain other than `cluster_root`, or the swap devices for a cluster member.

  The advantage to using this method is that no extra disk space is required. The disadvantage is that you might need to shut down and reboot the cluster or cluster member for the encapsulation to complete.

  Section 10.7.1 describes how to encapsulate the `/usr` file system on a cluster. For information on encapsulating other file systems, domains, or

existing data, see the Tru64 UNIX *Logical Storage Manager* manual and the volencap(8) reference page.

- Use the volmigrate command to migrate an AdvFS domain to an LSM volume.

  The volmigrate command creates LSM volumes on the disks that you specify, moves the domain to the volumes, and removes the original disk from the domain and leaves it unused. The advantage to using this method is that the migration occurs while the domain's filesets are mounted, and no reboot is required. The disadvantage is that the migration process temporarily uses additional disk space while the domain data is copied to the LSM volume.

  To place the cluster_root domain under LSM control, you must use the volmigrate command.

The following sections describe each method in more detail.

10.7.1 Encapsulating the /usr File System

The volencap command creates scripts to perform in-place encapsulation of disk partitions into LSM volumes.

Note

When encapsulating the cluster /usr file system, you must shut down the entire cluster. The scripts that are created by the volencap command are executed during the startup routine.

To encapsulate the /usr file system:

1. Enter the following command:

   ```
   # volencap cluster_usr
   ```

2. Shut down the cluster:

   ```
   # shutdown -chs time
   ```

3. Boot the member to multi-user mode.
4. Boot the remaining cluster members.

After you shut down the cluster, the volreconfig command is automatically executed during system boot from /etc/inittab.

10.7.2 Encapsulating Members' swap Devices

You can encapsulate:

- All the swap devices for a member at once, with the swap operand

- The above plus swap devices for other members, in the same command
- Only the swap devices you specify, for one or more members, in the same command

Note

All members whose swap devices you want to encapsulate must be running.

You can run the `volencap` command on one member to set up the encapsulation for several members at once. However, you must run the `volreconfig` command on each member whose swap devices you are encapsulating. The `volreconfig` command executes the scripts created by the `volencap` command and reboots the cluster member to complete the encapsulation.

The `swap` operand is member-specific; it is a shortcut for specifying all the swap devices for the member it is run on. To encapsulate swap devices for several members at a time, you must identify the other members' swap devices, for example by entering the `swapon` command on each member.

- To encapsulate all the swap devices for one member only, enter:

  ```
  # volencap swap
  # volreconfig
  ```

  After the member reboots, each swap device uses an LSM volume.

- To encapsulate all the swap devices for one member, plus the devices for one or more other members:

  1. Identify the cluster members:

     ```
     # clu_get_info
     ```

  2. Display the swap devices for each member by using one of the following methods:

     – On each member, enter:

       ```
       # swapon -s
       ```

     – On one member, display the swap devices for all members by entering the following command, where *n* is the member number:

       ```
       # more /cluster/members/membern/boot_partition/etc/sysconfigtab
       | grep swap
       ```

  3. Do one of the following:

     – To encapsulate all swap devices for the current member only, enter:

       ```
       # volencap swap
       ```

– To encapsulate all swap devices for the current member, plus devices for additional members, enter:

```
# volencap swap dsknp dsknp ...
```

– To encapsulate specific swap devices for the current member, plus (optionally) additional members, enter:

```
# volencap dsknp dsknp ...
```

4. Enter the following command on each member with queued-up encapsulation scripts:

```
# volreconfig
```

After each member reboots, its swap devices use LSM volumes.

See the volencap(8) reference page for more information on encapsulating swap in a cluster.

You can create an LSM volume for secondary swap space instead of encapsulating the member's swap devices, or in addition to doing so. See the Tru64 UNIX *Logical Storage Manager* manual for more information on creating volumes for secondary swap.

10.7.3 Migrating AdvFS Domains into LSM Volumes

You can place an AdvFS domain, including the cluster root domain cluster_root, into an LSM volume. This operation uses a different disk than the disk on which the domain originally resides, and therefore does not require a reboot. You cannot place the individual members' boot partitions (called root*memberID*_domain#root) into LSM volumes.

You can specify:

- The name of the volume (default is the name of the domain with the suffix vol)
- The number of stripes and mirrors that you want the volume to use

  Striping improves read performance, and mirroring ensures data availability in the event of a disk failure.

You must specify LSM disks by their disk media names on which to create the volume for the domain, and all the disks must belong to the same disk group. For the cluster_root domain, the disks must be simple or sliced disks (must have an LSM private region) and must belong to the rootdg disk group. The command fails if you specify disk media names that belong to a disk group other than rootdg.

There must be sufficient LSM disks and they must be large enough to contain the domain. See the volmigrate(8) reference page for more information on disk requirements and the options for striping and mirroring.

To migrate a domain into an LSM volume, enter:

```
# volmigrate [-g diskgroup] [-m num_mirrors] [-s num_stripes]
domain_name disk_media_name...
```

The `volmigrate` command creates a volume with the specified characteristics, moves the data from the domain into the volume, removes the original disk or disks from the domain, and leaves those disks unused. The volume is started and ready for use, and no reboot is required.

You can use LSM commands to manage the domain volume the same as for any other LSM volume.

If a disk in the volume fails, see the Troubleshooting section in the *Logical Storage Manager* manual for the procedure to replace a failed disk and recover the volumes on that disk. If a disk failure occurs in the `cluster_root` domain volume and the procedure does not solve the problem (specifically, if all members have attempted to boot, yet the volume that is associated with cluster root cannot be started), then you might have to restore the cluster root file system using a backup tape. After restoring the cluster, you can again migrate the cluster root domain to an LSM volume as described here.

10.7.4 Migrating Domains from LSM Volumes to Physical Storage

You can migrate any AdvFS domain onto physical disk storage and remove the LSM volume with the `volunmigrate` command. The cluster remains running during this process and no reboot is required.

You must specify one or more disk partitions that are not under LSM control, ideally on a shared bus, for the domain to use after the migration. These partitions must be large enough to accommodate the domain plus at least 10 percent additional space for file system overhead. The `volunmigrate` command examines the partitions that you specify to ensure they meet both qualifications, and returns an error if either or both is not met. See the `volunmigrate`(8) reference page for more information.

To migrate an AdvFS domain from an LSM volume to physical storage:

1. Determine the size of the domain volume:

   ```
   # volprint -vt domainvol
   ```

2. Find one or more disk partitions on a shared bus that are not under LSM control and are large enough to accommodate the domain plus file system overhead of at least 10 percent:

   ```
   # hwmgr -view devices -cluster
   ```

3. Migrate the domain, specifying the target disk partitions:

```
# volunmigrate domain_name dsknp [ dsknp...]
```

After migration, the domain resides on the specified disks and the LSM volume no longer exists.

10.7.5 Unencapsulating Swap Volumes

You can unencapsulate swap devices for a cluster member to move them from LSM volumes onto physical disk partitions.

To unencapsulate all members' swap devices, perform the following steps on all members with encapsulated swap devices:

1. Display the names of LSM volumes to determine the node name for the member and the disk name in the member's swap volume:

   ```
   # volprint -vht
   ```

   In the output, look for:

   - The *nodename*-swap*nn* volume name; for example, `larry-swap01`.
   - The disk name for the swap volume in the form `dsk`*np*; for example, `dsk10b`.

2. Display a list of LSM disks:

   ```
   # voldisk list
   ```

   In the output, look for the private region of the swap disk, in the form `dsk`*np*. The disk number (`dsk`*n*) will be the same as the disk name for the swap volume; for example, `dsk10f`.

3. Edit the `/etc/sysconfigtab` file for the member and remove the `/dev/vol/rootdg/`*nodename*`-swap`*nn* entry from the `swapdevice=` line.

4. Remove the swap disk's private region partition from the disk group and from LSM control:

   ```
   # voldg rmdisk dsknp
   # voldisk rm dsknp
   ```

5. Reboot the member:

   ```
   # shutdown -r now
   ```

   When the member starts again, it no longer uses the LSM swap volume.

6. Log back in to the same member.

7. Remove the swap volume:

   ```
   # voledit -rf rm nodename-swapnn
   ```

8. Remove the LSM disk for the swap volume from the disk group, and remove the disk from LSM control:

```
# voldg rmdisk dsknp
# voldisk rm dsknp
```

9. Set the cluster member to swap on the disk partition:

```
# swapon /dev/disk/dsknp
```

10. Edit the `/etc/sysconfigtab` file as follows:
    - Add the `/dev/disk/dsk`*np* entry to the line `swapdevice=` so that the line reads:

      ```
      swapdevice=/dev/disk/dsknp
      ```

    - If you removed the last LSM swap device for this member, set the value for `lsm_root_dev_is_volume=` to 0.

The cluster member uses the specified disk partition for its swap device, and the LSM swap volume no longer exists.

11

Troubleshooting Clusters

This chapter presents the following topics:

- Suggestions for resolving problems on a cluster (Section 11.1)
- Hints for configuring and managing a cluster (Section 11.2)

11.1 Resolving Problems

This section describes solutions to problems that can arise during the day-to-day operation of a cluster.

11.1.1 Booting Systems Without a License

You can boot a system that does not have a TruCluster Server license. The system joins the cluster and boots to multiuser mode, but only root can log in (with a maximum of two users). The cluster application availability (CAA) daemon, `caad`, is not started. The system displays a license error message reminding you to load the license. This policy enforces license checks while making it possible to boot, license, and repair a system during an emergency.

11.1.2 Shutdown Leaves Members Running

A cluster shutdown (`shutdown -c`) can leave one or more members running. In this situation, you must complete the cluster shutdown by shutting down all members.

Imagine a three-member cluster where each member has one vote and no quorum disk is configured. During cluster shutdown, quorum is lost when the second-to-last member goes down. If quorum checking is on, the last member running suspends all operations and cluster shutdown never completes.

To avoid an impasse in situations like this, quorum checking is disabled at the start of the cluster shutdown process. If a member fails to shut down during cluster shutdown, it might appear to be a normally functioning cluster member, but it is not, because quorum checking is disabled. You must manually complete the shutdown process.

The shutdown procedure depends on the state of the systems that are still running:

- If the systems are hung, not servicing commands from the console, then halt the systems and generate a crash dump.
- If the systems are not hung, then use the `/sbin/halt` command to halt the system.

11.1.3 Dealing with CFS Errors at Boot

During system boot when the clusterwide root (`/`) is mounted for the first time, CFS can generate the following warning message:

```
"WARNING:cfs_read_advfs_quorum_data: cnx_disk_read failed with error-number
```

Usually *error-number* is the `EIO` value.

This message is accompanied by the following message:

```
"WARNING: Magic number on ADVFS portion of CNX partition on quorum disk \
is not valid"
```

These messages indicate that the booting member is having problems accessing data on the CNX partition of the quorum disk, which contains the device information for the `cluster_root` domain. This can occur if the booting member does not have access to the quorum disk, either because the cluster is deliberately configured this way or because of a path failure. In the former case, the messages can be considered informational. In the latter case, you need to adddress the cause of the path failure.

The messages can mean that there are problems with the quorum disk itself. If hardware errors are also being reported for the quorum disk, then replace it. For information on replacing a quorum disk, see Section 4.5.1.

For a description of error numbers, see `errno`(5). For a description of `EIO`, see `errno`(2).

11.1.4 Backing Up and Repairing a Member's Boot Disk

A member's boot disk contains three partitions. Table 11–1 presents some details about these partitions.

Table 11–1: File Systems and Storage Differences

Partition	Content
`a`	Advanced File System (AdvFS) boot partition, member root file system (128 MB)

Table 11–1: File Systems and Storage Differences (cont.)

Partition	Content
b	Swap partition (all space between the a and h partitions)
h	CNX binary partition (1 MB) AdvFS and Logical Storage Manager (LSM) store information critical to their functioning on the h partition. This information includes whether the disk is a member or quorum disk, and the name of the device where the cluster root file system is located.

If a member's boot disk is damaged or becomes unavailable, you need the h partition information to restore the member to the cluster. The clu_bdmgr command enables you to configure a member boot disk, and to save and restore data on a member boot disk.

The clu_bdmgr command can do the following tasks:

- Configure a new member boot disk.
- Back up the information on the h partition of a member boot disk.
- Repair an h partition with data from a file or with data from the h partition of the boot disk of a currently available member.

For specifics on the command, see clu_bdmgr(8).

Whenever a member boots, clu_bdmgr automatically saves a copy of the h partition of that member's boot disk. The data is saved in /cluster/members/member*ID*/boot_partition/etc/clu_bdmgr.conf.

As a rule, the h partitions on all member boot disks contain the same data. There are two exceptions to this rule:

- The contents of the h partition somehow become corrupted.
- A member's boot disk is on the member's private bus and the member is down when an update occurs on the cluster that affects the contents of boot disk h partitions. Because the down member's disk is on a private bus, the h partition cannot be updated.

  We recommend that a member's boot disk be on a shared SCSI bus. In addition to ensuring that the h partition is up-to-date, this configuration enables you to diagnose and fix problems with the boot disk even though the member cannot be booted.

If a member's boot disk is damaged, you can use clu_bdmgr to repair or replace it. Even if the cluster is not up, as long as you can boot the clusterized kernel on at least one cluster member, you can use the clu_bdmgr command.

For a description of how to add a new disk to the cluster, see Section 9.2.3.

To repair a member's boot disk, you must first have backed up the boot partition. One method is to allocate disk space in the shared /var file system for a dump image of each member's boot partition.

To save a dump image for member3's boot partition in the member-specific file /var/cluster/members/member3/boot_part_vdump, enter the following command:

```
# vdump -0Df /var/cluster/members/member3/boot_part_vdump \
/cluster/members/member3/boot_partition
```

11.1.4.1 Example of Recovering a Member's Boot Disk

The following sequence of steps shows how to use the file saved by vdump to replace a boot disk. The sequence makes the following assumptions:

- The boot disk for member3 is dsk3.
- You have already added the new member boot disk to the cluster and the name of this replacement disk is dsk5.

  The process of adding the disk includes using the hwmgr -scan comp -cat scsi_bus command so that all members recognize the new disk. A description of how to add a disk to the cluster appears in Section 9.2.3.

Note

A member's boot disk should always be on a bus shared by all cluster members. This arrangement permits you to make repairs to any member's boot disk as long as you can boot at least one cluster member.

1. Use clu_get_info to determine whether member3 is down:

   ```
   # clu_get_info -m 3
   Cluster memberid = 3
   Hostname = member3.zk3.dec.com
   Cluster interconnect IP name = member3-mc0
   Member state = DOWN
   ```

2. Select a new disk (in this example, dsk5) as the replacement boot disk for member3. Because the boot disk for member3 is dsk3, you are instructed to edit member3's /etc/sysconfigtab so that dsk5 is used as the new boot disk for member3.

   To configure dsk5 as the boot disk for member3, enter the following command:

```
# /usr/sbin/clu_bdmgr  -c  dsk5 3
The new member's disk, dsk5, is not the same name as the original disk
configured for domain root3_domain.  If you continue the following
changes will be required in member3's/etc/sysconfigtab file:
        vm:
        swapdevice=/dev/disk/dsk5b
        clubase:
        cluster_seqdisk_major=19
        cluster_seqdisk_minor=175
```

3. Mount member3's root domain (now on dsk5) so you can edit member3's /etc/sysconfigtab and restore the boot partitions:

   ```
   # mount root3_domain#root /mnt
   ```

4. Restore the boot partition:

   ```
   # vrestore -xf /var/cluster/members/member3/boot_part_vdump -D /mnt
   ```

5. Edit member3's /etc/sysconfigtab

   ```
   # cd /mnt/etc
   # cp sysconfigtab sysconfigtab-bu
   ```

   As indicated in the output from the clu_bdmgr command, change the values of the swapdevice attribute in the vm stanza and the cluster_seqdisk_major and cluster_seqdisk_minor attributes in the clubase stanza:

   ```
   swapdevice=/dev/disk/dsk5b
   clubase:
   cluster_seqdisk_major=19
   cluster_seqdisk_minor=175
   ```

6. Restore the h partition CNX information:

   ```
   # /usr/sbin/clu_bdmgr -h  dsk5
   ```

   The h partition information is copied from the cluster member where you run the clu_bdmgr command to the h partition on dsk5.

   If the entire cluster is down, you need to boot one of the members from the clusterized kernel. After you have a single-member cluster running, you can restore the CNX h partition information to member3's new boot disk, dsk5, from /mnt/etc/clu_bdmgr.conf. Enter the following command:

   ```
   # /usr/sbin/clu_bdmgr -h  dsk5 /mnt/etc/clu_bdmgr.conf
   ```

7. Unmount the root domain for member3:

   ```
   # umount root3_domain#root /mnt
   ```

8. Boot member3 into the cluster.

9. Optionally, use the consvar -s bootdef_dev *disk_name* command on member3 to set the bootdef_dev variable to the new disk.

11.1.5 Specifying cluster_root at Boot Time

At boot time you can specify the device that the cluster uses for mounting cluster_root, the cluster root file system. Use this feature only for disaster recovery, when you need to boot with a new cluster root.

The Cluster File System (CFS) kernel subsystem supports six attributes for designating the major and minor numbers of up to three cluster_root devices. Because the cluster_root domain that is being used for disaster recovery may consist of multiple volumes, you can specify one, two, or three cluster_root devices:

- cluster_root_dev1_maj

  The device major number of one cluster_root device.

- cluster_root_dev1_min

  The device minor number of the same cluster_root device.

- cluster_root_dev2_maj

  The device major number of a second cluster_root device.

- cluster_root_dev2_min

  The device minor number of the second cluster_root device.

- cluster_root_dev3_maj

  The device major number of a third cluster_root device.

- cluster_root_dev3_min

  The device minor number of the third cluster_root device.

To use these attributes, shut down the cluster and boot one member interactively, specifying the appropriate cluster_root_dev major and minor numbers. When the member boots, the CNX partition (h partition) of the member's boot disk is updated with the location of the cluster_root devices. If the cluster has a quorum disk, its CNX partition is also updated. As other nodes boot into the cluster, their member boot disk information is also updated.

For example, assume that you want to use a cluster_root that is a two-volume file system that comprises dsk6b and dsk8g. Assume that the major/minor numbers of dsk6b are 19/227, and the major/minor numbers of dsk8g are 19/221. You boot the cluster as follows:

1. Boot one member interactively:

```
>>> boot -fl "ia"
 (boot dkb200.2.0.7.0 -flags ia)
 block 0 of dkb200.2.0.7.0 is a valid boot block
 reading 18 blocks from dkb200.2.0.7.0
 bootstrap code read in
 base = 200000, image_start = 0, image_bytes = 2400
```

```
initializing HWRPB at 2000
initializing page table at fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

.
.
.

Enter kernel_name [option_1 ... option_n]
Press Return to boot default kernel
'vmunix':vmunix cfs:cluster_root_dev1_maj=19 \
cfs:cluster_root_dev1_min=227 cfs:cluster_root_dev2_maj=19 \
cfs:cluster_root_dev2_min=221 Return
```

2. Boot the other cluster members.

For information about using these attributes to recover the cluster root file system, see Section 11.1.6 and Section 11.1.7.

11.1.6 Recovering the Cluster Root File System to a Disk Known to the Cluster

Use the recovery procedure described here when all of the following are true:

- The cluster root file system is corrupted or unavailable.
- You have a recent backup of the file system.
- A disk (or disks) on a shared bus that is accessible to all cluster members is available to restore the file system to, and this disk was part of the cluster configuration before the problems with the root file system occurred.

This procedure is based on the following assumptions:

- The `vdump` command was used to back up the cluster root (`cluster_root`) file system.

  If you used a different backup tool, use the appropriate tool to restore the file system.
- At least one member has access to:
  - A bootable base Tru64 UNIX disk.

    If a bootable base disk is not available, install Tru64 UNIX on a disk that is local to the cluster member. It must be the same version of Tru64 UNIX that was installed on the cluster.
  - The member boot disk for this member (`dsk2a` in this example)
  - The device with the backup of cluster root
- All members of the cluster have been halted.

To restore the cluster root, do the following:

1. Boot the system with the base Tru64 UNIX disk.

   For the purposes of this procedure, we assume this system to be member 1.

2. If this system's name for the device that will be the new cluster root is different than the name that the cluster had for that device, use the `dsfmgr -m` command to change the device name so that it matches the cluster's name for the device.

   For example, if the cluster's name for the device that will be the new cluster root is `dsk6b` and the system's name for it is `dsk4b`, rename the device with the following command:

   ```
   # dsfmgr -m dsk4 dsk6
   ```

3. If necessary, partition the disk so that the partition sizes and file system types will be appropriate after the disk is the cluster root.

4. Create a new domain for the new cluster root:

   ```
   # mkfdmn /dev/disk/dsk6d cluster_root
   ```

5. Make a root fileset in the domain:

   ```
   # mkfset cluster_root root
   ```

6. This restoration procedure allows for `cluster_root` to have up to three volumes. After restoration is complete, you can add additional volumes to the cluster root. For this example, we add only one volume, `dsk6b`:

   ```
   # addvol /dev/disk/dsk6b cluster_root
   ```

7. Mount the domain that will become the new cluster root:

   ```
   # mount cluster_root#root /mnt
   ```

8. Restore cluster root from the backup media. (If you used a backup tool other than `vdump`, use the appropriate restore tool in place of `vrestore`.)

   ```
   # vrestore -xf /dev/tape/tape0 -D /mnt
   ```

9. Change `/etc/fdmns/cluster_root` in the newly restored file system so that it references the new device:

   ```
   # cd /mnt/etc/fdmns/cluster_root
   # rm *
   # ln -s /dev/disk/dsk6b
   ```

10. Use the `file` command to get the major/minor numbers of the new `cluster_root` device. Make note of these major/minor numbers.

    For example:

```
# file /dev/disk/dsk6b
/dev/disk/dsk6b:        block special (19/221)
```

11. Shut down the system and reboot interactively, specifying the device major and minor numbers of the new cluster root:

```
>>> boot -fl "ia"
 (boot dkb200.2.0.7.0 -flags ia)
 block 0 of dkb200.2.0.7.0 is a valid boot block
 reading 18 blocks from dkb200.2.0.7.0
 bootstrap code read in
 base = 200000, image_start = 0, image_bytes = 2400
 initializing HWRPB at 2000
 initializing page table at fff0000
 initializing machine state
 setting affinity to the primary CPU
 jumping to bootstrap code

 .
 .
 .

 Enter kernel_name [option_1 ... option_n]
 Press Return to boot default kernel
 'vmunix':vmunix cfs:cluster_root_dev1_maj=19 \
 cfs:cluster_root_dev1_min=221 Return
```

12. Boot the other cluster members.

11.1.7 Recovering the Cluster Root File System to a New Disk

The process of recovering `cluster_root` to a disk that was previously unknown to the cluster is complicated. Before you attempt it, try to find a disk that was already installed on the cluster to serve as the new cluster boot disk, and follow the procedure in Section 11.1.6.

Use the recovery procedure described here when:

- The cluster root file system is corrupted or unavailable.
- You have a recent backup of the file system.
- No disk is available to restore to that is on a shared bus that is accessible to all cluster members and was part of the cluster configuration before the problems with the root file system occurred.

This procedure is based on the following assumptions:

- The `cluster_usr` and `cluster_var` file systems are not on the same disk as the `cluster_root` file system. The procedure describes recovering only the `cluster_root` file system.
- The `vdump` command was used to back up the cluster root (`cluster_root`) file system.

If you used a different backup tool, use the appropriate tool to restore the file system.

- At least one member has access to:
  - A bootable base Tru64 UNIX disk with the same version of Tru64 UNIX that was installed on the cluster.

    If a bootable base operating system disk is not available, install Tru64 UNIX on a disk that is local to the cluster member. Make sure that it is the same version of Tru64 UNIX that was installed on the cluster.
  - The member boot disk for this member (`dsk2a` in this example)
  - The device with the cluster root backup
  - The disk or disks for the new cluster root
- All members of the cluster have been halted.

To restore the cluster root, do the following:

1. Boot the system with the base Tru64 UNIX disk.

   For the purposes of this procedure, we assume this system to be member 1.
2. If necessary, partition the new disk so that the partition sizes and file system types will be appropriate after the disk is the cluster root.
3. Create a new domain for the new cluster root:

   ```
   # mkfdmn /dev/disk/dsk5b new_root
   ```

   As described in the TruCluster Server *Cluster Installation* guide, the cluster_root file system is often put on a `b` partition. In this case, `/dev/disk/dsk5b` is used for example purposes.
4. Make a root fileset in the domain:

   ```
   # mkfset new_root root
   ```
5. This restoration procedure allows for `new_root` to have up to three volumes. After restoration is complete, you can add additional volumes to the cluster root. For this example, we add one volume, `dsk8e`:

   ```
   # addvol /dev/disk/dsk8e new_root
   ```
6. Mount the domain that will become the new cluster root:

   ```
   # mount new_root#root /mnt
   ```
7. Restore cluster root from the backup media. (If you used a backup tool other than `vdump`, use the appropriate restore tool in place of `vrestore`.)

   ```
   # vrestore -xf /dev/tape/tape0 -D /mnt
   ```

8. Copy the restored cluster databases to the /etc directory of the base Tru64 UNIX system:

```
# cd /mnt/etc
# cp dec_unid_db dec_hwc_cdb dfsc.dat /etc
```

9. Copy the restored databases from the member-specific area of the current member to the /etc directory of the base Tru64 UNIX system:

```
# cd /mnt/cluster/members/member1/etc
# cp dfsl.dat /etc
```

10. If one does not already exist, create a domain for the member boot disk:

```
# cd /etc/fdmns
# ls
# mkdir root1_domain
# cd root1_domain
# ln -s /dev/disk/dsk2a
```

11. Mount the member boot partition:

```
# cd /
# umount /mnt
# mount root1_domain#root /mnt
```

12. Copy the databases from the member boot partition to the /etc directory of the base Tru64 UNIX system:

```
# cd /mnt/etc
# cp dec_devsw_db dec_hw_db dec_hwc_ldb dec_scsi_db /etc
```

13. Unmount the member boot disk:

```
# cd /
# umount /mnt
```

14. Update the database .bak backup files:

```
# cd /etc
# for f in dec_*db ; do cp $f $f.bak ; done
```

15. Reboot the system into single-user mode using the same base Tru64 UNIX disk so that it will use the databases that you copied to /etc.

16. After booting to single-user mode, scan the devices on the bus:

```
# hwmgr -scan scsi
```

17. Remount the root as writable:

```
# mount -u /
```

18. Verify and update the device database:

```
# dsfmgr -v -F
```

19. Use hwmgr to learn the current device naming.

```
# hwmgr -view devices
```

20. If necessary, update the local domains to reflect the device naming (especially `usr_domain`, `new_root`, and `root1_domain`).

    Do this by going to the appropriate `/etc/fdmns` directory, deleting the existing link and creating new links to the current device names. (You learned the current device names in the previous step.) For example:

```
# cd /etc/fdmns/root_domain
# rm *
# ln -s /dev/disk/dsk1a
# cd /etc/fdmns/usr_domain
# rm *
# ln -s /dev/disk/dsk1g
# cd /etc/fdmns/root1_domain
# rm *
# ln -s /dev/disk/dsk2a
# cd /etc/fdmns/new_root
# rm *
# ln -s /dev/disk/dsk5b
# ln -s /dev/disk/dsk8e
```

21. Run the `bcheckrc` command to mount local file systems, particularly `/usr`:

```
#  bcheckrc
```

22. Copy the updated cluster database files onto the cluster root:

```
# mount new_root#root /mnt
# cd /etc
# cp dec_unid_db* dec_hwc_cdb* dfsc.dat /mnt/etc
# cp dfsl.dat /mnt/cluster/members/member1/etc
```

23. Update the `cluster_root` domain on the new cluster root:

```
# rm /mnt/etc/fdmns/cluster_root/*
# cd /etc/fdmns/new_root
# tar cf - * | (cd /mnt/etc/fdmns/cluster_root && tar xf -)
```

24. Copy the updated cluster database files to the member boot disk:

```
# umount /mnt
# mount root1_domain#root /mnt
# cd /etc
# cp dec_devsw_db* dec_hw_db* dec_hwc_ldb* dec_scsi_db* /mnt/etc
```

25. Use the `file` command to get the major/minor numbers of the `cluster_root` devices. Write down these major/minor numbers for use in the next step.

    For example:

```
# file /dev/disk/dsk5b
/dev/disk/dsk5b:        block special (19/227)
# file /dev/disk/dsk8e
```

```
/dev/disk/dsk8e:        block special (19/221)
```

26. Halt the system and reboot interactively, specifying the device major and minor numbers of the new cluster root:

```
>>> boot -fl "ia"
 (boot dkb200.2.0.7.0 -flags ia)
 block 0 of dkb200.2.0.7.0 is a valid boot block
 reading 18 blocks from dkb200.2.0.7.0
 bootstrap code read in
 base = 200000, image_start = 0, image_bytes = 2400
 initializing HWRPB at 2000
 initializing page table at fff0000
 initializing machine state
 setting affinity to the primary CPU
 jumping to bootstrap code

 .
 .
 .

 Enter kernel_name [option_1 ... option_n]
 Press Return to boot default kernel
 'vmunix':vmunix cfs:cluster_root_dev1_maj=19 \
 cfs:cluster_root_dev1_min=227 cfs:cluster_root_dev2_maj=19 \
 cfs:cluster_root_dev1_min=221 Return
```

27. Boot the other cluster members.

    If during boot you encounter errors with device files, run the command `dsfmgr -v -F`.

11.1.8 Dealing with AdvFS Problems

This section describes some problems that can arise when you use AdvFS.

11.1.8.1 Responding to Warning Messages from addvol or rmvol

Under some circumstances, using `addvol` or `rmvol` on the `cluster_root` domain can cause the following warning message:

```
"WARNING:cfs_write_advfs_root_data: cnx_disk_write failed for
quorum disk with error-number."
```

Usually *error-number* is the `EIO` value.

This message indicates that the member where the `addvol` or `rmvol` executed cannot write to the CNX partition of the quorum disk. The CNX partition contains device information for the `cluster_root` domain.

The warning can occur if the member does not have access to the quorum disk, either because the cluster is deliberately configured this way or because of a path failure. In the former case, the message can be considered

informational. In the latter case, you need to adddress the cause of the path failure.

The message can mean that there are problems with the quorum disk itself. If hardware errors are also being reported for the quorum disk, then replace the disk. For information on replacing a quorum disk, see Section 4.5.1.

For a description of error numbers, see `errno`(5). For a description of `EIO`, see `errno`(2).

11.1.8.2 Resolving AdvFS Domain Panics Due to Loss of Device Connectivity

AdvFS can domain panic if one or more storage elements containing a domain or fileset become unavailable. The most likely cause of this problem is when a cluster member is attached to private storage that is used in an AdvFS domain, and that member leaves the cluster. A second possible cause is when a storage device has hardware trouble that causes it to become unavailable. In either case, because no cluster member has a path to the storage, the storage is unavailable and the domain panics.

Your first indication of a domain panic is likely to be I/O errors from the device, or panic messages written to the system console. Because the domain might be served by a cluster member that is still up, CFS commands such as `cfsmgr -e` might return a status of `OK` and not immediately reflect the problem condition.

```
# ls -l /mnt/mytst
/mnt/mytst: I/O error

# cfsmgr -e
Domain or filesystem name = mytest_dmn#mytst
Mounted On = /mnt/mytst
Server Name = deli
Server Status : OK
```

If you are able to restore connectivity to the device and return it to service, use the `cfsmgr` command to relocate the affected filesets in the domain to the same member that served them before the panic (or to another member) and then continue using the domain.

```
# cfsmgr -a SERVER=provolone -d mytest_dmn

# cfsmgr -e
Domain or filesystem name = mytest_dmn#mytests
Mounted On = /mnt/mytst
Server Name = provolone
Server Status : OK
```

11.1.8.3 Forcibly Unmounting an AdvFS File System or Domain

If you are not able to restore connectivity to the device and return it to service, TruCluster Server Version 5.1A includes the `cfsmgr -u` command that you can use to forcibly unmount an AdvFS file system or domain that is not being served by any cluster member. The unmount is not performed if the file system or domain is being served.

How you invoke this command depends on how the Cluster File System (CFS) currently views the domain:

- If the `cfsmgr -e` command indicates that the domain or file system is not served, use the `cfsmgr -u` command to forcibly unmount the domain or file system:

```
# cfsmgr -e
Domain or filesystem name = mytest_dmn#mytests
Mounted On = /mnt/mytst
Server Name = deli
Server Status : Not Served

# cfsmgr -u /mnt/mytst
```

- If the `cfsmgr -e` command indicates that the domain or file system is being served, you cannot use the `cfsmgr -u` command to unmount it because this command requires that the domain be not served.

  In this case, use the `cfsmgr` command to relocate the domain. Because the storage device is not available, the relocation fails; however, the operation changes the `Server Status` to `Not Served`.

  You can then use the `cfsmgr -u` command to forcibly unmount the domain.

```
# cfsmgr -e
Domain or filesystem name = mytest_dmn#mytests
Mounted On = /mnt/mytst
Server Name = deli
Server Status : OK

# cfsmgr -a SERVER=provolone -d mytest_dmn

# cfsmgr -e
Domain or filesystem name = mytest_dmn#mytests
Mounted On = /mnt/mytst
Server Status : Not Served

# cfsmgr -u /mnt/mytst
```

  You can also use the `cfsmgr -u -d` to forcibly unmount all mounted filesets in the domain.

```
# cfsmgr -u -d mytest_dmn
```

If there are nested mounts on the file system being unmounted, the forced unmount is not performed. Similarly, if there are nested mounts on any fileset when the entire domain is being forcibly unmounted, and the nested mount is not in the same domain, the forced unmount is not performed.

For detailed information on the cfsmgr command, see cfsmgr(8).

11.1.8.4 Avoiding Domain Panics

The AdvFS graphical user interface (GUI) agent, advfsd, periodically scans the system disks. If a metadata write error occurs, or if corruption is detected in a single AdvFS file domain, the advfsd daemon initiates a domain panic (rather than a system panic) on the file domain. This isolates the failed domain and allows a system to continue to serve all other domains.

From the viewpoint of the advfsd daemon running on a member of a cluster, any disk that contains an AdvFS domain and becomes inaccessible can trigger a domain panic. In normal circumstances, this is expected behavior. To diagnose such a panic, follow the instructions in the chapter on troubleshooting in the Tru64 UNIX *AdvFS Administration* manual. However, if a cluster member receives a domain panic because another member's private disk becomes unavailable (for instance, when that member goes down), the domain panic is an unnecessary distraction.

To avoid this type of domain panic, edit each member's /usr/var/advfs/daemon/disks.ignore file so that it lists the names of disks on other members' private storage that contain AdvFS domains. This will stop the advfsd daemon on the local member from scanning these devices.

To identify private devices, use the sms command to invoke the graphical interface for the SysMan Station, and then select Hardware from the Views menu.

11.1.9 Accessing Boot Partitions on Down Systems

When a member leaves the cluster, either cleanly through a shutdown or in an unplanned fashion, such as a panic, that member's boot partition is unmounted. If the boot partition is on the shared bus, any other member can gain access to the boot partition by mounting it.

Suppose the system provolone is down and you want to edit provolone's /etc/sysconfigtab. You can enter the following commands:

```
# mkdir /mnt
# mount root2_domain#root /mnt
```

Before rebooting provolone, you must unmount root2_domain#root. For example:

```
# umount root2_domain#root
```

11.1.10 Booting a Member While Its Boot Disk Is Already Mounted

Whenever the number of expected quorum votes or the quorum disk device is changed, the `/etc/sysconfigtab` file for each member is updated. In the case where a cluster member is down, the cluster utilities that affect quorum (`clu_add_member`, `clu_quorum`, `clu_delete_member`, and so forth) mount the down member's boot disk and make the update. If the down member tries to boot while its boot disk is mounted, it receives the following panic:

```
cfs_ mountroot: CFS server already exists for this nodes boot partition
```

The cluster utilities do the right thing and unmount the down member's boot disk after they complete the update.

In general, attempting to boot a member while another member has the first member's boot disk mounted causes the panic. For example, if you mount a down member's boot disk in order to make repairs, you generate the panic if you forget to unmount the boot disk before booting the repaired member.

11.1.11 Generating Crash Dumps

If a serious cluster problem occurs, crash dumps might be needed from all cluster members. To get crash dumps from functioning members, use the `dumpsys` command, which saves a snapshot of the system memory to a dump file.

To generate the crash dumps, log in to each running cluster member and run `dumpsys`. By default, `dumpsys` writes the dump to the member-specific directory `/var/adm/crash`.

For more information, see `dumpsys`(8).

11.1.12 Fixing Network Problems

This section describes potential networking problems in a cluster and solutions to resolve them.

Symptoms

- Cannot `ping` cluster
- Cannot `rlogin` to or from cluster
- Cannot `telnet` from cluster

Things to Verify

- Make sure that all cluster members are running `gated`.

Additionally, make sure that/etc/rc.config contains the following lines:

```
GATED="yes"
export GATED
```

- Make sure that /etc/rc.config contains the following lines:

```
ROUTER="yes"
export ROUTER
```

- Make sure that /etc/hosts has correct entries for the cluster default alias and cluster members.

  At a minimum, ensure that /etc/hosts has the following:

  – IP address and name for the cluster alias

Note

A cluster alias address should not be a broadcast address or a multicast address, nor can it reside in the subnet used by the cluster interconnect. In addition, although cluster members can use and advertise IPv6 addresess, they are not supported by the cluster alias subsystem. Therefore, you cannot assign IPv6 addresses to cluster aliases.

Although you can assign a cluster alias an IP address that resides in one of the private address spaces defined in RFC 1918, you must do the following in order for the alias subsystem to advertise a route to the alias address:

```
# rcmgr -c set CLUAMGR_ROUTE_ARGS resvok
# cluamgr -r resvok
```

Repeat the cluamgr command for each cluster member. For more information on the resvok flag, see cluamgr(8).

  – IP address and name for each cluster member

  – IP address and interface name associated with each member's cluster interconnect interface

  For example:

```
127.0.0.1       localhost
16.140.102.238  trees.tyron.com      trees
16.140.102.176  birch.tyron.com      birch
16.140.102.237  oak.tyron.com        oak
16.140.102.3    hickory.tyron.com    hickory
10.0.0.1        birch-mc0
10.0.0.2        oak-mc0
10.0.0.3        hickory-mc0
```

- Make sure `aliasd` is running on every cluster member.
- Make sure that all cluster members are members of the default alias (`joined` and `enabled`). You can verify this by entering the following command:

  ```
  #  cluamgr -s default_alias
  ```

  To make one member a member of the default alias, run the `cluamgr` command on that member. For example:

  ```
  #  cluamgr -a alias=default_alias,join
  ```

  Then on each member run the following command:

  ```
  # cluamgr -r start
  ```

- Make sure a member is routing for the default alias. You can verify this by running the following command on each member:

  ```
  # arp default_alias
  ```

  The result should include the phrase `permanent published`. One member should have a permanent published route for the cluster default alias.

- Make sure that the IP addresses of the cluster aliases are not already in use by another system.

  If you accidentally configure the cluster alias daemon, `aliasd`, with an alias IP address that is already used by another system, the cluster can experience connectivity problems: some machines might be able to reach the cluster alias and others might fail. Those that cannot reach the alias might appear to get connected to a completely different machine.

  An examination of the `arp` caches on systems that are outside the cluster might reveal that the affected alias IP address maps to two or more different hardware addresses.

  If the cluster is configured to log messages of severity `err`, then look at the system console and kernel log files for the following message:

  `local IP address` *nnn.nnn.nnn.nnn* `in use by hardware address` *xx-xx-xx-xx-xx*

After you have made sure that the entries in `/etc/rc.config` and `/etc/hosts` are correct and have fixed any other problems, try stopping and then restarting the `gateway` and `inet` daemons. Do this by entering the following commands on each cluster member:

```
# /sbin/init.d/gateway stop
# /sbin/init.d/gateway start
```

11.1.13 Running routed in a Cluster

Although it is technically possible to run `routed` in a cluster, doing so can cause the loss of failover support in the event of a cluster member failure. Running `routed` is considered a misconfiguration of the cluster and generates console and Event Manager (EVM) warning messages.

The only supported router is `gated`.

11.2 Hints for Managing Clusters

This section contains hints and suggestions for configuring and managing clusters.

11.2.1 Moving /tmp

By default, member-specific `/tmp` areas are in the same file system, but they can be moved to separate file systems. In some cases, you may want to move each member's `/tmp` area to a disk local to the member in order to reduce traffic on the shared SCSI bus.

If you want a cluster member to have its own `/tmp` directory on a private bus, you can create an AdvFS domain on a disk on the bus local to that cluster member and add an entry in `/etc/fstab` for that domain with a mountpoint of `/tmp`.

For example, the following `/etc/fstab` entries are for the `/tmp` directories for two cluster members, `tcr58` and `tcr59`, with member IDs of `58` and `59`, respectively.

```
tcr58_tmp#tmp   /cluster/members/member58/tmp   advfs rw 0 0
tcr59_tmp#tmp   /cluster/members/member59/tmp   advfs rw 0 0
```

The `tcr58_tmp` domain is on a bus that only member `tcr58` has connectivity to. The `tcr59_tmp` domain is on a disk that only member `tcr59` has connectivity to.

When each member boots, it attempts to mount all file systems in `/etc/fstab` but it can mount only those domains that are not already mounted and for which a path to the device exists. In this example, only `tcr58` can mount `tcr58_tmp#tmp` and only `tcr59` can mount `tcr59_tmp#tmp`.

You could have put the following in `/etc/fstab`:

```
tcr58_tmp#tmp   /tmp    advfs rw 0 0
tcr59_tmp#tmp   /tmp    advfs rw 0 0
```

Because /tmp is a context-dependent symbolic link (CDSL), it will be resolved to /cluster/members/member*n*/tmp. However, putting the full pathname in /etc/fstab is clearer and less likely to cause confusion.

11.2.2 Running the MC_CABLE Console Command

All members must be shut down to the console prompt before you run the MC_CABLE Memory Channel diagnostic command on any member. This is normal operation.

Running the MC_CABLE command from the console of a down cluster member when other members are up crashes the cluster.

11.2.3 Korn Shell Does Not Record True Path to Member-Specific Directories

The Korn shell (ksh) remembers the path that you used to get to a directory and returns that pathname when you enter a pwd command. This is true even if you are in some other location because of a symbolic link somewhere in the path. Because TruCluster Server uses CDSLs to maintain member-specific directories in a clusterwide namespace, the Korn shell does not return the true path when the working directory is a CDSL.

If you depend on the shell interpreting symbolic links when returning a pathname, use a shell other than the Korn shell. For example:

```
# ksh
# ls -l /var/adm/syslog
lrwxrwxrwx   1 root system  36 Nov 11 16:17 /var/adm/syslog
->../cluster/members/{memb}/adm/syslog
# cd /var/adm/syslog
# pwd
/var/adm/syslog
# sh
# pwd
/var/cluster/members/member1/adm/syslog
```

A

Cluster Events

Cluster events are Event Manager (EVM) events that are posted on behalf of the cluster, not for an individual member.

To get a list of all the cluster events, use the following command:

```
# evmwatch -i | evmshow -t "@name @cluster_event" | \
grep True$ | awk '{print $1}'
```

To get the EVM priority and a description of an event, use the following command:

```
# evmwatch -i -f '[name event_name]' | \
evmshow -t "@name @priority" -x
```

For example:

```
# evmwatch -i -f '[name sys.unix.clu.cfs.fs.served]' | \
evmshow -t "@name @priority" -x
sys.unix.clu.cfs.fs.served 200
    This event is posted by the cluster filesystem (CFS) to
    indicate that a filesystem has been mounted in the cluster,
    or that a filesystem for which this node is the server has
    been relocated or failed over.
```

For a description of EVM priorities, see EvmEvent(5). For more information on event management, see EVM(5).

B

Configuration Variables

Table B–1 contains a partial list of cluster configuration variables that can appear in the member-specific `rc.config` file.

After making a change to `rc.config` or `rc.config.common`, make the change active by rebooting each member individually.

For more information about `rc.config`, see Section 5.1.

Table B–1: Cluster Configuration Variables

Variable	Description
`CLU_BOOT_FILESYSTEM`	Specifies the domain and fileset for this member's boot disk.
`CLU_NEW_MEMBER`	Specifies whether this is the first time this member has booted. A value of 1 indicates a first boot. A value of 0 (zero) indicates the member has booted before.
`CLU_VERSION`	Specifies the version of the TruCluster Server software installed on the cluster.
`CLUSTER_NET`	Specifies the name of the system's primary network interface.
`IMC_AUTO_INIT`	When this variable is set to 1, the Memory Channel API library is automatically initialized at boot time. This initialization involves reserving approximately 4.5 MB for the Memory Channel API library. The default value of `IMC_AUTO_INIT` is 0.
`IMC_MAX_ALLOC`	Determines the maximum aggregate amount of Memory Channel address space the Memory Channel API library can allocate for its use across the cluster. If the value of this variable differs among cluster members, the largest value specified on any individual member determines the value set for the cluster. The default amount of address space is 10 MB.
`IMC_MAX_RECV`	Determines the maximum amount of physical memory the Memory Channel API library can map for reading Memory Channel address space. This limit is node-specific and can vary from member to member. The default amount of address space is 10 MB.

Table B–1: Cluster Configuration Variables (cont.)

Variable	Description
TCR_INSTALL	Indicates a successful installation when equal to TCR. Indicates an unsuccessful installation when equal to BAD.
TCR_PACKAGE	Indicates a successful installation when equal to TCR.

C

clu_delete_member Log

Each time that you run `clu_delete_member`, it writes log messages to `/cluster/admin/clu_delete_member.log`. The following is a sample `clu_delete_member` log file:

```
This is the TruCluster Delete Member Program

You will need the following information in order to delete a member from
the cluster:

    - Member ID (1-63)

The program will prompt for this information, offering a default
value when one is available.  To accept the default value, press Return
If you need help responding to a prompt, either type the word 'help'
or type a question mark (?) at the prompt.

The program does not begin to delete the member until you answer
all the prompts, and confirm that the answers are correct.

Deleting a member involves the following steps:

    Removing the files from the member boot disk. (If accessible)
    Removing member specific areas from the /, /usr, and /var file systems.
    Removing the deleted members entries in shared configuration files.

Do you want to continue deleting a member from this cluster? [yes]: y

A member ID is used to identify each member in a cluster.
Each member must have a unique member ID, which is an integer in
the range 1-63, inclusive.

Enter a cluster member ID []: 2
Checking cluster member ID: 2

You entered '2' as the member ID.
Is this correct? [yes]: y

You entered the following information:

    Member's ID: 2

If any of this information is incorrect, answer 'n' to the following
prompt.  You can then enter the correct information.

Do you want to continue to delete a cluster member?:[no] y

Deleting member disk boot partition files
  Member disk boot partition files deleted

Initial cluster deletion successful, member '2' can no longer join the
cluster. Deletion continuing with cleanup
cluster_expected_votes: reconfigured
```

```
Removing deleted member entries from shared configuration files
  Removing cluster interconnect interface 'polishham-mc0' from /.rhosts
 .
 .
 .

Deleting Member Specific Directories
  Deleting: /cluster/members/member2/
  Deleting: /usr/cluster/members/member2/
  Deleting: /var/cluster/members/member2/

clu_delete_member: The deletion of cluster member '2' completed successfully.
```

Index

Numbers and Special Characters

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y