HP AlphaServer SC

System Administration Guide

Order Number: AA-RLAWE-TE

November 2003

This document describes how to administer an AlphaServer SC system from the Hewlett-Packard Company (“HP”).

Revision/Update Information	This version supersedes the *HP AlphaServer SC System Administration Guide* issued in September 2002 for HP AlphaServer SC Version 2.5.
Operating System and Version:	HP Tru64 UNIX Version 5.1B, Patch Kit 3
Software Version:	Version 2.6
Maximum Node Count:	1024 nodes
Node Type:	HP AlphaServer ES45 HP AlphaServer ES40 HP AlphaServer DS20L

Legal Notices

The information in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY
3000 Hanover Street
Palo Alto, California 94304 U.S.A.

Use of this manual and media is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs, in their present form or with alterations, is expressly prohibited.

Copyright Notices

© Copyright 2002, 2003 Hewlett-Packard Development Company, L.P.

Some information in this document is based on Platform documentation, which includes the following copyright notice: Copyright 1994 - 2003 Platform Computing Corporation.

The Load Sharing Facility (LSF) software product is developed by Platform Computing Corporation ("Platform"). LSF has been licensed by Platform to HP for inclusion in HP AlphaServer SC systems.

The HP MPI software that is included in this HP AlphaServer SC software release is based on the MPICH V1.2.4 implementation of MPI, which includes the following copyright notice:

© 1993 University of Chicago
© 1993 Mississippi State University

Permission is hereby granted to use, reproduce, prepare derivative works, and to redistribute to others. This software was authored by:

Argonne National Laboratory Group
W. Gropp: (630) 252-4318; FAX: (630) 252-7852; e-mail: gropp@mcs.anl.gov
E. Lusk: (630) 252-5986; FAX: (630) 252-7852; e-mail: lusk@mcs.anl.gov
Mathematics and Computer Science Division, Argonne National Laboratory, Argonne IL 60439

Mississippi State Group
N. Doss and A. Skjellum: (601) 325-8435; FAX: (601) 325-8997; e-mail: tony@erc.msstate.edu
Mississippi State University, Computer Science Department & NSF Engineering Research Center for Computational Field Simulation, P.O. Box 6176, Mississippi State MS 39762

GOVERNMENT LICENSE

Portions of this material resulted from work developed under a U.S. Government Contract and are subject to the following license: the Government is granted for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable worldwide license in this computer software to reproduce, prepare derivative works, and perform publicly and display publicly.

DISCLAIMER

This computer code material was prepared, in part, as an account of work sponsored by an agency of the United States Government. Neither the United States, nor the University of Chicago, nor Mississippi State University, nor any of their employees, makes any warranty express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

Trademark Notices

Microsoft®, Windows®, and Windows NT® are trademarks of Microsoft Corporation in the U.S. and /or other countries.

UNIX® is a registered trademark of The Open Group in the U.S. and /or other countries.

The following product names refer to specific versions of products developed by Quadrics. These products combined with technologies from HP form an integral part of the supercomputing systems produced by HP and Quadrics. These products have been licensed by Quadrics to HP for inclusion in HP AlphaServer SC systems.

- *Interconnect hardware developed by Quadrics, including switches and adapter cards*
- *Elan, which describes the PCI host adapter for use with the interconnect technology developed by Quadrics*
- *PFS or Parallel File System*
- *RMS or Resource Management System*

Contents

Preface xxxiii

PART 1: SYSTEMWIDE ADMINISTRATION

1 HP AlphaServer SC System Overview

1.1 Configuration Overview 1–2
1.1.1 Assigning IP Addresses 1–10
1.1.1.1 Node IP Addresses 1–11
1.2 HP AlphaServer SC Nodes 1–12
1.3 Graphics Consoles 1–13
1.4 Domains 1–13
1.5 Local Disks 1–15
1.6 Console Network 1–15
1.7 Management LAN 1–16
1.8 HP AlphaServer SC Interconnect 1–16
1.8.1 Single-Rail Configurations and Dual-Rail Configurations 1–17
1.9 External Network 1–18
1.10 Management Server 1–18
1.11 Physical Storage 1–19
1.11.1 Local Storage 1–20
1.11.2 External Storage 1–20
1.12 Cluster File System (CFS) 1–21
1.13 Device Request Dispatcher (DRD) 1–23
1.14 Resource Management System (RMS) 1–24
1.15 Parallel File System (PFS) 1–25
1.16 SC File System (SCFS) 1–25
1.17 Managing an HP AlphaServer SC System 1–25
1.18 Monitoring System Activity 1–27
1.19 Differences between HP AlphaServer SC and TruCluster Server 1–28
1.19.1 Restrictions on TruCluster Server Features 1–29
1.19.2 Changes to TruCluster Server Utilities and Commands 1–29

2 Booting and Shutting Down the HP AlphaServer SC System

2.1 Booting the Entire HP AlphaServer SC System 2–2
2.1.1 Booting an HP AlphaServer SC System That Has a Single Management Server 2–3
2.1.2 Booting an HP AlphaServer SC System That Has a Clustered Management Server 2–3
2.1.3 Booting an HP AlphaServer SC System That Has No Management Server 2–4
2.2 Booting One or More Domains 2–4
2.3 Booting One or More Domain Members 2–5
2.4 The BOOT_RESET Console Variable 2–5
2.5 Booting a Domain Member to Single-User Mode 2–5
2.6 Rebooting an HP AlphaServer SC System 2–6
2.7 Defining a Node to be Not Bootable 2–6
2.8 Managing Boot Disks 2–7
2.8.1 The Alternate Boot Disk 2–8
2.8.2 Configuring and Using the Alternate Boot Disk 2–9
2.8.2.1 How to Use an Already-Configured Alternate Boot Disk 2–9
2.8.2.2 How to Configure and Use an Alternate Boot Disk After Installation 2–10
2.8.2.3 How to Stop Using the Alternate Boot Disk 2–12
2.8.3 Booting from the Alternate Boot Disk 2–12
2.8.4 The server_only Mount Option 2–13
2.8.5 Creating a New Boot Disk from the Alternate Boot Disk 2–14
2.9 Shutting Down the Entire HP AlphaServer SC System 2–15
2.9.1 Shutting Down an HP AlphaServer SC System That Has a Single Management Server 2–15
2.9.2 Shutting Down an HP AlphaServer SC System That Has a Clustered Management Server 2–16
2.9.3 Shutting Down an HP AlphaServer SC System That Has No Management Server 2–16
2.10 The Shutdown Grace Period 2–16
2.11 Shutting Down One or More Domain Members 2–17
2.11.1 Shutting Down One or More Non-Voting Members 2–17
2.11.2 Shutting Down Voting Members 2–17
2.12 Shutting Down a Domain Member to Single-User Mode 2–18
2.13 Resetting Members 2–19
2.14 Halting Members 2–19
2.15 Powering Off or On a Member 2–19
2.16 Configuring Nodes In or Out When Booting or Shutting Down 2–20

3 Managing the SC Database

3.1 Backing Up the SC Database 3–2
3.1.1 Back Up the Complete SC Database Using the rmsbackup Command 3–2
3.1.2 Back Up the SC Database, or a Table, Using the rmstbladm Command 3–3
3.1.3 Back Up the SC Database Directory 3–4
3.2 Reducing the Size of the SC Database by Archiving 3–4
3.2.1 Deciding What Data to Archive 3–4

3.2.2 Data Archived by Default . . . 3–5
3.2.3 The archive_tables Table . . . 3–5
3.2.3.1 Description of the archive_tables Table . . . 3–6
3.2.3.2 Adding Entries to the archive_tables Table . . . 3–6
3.2.3.3 Deleting Entries from the archive_tables Table . . . 3–6
3.2.3.4 Changing Entries in the archive_tables Table . . . 3–7
3.2.4 The rmsarchive Command . . . 3–7
3.3 Restoring the SC Database . . . 3–7
3.3.1 Restore the Complete SC Database . . . 3–8
3.3.2 Restore a Specific Table . . . 3–9
3.3.3 Restore the SC Database Directory . . . 3–9
3.3.4 Restore Archived Data . . . 3–10
3.4 Deleting the SC Database . . . 3–10
3.5 Monitoring /var . . . 3–12
3.6 Cookie Security Mechanism . . . 3–12
3.7 SC Database Integrity Check Utility . . . 3–13
3.7.1 Overview of the Database Integrity Check Utility . . . 3–13
3.7.2 Possible Causes of Inconsistency in the SC Database . . . 3–14
3.7.3 Consistency Checks Performed by the SC Database Integrity Check . . . 3–15
3.7.3.1 Consistency of System Attributes . . . 3–16
3.7.3.2 Consistency of the SRA Configuration . . . 3–17
3.7.3.3 Consistency of Terminal Servers/Ports/CMF Information . . . 3–18
3.7.3.4 Consistency of Disk and Storage Information . . . 3–19
3.7.3.5 Consistency of Information on AdvFS/PFS/SCFS File Systems . . . 3–20
3.7.3.6 Consistency of Information in the Monitoring Subsystem . . . 3–22
3.7.3.7 Consistency of HSV Information . . . 3–23
3.7.4 Running the SC Database Integrity Check . . . 3–23
3.7.4.1 sra dbcheck Syntax . . . 3–24
3.7.5 Output from the SC Database Integrity Check . . . 3–25
3.7.5.1 Example 1: Summary Output . . . 3–25
3.7.5.2 Example 2: Formatted Output . . . 3–27
3.7.6 Database Integrity Check Events and Automatic E-mail Alerts . . . 3–27

4 Managing the Load Sharing Facility (LSF)

4.1 Introduction to LSF . . . 4–2
4.1.1 Installing LSF on an HP AlphaServer SC System . . . 4–2
4.1.2 LSF Directory Structure on an HP AlphaServer SC System . . . 4–2
4.1.3 Compatibility with Earlier Releases of LSF . . . 4–3
4.1.4 Using NFS to Share LSF Configuration Information . . . 4–4
4.1.5 Using LSF Commands . . . 4–4
4.2 Setting Up Virtual Hosts . . . 4–5
4.3 Starting the LSF Daemons . . . 4–5
4.3.1 Starting the LSF Daemons on a Management Server or Single Host . . . 4–6

4.3.2 Starting the LSF Daemons on A Number of Real Hosts 4–6
4.3.3 Starting the LSF Daemons on a Virtual Host 4–6
4.3.4 Starting the LSF Daemons on a Number of Virtual Hosts 4–7
4.3.5 Checking that the LSF Daemons Are Running 4–7
4.4 Shutting Down the LSF Daemons 4–7
4.4.1 Shutting Down the LSF Daemons on a Management Server or Single Host 4–7
4.4.2 Shutting Down the LSF Daemons on a Number of Real Hosts 4–8
4.4.3 Shutting Down the LSF Daemons on a Virtual Host. 4–8
4.4.4 Shutting Down the LSF Daemons on A Number of Virtual Hosts 4–8
4.5 Checking the LSF Configuration 4–9
4.6 Setting Dedicated LSF Partitions 4–9
4.7 Customizing Job Control Actions (Optional) 4–9
4.8 Configuration Notes 4–10
4.8.1 Maximum Job Slot Limit 4–10
4.8.2 Per-Processor Job Slot Limit 4–10
4.8.3 Management Servers 4–10
4.8.4 Default Queue 4–11
4.8.5 Maximum Number of sbatchd Connections 4–11
4.9 LSF Topology Scheduler 4–11
4.9.1 Syntax 4–12
4.9.1.1 Allocation Type 4–12
4.9.1.2 Topology. 4–13
4.9.1.3 Flags 4–14
4.9.1.4 LSF Configuration Parameters 4–14
4.9.2 DEFAULT_EXTSCHED 4–14
4.9.3 MANDATORY_EXTSCHED 4–15
4.10 Operating LSF for HP AlphaServer SC 4–16
4.10.1 Remote LSF Adapter for RMS (RLA) 4–17
4.10.2 Running Jobs on Any Host Type 4–17
4.10.3 Using the brun Command to Run a Job on an RMS Host 4–18
4.10.4 LSF Licensing 4–18
4.10.4.1 How to Get Additional LSF Licenses. 4–18
4.10.5 Maintaining a Pool of Nodes in Reserve 4–19
4.10.6 Running Pre-Execution Programs 4–19
4.10.7 LSF Job Termination Reasons 4–20
4.10.7.1 Termination Reasons Displayed by the bacct Command 4–21
4.10.8 User Information for Interactive Batch Jobs. 4–24
4.10.9 Using LSF/HPC Extensions. 4–24
4.10.10 Limitations on User-Level Checkpointing 4–26
4.11 The lsf.conf File 4–26
4.11.1 LSB_RLA_UPDATE 4–27
4.11.2 LSB_RMSACCT_DELAY 4–27
4.11.3 LSB_RMS_MAXNUMNODES 4–27
4.11.4 LSB_RMS_MAXNUMRAILS 4–27

4.11.5 LSB_RMS_MAXPTILE 4–28
4.11.6 LSB_SHORT_HOSTLIST 4–28
4.11.7 LSF_ENABLE_EXTSCHEDULER 4–28
4.11.8 LSF_HPC_EXTENSIONS 4–28
4.11.9 RLA_RMS_ONLINENODES 4–29
4.12 Known Problems or Limitations 4–29

5 Managing the Resource Management System (RMS)

5.1 RMS Overview 5–2
5.1.1 RMS Concepts 5–2
5.1.2 RMS Tasks 5–3
5.2 RMS Accounting 5–3
5.2.1 Accessing Accounting Data 5–6
5.3 Monitoring RMS 5–6
5.3.1 rinfo 5–6
5.3.2 rcontrol 5–8
5.3.3 rmsquery 5–8
5.4 Basic Partition Management 5–8
5.4.1 Creating Partitions 5–9
5.4.2 Specifying Configurations 5–10
5.4.3 Starting Partitions 5–12
5.4.4 Reloading Partitions 5–13
5.4.5 Stopping Partitions 5–13
5.4.6 Deleting Partitions 5–15
5.5 Resource and Job Management 5–15
5.5.1 Resource and Job Concepts 5–16
5.5.2 Viewing Resources and Jobs 5–17
5.5.3 Suspending Resources 5–20
5.5.4 Killing and Signalling Resources 5–21
5.5.5 Resource Cleanup 5–22
5.5.6 Running Jobs as Root 5–23
5.5.7 Managing Exit Timeouts 5–24
5.5.8 Idle Timeout 5–24
5.5.9 Managing Core Files 5–25
5.5.9.1 Location of Core Files 5–25
5.5.9.2 Backtrace Printing 5–26
5.5.9.3 Preservation and Cleanup of Core Files 5–27
5.5.10 Resources and Jobs during Node and Partition Transitions 5–28
5.5.10.1 Partition Transition 5–28
5.5.10.2 Node Transition 5–32
5.5.10.3 Orphan Job Cleanup 5–33
5.5.11 Running User-Defined Scripts 5–34
5.6 Advanced Partition Management 5–35

5.6.1 Partition Types . . . 5–35
5.6.2 Controlling User Access to Partitions . . . 5–37
5.6.2.1 Concepts . . . 5–37
5.6.2.2 RMS Projects and Access Controls Menu . . . 5–38
5.6.2.3 Using the rcontrol Command . . . 5–43
5.6.3 Controlling Whether the Partition Uses HP AlphaServer SC Elan Adapter Cards . . . 5–45
5.7 Controlling Resource Usage . . . 5–45
5.7.1 Resource Priorities . . . 5–46
5.7.2 Memory Limits . . . 5–47
5.7.2.1 Memory Limits Overview . . . 5–47
5.7.2.2 Setting Memory Limits . . . 5–49
5.7.2.3 Memory Limits Precedence Order . . . 5–49
5.7.2.4 How Memory Limits Affect Resource and Job Scheduling . . . 5–50
5.7.2.5 Memory Limits Applied to Processes . . . 5–51
5.7.3 Minimum Number of CPUs . . . 5–53
5.7.4 Maximum Number of CPUs . . . 5–53
5.7.5 Time Limits . . . 5–55
5.7.6 Enabling Timesliced Gang Scheduling . . . 5–56
5.7.7 Partition Queue Depth . . . 5–59
5.7.8 Processor Sets . . . 5–60
5.8 Node Management . . . 5–61
5.8.1 Configure Nodes In or Out . . . 5–61
5.8.2 Booting Nodes . . . 5–63
5.8.3 Shutting Down Nodes . . . 5–63
5.8.4 Node Failure . . . 5–64
5.8.4.1 Node Status . . . 5–64
5.8.4.2 Partition Status . . . 5–65
5.8.5 Draining Nodes . . . 5–66
5.9 RMS Servers and Daemons . . . 5–67
5.9.1 Overview . . . 5–67
5.9.2 Stopping the RMS System and mSQL . . . 5–69
5.9.3 Manually Starting RMS . . . 5–71
5.9.4 Stopping and Starting RMS Servers . . . 5–72
5.9.5 Running the Switch Manager . . . 5–72
5.9.6 Log Files . . . 5–73
5.10 Site-Specific Modifications to RMS: the pstartup Script . . . 5–74
5.11 RMS and CAA Failover Capability . . . 5–74
5.11.1 Determining Whether RMS is Set Up for Failover . . . 5–74
5.11.2 Removing CAA Failover Capability from RMS . . . 5–75
5.12 Using Dual Rail . . . 5–76
5.13 Useful SQL Commands . . . 5–76

6 Overview of File Systems and Storage

6.1 Introduction . . . 6–2
6.2 Changes in HP AlphaServer SC File Systems in Version 2.6 . . . 6–2
6.3 SCFS . . . 6–3
6.3.1 Selection of FAST Mode . . . 6–4
6.3.2 Getting the Most Out of SCFS . . . 6–4
6.4 PFS . . . 6–5
6.4.1 PFS and SCFS . . . 6–6
6.4.1.1 User Process Operation . . . 6–7
6.4.1.2 System Administrator Operation . . . 6–7
6.5 Preferred File Server Nodes and Failover . . . 6–8
6.6 Storage Overview . . . 6–9
6.6.1 Local or Internal Storage . . . 6–10
6.6.1.1 Using Local Storage for Application I/O . . . 6–10
6.6.2 Global or External Storage . . . 6–11
6.6.2.1 System Storage . . . 6–12
6.6.2.2 Data Storage . . . 6–13
6.6.2.3 External Storage Hardware Products . . . 6–13
6.7 External Data Storage Configuration . . . 6–13
6.7.1 MSA1000 Controllers — Active/Standby Mode . . . 6–13
6.7.2 HSG-Based Storage Arrays — Multiple-Bus Failover Mode . . . 6–15
6.7.3 HSV-Based Storage Arrays — Multipathing Support . . . 6–17

7 Managing the SC File System (SCFS)

7.1 SCFS Overview . . . 7–2
7.2 SCFS Configuration Attributes . . . 7–3
7.2.1 Changing SCFS File System Mode (FAST/UBC) . . . 7–7
7.3 Creating SCFS File Systems . . . 7–7
7.4 The scfsmgr Command . . . 7–8
7.4.1 scfsmgr create . . . 7–9
7.4.1.1 Specifying Mount Options . . . 7–11
7.4.2 scfsmgr destroy . . . 7–13
7.4.3 scfsmgr export . . . 7–14
7.4.4 scfsmgr offline . . . 7–15
7.4.5 scfsmgr online . . . 7–15
7.4.6 scfsmgr scan . . . 7–16
7.4.7 scfsmgr server . . . 7–16
7.4.8 scfsmgr show . . . 7–17
7.4.9 scfsmgr status . . . 7–18
7.4.10 scfsmgr sync . . . 7–19
7.4.11 scfsmgr sync_domain . . . 7–19
7.4.12 scfsmgr upgrade . . . 7–20
7.4.13 scfsmgr try_mount . . . 7–20

7.4.14 scfsmgr try_unmount 7–21
7.4.15 scfsmgr Quota Commands 7–22
7.4.15.1 scfsmgr quota 7–23
7.4.15.2 scfsmgr quotaon 7–24
7.4.15.3 scfsmgr quotaoff 7–24
7.4.15.4 scfsmgr setquota 7–25
7.4.15.5 scfsmgr repquota 7–26
7.5 Access Control Lists 7–27
7.6 SysMan Menu 7–27
7.7 Monitoring and Correcting File-System Failures 7–28
7.7.1 Overview of the File-System Management System 7–28
7.7.2 Monitoring File-System State 7–29
7.7.3 File-System Events 7–30
7.7.4 Interpreting and Correcting File-System Failures 7–30
7.8 Tuning SCFS 7–32
7.8.1 Tuning SCFS Kernel Subsystems 7–32
7.8.2 Tuning SCFS Server Operations 7–32
7.8.2.1 SCFS I/O Transfers 7–32
7.8.2.2 SCFS Synchronization Management 7–33
7.8.2.3 SCFS Multiple Rails 7–34
7.8.3 Tuning SCFS Client Operations 7–34
7.8.4 Monitoring SCFS Activity 7–35
7.9 SC Database Tables Supporting SCFS File Systems 7–35
7.9.1 The sc_scfs Table 7–36
7.9.2 The sc_scfs_mount Table 7–37
7.9.3 The sc_advfs_vols Table 7–38
7.9.4 The sc_advfs_filesets Table 7–38
7.9.5 The sc_disk Table 7–38
7.9.6 The sc_disk_server Table 7–39
7.9.7 The sc_lsm_vols Table 7–40

8 Managing the Parallel File System (PFS)

8.1 PFS Overview 8–2
8.1.1 PFS Attributes 8–3
8.1.2 Structure of a PFS Component File System 8–4
8.1.3 Storage Capacity of a PFS File System 8–4
8.2 Planning a PFS File System to Maximize Performance 8–5
8.3 Managing a PFS File System 8–7
8.3.1 Creating and Mounting a PFS File System 8–7
8.3.1.1 Example 1: Four-Component PFS File System — /scratch 8–8
8.3.1.2 Example 2: 32-Component PFS File System — /data3t 8–9
8.3.2 Increasing the Capacity of a PFS File System 8–10
8.3.3 Upgrading a PFS File System 8–11

8.3.4 Checking a PFS File System . . . 8–11
8.3.5 Exporting a PFS File System . . . 8–12
8.3.6 Backing up a PFS File System . . . 8–12
8.4 The PFS Management Utility: pfsmgr . . . 8–13
8.4.1 PFS Configuration Attributes . . . 8–13
8.4.2 pfsmgr Commands . . . 8–14
8.4.2.1 pfsmgr create . . . 8–15
8.4.2.2 pfsmgr delete . . . 8–16
8.4.2.3 pfsmgr offline . . . 8–16
8.4.2.4 pfsmgr online . . . 8–17
8.4.2.5 pfsmgr show . . . 8–18
8.4.2.6 pfsmgr try_mount . . . 8–19
8.4.2.7 pfsmgr try_unmount . . . 8–20
8.4.3 Managing PFS File Systems Using sysman . . . 8–21
8.5 Using a PFS File System . . . 8–22
8.5.1 Creating PFS Files . . . 8–23
8.5.2 Optimizing a PFS File System . . . 8–23
8.5.3 PFS Ioctl Calls . . . 8–25
8.5.3.1 PFSIO_GETFSID . . . 8–26
8.5.3.2 PFSIO_GETMAP . . . 8–26
8.5.3.3 PFSIO_SETMAP . . . 8–26
8.5.3.4 PFSIO_GETDFLTMAP . . . 8–26
8.5.3.5 PFSIO_SETDFLTMAP . . . 8–27
8.5.3.6 PFSIO_GETFSMAP . . . 8–27
8.5.3.7 PFSIO_GETLOCAL . . . 8–28
8.5.3.8 PFSIO_GETFSLOCAL . . . 8–29
8.6 SC Database Tables Supporting PFS File Systems . . . 8–29
8.6.1 The sc_pfs Table . . . 8–30
8.6.2 The sc_pfs_mount Table . . . 8–30
8.6.3 The sc_pfs_components Table . . . 8–31
8.6.4 The sc_pfs_filesystems Table . . . 8–31

9 Managing Events

9.1 Event Overview . . . 9–2
9.1.1 Event Category . . . 9–3
9.1.2 Event Class . . . 9–3
9.1.3 Event Severity . . . 9–6
9.2 HP AlphaServer SC Event Filter Syntax . . . 9–6
9.2.1 Filter Operators . . . 9–9
9.3 Viewing Events . . . 9–9
9.3.1 Using the SC Viewer to View Events . . . 9–9
9.3.2 Using the scevent Command to View Events . . . 9–9
9.3.2.1 scevent Command Syntax . . . 9–10

9.4 Event Examples 9–10
9.5 Notification of Events 9–13
9.5.1 Using the scalertmgr Command 9–13
9.5.1.1 Add an Alert 9–14
9.5.1.2 Remove an Alert 9–14
9.5.1.3 List the Existing Alerts 9–14
9.5.1.4 Change the E-Mail Addresses Associated with Existing Alerts 9–15
9.5.1.5 Example E-Mail Alert 9–15
9.5.2 Event Handlers 9–16
9.5.2.1 rmsevent_node Event Handler 9–17
9.5.2.2 rmsevent_env Event Handler 9–17
9.5.2.3 rmsevent_escalate Event Handler 9–18
9.6 Event Handler Scripts 9–18

10 Viewing System Status

10.1 SC Viewer 10–2
10.1.1 Invoking SC Viewer 10–2
10.1.2 SC Viewer Menus 10–3
10.1.2.1 The File Menu 10–3
10.1.2.2 The View Menu 10–4
10.1.2.3 The Help Menu 10–4
10.1.3 SC Viewer Icons 10–4
10.1.3.1 Overview Status Icons 10–5
10.1.3.2 Object Icons 10–6
10.1.3.3 Status Icons 10–6
10.1.3.4 Event Severity Icons 10–7
10.1.3.5 Object Panels 10–7
10.1.4 SC Viewer Tabs 10–9
10.1.5 Properties Pane 10–11
10.2 Overview Tab 10–12
10.3 Failures Tab 10–14
10.4 Domains Tab 10–16
10.4.1 Nodes Window 10–17
10.5 Infrastructure Tab 10–19
10.5.1 Switch 10–20
10.5.2 Terminal Server 10–21
10.5.3 SANworks Management Appliance 10–22
10.5.4 HSG80 RAID System 10–23
10.5.5 HSV110 RAID System 10–24
10.6 Physical Tab 10–25
10.7 Events Tab 10–27
10.8 Interconnect Tab 10–30

11 SC Performance Visualizer

11.1 Using SC Performance Visualizer . . . 11–2
11.2 Personal Preferences . . . 11–2
11.3 Online Help . . . 11–2
11.4 The scload Command . . . 11–3
11.4.1 scload Options . . . 11–3
11.4.2 scload Metrics . . . 11–4
11.4.3 Example scload Output . . . 11–5
11.4.3.1 Resource Output . . . 11–5
11.4.3.2 Overlapping Resource Output . . . 11–7
11.4.3.3 Domain-Level Output . . . 11–7

12 Managing Multiple Domains

12.1 Overview of the scrun Command . . . 12–2
12.2 scrun Command Syntax . . . 12–2
12.3 scrun Examples . . . 12–6
12.4 Interrupting an scrun Command . . . 12–7

13 User Administration

13.1 Adding Local Users . . . 13–2
13.2 Removing Local Users . . . 13–2
13.3 Managing Local Users Across Domains . . . 13–3
13.4 Managing User Home Directories . . . 13–3

14 Managing the Console Network

14.1 Console Network Configuration . . . 14–2
14.2 Console Logger Daemon (cmfd) . . . 14–2
14.3 Configurable CMF Information in the SC Database . . . 14–4
14.4 Console Logger Configuration and Output Files . . . 14–5
14.5 Console Log Files . . . 14–8
14.6 Configuring the Terminal-Server Ports . . . 14–9
14.7 Reconfiguring or Replacing a Terminal Server . . . 14–9
14.8 Manually Configuring a Terminal-Server Port . . . 14–10
14.9 Changing the Terminal-Server Password . . . 14–12
14.10 Configuring the Terminal-Server Ports for New Members . . . 14–12
14.11 Starting and Stopping the Console Logger . . . 14–13
14.12 User Communication with the Terminal Server . . . 14–14
14.12.1 Disconnect a User Connection from CMF . . . 14–14
14.12.2 Disconnect a Connection Between CMF and the Terminal Server . . . 14–14
14.12.3 Bypass CMF and Log Out a Terminal Server Port . . . 14–14
14.13 Backing Up or Deleting Console Log Files . . . 14–15

14.14 Connecting to a Node's Console . . . 14–15
14.15 Connecting to a Terminal Server . . . 14–16
14.16 Monitoring a Node's Console Output . . . 14–16
14.17 Changing the CMF Port Number . . . 14–17
14.18 CMF and CAA Failover Capability . . . 14–17
14.18.1 Determining Whether CMF is Set Up for Failover . . . 14–18
14.18.2 Enabling CMF as a CAA Application . . . 14–18
14.18.3 Disabling CMF as a CAA Application . . . 14–20
14.19 Changing the CMF Host . . . 14–21

15 System Log Files

15.1 Log Files Overview . . . 15–2
15.2 LSF Log Files . . . 15–3
15.3 RMS Log Files . . . 15–3
15.4 System Event Log Files . . . 15–4
15.5 Crash Dump Log Files . . . 15–4
15.6 Console Log Files . . . 15–4
15.7 Log Files Created by sra Commands . . . 15–6
15.8 SCFS and PFS File-System Management Log Files . . . 15–7

16 The sra Command

16.1 sra . . . 16–2
16.1.1 Nodes, Domains, and Members . . . 16–3
16.1.2 Syntax . . . 16–4
16.1.3 Description . . . 16–8
16.1.4 Options . . . 16–12
16.1.5 Error Messages From sra console Command . . . 16–20
16.1.6 The sramon Command . . . 16–20
16.2 sra edit . . . 16–22
16.2.1 Usage . . . 16–23
16.2.2 Node Submenu . . . 16–24
16.2.2.1 Show Node Attributes . . . 16–25
16.2.2.2 Add Nodes to, and Delete Nodes from, the SC Database . . . 16–27
16.2.2.3 Edit Node Attributes . . . 16–28
16.2.3 System Submenu . . . 16–30
16.2.3.1 Show System Attributes . . . 16–31
16.2.3.2 Edit System Attributes . . . 16–34
16.2.3.3 Update System Files and Restart Daemons . . . 16–37
16.2.3.4 Add or Delete a Terminal Server, Image, or Cluster . . . 16–38
16.3 sra-display . . . 16–39

PART 2: DOMAIN ADMINISTRATION

17 Overview of Managing Domains

17.1 Commands and Utilities for Domains . . . 17–2
17.2 Commands Not Supported in HP AlphaServer SC . . . 17–3
17.3 Tru64 UNIX Commands that are Different in a Domain . . . 17–4

18 Tools for Managing Domains

18.1 Introduction . . . 18–2
18.1.1 Domain Tools Quick Start . . . 18–2
18.2 Domain Configuration Tools and SysMan . . . 18–3
18.3 SysMan Management Options . . . 18–4
18.3.1 Introduction to SysMan Menu . . . 18–4
18.3.2 Introduction to the SysMan Command Line . . . 18–5
18.4 Using SysMan Menu in a Domain . . . 18–5
18.4.1 Getting in Focus . . . 18–5
18.4.2 Specifying a Focus on the Command Line . . . 18–6
18.4.3 Invoking SysMan Menu . . . 18–6
18.5 Using the SysMan Command-Line Interface in a Domain . . . 18–7

19 Managing the Cluster Alias Subsystem

19.1 Summary of Alias Features . . . 19–2
19.2 Configuration Files . . . 19–5
19.3 Planning for Cluster Aliases . . . 19–6
19.4 Preparing to Create Cluster Aliases . . . 19–8
19.5 Specifying and Joining a Cluster Alias . . . 19–9
19.6 Modifying Cluster Alias and Service Attributes . . . 19–11
19.7 Monitoring Cluster Aliases . . . 19–12
19.8 Leaving a Cluster Alias . . . 19–12
19.9 Deleting a Cluster Alias . . . 19–13
19.10 Modifying Domainwide Port Space . . . 19–14
19.11 Changing the Default Cluster Alias IP Name . . . 19–15
19.12 Changing the Default Cluster Alias IP Address . . . 19–18
19.13 Cluster Alias and NFS . . . 19–19
19.14 Cluster Alias and Cluster Application Availability . . . 19–20
19.15 Preferred Server Cluster Aliases . . . 19–23
19.15.1 Overview . . . 19–24
19.15.2 Availability Configuration . . . 19–24
19.15.3 Proxy ARP Master Configuration . . . 19–25
19.15.4 Routing Configuration . . . 19–25
19.15.5 NFS Export Configuration . . . 19–26

19.15.6 Validating the Preferred Server Cluster Aliases Configuration 19–26
19.15.7 Possible Customizations for Preferred Server Cluster Aliases 19–27
19.15.8 Debugging Tools for Preferred Server Cluster Aliases 19–28
19.16 Third-Party License Managers 19–29

20 Managing Domain Membership

20.1 Connection Manager 20–2
20.2 Quorum and Votes 20–2
20.2.1 How a Node Becomes a Domain Member 20–3
20.2.2 Expected Votes 20–3
20.2.3 Current Votes 20–4
20.2.4 Node Votes 20–4
20.3 Calculating Quorum 20–5
20.4 A Connection Manager Example 20–7
20.5 The clu_quorum Command 20–10
20.5.1 Using the clu_quorum Command to Manage Cluster Votes 20–10
20.5.2 Using the clu_quorum Command to Display Cluster Vote Information 20–10
20.6 Monitoring the Connection Manager 20–12
20.7 Connection Manager Panics 20–13
20.8 Troubleshooting 20–13

21 Managing Domain Members

21.1 Managing Configuration Variables 21–2
21.2 Managing Kernel Attributes 21–3
21.3 Managing Remote Access Within and From the Domain 21–5
21.4 Adding Domain Members After Installation 21–6
21.4.1 Adding Domain Members After Initial Installation — Phased Installation 21–6
21.4.2 Adding Domain Members After Initial Installation — System Expansion 21–7
21.4.3 Changing the Interconnect Nodeset Mask 21–9
21.5 Converting a CS Domain to an FS Domain 21–12
21.6 Deleting a Domain Member 21–15
21.7 Adding a Deleted Member Back into the Domain 21–17
21.8 Reinstalling a Domain 21–17
21.8.1 Reinstalling a Domain From the Tru64 UNIX Disk 21–17
21.8.2 Reinstalling a Domain Completely 21–18
21.9 Moving a Domain to Another IP Subnet 21–18
21.9.1 File Edit Tables 21–22
21.9.1.1 Changing External IP Addresses Only 21–22
21.9.1.2 Changing External IP Addresses and Host Names 21–23
21.9.1.3 Changing External and Internal IP Addresses and Host Names 21–24
21.9.2 Attribute and Checklist Tables 21–26
21.9.2.1 External Host Names and IP Addresses 21–26

21.9.2.2 Domain Name and Cluster Aliases 21–27
21.9.2.3 Interface IP Aliases 21–27
21.9.2.4 External Servers 21–28
21.9.2.5 Checklist 21–29
21.9.3 Verifying Success 21–31
21.9.4 Troubleshooting 21–32
21.10 Updating ifaccess.conf When Changing Network Interfaces 21–32
21.11 Managing Software Licenses 21–34
21.12 Updating System Firmware 21–34
21.12.1 Updating System Firmware When Using a Management Server 21–35
21.12.2 Updating System Firmware When Not Using a Management Server 21–35
21.13 Updating the Generic Kernel After Installation 21–36
21.14 Changing a Node's Ethernet Card 21–36
21.15 Managing Swap Space 21–37
21.15.1 Increasing Swap Space 21–38
21.15.1.1 Increasing Swap Space on an HP AlphaServer DS20L Node 21–39
21.15.1.2 Increasing Swap Space by Resizing the Primary Boot Disk 21–39
21.15.1.3 Increasing Swap Space by Resizing the Alternate Boot Disk 21–40
21.16 Installing and Deleting Layered Applications 21–42
21.16.1 Installing an Application 21–42
21.16.2 Deleting an Application 21–42
21.17 Managing Accounting Services 21–43
21.17.1 Setting Up HP AlphaServer SC-Specific UNIX Accounting 21–44
21.17.2 Setting Up HP AlphaServer SC-Specific UNIX Accounting on a Newly Added Member 21–46
21.17.3 Removing HP AlphaServer SC-Specific UNIX Accounting 21–46

22 Networking and Network Services

22.1 Default Route Configuration 22–2
22.1.1 Validating the Default Route Configuration 22–4
22.2 Running IP Routers 22–6
22.3 Configuring the Network 22–7
22.4 Configuring DNS/BIND 22–8
22.5 Managing Time Synchronization 22–9
22.5.1 Configuring NTP 22–9
22.5.2 All Members Should Use the Same External NTP Servers 22–10
22.5.2.1 Time Drift 22–10
22.6 Configuring NFS 22–11
22.6.1 The HP AlphaServer SC System as an NFS Client 22–11
22.6.2 The HP AlphaServer SC System as an NFS Server 22–13
22.6.3 How to Configure NFS 22–13
22.6.4 Considerations for Using NFS in a Domain 22–15
22.6.4.1 CFS Support of NFS File Systems 22–15

22.6.4.2 Clients Must Use a Cluster Alias . . . 22–15
22.6.4.3 Loopback Mounts Are Not Supported . . . 22–15
22.6.4.4 Do Not Mount Non-NFS File Systems on NFS-Mounted Paths . . . 22–16
22.6.4.5 Use AutoFS to Mount File Systems . . . 22–16
22.6.5 Mounting NFS File Systems using AutoFS . . . 22–16
22.6.6 Forcibly Unmounting File Systems Mounted by AutoFS . . . 22–18
22.6.6.1 Determining Whether a Forced Unmount is Required . . . 22–18
22.6.6.2 Correcting the Problem . . . 22–19
22.7 Configuring NIS . . . 22–21
22.7.1 Configuring a NIS Master in a Domain with Enhanced Security . . . 22–22
22.8 Managing Mail . . . 22–22
22.8.1 Configuring Mail . . . 22–23
22.8.2 Mail Files . . . 22–23
22.8.3 The Cw Macro (System Nicknames List) . . . 22–24
22.8.4 Configuring Mail at Domain Creation . . . 22–24
22.9 Managing inetd Configuration . . . 22–25
22.10 Displaying X Window Applications Remotely . . . 22–25

23 Managing Highly Available Applications

23.1 Introduction . . . 23–2
23.2 Learning the Status of a Resource . . . 23–3
23.2.1 Learning the State of a Resource . . . 23–5
23.2.2 Learning Status of All Resources on One Domain Member . . . 23–6
23.2.3 Learning Status of All Resources on All Domain Members . . . 23–7
23.2.4 Getting Number of Failures and Restarts and Target States . . . 23–8
23.3 Reporting Resource Availability Measurements . . . 23–9
23.4 Relocating Applications . . . 23–11
23.4.1 Manual Relocation of All Applications on a Domain Member . . . 23–11
23.4.2 Manual Relocation of a Single Application . . . 23–12
23.4.3 Manual Relocation of Dependent Applications . . . 23–12
23.5 Starting and Stopping Application Resources . . . 23–13
23.5.1 Starting Application Resources . . . 23–13
23.5.2 Stopping Application Resources . . . 23–14
23.5.3 No Multiple Instances of an Application Resource . . . 23–15
23.5.4 Using caa_stop to Reset UNKNOWN State . . . 23–15
23.6 Balancing Application Resources . . . 23–15
23.6.1 Balancing All Applications on a Domain . . . 23–16
23.6.2 Balancing All Applications on a Member . . . 23–16
23.6.3 Balancing Specific Applications . . . 23–17
23.6.4 Balancing an Application at Specific Time . . . 23–17
23.7 Registering and Unregistering Resources . . . 23–18
23.7.1 Registering Resources . . . 23–18
23.7.2 Unregistering Resources . . . 23–18

23.7.3 Updating Registration 23–19
23.8 HP AlphaServer SC Resources 23–19
23.9 Managing Network, Tape, and Media Changer Resources 23–20
23.10 Managing CAA with SysMan Menu 23–21
23.10.1 CAA Management Dialog Box 23–23
23.10.1.1 Start Dialog Box 23–24
23.10.1.2 Setup Dialog Box 23–25
23.11 Understanding CAA Considerations for Startup and Shutdown 23–25
23.12 Managing the CAA Daemon (caad) 23–26
23.12.1 Determining Status of the Local CAA Daemon 23–26
23.12.2 Restarting the CAA Daemon 23–27
23.12.3 Monitoring CAA Daemon Messages 23–27
23.13 Using EVM to View CAA Events 23–27
23.13.1 Viewing CAA Events 23–28
23.13.2 Monitoring CAA Events 23–29
23.14 Troubleshooting with Events 23–29
23.14.1 Action Script Has Timed Out 23–29
23.14.2 Action Script Stop Entry Point Not Returning 0 23–30
23.14.3 Network Failure 23–30
23.14.4 Lock Preventing Start of CAA Daemon 23–30
23.15 Troubleshooting a Command-Line Message 23–30

24 Managing the Cluster File System (CFS), the Advanced File System (AdvFS), and Devices

24.1 CFS Overview 24–2
24.1.1 File System Topology 24–4
24.2 Working with CDSLs 24–4
24.2.1 Making CDSLs 24–5
24.2.2 Maintaining CDSLs 24–6
24.2.3 Kernel Builds and CDSLs 24–6
24.2.4 Exporting and Mounting CDSLs 24–6
24.3 Managing Devices 24–7
24.3.1 The Hardware Management Utility (hwmgr) 24–8
24.3.2 The Device Special File Management Utility (dsfmgr) 24–8
24.3.3 The Device Request Dispatcher Utility (drdmgr) 24–8
24.3.3.1 Direct-Access I/O and Single-Server Devices 24–9
24.3.3.2 Devices Supporting Direct-Access I/O 24–10
24.3.3.3 Replacing RZ26, RZ28, RZ29, or RZ1CB-CA as Direct-Access I/O Disks 24–10
24.3.3.4 HSZ Hardware Supported on Shared Buses 24–10
24.3.4 Detecting Storage Devices Using SRM Commands 24–11
24.3.5 Determining Device Locations 24–11
24.3.6 Adding a Disk to the Domain 24–12
24.3.7 Managing Third-Party Storage 24–13

24.3.8 Replacing a Failed Disk . . . 24–14
24.3.9 Diskettes. . . . 24–15
24.3.10 CD-ROM and DVD-ROM. . . . 24–15
24.4 Managing the Cluster File System . . . 24–15
24.4.1 Mounting CFS File Systems . . . 24–15
24.4.1.1 fstab and member_fstab Files . . . 24–16
24.4.1.2 Start Up Scripts. . . . 24–16
24.4.1.3 CFS Load Balancing . . . 24–18
24.4.1.4 Distributing File Systems *via* the mount -o Command . . . 24–19
24.4.2 Freezing an AdvFS Domain Before Cloning . . . 24–21
24.4.2.1 Determining Whether an AdvFS Domain Is Frozen . . . 24–22
24.4.3 File System Availability . . . 24–22
24.4.3.1 When File Systems Cannot Fail Over . . . 24–23
24.4.4 Optimizing CFS — Locating and Migrating File Servers . . . 24–24
24.4.4.1 Automatically Distributing CFS Server Load . . . 24–25
24.4.4.2 Tuning the Block Transfer Size. . . . 24–25
24.4.4.3 Changing the Number of Read-Ahead and Write-Behind Threads. . . . 24–26
24.4.4.4 Taking Advantage of Direct I/O . . . 24–27
24.4.4.5 Adjusting CFS Memory Usage. . . . 24–31
24.4.4.6 Using Memory Mapped Files . . . 24–33
24.4.4.7 Avoid Full File Systems . . . 24–33
24.4.4.8 Other Strategies . . . 24–33
24.4.5 MFS and UFS File Systems Supported . . . 24–33
24.4.6 Partitioning File Systems . . . 24–34
24.4.7 Block Devices and Cache Coherency . . . 24–36
24.4.8 CFS Restrictions . . . 24–36
24.5 Managing AdvFS in a Domain . . . 24–37
24.5.1 Create Only One Fileset in Cluster Root AdvFS Domain . . . 24–37
24.5.2 Adding Filesets to a Member's Boot Partition Not Recommended . . . 24–37
24.5.3 Do Not Add a Volume to a Member's Root AdvFS Domain . . . 24–38
24.5.4 Using the addvol and rmvol Commands in a Domain. . . . 24–38
24.5.5 User and Group File Systems Quotas Are Supported . . . 24–39
24.5.5.1 Quota Hard Limits . . . 24–40
24.5.5.2 Setting the quota_excess_blocks Value. . . . 24–41
24.5.6 Storage Connectivity and AdvFS Volumes . . . 24–42
24.6 Considerations When Creating New File Systems . . . 24–42
24.6.1 Checking for Disk Connectivity . . . 24–43
24.6.2 Checking for Available Disks. . . . 24–43
24.6.2.1 Checking for Member Boot Disks and Domainwide AdvFS File Systems . . . 24–44
24.6.2.2 Checking for Member Swap Areas . . . 24–45
24.7 Backing Up and Restoring Files . . . 24–45
24.7.1 Suggestions for Files to Back Up . . . 24–46
24.7.2 Booting the Domain Using the Backup Cluster Disk . . . 24–46
24.8 Managing CDFS File Systems. . . . 24–47

24.9 Using the verify Command in a Domain 24–48
24.9.1 Using the verify Command on Cluster Root 24–49

25 Using Logical Storage Manager (LSM) in an HP AlphaServer SC System
25.1 Overview 25–2
25.2 Differences Between Managing LSM on an HP AlphaServer SC Domain and on a Standalone System 25–2
25.3 Storage Connectivity and LSM Volumes 25–3
25.4 Configuring LSM on an HP AlphaServer SC Domain 25–4
25.5 Creating an LSM Volume with a Single Striped Plex 25–4

26 Managing Security
26.1 General Guidelines 26–2
26.1.1 RSH 26–2
26.1.2 sysconfig 26–2
26.2 Configuring Enhanced Security 26–3
26.3 Secure Shell Software Support 26–4
26.3.1 Sample Default Configuration 26–4
26.3.2 Secure Shell Software Commands 26–10
26.3.3 Client Security 26–11
26.3.4 Host-Based Security 26–11
26.3.4.1 Disabling Root Login 26–12
26.3.4.2 Host Restrictions 26–12
26.3.4.3 User Restrictions 26–13

PART 3: SYSTEM VALIDATION AND TROUBLESHOOTING

27 SC Monitor
27.1 Hardware Components Managed by SC Monitor 27–2
27.2 SC Monitor Events 27–4
27.2.1 Hardware Component Events 27–5
27.2.2 EVM Events 27–5
27.3 Managing SC Monitor 27–6
27.3.1 SC Monitor Attributes 27–6
27.3.2 Specifying Which Hardware Components Should Be Monitored 27–7
27.3.3 Distributing the Monitor Process 27–9
27.3.3.1 Overview 27–10
27.3.3.2 Managing the Distribution of HSG80 RAID Systems 27–11
27.3.3.3 Managing the Distribution of HSV110 RAID Systems 27–12
27.3.4 Managing the Impact of SC Monitor 27–13

27.3.5 Monitoring the SC Monitor Process . . . 27–14
27.4 Viewing Hardware Component Properties . . . 27–15
27.4.1 The scmonmgr Command . . . 27–16

28 Using System Event Analyzer to Diagnose Node Problems

28.1 Overview of Node Diagnostics . . . 28–2
28.2 Obtaining System Event Analyzer . . . 28–3
28.3 Installing System Event Analyzer . . . 28–3
28.3.1 Installing System Event Analyzer on a Management Server . . . 28–3
28.3.2 Installing System Event Analyzer on a Domain Member . . . 28–5
28.4 Producing Consolidated Error Reports Using System Event Analyzer . . . 28–7
28.4.1 Advanced scca_setup Command Options . . . 28–8
28.5 Using the sra diag Command on Individual Nodes . . . 28–10
28.5.1 Running the sra diag Command . . . 28–10
28.5.2 Diagnostics Performed by the sra diag Command . . . 28–11
28.5.3 Reviewing the Reports . . . 28–12
28.6 Using the System Event Analyzer Command Line Interface . . . 28–14
28.7 Using the System Event Analyzer Web User Interface . . . 28–14
28.7.1 The WEBES Director . . . 28–14
28.7.1.1 Starting the Director at Boot Time . . . 28–15
28.7.1.2 Starting the Director Manually . . . 28–15
28.7.2 Invoking the System Event Analyzer WUI . . . 28–16
28.8 Managing the Size of the Error Log Files . . . 28–16
28.9 Checking the Status of the System Event Analyzer Processes . . . 28–19
28.10 Stopping the System Event Analyzer Processes . . . 28–19
28.11 Removing System Event Analyzer . . . 28–19

29 Troubleshooting

29.1 Booting Nodes Without a License . . . 29–3
29.2 Shutdown Leaves Members Running . . . 29–3
29.3 Backing Up and Restoring Files . . . 29–3
29.3.1 Suggestions for Files to Back Up . . . 29–4
29.3.2 Backing Up and Restoring the Management Server Root (/), /usr, and /var File Systems . . . 29–4
29.3.3 Booting the Domain Using the Backup Cluster Disk . . . 29–4
29.4 CFS Problems . . . 29–5
29.4.1 Specifying cluster_root at Boot Time . . . 29–6
29.4.2 Recovering the Cluster Root File System to a Disk Known to the Domain . . . 29–6
29.4.3 Recovering the Cluster Root File System to a New Disk . . . 29–8
29.4.4 Recovering When Both Boot Disks Fail . . . 29–12
29.4.5 Resolving AdvFS Domain Panics Due to Loss of Device Connectivity . . . 29–12
29.4.6 Forcibly Unmounting an AdvFS File System or Domain . . . 29–12

29.4.7 /var is Full 29–13
29.5 Crashes and Hangs 29–14
29.5.1 Application Hangs 29–14
29.5.1.1 Application Has Hung in User Space 29–14
29.5.1.2 Application Has Hung in Kernel Space 29–16
29.5.2 Identifying and Booting Crashed Nodes 29–17
29.5.3 Generating Crash Dumps from Responsive Domain Members 29–17
29.5.4 Crashing Unresponsive Domain Members to Generate Crash Dumps 29–17
29.5.5 LSM Hangs at Boot Time 29–19
29.6 Network Problems 29–19
29.6.1 Access Problems 29–20
29.6.1.1 Accessing the Cluster Alias from Outside the Domain 29–20
29.6.1.2 Accessing External Networks from Externally Connected Members 29–20
29.6.1.3 Accessing External Networks from Internally Connected Members 29–21
29.6.1.4 Additional Access Checks 29–21
29.6.2 NFS Problems 29–22
29.6.2.1 Node Failure of Client to External NFS Server 29–23
29.6.2.2 File-Locking Operations on NFS File Systems Hang Permanently 29–23
29.6.3 Cluster Alias Problems 29–24
29.6.3.1 Using the ping Command in a Domain 29–24
29.6.3.2 Running routed in a Domain 29–24
29.6.3.3 Incorrect Cabling Causes “Cluster Alias IP Address in Use” Warning 29–24
29.6.4 Changes to TCP/IP Ephemeral Port Numbers 29–25
29.7 RMS Problems 29–25
29.7.1 RMS Core Files 29–26
29.7.2 rmsquery Fails 29–27
29.7.3 prun Fails with “Operation Would Block” Error 29–27
29.7.4 Identifying the Causes of Load on msqld 29–27
29.7.5 RMS May Generate “Hostname / IP address mismatch” Errors 29–27
29.7.6 Management Server Reports rmsd Errors 29–28
29.8 Console Logger Problems 29–28
29.8.1 Port Not Connected Error 29–28
29.8.2 CMF Daemon Reports connection.refused Event 29–29
29.8.3 Console Output Wraps Strangely 29–30
29.9 Domain Member Fails and Domain Loses Quorum 29–30
29.10 Kernel Crashes 29–32
29.11 Console Messages 29–33
29.12 Korn Shell Does Not Record True Path to Member-Specific Directories 29–35
29.13 Pressing Ctrl/C Does Not Stop scrun Command 29–36
29.14 Setting the HiPPI Tuning Parameters 29–36
29.15 SSH Conflicts with sra shutdown -domain Command 29–37
29.16 FORTRAN: How to Produce Core Files 29–37
29.17 Checking the Status of the SRA Daemon 29–37
29.18 Accessing the HP AlphaServer SC Interconnect Control Processor Directly 29–38

29.19 SC Monitor Fails to Detect or Monitor HSG80 RAID Arrays . . . 29–38
29.20 Changing the Kernel Communications Rail . . . 29–40
29.21 SCFS/PFS File System Problems . . . 29–40
29.21.1 Mount State for Domain Is Unknown . . . 29–41
29.21.2 PFS File System Cannot be Deleted if Underlying SCFS File System is Broken . . . 29–42
29.21.3 Mount State Is mounted-busy . . . 29–43
29.21.4 PFS Mount State Is mounted-partial . . . 29–44
29.21.5 Mount State Remains unknown After Reboot . . . 29–45
29.21.6 SCFS File Systems Not Mounted by Newly Created FS Domain . . . 29–45
29.21.7 SCFS: scfsmgr May Fail to Create an Entry for a Disk . . . 29–46
29.22 System Check Tool . . . 29–47
29.22.1 System Check Tool Overview . . . 29–47
29.22.2 Installing the System Check Tool . . . 29–48
29.22.3 Running the System Check Tool . . . 29–48

PART 4: APPENDIXES

Appendix A Cluster Events . . . A–1

Appendix B Configuration Variables . . . B–1

Appendix C SC Daemons . . . C–1

C.1 HP AlphaServer SC Daemons . . . C–2
C.2 LSF Daemons . . . C–2
C.3 RMS Daemons . . . C–3
C.4 Domain Daemons . . . C–3
C.5 Tru64 UNIX Daemons . . . C–4
C.6 Daemons Not Supported in an HP AlphaServer SC System . . . C–5

Appendix D Example Output . . . D–1

D.1 Sample Output from sra delete_member Command . . . D–1
D.2 Sample clu_delete_member Log File . . . D–2

Index

List of Figures

Figure 1–1: HP AlphaServer SC Configuration for a Single-Rail 16-Node System 1–4
Figure 1–2: Node Network Connections: HP AlphaServer SC 16-Port Switch, HP AlphaServer ES40 Nodes 1–5
Figure 1–3: Node Network Connections: HP AlphaServer SC 16-Port Switch, HP AlphaServer DS20L Nodes 1–6
Figure 1–4: Node Network Connections When Using an HP AlphaServer SC 128-Way Switch 1–7
Figure 1–5: Node Network Connections: HP AlphaServer SC 128-Way Switch, HP AlphaServer DS20L Nodes 1–8
Figure 1–6: Node Network Connections: Federated HP AlphaServer SC Interconnect Configuration. 1–9
Figure 1–7: CFS Makes File Systems Available to All Domain Members 1–23
Figure 5–1: RMS User Dialog Box 5–40
Figure 5–2: Manage Partition Access and Limits Dialog Box 5–41
Figure 6–1: Example PFS/SCFS Configuration 6–7
Figure 6–2: HP AlphaServer SC Storage Configuration 6–9
Figure 6–3: Typical Active/Standby Mode Configuration on MSA1000 6–14
Figure 6–4: Typical Multiple-Bus Failover Configuration 6–16
Figure 6–5: Overview of Enterprise Virtual Array Component Connections 6–18
Figure 8–1: Parallel File System. 8–2
Figure 10–1: SC Viewer GUI 10–3
Figure 10–2: SC Viewer Menus 10–3
Figure 10–3: SC Viewer Overview Status Icons 10–5
Figure 10–4: SC Viewer Object Icons 10–6
Figure 10–5: SC Viewer Status Icons. 10–6
Figure 10–6: SC Viewer Event Severity Icons. 10–7
Figure 10–7: Example Object Panels 10–8
Figure 10–8: SC Viewer Tabs — General Layout 10–9
Figure 10–9: Example Overview Tab. 10–12
Figure 10–10: Example Failures Tab 10–14
Figure 10–11: Example Failures Tab with Object Selected 10–15
Figure 10–12: Example Domains Tab 10–16
Figure 10–13: Example Domains Tab with Domain Selected 10–17
Figure 10–14: Example Nodes Window for an HP AlphaServer ES40 10–18
Figure 10–15: Example Infrastructure Tab. 10–19
Figure 10–16: Example Properties Pane for a Switch 10–20
Figure 10–17: Example Properties Pane for a Terminal Server 10–21
Figure 10–18: Example Properties Pane for a SANworks Management Appliance 10–22
Figure 10–19: Example Properties Pane for an HSG80 RAID System 10–23

Figure 10–20: Example Properties Pane for an HSV110 RAID System 10–24
Figure 10–21: Example Physical Tab 10–25
Figure 10–22: Example Physical Tab with Cabinet Selected 10–26
Figure 10–23: Example Physical Tab with Node Selected Within Cabinet 10–27
Figure 10–24: Example Events Tab 10–28
Figure 10–25: Event Filter Dialog Box. 10–28
Figure 10–26: Example Event Tab with Event Selected 10–29
Figure 10–27: Example Interconnect Tab. 10–30
Figure 16–1: sramon GUI 16–21
Figure 16–2: sra-display Output 16–40
Figure 18–1: The SysMan Menu Hierarchy 18–4
Figure 20–1: The Three-Member atlasD0 Domain. 20–7
Figure 20–2: Three-Member atlasD0 Domain Loses a Member 20–9
Figure 23–1: CAA Branch of SysMan Menu 23–22
Figure 23–2: CAA Management Dialog Box 23–23
Figure 23–3: Start Dialog Box 23–24
Figure 23–4: Setup Dialog Box 23–25

List of Tables

Table 0–1: Abbreviations xxxviii
Table 0–2: Documentation Conventions xlii
Table 0–3: HP-Specific Names and Part Numbers for Quadrics Components xliii
Table 0–4: Network Adapters and Device Names xliv
Table 1–1: How to Connect the Components of an HP AlphaServer SC System 1–3
Table 1–2: HP AlphaServer SC IP Addresses 1–10
Table 1–3: Calculating Node IP Addresses 1–11
Table 1–4: Node and Member Numbering in an HP AlphaServer SC System 1–14
Table 1–5: Useful hwmgr Options 1–27
Table 2–1: Effect of Using an Alternate Boot Disk 2–8
Table 3–1: Tables In Which RMS Stores Operational or Transactional Records 3–4
Table 3–2: Records Archived by Default by the rmsarchive Command 3–5
Table 3–3: System Attributes Consistency Checks 3–17
Table 3–4: SRA Configuration Consistency Checks 3–17
Table 3–5: Terminal Servers/Ports/CMF Information Consistency Checks 3–18
Table 3–6: Disk and Storage Consistency Checks 3–20
Table 3–7: AdvFS/PFS/SCFS File Systems Consistency Checks 3–21
Table 3–8: Monitoring Subsystem Consistency Checks 3–23
Table 3–9: Miscellaneous Consistency Check 3–23
Table 3–10: sra dbcheck Options 3–24
Table 3–11: sra dbcheck Events 3–28
Table 4–1: LSF Job Termination Reasons 4–21
Table 4–2: Example Output from bacct and bhist Commands 4–22
Table 5–1: Fields in acctstats Table 5–4
Table 5–2: Fields in resources Table 5–5
Table 5–3: Example Partition Layout 1 5–9
Table 5–4: Example Partition Layout 2 5–11
Table 5–5: Effect on Active Resources of Partition Stop/Start 5–29
Table 5–6: Effect on Active Resources of Node Failure or Node Configured Out 5–33
Table 5–7: Actions Taken by pstartup.OSF1 Script 5–36
Table 5–8: Specifying Attributes When Creating Objects 5–43
Table 5–9: RMS Limits 5–52
Table 5–10: Scripts that Start RMS Daemons 5–68
Table 5–11: RMS Log Files 5–73
Table 6–1: Supported RAID Products 6–13
Table 7–1: SCFS Mount Status Values 7–5
Table 7–2: File-System Event Classes 7–30

Table 7–3: The sc_scfs Table 7–36
Table 7–4: The sc_scfs_mount Table 7–37
Table 7–5: The sc_advfs_vols Table 7–38
Table 7–6: The sc_advfs_filesets Table 7–38
Table 7–7: The sc_disk Table 7–38
Table 7–8: The sc_disk_server Table 7–39
Table 7–9: The sc_lsm_vols Table 7–40
Table 8–1: PFS Component File System Directory Structure 8–4
Table 8–2: Component File Systems for /scratch 8–8
Table 8–3: Component File Systems for /data3t. 8–9
Table 8–4: PFS Mount Status Values 8–14
Table 8–5: The sc_pfs Table 8–30
Table 8–6: The sc_pfs_mount Table. 8–30
Table 8–7: The sc_pfs_components Table 8–31
Table 8–8: The sc_pfs_filesystems Table 8–31
Table 9–1: HP AlphaServer SC Event Categories 9–3
Table 9–2: HP AlphaServer SC Event Classes 9–3
Table 9–3: HP AlphaServer SC Event Severities 9–6
Table 9–4: HP AlphaServer SC Event Filter Syntax 9–7
Table 9–5: Supported HP AlphaServer SC Event Filter Operators 9–9
Table 9–6: scevent Command-Line Options 9–10
Table 9–7: RMS Event Handlers 9–16
Table 9–8: Events that Trigger the rmsevent_env Handler 9–17
Table 10–1: Overview Tab Properties Pane 10–13
Table 10–2: Switch Properties Pane 10–20
Table 10–3: Terminal Server Properties Pane 10–21
Table 10–4: SANworks Management Appliance Properties Pane 10–22
Table 10–5: HSG80 RAID System Properties Pane 10–23
Table 10–6: HSV110 RAID System Properties Pane 10–24
Table 11–1: scload Options. 11–3
Table 11–2: scload Metrics 11–4
Table 12–1: scrun Command Options 12–3
Table 12–2: scrun Examples 12–6
Table 14–1: CMF Interpreter Commands 14–7
Table 14–2: cmfd Options. 14–13
Table 15–1: HP AlphaServer SC Log Files 15–2
Table 16–1: sra Command Syntax 16–5
Table 16–2: sra Command Description 16–8
Table 16–3: sra Options 16–12
Table 16–4: sra edit Subcommands 16–23
Table 16–5: sra edit Quick Reference 16–23
Table 16–6: Node Submenu Options 16–24
Table 16–7: System Submenu Options 16–30
Table 17–1: Domain Commands 17–2

Table 17–2: Features Not Supported in HP AlphaServer SC 17–3
Table 17–3: File Systems and Storage Differences 17–4
Table 17–4: Networking Differences 17–6
Table 17–5: Printing Differences 17–7
Table 17–6: Security Differences 17–8
Table 17–7: General System Management Differences 17–9
Table 18–1: Domain Tools Quick Start 18–2
Table 18–2: Domain Management Tools 18–3
Table 18–3: Invoking SysMan Menu 18–6
Table 19–1: Cluster Alias Configuration Files 19–5
Table 19–2: Preferred Server Cluster Aliases Configuration 19–26
Table 21–1: /etc/rc.config* Files 21–2
Table 21–2: Kernel Attributes to be Left Unchanged — vm Subsystem 21–3
Table 21–3: Configurable TruCluster Server Subsystems 21–3
Table 21–4: Configurable SC Subsystems 21–4
Table 21–5: Example System — Node Layout 21–10
Table 21–6: Example System — Nodeset Values 21–11
Table 21–7: Changing External IP Addresses - Items to Edit 21–22
Table 21–8: Changing External IP Addresses and Host Names - Items to Edit 21–23
Table 21–9: Changing External and Internal IP Addresses and Host Names - Items to Edit 21–24
Table 21–10: External Host Names and IP Addresses 21–26
Table 21–11: Domain Name and Cluster Aliases 21–27
Table 21–12: Interface IP Aliases 21–27
Table 21–13: External Servers 21–28
Table 21–14: Checklist 21–29
Table 21–15: Possible Solutions for Problems Moving Domain to Another IP Subnet 21–32
Table 21–16: Minimum System Firmware Versions 21–34
Table 22–1: Default Route Configuration 22–5
Table 22–2: Supported NIS Configurations 22–21
Table 23–1: Target and State Combinations for Application Resources 23–4
Table 23–2: Target and State Combinations for Network Resources 23–5
Table 23–3: Target and State Combinations for Tape Device and Media Changer Resources 23–5
Table 23–4: HP AlphaServer SC Resources 23–19
Table 26–1: Commonly Used SSH Commands 26–10
Table 26–2: Host Restrictions 26–13
Table 26–3: User Restrictions 26–13
Table 27–1: Hardware Components Managed by SC Monitor 27–2
Table 27–2: Hardware Component Events 27–5
Table 27–3: SC Monitor Attributes 27–6
Table 27–4: Hardware Components Monitored by SC Monitor 27–7
Table 27–5: Name Field Values in sc_classes Table 27–13
Table 27–6: Monitoring the SC Monitor Process 27–14
Table 27–7: scmonmgr Commands 27–16
Table 27–8: scmonmgr Command Options 27–17

Table 28–1: scca_setup Options 28–8
Table B–1: Cluster Configuration Variables B–1
Table C–1: HP AlphaServer SC Daemons C–2
Table C–2: LSF Daemons C–2
Table C–3: RMS Daemons C–3
Table C–4: Domain Daemons C–3
Table C–5: Tru64 UNIX Daemons C–4

Preface

Purpose of this Guide

This document describes how to administer an AlphaServer SC system from the Hewlett-Packard Company ("HP").

Intended Audience

This document is for those who maintain HP AlphaServer SC systems. Some sections will be helpful to end-users. Instructions in this document assume that you are an experienced UNIX® administrator who can configure and maintain hardware, operating systems, and networks.

Structure of This Guide

This document is organized as follows:

- Part 1: Systemwide Administration
  - Chapter 1: HP AlphaServer SC System Overview
  - Chapter 2: Booting and Shutting Down the HP AlphaServer SC System
  - Chapter 3: Managing the SC Database
  - Chapter 4: Managing the Load Sharing Facility (LSF)
  - Chapter 5: Managing the Resource Management System (RMS)
  - Chapter 6: Overview of File Systems and Storage
  - Chapter 7: Managing the SC File System (SCFS)
  - Chapter 8: Managing the Parallel File System (PFS)

- Chapter 9: Managing Events
- Chapter 10: Viewing System Status
- Chapter 11: SC Performance Visualizer
- Chapter 12: Managing Multiple Domains
- Chapter 13: User Administration
- Chapter 14: Managing the Console Network
- Chapter 15: System Log Files
- Chapter 16: The sra Command

- Part 2: Domain Administration
  - Chapter 17: Overview of Managing Domains
  - Chapter 18: Tools for Managing Domains
  - Chapter 19: Managing the Cluster Alias Subsystem
  - Chapter 20: Managing Domain Membership
  - Chapter 21: Managing Domain Members
  - Chapter 22: Networking and Network Services
  - Chapter 23: Managing Highly Available Applications
  - Chapter 24: Managing the Cluster File System (CFS), the Advanced File System (AdvFS), and Devices
  - Chapter 25: Using Logical Storage Manager (LSM) in an HP AlphaServer SC System
  - Chapter 26: Managing Security
- Part 3: System Validation and Troubleshooting
  - Chapter 27: SC Monitor
  - Chapter 28: Using System Event Analyzer to Diagnose Node Problems
  - Chapter 29: Troubleshooting
- Part 4: Appendixes
  - Appendix A: Cluster Events
  - Appendix B: Configuration Variables
  - Appendix C: SC Daemons
  - Appendix D: Example Output

Terms Used in This Guide

Clusters of nodes within an HP AlphaServer SC system are called domains (except in the case of clustered management servers). However, as domain management in HP AlphaServer SC systems is based on cluster management in TruCluster Server, many of the terms used in domain management include the word “cluster” — for example, cluster alias, cluster quorum, cluster application availability (CAA), cluster root, cluster disk.

The term “LSF cluster” is used to describe a group of machines controlled by LSF. An LSF cluster can consist of one or more HP AlphaServer SC systems, or part of an HP AlphaServer SC system; it can also include additional machines of arbitrary architectures.

TruCluster Server Documentation

The HP TruCluster Server documentation set provides a wealth of information about clusters and cluster management; however, there are a number of differences between the management of domains in an HP AlphaServer SC system, and the management of TruCluster Server clusters. These differences are described in the *HP AlphaServer SC System Administration Guide* (this document).

You should use the HP TruCluster Server documentation set to supplement the HP AlphaServer SC documentation set — if there is a conflict of information, use the instructions provided in the HP AlphaServer SC documentation set.

Related Documentation

You should have a hard copy or soft copy of the following documents:

- *HP AlphaServer SC Release Notes*
- *HP AlphaServer SC Installation Guide*
- *HP AlphaServer SC Best Practices I/O Guide*
- *HP AlphaServer SC Interconnect Installation and Diagnostics Manual*
- *HP AlphaServer SC RMS Reference Manual*
- *HP AlphaServer SC User Guide*
- *Administering HP AlphaServer SC Platform LSF®*
- *HP AlphaServer SC Platform LSF® Reference*
- *Running Jobs with HP AlphaServer SC Platform LSF®*
- *HP AlphaServer SC Platform LSF® Quick Reference*

- HP *AlphaServer ES45 Owner's Guide*
- HP *AlphaServer ES40 Owner's Guide*
- HP *AlphaServer DS20L User's Guide*
- HP *StorageWorks HSG80 Array Controller CLI Reference Guide*
- HP *StorageWorks HSG80 Array Controller Configuration Guide*
- HP *StorageWorks Fibre Channel Storage Switch User's Guide*
- HP *StorageWorks Enterprise Virtual Array HSV Controller User Guide*
- HP *StorageWorks Enterprise Virtual Array Initial Setup User Guide*
- HP *StorageWorks Modular SAN Array 1000 User Guide*
- HP *SANworks Tru64 UNIX Kit for Enterprise Virtual Array Release Notes*
- HP *SANworks Tru64 UNIX Kit for Enterprise Virtual Array Installation and Configuration Guide*
- HP *SANworks Scripting Utility for Enterprise Virtual Array Reference Guide*
- HP *TruCluster Server Cluster Release Notes*
- HP *TruCluster Server Cluster Technical Overview*
- HP *TruCluster Server Cluster Hardware Configuration*
- HP *TruCluster Server Cluster Administration*
- HP *TruCluster Server Cluster Highly Available Applications*
- HP *Tru64 UNIX Release Notes*
- HP *Tru64 UNIX Installation Guide*
- HP *Tru64 UNIX Network Administration: Connections*
- HP *Tru64 UNIX Network Administration: Services*
- HP *Tru64 UNIX System Administration*
- HP *Tru64 UNIX System Configuration and Tuning*
- HP *Tru64 UNIX AdvFS Administration*
- HP *Tru64 UNIX Logical Storage Manager*
- HP *Tru64 UNIX Advanced Printing Software User Guide*
- HP *Management and Configuration Guide for the HP ProCurve Series 4100GL Switches, Series 2600 Switches, and Switch 6108*

- *Summit Switch Hardware Installation Guide* from Extreme Networks, Inc.
- *ExtremeWare Software User Guide* from Extreme Networks, Inc.

The following is a list of the HP AlphaServer SC-specific reference pages:

- `fsck_pfs(8)`
- `mkfs_pfs(8)`
- `mount_pfs(8)`
- `pfs(8)`
- `umount_pfs(8)`
- `pvis(1)`
- `scload(1)`
- `scpvis(1)`
- `scfs(4)`
- `sys_attrs_scfs(5)`
- `sys_attrs_scfs_client(5)`
- `scfs_xfer_stats(8)`
- `cmfd(8)`
- `sra(8)`
- `srad(8)`
- `sramon(8)`
- `scrun(8)`
- `consistmgr(8)`
- `pfsmgr(8)`
- `scalertmgr(8)`
- `scevent(8)`
- `scfsmgr(8)`
- `scmonmgr(8`
- `scps(8)`
- `scviewer(8)`

Online documentation for the Message Passing Interface (MPI) is available on a `Documentation` link at `http://www-unix.mcs.anl.gov/mpi/mpich/`.

Abbreviations

Table 0–1 lists the abbreviations that are used in this document.

Table 0–1 Abbreviations

Abbreviation	Description
ACL	Access Control List
AdvFS	Advanced File System
API	Application Programming Interface
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
AUI	Attachment Unit Interface
BIND	Berkeley Internet Name Domain
CAA	Cluster Application Availability
CD-ROM	Compact Disc — Read-Only Memory
CDE	Common Desktop Environment
CDFS	CD-ROM File System
CDSL	Context-Dependent Symbolic Link
CFS	Cluster File System
CLI	Command Line Interface
CMF	Console Management Facility
CPU	Central Processing Unit
CS	Compute-Serving
DHCP	Dynamic Host Configuration Protocol
DMA	Direct Memory Access
DMS	Dataless Management Services
DNS	Domain Name System
DRD	Device Request Dispatcher

Table 0–1 Abbreviations

Abbreviation	Description
DRL	Dirty Region Logging
DVD	Digital Versatile Disc
DRM	Distributed Resource Management
DVDFS	DVD-ROM File System
EEPROM	Electrically Erasable Programmable Read-Only Memory
ELM	Elan License Manager
EVM	Event Manager
FastFD	Fast, Full Duplex
FC	Fibre Channel
FDDI	Fiber-optic Digital Data Interface
FRU	Field Replaceable Unit
FS	File-Serving
GUI	Graphical User Interface
HBA	Host Bus Adapter
HiPPI	High-Performance Parallel Interface
HPSS	High-Performance Storage System
HWID	Hardware (component) Identifier
ICMP	Internet Control Message Protocol
ICS	Internode Communications Service
IP	Internet Protocol
JBOD	Just a Bunch of Disks
JTAG	Joint Test Action Group
KVM	Keyboard-Video-Mouse
LAN	Local Area Network
LIM	Load Information Manager
LMF	License Management Facility

Table 0–1 Abbreviations

Abbreviation	Description
LSF	Load Sharing Facility
LSM	Logical Storage Manager
MAU	Multiple Access Unit
MB3	Mouse Button 3
MFS	Memory File System
MIB	Management Information Base
MPI	Message Passing Interface
MTS	Message Transport System
NFS	Network File System
NIFF	Network Interface Failure Finder
NIS	Network Information Service
NTP	Network Time Protocol
NVRAM	Non-Volatile Random Access Memory
OCP	Operator Control Panel
OS	Operating System
OSPF	Open Shortest Path First
PAK	Product Authorization Key
PBS	Portable Batch System
PCMCIA	Personal Computer Memory Card International Association
PE	Process Element
PFS	Parallel File System
PID	Process Identifier
PPID	Parent Process Identifier
RAID	Redundant Array of Independent Disks
RCM	Remote Console Monitor
RES	Remote Execution Server Daemon

Table 0–1 Abbreviations

Abbreviation	Description
RIP	Routing Information Protocol
RIS	Remote Installation Services
RLA	Remote LSF Adapter
RMC	Remote Management Console
RMS	Resource Management System
RPM	Revolutions Per Minute
SC	SuperComputer
SCFS	HP AlphaServer SC File System
SCSI	Small Computer System Interface
SMP	Symmetric Multiprocessing
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SRM	System Resources Manager
SROM	Serial Read-Only Memory
SSH	Secure Shell
TCL	Tool Command Language
UBC	Unified Buffer Cache
UDP	User Datagram Protocol
UFS	UNIX File System
UID	User Identifier
UTP	Unshielded Twisted Pair
UUCP	UNIX-to-UNIX Copy Program
WEBES	Web-Based Enterprise Service
WUI	Web User Interface

Documentation Conventions

Table 0–2 lists the documentation conventions that are used in this document.

Table 0–2 Documentation Conventions

Convention	Description
%	A percent sign represents the C shell system prompt.
$	A dollar sign represents the system prompt for the Bourne and Korn shells.
#	A number sign represents the superuser prompt.
P00>>>	A P00>>> sign represents the SRM console prompt.
`Monospace type`	Monospace type indicates file names, commands, system output, and user input.
Boldface type	Boldface type in interactive examples indicates typed user input. Boldface type in body text indicates the first occurrence of a new term.
Italic type	Italic (slanted) type indicates emphasis, variable values, placeholders, menu options, function argument names, and complete titles of documents.
UPPERCASE TYPE	Uppercase type indicates variable names and RAID controller commands.
Underlined type	Underlined type emphasizes important information.
[|] {|}	In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.
...	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
⋮	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.
Ctrl/x	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash.
Note	A note contains information that is of special importance to the reader.
`atlas`	`atlas` is an example system name.

HP-Specific Names and Part Numbers for Quadrics Components

Several HP AlphaServer SC Interconnect components are created by Quadrics. HP documents refer to Quadrics components using HP-specific names. Several Quadrics components also have a (different) Quadrics name. Table 0–3 shows how the HP-specific names and part numbers map to the equivalent Quadrics names.

Table 0–3 HP-Specific Names and Part Numbers for Quadrics Components

HP Part#	HP Name	Quadrics Name
3X-CCNBA-AA	HP AlphaServer SC 16-Port Switch	QM-S16
3X-CCNXA-BA	HP AlphaServer SC 128-Way Switch (new-type)	QM-S128F[1]
3X-CCNXE-CA	HP AlphaServer SC Top-Level Switch	QM-S128F[1]
3X-CCNXA-CA	HP AlphaServer SC Node-Level Switch	QM-S128F[1]
3X-CCNXA-AA	HP AlphaServer SC 128-Way Switch (old-type)	QM-S128
3X-CCNNA-AA	HP AlphaServer SC Elan Adapter Card	QM-400
3X-CCNXF-BA	HP AlphaServer SC 16-Port Switch Card (new-type)	QM-401X[2]
3X-CCNXF-AA	HP AlphaServer SC 16-Port Switch Card (old-type)	QM-401X[2]
3X-CCNXR-AA	HP AlphaServer SC High-Level Switch Card	QM-402[3]
3X-CCNCR-BA	HP AlphaServer SC Clock Card (new-type)	QM-408
3X-CCNCR-AA	HP AlphaServer SC Clock Card (old-type)	QM-403
3X-CCNXN-AA	HP AlphaServer SC 16-Link Null Card	QM-407
3X-CCNXP-AA	HP AlphaServer SC Interconnect Control Processor	QM-410
3X-CCNXC-AA	HP AlphaServer SC Clock Distribution Box	QM-SCLK

[1]The Quadrics part number QM-S128F corresponds to several components. The Quadrics part number refers to the basic empty chassis. Use the HP part numbers to distinguish between the different ways in which the chassis may be populated.

[2]The Quadrics part number QM-401X was not updated when this component was updated. Use the HP part number to distinguish between the new-type and old-type versions of this component.

[3]The Quadrics part number QM-402 and the HP part number 3X-CCNXR-AA were not updated when this component was updated. Use the revision number to distinguish between the new-type and old-type versions of this component.

Supported Network Adapters

Table 0–4 lists the associated device names for each supported network adapter. The examples in this guide refer to the DE602 network adapter.

Table 0–4 Network Adapters and Device Names

Network Adapter	SRM Device Name	UNIX Device Name
DE60*x*	`eia0`	`ee0`
DE50*x*	`ewa0`	`tu0`
Gigabit Ethernet	SRM cannot use this device	`alt0`
Gigabit Ethernet	SRM cannot use this device	`bcm0`
HiPPI[1, 2]	SRM cannot use this device	`hip0`
ATM[2]	SRM cannot use this device	`lis0`
FDDI	SRM cannot use this device	`fta0`

[1]HiPPI is only available if you install an additional HiPPI subset — for Tru64 UNIX Version 5.1B, the minimum supported version is HiPPI kit 222.

[2]The `sra install` command does not configure HiPPI and ATM interfaces — you must configure such interfaces manually.

Supported Node Types

HP AlphaServer SC Version 2.6 supports the following node types:

- HP AlphaServer ES45
- HP AlphaServer ES40
- HP AlphaServer DS20L

Location of Code Examples

Code examples are located in the `/Examples` directory of the *HP AlphaServer SC System Software* CD-ROM.

Location of Online Documentation

Online documentation is located in the `/docs` directory of the *HP AlphaServer SC System Software* CD-ROM.

Comments on this Document

HP welcomes any comments and suggestions that you have on this document. Please send all comments and suggestions to your HP Customer Support representative.

Part 1:

Systemwide Administration

1

HP AlphaServer SC System Overview

This guide does not attempt to cover all aspects of normal UNIX system administration (these are covered in detail in the HP *Tru64 UNIX System Administration* manual), but rather focuses on aspects that are specific to HP AlphaServer SC systems.

This chapter is organized as follows:

- Configuration Overview (see Section 1.1 on page 1–2)
- HP AlphaServer SC Nodes (see Section 1.2 on page 1–12)
- Graphics Consoles (see Section 1.3 on page 1–13)
- Domains (see Section 1.4 on page 1–13)
- Local Disks (see Section 1.5 on page 1–15)
- Console Network (see Section 1.6 on page 1–15)
- Management LAN (see Section 1.7 on page 1–16)
- HP AlphaServer SC Interconnect (see Section 1.8 on page 1–16)
- External Network (see Section 1.9 on page 1–18)
- Management Server (see Section 1.10 on page 1–18)
- Physical Storage (see Section 1.11 on page 1–19)
- Cluster File System (CFS) (see Section 1.12 on page 1–21)
- Device Request Dispatcher (DRD) (see Section 1.13 on page 1–23)
- Resource Management System (RMS) (see Section 1.14 on page 1–24)
- Parallel File System (PFS) (see Section 1.15 on page 1–25)
- SC File System (SCFS) (see Section 1.16 on page 1–25)
- Managing an HP AlphaServer SC System (see Section 1.17 on page 1–25)
- Monitoring System Activity (see Section 1.18 on page 1–27)
- Differences between HP AlphaServer SC and TruCluster Server (see Section 1.19 on page 1–28)

1.1 Configuration Overview

An HP AlphaServer SC system is a scalable, distributed-memory, parallel computer system that can expand to up to 4096 CPUs. An HP AlphaServer SC system can be used as a single compute platform to host parallel jobs that consume up to the total compute capacity. An HP AlphaServer SC system is built primarily from standard components. This section provides a brief description of those components, and the following sections describe the components in more detail.

The most important hardware components of the HP AlphaServer SC system are the nodes and the high-speed system interconnect. The HP AlphaServer SC system is constructed through the tight coupling of up to 1024 HP AlphaServer ES45 or HP AlphaServer ES40 nodes, or up to 128 HP AlphaServer DS20L nodes. The nodes are interconnected using a high-bandwidth (340 MB/s), low-latency (~3 μs) switched fabric (this fabric is called a **rail**). The bandwidth (both point-to-point and bi-section) and latency characteristics of this network are key to providing the parallel compute power of the HP AlphaServer SC system. Additional high-speed interconnect bandwidth can be obtained by using an optional second rail.

In addition to the high-speed system interconnect, the HP AlphaServer SC system uses two further internal networks, as follows:

- A 100Mbps switched Ethernet. This connects all of the nodes into a single management domain.
- A console network. This integrates all of the HP AlphaServer SC node console ports and allows management software to control the individual nodes (for boot, hardware diagnostics, and so on).

Other key hardware components are as follows:

- System Storage
- Per-node disks
- System console (or consoles)

Clusters of nodes within an HP AlphaServer SC system are called domains (except in the case of clustered management servers). For ease of management, the HP AlphaServer SC nodes are organized into multiple domains. Each domain shares a common domain file system. This is served by the system storage and provides a common image of the operating system (OS) files to all nodes within a domain. Each node has a locally attached disk, which is used to hold the per-node boot image, swap space, and other temporary files.

A system can be configured with a front-end management server, and in some cases this is mandatory, as follows:

- If the system uses a 16-port switch, a management server is optional.
- If the system uses a 128-way switch, a management server is mandatory.
- If the system contains HP AlphaServer DS20L nodes, a management server is mandatory.

HP AlphaServer SC Version 2.6 also supports a clustered management server. This is a standard TruCluster Server implementation operating over a Gigabit Ethernet Interconnect, and should not be confused with the HP AlphaServer SC system which operates over the HP AlphaServer SC Interconnect. In HP AlphaServer SC Version 2.6, the clustered management server is qualified at two servers. For more information, see Chapter 3 of the *HP AlphaServer SC Installation Guide*.

If the front-end management server is configured, certain housekeeping functions run on this node. This node is not connected to the high-speed interconnect. If the front-end management server is not configured, the housekeeping functions run on Node 0 (zero).

Figure 1–1 on page 1–4 shows an example HP AlphaServer SC configuration, for a single-rail 16-node system.

Figure 1–2 to Figure 1–6 show how the first nodes are connected to the networks of the HP AlphaServer SC system, depending on the type of HP AlphaServer SC Interconnect switch used. See Table 1–1 to identify which figure applies to your system.

Table 1–1 How to Connect the Components of an HP AlphaServer SC System

If you are using...	See...
HP AlphaServer SC 16-port switch	Figure 1–2 on page 1–5 Figure 1–3 on page 1–6
HP AlphaServer SC 128-way switch	Figure 1–4 on page 1–7 Figure 1–5 on page 1–8
HP AlphaServer SC 128-way switches in a federated configuration	Figure 1–6 on page 1–9

Note:

These diagrams are not to scale.

The nodes in some of these diagrams have been re-arranged to show the cables more clearly — in reality, node numbers increase from bottom to top in a network cabinet.

The rest of this section provides more detail on the system components.

Figure 1–1 shows an example HP AlphaServer SC configuration, for a 16-node system. In this diagram, the ES4*x* value is used to represent HP AlphaServer ES40, HP AlphaServer ES45, or HP AlphaServer DS20L nodes. KVM switch is a Keyboard-Video-Mouse switch.

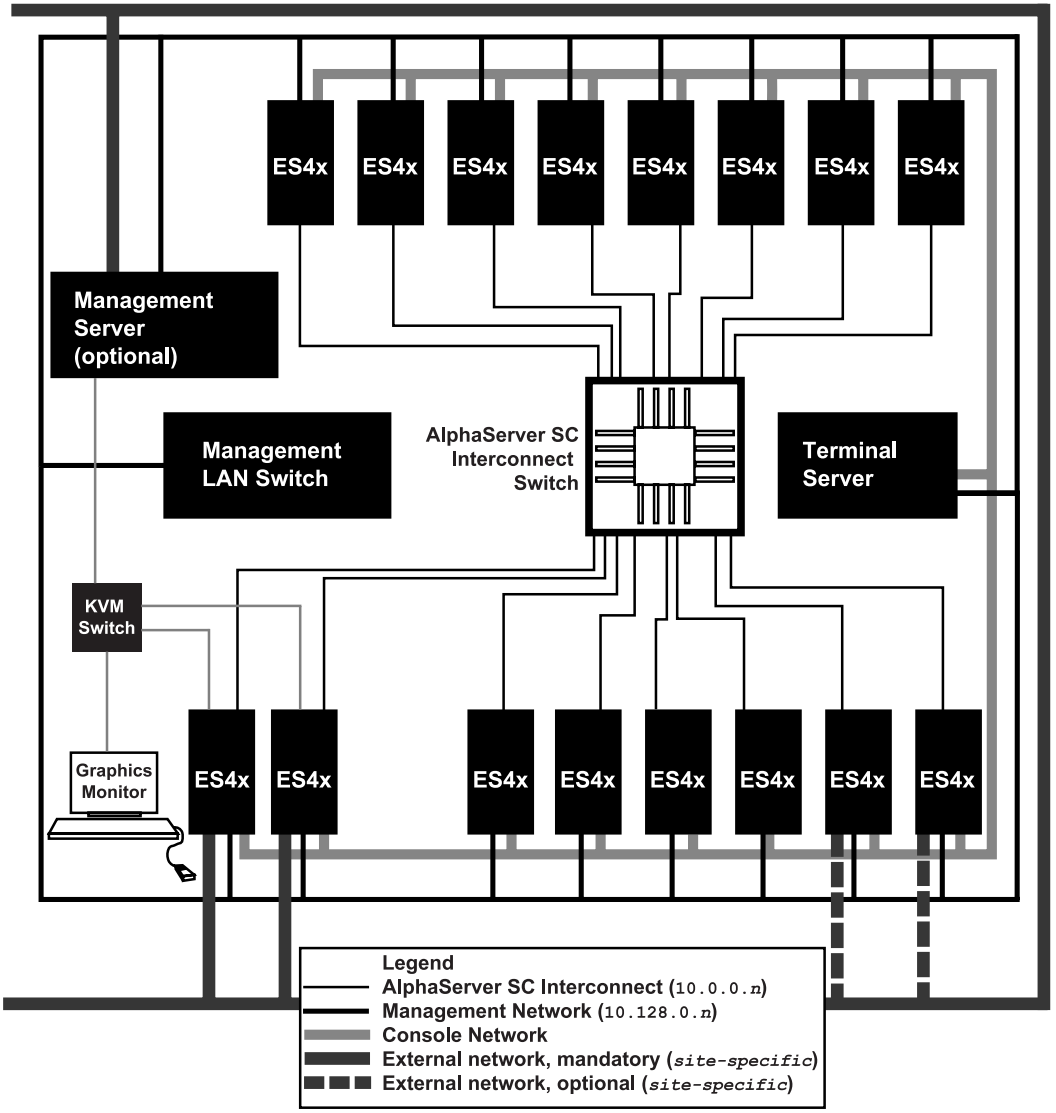

Figure 1–1 HP AlphaServer SC Configuration for a Single-Rail 16-Node System

Figure 1–2 shows how the first three HP AlphaServer ES40 nodes are connected to the networks of the HP AlphaServer SC system containing an HP AlphaServer SC 16-port switch, an optional management server, and an optional second rail.

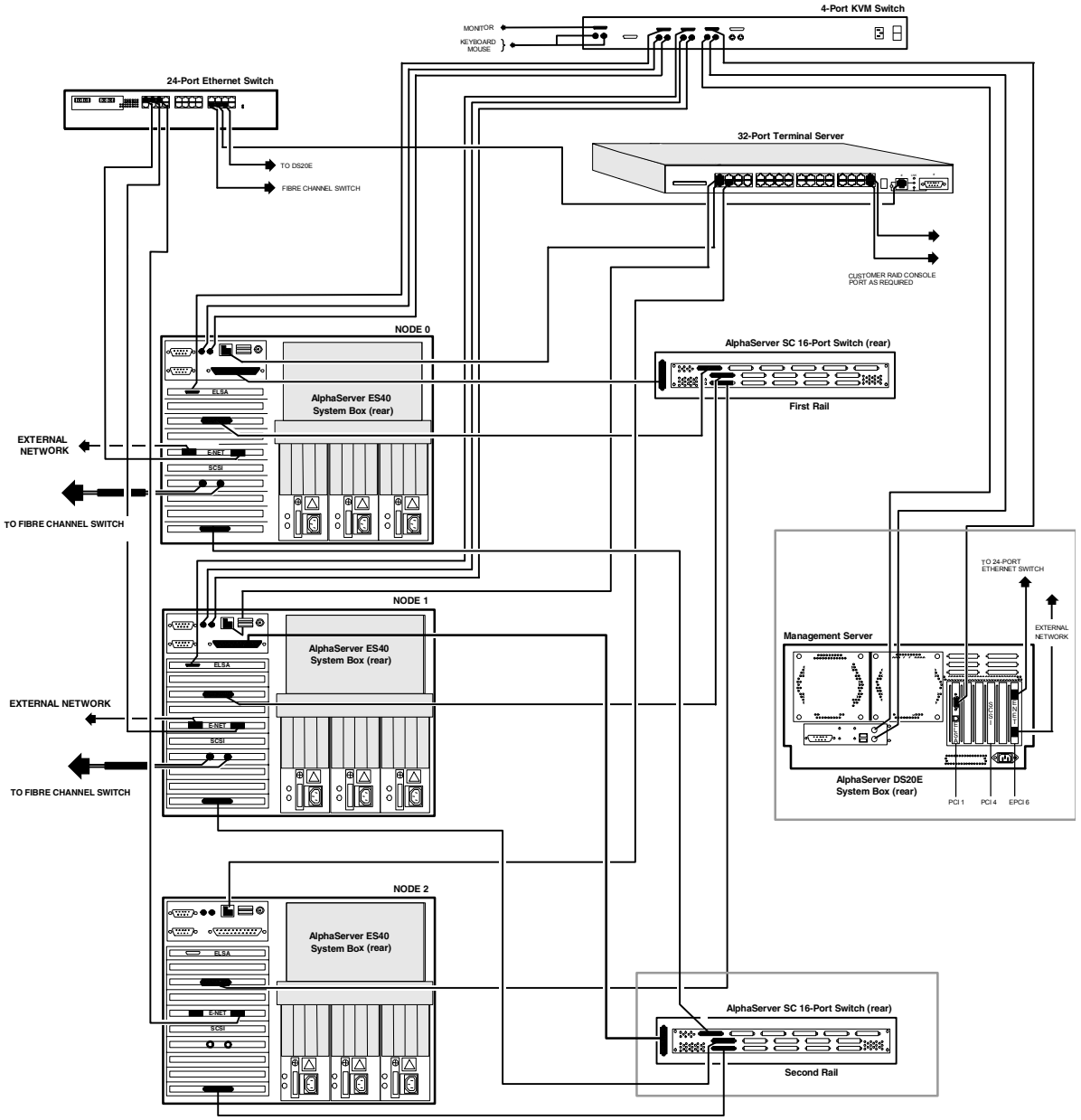

Figure 1–2 Node Network Connections: HP AlphaServer SC 16-Port Switch, HP AlphaServer ES40 Nodes

Figure 1–3 shows how the first two HP AlphaServer DS20L nodes are connected to the networks of the HP AlphaServer SC system containing an HP AlphaServer SC 16-port switch and a management server.

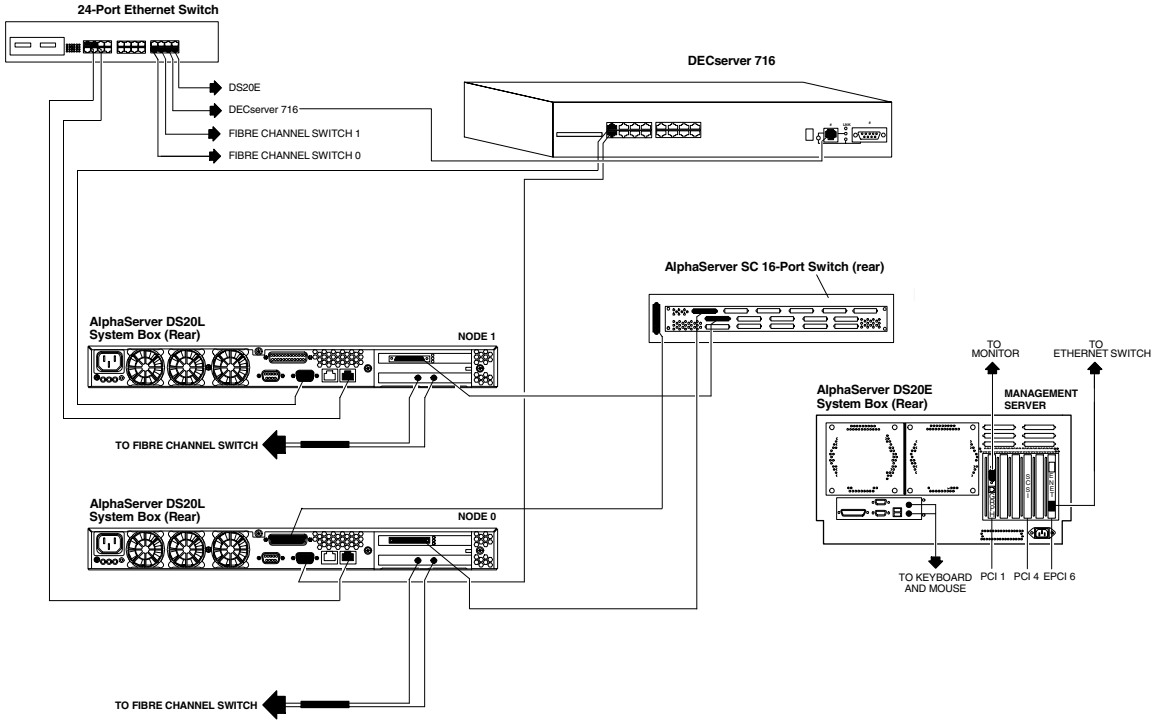

Figure 1–3 Node Network Connections: HP AlphaServer SC 16-Port Switch, HP AlphaServer DS20L Nodes

Figure 1–4 shows how the first three nodes are connected to the networks of the HP AlphaServer SC system containing an HP AlphaServer SC 128-way switch, a management server, and an optional second rail.

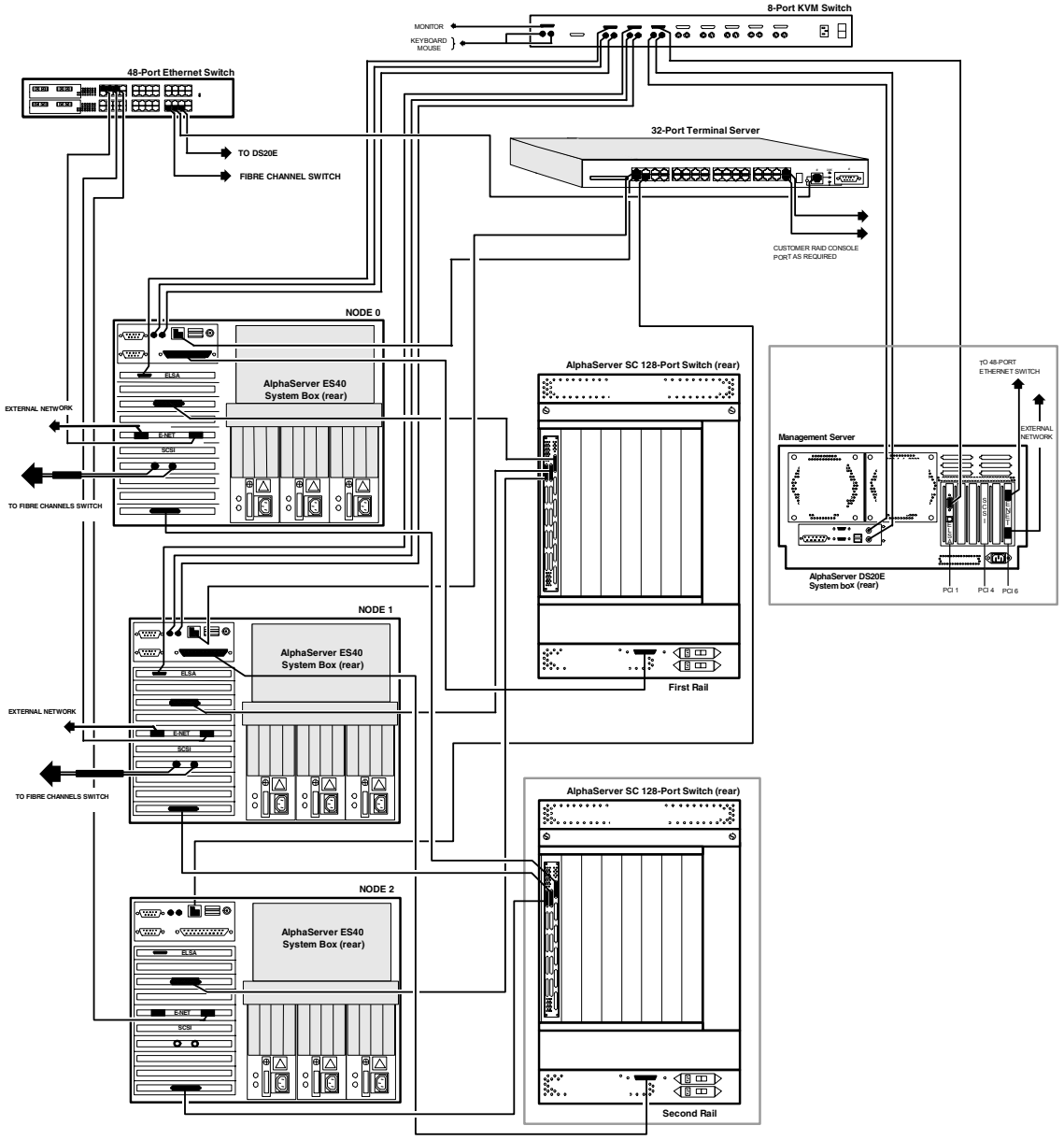

Figure 1–4 Node Network Connections When Using an HP AlphaServer SC 128-Way Switch

Figure 1–5 shows how the first two HP AlphaServer DS20L nodes are connected to the networks of the HP AlphaServer SC system containing an HP AlphaServer SC 128-way switch and a management server.

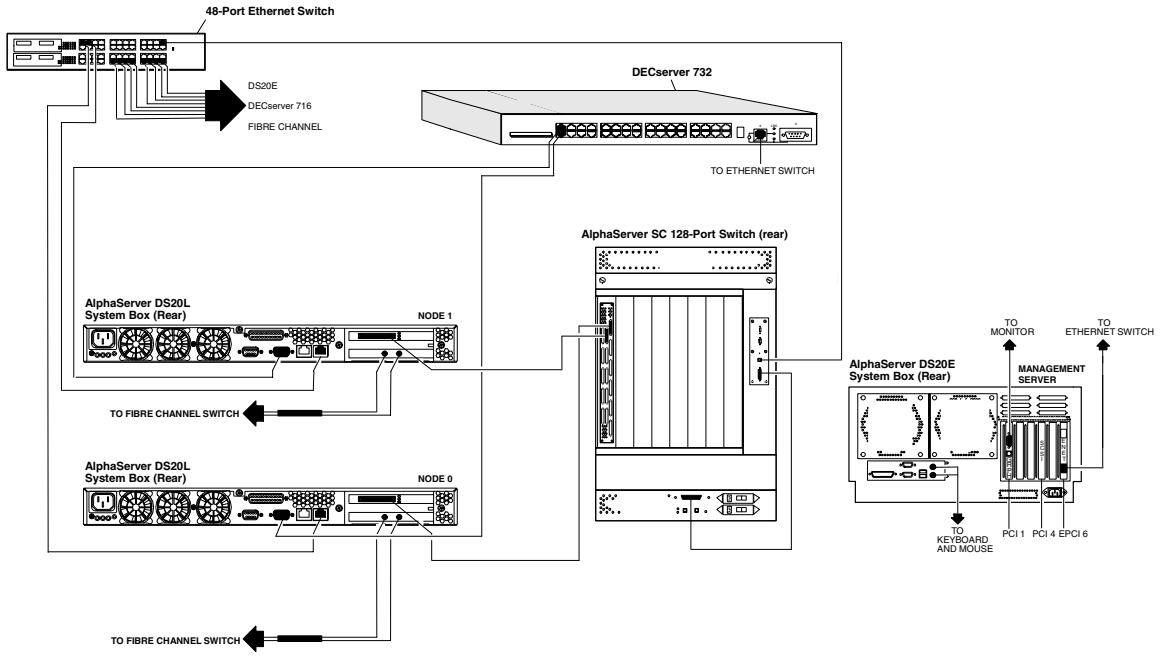

Figure 1–5 Node Network Connections: HP AlphaServer SC 128-Way Switch, HP AlphaServer DS20L Nodes

Figure 1–6 shows the hardware connections when using a federated HP AlphaServer SC Interconnect configuration.

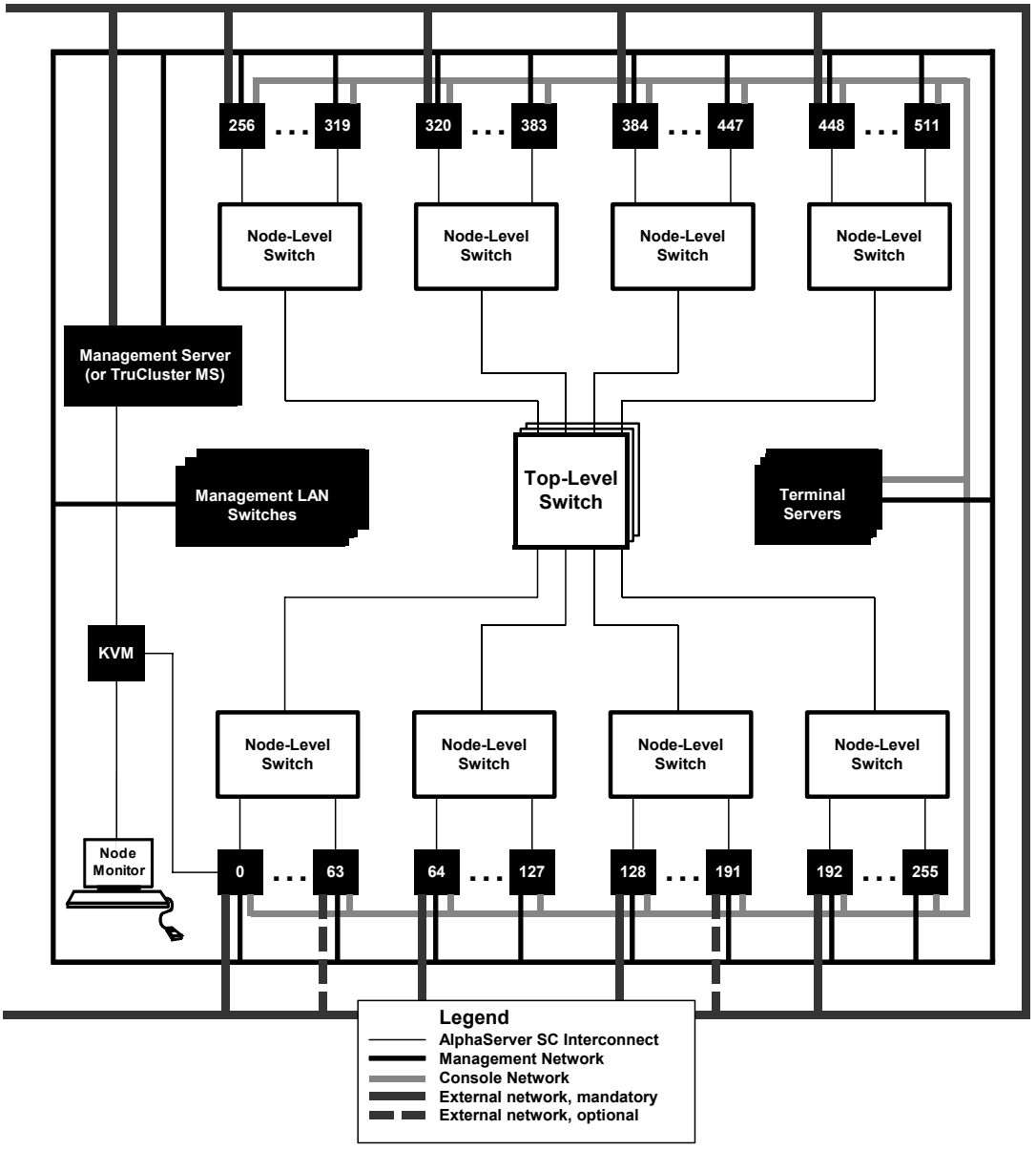

Figure 1–6 Node Network Connections: Federated HP AlphaServer SC Interconnect Configuration

1.1.1 Assigning IP Addresses

As mentioned above, the system is connected using an internal management local area network (LAN). This connects each of the nodes and other key hardware components.

The HP AlphaServer SC Interconnect is also used internally as an Ethernet-type device. Both of these LANs are configured as internal networks; that is, they are not connected to external networks. They use a 10.x.x.x address notation.

Note that the cluster alias IP address should not be a 10.x.x.x address. However, if a 10.x.x.x address is required, refer to Appendix G of the *HP AlphaServer SC Installation Guide*.

Table 1–2 shows the network address convention used for these networks and attached devices.

Table 1–2 HP AlphaServer SC IP Addresses

Component	IP Address Range
Net mask	`255.255.0.0`
Cluster Interconnect (IP suffix: `-ics0`)	`10.0.`*`x.y`*
System Interconnect (IP suffix: `-eip0`)	`10.64.`*`x.y`*
Management network interface card	`10.128.`*`x.y`*
Terminal server *t*, where *t* is 1–254	`10.128.100.`*`t`*
Management server *m* on management LAN, where *m* is 1, or 2 for the second member of a clustered management server	`10.128.101.`*`m`*
Management server *m* Cluster Interconnect (IP suffix: `-ics0`)	`10.32.0.`*`m`*
Management server *m* Gigabit Interconnect (IP suffix: `-icstcp0`)	`10.33.0.`*`m`*
Summit switch *g*, where *g* is 1–11	`10.128.103.`*`g`*
HP SANworks Management Appliance or Fibre Channel switch *f*, where *f* is 1–254	`10.128.104.`*`f`*
RAID array controller *a*, where *a* is 1, 2, and so on	`10.128.105.`*`a`*
HP AlphaServer SC Interconnect Control Card for Node-Level switch *N*, where *r* is the rail number and *N* is 0–31	`10.128.(128+`*`r`*`).(`*`N`*`+1)`
HP AlphaServer SC Interconnect Control Card for Top-Level switch *T*, where *r* is the rail number and *T* is 0–15	`10.128.(128+`*`r`*`).(`*`T`*`+128)`
Preferred server cluster alias addresses (where *FS* is 0-3, thus accommodating a maximum of four FS domains, and *y* is the member ID within the FS domain)	`10.128.(106+`*`FS`*`).`*`y`

1.1.1.1 Node IP Addresses

The IP addresses for the Cluster Interconnect, the System Interconnect, and the management network interface cards are assigned automatically during installation. These IP addresses are of the form 10.*Z*.*x*.*y*, where

- *Z* is fixed for a particular network, as follows:
  - *Z* = 0 for the Cluster Interconnect
  - *Z* = 64 for the System Interconnect
  - *Z* = 128 for the management network interface card
- *x* and *y* are deduced by dividing the node number *n* by 128, where:
  - *x* is the integer part of the quotient
  - *y* is the remainder +1

Table 1–3 shows some examples of how to use this formula to calculate node IP addresses.

Table 1–3 Calculating Node IP Addresses

Node (n)	Calculation	x	y	IP Address
0	(0/128) = 0, remainder = 0	0	1	10.*Z*.0.1
1	(1/128) = 0, remainder = 1	0	2	10.*Z*.0.2
127	(127/128) = 0, remainder = 127	0	128	10.*Z*.0.128
128	(128/128) = 1, remainder = 0	1	1	10.*Z*.1.1
250	(250/128) = 1, remainder = 122	1	123	10.*Z*.1.123
256	(256/128) = 2, remainder = 0	2	1	10.*Z*.2.1
384	(384/128) = 3, remainder = 0	3	1	10.*Z*.3.1
500	(500/128) = 3, remainder = 116	3	117	10.*Z*.3.117
512	(512/128) = 4, remainder = 0	4	1	10.*Z*.4.1
640	(640/128) = 5, remainder = 0	5	1	10.*Z*.5.1
750	(750/128) = 5, remainder = 110	5	111	10.*Z*.5.111
768	(768/128) = 6, remainder = 0	6	1	10.*Z*.6.1
896	(896/128) = 7, remainder = 0	7	1	10.*Z*.7.1
1000	(1000/128) = 7, remainder = 104	7	105	10.*Z*.7.105
1023	(1023/128) = 7, remainder = 127	7	128	10.*Z*.7.128

1.2 HP AlphaServer SC Nodes

An HP AlphaServer SC Version 2.6 system may contain up to 1024 HP AlphaServer ES45 or HP AlphaServer ES40 nodes, or up to 128 HP AlphaServer DS20L nodes. In general, this document refers to the HP AlphaServer ES45 node type.

Observe the following guidelines:

- Do not mix the node types in a domain.
- Do not mix the node types in a Resource Management System (RMS) partition.

Each node has the following components:

- An HP AlphaServer ES45 has up to four 1000 MHz or 1250 MHz Alpha EV68 CPUs, and up to 32GB of memory.

  An HP AlphaServer ES40 has up to four 667 MHz Alpha EV67 CPUs or up to four 833 MHz Alpha EV68 CPUs, and up to 16GB of memory.

  An HP AlphaServer DS20L has up to two 833 MHz Alpha EV68 CPUs, and up to 2GB of memory.
- At least one HP AlphaServer SC PCI Elan adapter card with cable connected to a switch. The size of the switch depends on the number of nodes — the maximum number of nodes supported in HP AlphaServer SC Version 2.6 is 1024 nodes. In this document, this network is called the **HP AlphaServer SC Interconnect**; the adapter is called the **HP AlphaServer SC Elan adapter card**; and the switch is called the **HP AlphaServer SC Interconnect switch**. See also Section 1.8 on page 1–16.
- A 100BaseT adapter connected to the FastEthernet network. In this document, this is called the **management network**. On an HP AlphaServer DS20L, this adapter is built onto the motherboard.
- A connection from the COM 1 serial interface MMJ-style connector to a terminal server. In this document, this network is called the **console network**, and the COM 1 serial interface MMJ-style connector is called the **console port** (also known as the **SRM console port** or the **SRM and RMC console port**).

Note:

These specifications are correct at the time of writing. For the latest CPU specification, and for information about the supported network adapters and disk sizes, please check the relevant *QuickSpecs*.

See Chapter 3 of the *HP AlphaServer SC Installation Guide* for information on how to populate the PCI slots in an HP AlphaServer SC system.

Conceptually, the 1024 nodes form one **system**. The system has a name; for example, `atlas`.

Each HP AlphaServer ES45, HP AlphaServer ES40, or HP AlphaServer DS20L is called a **node**. Nodes are numbered from 0 to 1023. Each node is named by appending its node number to the system name. For example, if the system name is `atlas`, the name of Node 7 (the eighth node) is `atlas7`.

Note:

In this guide, the terms "node" and "member" both refer to an HP AlphaServer ES45, HP AlphaServer ES40, or HP AlphaServer DS20L. However, the term **member** exclusively refers to an HP AlphaServer ES45, HP AlphaServer ES40, or HP AlphaServer DS20L that is a member of a domain (see Section 1.4).

1.3 Graphics Consoles

The network cabinet of an HP AlphaServer SC system contains a flat panel graphics monitor, a keyboard, a mouse, and a Keyboard-Video-Mouse (KVM) switch. By changing the setting of the KVM switch, you can connect the monitor/keyboard/mouse to the management server (if used), to Node 0, or to Node 1.

During the initial installation process, set the KVM switch to connect to the management server (if used), and install and configure the management server. If not using a management server, set the KVM switch to connect to Node 0, and install and configure Node 0.

Once you have installed and configured either the management server or Node 0, you can access the console ports of all nodes using the `sra -cl` command. The monitor/keyboard/mouse can then be used as a regular graphics console to log into the system.

If Node 0 fails, set the KVM switch to connect to Node 1 to control the system (assuming that the system maintains quorum despite the failure of Node 0).

Note:

On an HP AlphaServer SC system composed of HP AlphaServer DS20L nodes, the nodes do not have any graphics capability. Therefore, there is no KVM strategy between the management server (if used), Node 0, and Node 1.

1.4 Domains

HP AlphaServer SC Version 2.6 supports multiple domains. Each domain can contain up to 32 HP AlphaServer ES45, HP AlphaServer ES40, or HP AlphaServer DS20Ls nodes, providing a maximum of 1024 HP AlphaServer SC nodes.

Nodes are numbered from 0 to 1023 within the overall system (see Section 1.2), but members are numbered from 1 to 32 within a domain, as shown in Table 1–4, where `atlas` is an example system name.

Table 1–4 Node and Member Numbering in an HP AlphaServer SC System

Node	Member	Domain
atlas0	member1	atlasD0
...	...	
atlas31	member32	
atlas32	member1	atlasD1
...	...	
atlas63	member32	
atlas64	member1	atlasD2
...	...	...
atlas991	member32	
atlas992	member1	atlasD31
...	...	
atlas1023	member32	

System configuration operations must be performed on each of the domains. Therefore, from a system administration point of view, a 1024-node HP AlphaServer SC system may entail managing a single system or managing several domains — this can be contrasted with managing 1024 individual nodes.

The first two nodes of each domain provide a number of services to the rest of the nodes in their respective domains — the second node also acts as a root file server backup in case the first node fails to operate correctly.

The services provided by the first two nodes of each domain are as follows:

- Serves as the root of the Cluster File System (CFS). The first two nodes in each domain are directly connected to a Redundant Array of Independent Disks (RAID) subsystem.
- Provides a gateway to an external Local Area Network (LAN). The first two nodes of each domain should be connected to an external LAN.

In HP AlphaServer SC Version 2.6, there are two domain types:

- File-Serving (FS) domain
- Compute-Serving (CS) domain

HP AlphaServer SC Version 2.6 supports a maximum of four FS domains. The SCFS file system exports file systems from an FS domain to the other domains. Although the FS domains can be located anywhere in the HP AlphaServer SC system, HP recommends that you configure either the first domain(s) or the last domain(s) as FS domains — this provides a

contiguous range of CS nodes for MPI jobs. It is not mandatory to create an FS domain, but you will not be able to use SCFS if you have not done so. For more information about SCFS, see Chapter 7.

1.5 Local Disks

Each node contains two disks that provide swap, local, and temporary storage space. (The first node in each domain has a third local disk, for the Tru64 UNIX operating system — prior to domain creation, this disk is used as the boot disk.)

Each disk has a boot partition. Under normal operation, the first disk's boot partition is used. The second disk allows the system to be booted in the case of failure of the first one. See Chapter 2 for more information about boot disks.

Note:

On an HP AlphaServer SC system composed of HP AlphaServer DS20L nodes:

- There is only one local disk for swap, local and temporary storage space.
- The Tru64 UNIX operating system disk is within the external storage configuration.
- There is no alternate boot disk.

1.6 Console Network

The console network comprises console cables and several terminal servers (depending on the number of nodes and other components).

Each terminal server manages up to 32 console ports — each domain should have its own terminal server. The COM 1 MMJ-style connector serial port of each system node is connected to a port on the terminal server. The order of connections is important: Node 0 is connected to port 1 of the terminal server, Node 1 to port 2, and so on.

The terminal server is in turn connected to the management network. This configuration enables each node's console port to be accessed using IP over the management network. This facility provides management software with access to a node's console port (for boot, power control, configuration probes, firmware upgrade, and so on).

The IP naming convention for the terminal server *t* is `10.128.100.`*t*, where *t* is 1–254.

1.7 Management LAN

The management network is based on a FastEthernet switch, and comprises 100BaseT Ethernet adapters, cables, and one or more switches, depending on the number of nodes. Switches may be Extreme Network Summit switches, or HP ProCurve switches.

The management network provides the connections for management traffic during both system installation and normal operations. This traffic is separated from the HP AlphaServer SC Interconnect to avoid interfering with parallel application performance. This network is heavily used during the system installation process. During normal operation, usage of this network is light.

For security reasons, the management network should not be connected directly or *via* a gateway to any other network.

The following components are connected on the management network:

- Nodes (configured at 100Mbps full duplex)
- Terminal servers (configured at 10Mbps half duplex)
- Management servers (configured at 100Mbps full duplex)
- HP AlphaServer SC Interconnect switches (configured at 100Mbps full duplex)
- Extreme Network (24- or 48-Port) Summit switch
- Extreme Network Summit 5i or 7i switch (configured at 1Gbps full duplex)
- HP ProCurve (48-Port) switch
- HP SANworks Management Appliances or Fibre Channel switches (configured at 10Mbps half duplex)

Table 1–2 on page 1–10 shows the convention used to assign IP addresses in an HP AlphaServer SC system.

1.8 HP AlphaServer SC Interconnect

The HP AlphaServer SC Interconnect provides the high-speed message passing and remote memory access capability for parallel applications.

The HP AlphaServer SC Interconnect comprises adapters, cables, and at least one HP AlphaServer SC Interconnect switch. The size of the switch, and the number of switches, depends on the number of nodes, as follows:

- If the HP AlphaServer SC system contains 16 nodes or less, a 16-port switch may be used.
- If the HP AlphaServer SC system contains 17 to 128 nodes, a 128-port switch cabinet may be used.

- If the HP AlphaServer SC system contains more than 128 nodes, a **federated** switch configuration should be used.

There is a parallel interface, or **JTAG port**, on each HP AlphaServer SC Interconnect switch. Some switches have a control card to handle the JTAG port. For a switch with a control card, the switch-management-server function runs on the control card.

Other switches must be connected to the parallel interface of a node — these are **directly connected** switches. For directly connected switches, the switch-management-server functions run on the nodes directly connected to the switches; these nodes are defined by the `swmserver` entries in the `servers` table of the SC database. By default, Node 0 performs switch management tasks for the first rail of the HP AlphaServer SC Interconnect network, and Node 1 manages the switch for the second rail. You can change this by moving the JTAG cables and updating the `servers` table in the SC database.

1.8.1 Single-Rail Configurations and Dual-Rail Configurations

A **single-rail** configuration is a configuration in which there is one HP AlphaServer SC Elan adapter card in each node, and all nodes are connected to one HP AlphaServer SC Interconnect switch.

A **dual-rail** configuration is a configuration in which there are two HP AlphaServer SC Interconnects. Each node contains two HP AlphaServer SC Elan adapter cards, with each card connected to a different HP AlphaServer SC Interconnect switch. For any given node in a dual-rail configuration, the same port must be used on each HP AlphaServer SC Interconnect — the node must have the same Elan ID on each rail.

Additionally, each HP AlphaServer SC Interconnect must use similar hardware. For example, if the first rail uses a 16-port switch, the second rail must also use a 16-port switch — do not use a 16-port switch for one rail while using a 128-port switch for another rail.

The dual-rail configuration provides a high bandwidth solution for application programs — it does not provide a failover solution. In addition, intra-domain communications associated with system operation (domain operations) are performed on the first rail only. Inter-domain SCFS file system operations make use of both rails by default, load balancing I/O transfers over both rails. The second rail is available for use by applications.

Dual rail impacts the following areas:

- Configuration
  See the *HP AlphaServer SC Installation Guide*.
- Using dual rail in applications
  See Section 5.12 on page 5–76.

- Administration
  See the *HP AlphaServer SC Interconnect Installation and Diagnostics Manual*.

Table 1–2 on page 1–10 shows the IP naming convention for the HP AlphaServer SC Interconnect.

Note:

An HP AlphaServer SC system composed of HP AlphaServer DS20L nodes supports only a single HP AlphaServer SC Interconnect rail, because of the limited number of PCI slots.

1.9 External Network

The first two nodes of each domain — that is, Nodes 0 and 1, 32 and 33, 64 and 65, 96 and 97, and so on — should be connected to an external network using Ethernet (the default), ATM, FDDI, Gigabit Ethernet, or HiPPI. This allows users to log into the system, and provides general external connection capability. Other nodes may also be connected to an external network, but this is not a requirement. If other nodes do not have an external connection, external traffic is routed through the connected nodes.

Note:

On an HP AlphaServer SC system composed of HP AlphaServer DS20L nodes, the first two nodes in each domain do not have any spare PCI slots for additional network adapters. On such domains, use Member 3 and Member 4 for this purpose.

This guide does not provide any debug information for external network issues, as such issues are site-specific. If you experience problems with your external network, contact your site network manager.

1.10 Management Server

A system can be configured with a front-end management server, and in some cases this is mandatory, as follows:

- If the system uses a 16-port switch, a management server is optional.
- If the system uses a 128-way switch, a management server is mandatory.
- If the system contains HP AlphaServer DS20L nodes, a management server is mandatory.

The management server is attached to the management network, and can be used to initiate user jobs (for example, `prun` or `allocate`) but cannot run jobs (that is, cannot be included in a parallel partition).

Table 1–2 on page 1–10 lists the IP naming convention for management servers.

When configured, the management server serves the following functions:

- Interactive Development (for example, program compile). Note that this provides the option of not using some of the system nodes for interactive access, but does not preclude doing so. Parallel jobs can be submitted from the management server using the `prun` command.

  Note:

  The pathname for commands specified to `prun` must be the same on the management server as on the system, and the current working directory must be the same on the management server as on the system.

- RMS master node (`rmshost`): hosts the SC database and central management functions. Runs the RMS central daemons — removes "one-off" management processes from Node 0.
- Performs switch management tasks for the HP AlphaServer SC Interconnect switch.
- Server for the installation process.
- RIS server for initial operating system boot step of the system node installation process.
- Runs the console manager — you can still access the systems' consoles even if all nodes are down. If you do not have a management server, you can not access other systems' consoles if the node running the console manager (usually Node 0) is down.
- Runs System Event Analyzer to debug hardware faults on itself and other nodes.

HP AlphaServer SC Version 2.6 also supports a clustered management server, qualified at two servers. This is a standard TruCluster Server implementation operating over a Gigabit Ethernet Interconnect.

1.11 Physical Storage

Storage within an HP AlphaServer SC system is classified as either local or external storage. This is merely an organizational feature of the HP AlphaServer SC system; it is not a CFS attribute. As stated in Section 1.12 on page 1–21, all CFS storage is domainwide. This organization simply reflects how we deploy storage within the system.

1.11.1 Local Storage

In HP AlphaServer SC Version 2.6, each HP AlphaServer SC node is configured with two drives[1]— typically 36GB. The HP AlphaServer SC system uses approximately 50MB of each drive, to hold the primary boot partition (first drive) and a backup boot partition (second drive). The backup disk can be booted if the primary disk fails.

The remainder of the disk capacity is shared between swap, `/tmp` and `/local` space that is specific to the node. These drives are configured during system installation.

All data stored on local storage is considered to be volatile, because the devices themselves are not highly available; that is, the devices are neither RAIDed nor multi-host connected. The failure of a SCSI card, for instance, will render the storage inaccessible. Likewise, the loss of a node will render its local file systems inaccessible.

As mentioned above, even though all file systems are part of the domainwide CFS, some nodes do not *actively* serve file systems into the CFS. For example, if not mounted `server_only`, a node's `/tmp` can be seen and accessed from any other node, using the CFS path `/cluster/members/member`*M*`/tmp`. Note, however, that `/tmp` is normally mounted `server_only`.

Note:

On an HP AlphaServer SC system composed of HP AlphaServer DS20L nodes:

- There is only one local disk for swap, local and temporary storage space.
- The Tru64 UNIX operating system disk is within the external storage configuration.
- There is no alternate boot disk.

1.11.2 External Storage

In an HP AlphaServer SC system, external storage is configured to be highly available. This is achieved by the use of:

- RAID storage
- Physical connectivity to multiple nodes

1. The first node of each domain requires a third local drive to hold the base Tru64 UNIX operating system.

One class of external storage is system storage. This is the storage that is used to host the mandatory file systems: /, /usr, and /var. These file systems hold the required system files (binaries, configuration files, libraries, and so on). This storage must remain available in order for the domain to remain viable.

System storage is configured at system installation time and comprises individual RAID subsystems — one for each 128 nodes (four domains) — which ensures that the system is highly available. A RAID subsystem is connected *via* Fibre Channel to the first two nodes of each domain; that is, to node pairs 0 and 1, 32 and 33, 64 and 65, 96 and 97, and so on.

At the storage array level, multiple physical disks are aggregated into RAID storagesets (to increase performance and availability).

At the UNIX disk level, RAID units are seen as disk devices (for example, /dev/disk/dsk3c). UNIX disks can be subdivided into UNIX partitions. These partitions are denoted by the suffixes a, b, c, d, e, f, g, and h. The 'c' partition, by definition, refers to the entire disk.

The RAID subsystems are configured in multiple-bus failover mode. Several types of RAID products are supported. The system storage serves the CFS and other user data. Other nodes can connect to the same storage arrays.

This storage, and associated file systems, is resilient to the following:

- Loss of an access path to the storage (that is, failure of a host adapter)
- Physical disk failure
- File-serving node failure

See Chapter 6 for more information about physical storage in an HP AlphaServer SC system.

1.12 Cluster File System (CFS)

CFS is a file system that is layered on top of underlying per-node AdvFS file systems. CFS does not change or manage on-disk file system data; rather, it is a value-add layer that provides the following capabilities:

- Shared root file system
  CFS provides each member of the domain with coherent access to all file systems, including the root (/) file system. All nodes in the file system share the same root.
- Coherent name space
  CFS provides a unifying view of all of the file systems served by the constituent nodes of the domain. All nodes see the same path names. A mount operation by any node is immediately visible to all other nodes. When a node boots into a domain, its file systems are mounted into the domainwide CFS.

Note:

One of the nodes physically connected to the root file system storage must be booted first (typically the first or second node of a domain). If another node boots first, it will pause in the boot sequence until the root file server is established.

- High availability and transparent failover
  CFS, in combination with the device request dispatcher, provides disk and file system failover. The loss of a file server node does not mean the loss of its served file systems. As long as one other node in the domain has physical connectivity to the relevant storage, CFS will — transparently — migrate the file service to the new node.
- Scalability
  The system is scalable, with the ability to add extra file server nodes up to the maximum domain size (currently 32).

A key feature of CFS is that every node in the domain is simultaneously a server and a client of the CFS file system. However, this does not mandate a particular operational mode; for example, a specific node can have file systems that are potentially visible to other nodes, but not actively accessed by them. In general, the fact that every node is simultaneously a server and a client is a theoretical point — normally, a subset of nodes will be active servers of file systems into the CFS, while other nodes will primarily act as clients (see Section 1.11 on page 1–19).

Figure 1–7 shows the relationship between file systems contained by disks on a shared SCSI bus and the resulting cluster directory structure. Each member boots from its own boot partition, but then mounts that file system at its mount point in the domainwide file system. Note that this figure is only an example to show how each domain member has the same view of file systems in a domain. Many physical configurations are possible, and a real domain would provide additional storage to mirror the critical root (`/`), `/usr`, and `/var` file systems.

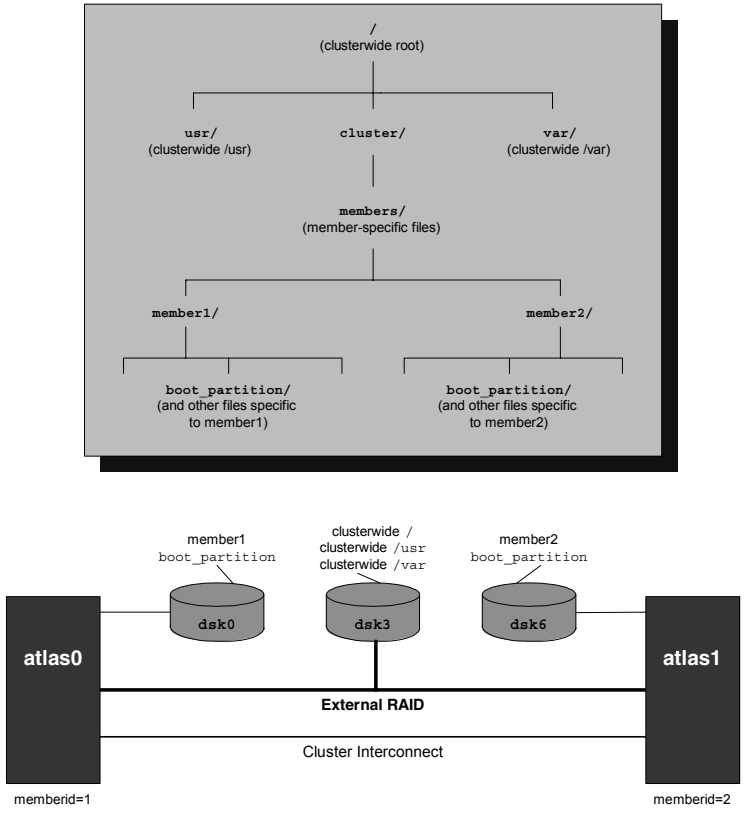

Figure 1–7 CFS Makes File Systems Available to All Domain Members

See Chapter 24 for more information about the Cluster File System.

1.13 Device Request Dispatcher (DRD)

DRD is a system software component that abstracts the physical storage devices in a CFS system. It understands which physical storage is connected to which nodes, and which storage is connected to multiple nodes.

DRD presents the higher level file system, and the administrator, with a domainwide view of the device space that can be seen from any node. For example, the `/dev/disk` directory will list all of the devices in the domain, not just those connected to a particular node. It is possible, although not recommended under normal circumstances, to have node I act as a CFS server for a file system that uses storage that is only accessible from node J.

In such a case, the DRD on node I will transfer I/O requests to its peer on node J. This would happen automatically for a file system being served on Node I, if its storage adapter or

physical path to the storage were lost. The file system would remain on node I, but raw I/O requests would be directed to node J.

A consequence of DRD is that the device name space is domainwide, and device access is highly available.

See Section 24.3.3 on page 24–8 for more information about the Device Request Dispatcher.

1.14 Resource Management System (RMS)

RMS provides the job management services for the entire system. It is responsible for running and scheduling jobs on the system. RMS runs jobs on a partition; the system administrator defines which nodes make up a partition. A partition consists of a series of consecutive nodes. The system administrator can use the `rcontrol` command to define a single partition encompassing the entire system, or a series of smaller partitions.

For example, the following set of commands will create three partitions, `fs`, `big`, and `small`:

```
# rcontrol create partition=fs configuration=day nodes='atlas[0-1]'
# rcontrol create partition=big configuration=day nodes='atlas[2-29]'
# rcontrol create partition=small configuration=day nodes='atlas[30-31]'
```

When operational, each node runs two RMS daemons: `rmsmhd` and `rmsd`. The `rmsmhd` daemon is responsible for monitoring the status of the `rmsd` daemon. The `rmsd` daemon is responsible for loading and scheduling the processes that constitute a job's processes on a particular node. The RMS daemons communicate using the management LAN.

RMS uses a number of central daemons to manage the overall system. These daemons run on either the management server (if present) or on Node 0. These daemons manage the RMS tables (in the SC database), RMS partitions, and so on.

To run a job, the user executes the `prun` command. Using the appropriate arguments to the `prun` command, the user specifies (at a minimum) the number of CPUs required, the partition on which to run the job, and the executable that is to be executed. For example, the following command runs the executable `myprog` on 512 CPUs in the partition named `parallel`:

```
# prun -n 512 -p parallel myprog
```

RMS is responsible for starting up the requisite processes on the nodes selected to run the job. The `prun` command provides many options that allow the user to control the job deployment.

While an HP AlphaServer SC system is comprised of multiple domains, the RMS system operates at the level of the complete system. The nodes that comprise a partition can span multiple domains, with only the following constraints:

- Partitions cannot overlap within a configuration.

- Imported file systems must be consistently mounted in all domains.

See Chapter 5 for more information about RMS.

1.15 Parallel File System (PFS)

PFS is a higher-level file system, which allows a number of file systems to be accessed and viewed as a single file system view. PFS can be used to provide a parallel application with scalable file system performance. This works by striping the PFS file system over multiple underlying component file systems, where the component file systems are served by different nodes.

A system does not have to use PFS; where it does, PFS will co-exist with CFS.

See Chapter 8 for more information about PFS.

1.16 SC File System (SCFS)

SCFS provides a global file system for the HP AlphaServer SC system.

The SCFS file system exports file systems from the FS domains to the other domains. It replaces the role of NFS for inter-domain sharing of files within the HP AlphaServer SC system. The SCFS file system is a high-performance system that uses the HP AlphaServer SC Interconnect.

See Chapter 7 for more information about SCFS.

1.17 Managing an HP AlphaServer SC System

In most cases, the fact that you are managing an HP AlphaServer SC system rather than a single system becomes apparent because of the occasional need to manage one of the following aspects of the system:

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

- Cluster file system (CFS), which provides domainwide coherent access to all file systems, including the root (/) file system. CFS, in combination with the device request dispatcher, provides disk and file system failover.
- HP AlphaServer SC Interconnect, which provides the systemwide communications path interconnect between system nodes.
- The console network, which allows you to connect to any node's console from any node.
- HP AlphaServer SC Parallel File System (PFS), which allows a number of data file systems to be accessed and viewed as a single file system view.
- HP AlphaServer SC File System (SCFS), which provides a global file system for the HP AlphaServer SC system.
- RAID storage, which ensures that the system is highly available.
- HP AlphaServer SC Resource Management System (RMS), which provides a programming environment for running parallel programs.
- LSF, which acts primarily as the workload scheduler, providing policy and topology-based scheduling.
- SC Performance Visualizer, which provides a graphical user interface (GUI) using the `scpvis` command, and a command line interface (CLI) using the `scload` command, to monitor performance. This provides a systemwide view of performance utilization.
- Devices
  The device name space within a domain is domainwide. Therefore, the `/dev/disk` directory will show all of the physical disk devices in the domain, not just those attached to a specific node. The `hwmgr` command shows the physical hardware on each node. The `drdmgr` command shows which nodes are attached to which node. See the `hwmgr(8)` and `drdmgr(8)` reference pages for more details.
- SC Monitor, which monitors critical hardware components in an HP AlphaServer SC system.
- SC Viewer, which displays status information for various components of the HP AlphaServer SC system.

In addition to the previous items, there are some command-level exceptions when a domain 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 domain member where the command executes. To send a message to all users who are logged in on all domain members, use the `wall -c` command.

1.18 Monitoring System Activity

Use the following commands to monitor the status of an HP AlphaServer SC system:

- `scpvis` and `scload` (SC Performance Visualizer)
  SC Performance Visualizer enables developers to monitor application execution, and enables system managers to see system resource usage. The `scload` command displays similar information to that displayed by the `scpvis` command, but in a CLI format instead of a GUI format. For more information about SC Performance Visualizer, see Chapter 11.

- `bhosts`, `bjobs`, and `xlsf`
  If you use the LSF system to manage jobs, the `bhosts` command shows the current status of hosts used by LSF. The `bjobs` command shows the status of jobs. The `xlsf` command provides a graphical interface to LSF management.

- `rinfo` and `rcontrol`
  These commands monitor the RMS system. The `rinfo` command shows the status of nodes, partitions, resources, and jobs. The `rcontrol` command shows more detailed information than the `rinfo` command. For more information about the `rinfo` and `rcontrol` commands, see Section 5.3 on page 5–6.

- `evmwatch` and `evmshow` (Tru64 UNIX Event Manager)
  If you suspect that there is a significant level of non-quiescent activity in the system, it can be useful to occasionally monitor `evm` events by logging into the node and issuing the following command:
  ```
  # evmwatch | evmshow
  ```
  For more information about the Tru64 UNIX Event Manager, see the HP *Tru64 UNIX System Administration* manual.

- `hwmgr` (Hardware Manager)
  This is a command line interface to hardware device data. Some useful options include those listed in Table 1–5.

Table 1–5 Useful hwmgr Options

Option	Description
`-view cluster`	Displays the status of all nodes in the domain
`-view hierarchy`	Displays hardware hierarchy for the entire system
`-view devices`	Shows every device and pseudodevice on the current node
`-view devices -cluster`	Shows every device and pseudodevice in the domain
`-get attribute`	Returns the attribute values for a device

For more information about `hwmgr`, see the HP *Tru64 UNIX Hardware Manager* manual.

- `scmonmgr`
  You can use the `scmonmgr` command to view the properties of hardware components. For more information about the `scmonmgr` command, see Chapter 27.
- `scevent`
  This command allows you to view the events stored in the SC database. These events indicate that something has happened to either the hardware or software of the system. For more information about HP AlphaServer SC events, see Chapter 9.
- `scviewer`
  This command provides a graphical interface that shows the status of the hardware and software in an HP AlphaServer SC system, including any related events. For more information about the `scviewer` command, see Chapter 10.
- `sra info` and `sra diag`

  To find out whether the system is up or at the SRM prompt, run the following command:
  `# sra info -nodes nodes`

  To perform more extensive system checking, run the following command:
  `# sra diag -nodes nodes`

  For more information about the `sra info` command, see Chapter 16. For more information about the `sra diag` command, see Chapter 28.

1.19 Differences between HP AlphaServer SC and TruCluster Server

The HP AlphaServer SC file system and system management capabilities are based on the TruCluster Server product. Many of the features of HP AlphaServer SC are inherited from TruCluster Server. However, there are differences between the systems:

- An HP AlphaServer SC system is comprised of several distinct underlying clusters. To distinguish between the whole HP AlphaServer SC system and the underlying clusters, we use the term “domain” when it is necessary to refer to a specific underlying cluster. A domain can have up to 32 nodes.
- HP AlphaServer SC systems of up to 32 nodes need only one domain. Larger HP AlphaServer SC systems need several domains. A domain can have up to 32 member nodes; a TruCluster Server can have up to 8 members.
- TruCluster Server uses the Memory Channel interconnect or Gigabit Ethernet interconnect to provide its clusterwide services. In the HP AlphaServer SC, each domain uses the HP AlphaServer SC Interconnect network.

- Although TruCluster Server supports multiple networks, generally all members of a TruCluster Server have network interfaces on the same network. An HP AlphaServer SC system has a more complex setup — all members of a domain have an interface on the Management Network, and only some nodes have an external network interface. This places restrictions on the way various services can be configured. These are documented in the relevant sections of the documentation.
- A small number of changes have been made to the standard TruCluster Server utilities and commands. These are described in Section 1.19.2.

1.19.1 Restrictions on TruCluster Server Features

The following restrictions apply to the way in which services can be configured in a domain:

- Do not use a quorum disk.
- TruCluster Server Reliable Datagram (RDG) is not supported.
- The Memory Channel application programming interface (API) is not supported.

1.19.2 Changes to TruCluster Server Utilities and Commands

Several HP AlphaServer SC utilities and commands are not the same as the equivalent TruCluster Server utilities and commands. These are as follows:

- Do not use `clu_create`. Instead use the `sra install` command, which in turn invokes a `clu_create` command that has been modified to work in the HP AlphaServer SC environment.
- Do not use `clu_add_member`. Instead, use the `sra install` command.
- Do not use `clu_delete_member`. Instead, use the `sra delete_member` command.
- The file `/etc/member_fstab` is provided as an alternative to `/etc/fstab` for mounting NFS file systems. `/sbin/init.d/nfsmount` uses this new file.
- In TruCluster Server systems, the cluster alias subsystem monitors network interfaces by configuring Network Interface Failure Finder (NIFF), and updates routing tables on interface failure. HP AlphaServer SC systems implement a pseudo-Ethernet interface, which spans the entire HP AlphaServer SC Interconnect. The IP suffix of this network is `-eip0`. HP AlphaServer SC systems disable NIFF monitoring on this network, to avoid unnecessary traffic on this network.

Do not use the `cfsd` daemon with SCFS file systems. Instead, SCFS filesystems exported from an FS domain can be mounted from preferred server cluster aliases (for more information, see Section 6.2 on page 6–2).

2

Booting and Shutting Down the HP AlphaServer SC System

This chapter describes how to boot and shut down the HP AlphaServer SC system.

The information in this chapter is organized as follows:

- Booting the Entire HP AlphaServer SC System (see Section 2.1 on page 2–2)
- Booting One or More Domains (see Section 2.2 on page 2–4)
- Booting One or More Domain Members (see Section 2.3 on page 2–5)
- The BOOT_RESET Console Variable (see Section 2.4 on page 2–5)
- Booting a Domain Member to Single-User Mode (see Section 2.5 on page 2–5)
- Rebooting an HP AlphaServer SC System (see Section 2.6 on page 2–6)
- Defining a Node to be Not Bootable (see Section 2.7 on page 2–6)
- Managing Boot Disks (see Section 2.8 on page 2–7)
- Shutting Down the Entire HP AlphaServer SC System (see Section 2.9 on page 2–15)
- The Shutdown Grace Period (see Section 2.10 on page 2–16)
- Shutting Down One or More Domain Members (see Section 2.11 on page 2–17)
- Shutting Down a Domain Member to Single-User Mode (see Section 2.12 on page 2–18)
- Resetting Members (see Section 2.13 on page 2–19)
- Halting Members (see Section 2.14 on page 2–19)
- Powering Off or On a Member (see Section 2.15 on page 2–19)
- Configuring Nodes In or Out When Booting or Shutting Down (see Section 2.16 on page 2–20)

2.1 Booting the Entire HP AlphaServer SC System

To boot an entire HP AlphaServer SC system, use the `sra boot` command.

The `sra boot` command requires access to the console network *via* the console manager. The console manager runs on the management server (if used) or on Node 0 (if not using a management server).

When booting a member, you must boot from the boot disk created by the `sra install` command or the `sra copy_boot_disk` command — you cannot boot from a manually-created copy of the boot disk.

By default, nodes are booted and shut down eight nodes at a time (per domain). This number is set by the `boot` and `halt` limits respectively, in the `sc_limits` table. To change these limits for a single command, use the `-width` option on the command line. To change these limits for all commands, run the `sra edit` command. The following example shows how to use the `sra edit` command to change the `boot` limit:

```
# sra edit
sra> sys
sys> edit widths

Id      Description                                                  Value
---------------------------------------------------------------------
[0  ]   RIS Install Tru64 UNIX                                       32
[1  ]   Configure Tru64 UNIX                                         32
[2  ]   Install Tru64 UNIX patches                                   32
[3  ]   Install AlphaServer SC Software Subsets                      32
[4  ]   Install AlphaServer SC Software Patches                      32
[5  ]   Install New Hardware Delivery Subsets                        32
[6  ]   Create a One Node Cluster                                    32
[7  ]   Add Member to Cluster                                        8
[8  ]   RIS Download the New Members Boot Partition                  8
[9  ]   Boot the New Member using the GENERIC Kernel                 8
[10 ]   Boot                                                         8
[11 ]   Shutdown                                                     8
[12 ]   Cluster Shutdown                                             8
[13 ]   Cluster Boot to Single User Mode                             8
[14 ]   Cluster Boot Mount Local Filesystems                         4
[15 ]   Cluster Boot to Multi User Mode                              32

---------------------------------------------------------------------

Select attributes to edit, q to quit
eg. 1-5 10 15

edit? 10
Boot                                              [8]
new value? 2

Boot                                              [2]
Correct? [y|n] y
sys>
```

If you use the default width when booting, cluster availability is not an issue for the remaining domains. However, using a width of 1 (one) will not allow the remaining domains to attain quorum: the first node will wait, partially booted, to attain quorum before completing the boot, and the `sra` command will not boot any other nodes. Do not use a width greater than 8.

2.1.1 Booting an HP AlphaServer SC System That Has a Single Management Server

To boot an HP AlphaServer SC system that has a single management server, perform the following steps:

1. Boot the management server, as follows:
   ```
   P00>>> boot boot_device
   ```
2. Boot the domains.

   If your system has more than 256 nodes, HP strongly recommends that you boot 8 domains at a time, as shown in the following example (where `atlas` is an example system name):
   ```
   atlasms# sra boot -domains atlasD[0-7]
   atlasms# sra boot -domains atlasD[8-15]
   .
   .
   .
   ```
   If your system has 256 nodes or fewer, you can boot all domains at the same time, as shown in the following example:
   ```
   atlasms# sra boot -domains all
   ```

2.1.2 Booting an HP AlphaServer SC System That Has a Clustered Management Server

To boot an HP AlphaServer SC system that has a clustered management server, perform the following steps:

1. Boot the two nodes of the management server cluster at the same time, by entering the following command on the graphics console of each management server node:
   ```
   P00>>> boot boot_device
   ```
2. Boot the domains.

   If your system has more than 256 nodes, HP strongly recommends that you boot 8 domains at a time, as shown in the following example (where `atlas` is an example system name):
   ```
   atlasms# sra boot -domains atlasD[0-7]
   atlasms# sra boot -domains atlasD[8-15]
   .
   .
   .
   ```

If your system has 256 nodes or fewer, you can boot all domains at the same time, as shown in the following example:

```
atlasms# sra boot -domains all
```

2.1.3 Booting an HP AlphaServer SC System That Has No Management Server

To boot an HP AlphaServer SC system that does not have a management server, perform the steps shown in the following example (where `atlas` is an example system name):

1. Boot the first two nodes of the first domain (`atlasD0` in this example) at the same time, by typing the following command on the graphics console for Node 0 and on the graphics console for Node 1:

   ```
   P00>>> boot boot_device
   ```

   If not using a management server, the console manager runs on the first domain (`atlasD0`). The console manager will not start until `atlasD0` is established. As each of the first three nodes of each domain has a vote, two of these three nodes in `atlasD0` must be booted before the console manager will function.

2. When these nodes have booted, boot the remaining nodes.

   If your system has more than 256 nodes, HP strongly recommends that you boot 8 domains at a time, as shown in the following example:

   ```
   atlasms# sra boot -domains atlasD[0-7]
   atlasms# sra boot -domains atlasD[8-15]
   .
   .
   .
   ```

   If your system has 256 nodes or fewer, you can boot all domains at the same time, as shown in the following example:

   ```
   atlasms# sra boot -domains all
   ```

2.2 Booting One or More Domains

Use the `sra boot` command to boot one or more domains, as shown in the following examples:

- Example 1: Booting a single domain:

  ```
  # sra boot -domains atlasD1
  ```

- Example 2: Booting multiple domains:

  ```
  # sra boot -domains 'atlasD[2-4,6-9]'
  ```

2.3 Booting One or More Domain Members

Use the `sra boot` command to boot one or more domain members, as shown in the following examples:

- Example 1: Booting a single node:
  `# sra boot -nodes atlas5`
- Example 2: Booting multiple nodes:
  `# sra boot -nodes 'atlas[5-10,12-15]'`

2.4 The BOOT_RESET Console Variable

The HP AlphaServer SC software sets the BOOT_RESET console variable on all nodes to OFF. This means that when a system boots, it does not perform a full reset (which includes running memory diagnostics). The BOOT_RESET console variable is set to OFF because the accumulated effect of resetting each node would noticeably increase the system boot time.

If you would prefer all nodes to reset before a boot, it is more efficient to use the `sra` command to initialize the nodes in parallel and then boot them, as shown in the following example:

1. Initialize the nodes (which should be at the SRM console prompt), as follows:
   `atlasms> sra command -nodes all -command 'INIT'`
2. When the initialization has completed, boot the nodes, as follows:
   `atlasms> sra boot -nodes all`

2.5 Booting a Domain Member to Single-User Mode

You can use the `sra boot` command to boot a domain member to single-user mode, as shown in the following example:

```
# sra boot -nodes atlas5 -single yes
```

To boot a system from single-user mode to multi-user mode, use the standard `sra boot` command as follows:

```
# sra boot -nodes atlas5
```

2.6 Rebooting an HP AlphaServer SC System

To reboot an entire HP AlphaServer SC system, run the following command:

```
# sra shutdown -domains all -reboot yes
```

Note:

Do not run this command if your system does not have a management server.

Instead, shut down the system as described in Section 2.9.3 on page 2–16, and then boot the system as described in Section 2.1.3 on page 2–4.

2.7 Defining a Node to be Not Bootable

If, for any reason, you do not wish to boot a particular member — for example, if the node is shut down for maintenance reasons — you can use the `sra edit` command to indicate that the node is not bootable. The following example shows how to indicate that `atlas7` is not bootable (where `atlas` is an example system name).

```
# sra edit
sra> node
node> edit atlas7

Id      Description                                     Value
------------------------------------------------------------------
.
.
.
[11 ]   Bootable or not                                 1               *
.
.
.
* = default generated from system
# = no default value exists

------------------------------------------------------------------

Select attributes to edit, q to quit
eg. 1-5 10 15

edit? 11

enter a new value, probe or auto
auto  = generate value from system
probe = probe hardware for value

Bootable or not                         [1]                (auto)
new value? 0

Bootable or not                         [0]                 (new)
correct? [y|n] y
node> quit
sra> quit
```

Setting the `Bootable or not` value to `0` will allow you to boot all of the other nodes in the domain using the `-domains atlasD0` value, instead of the more difficult specification `-nodes atlas[0-6,8-31]`, as follows:

```
# sra boot -domains atlasD0
```

In HP AlphaServer SC Version 2.6, you can also set the bootable state of a node by specifying the `-bootable` option when running the `sra boot` or `sra shutdown` command.

In the following example, the specified nodes are shut down and marked as not bootable so that they cannot be booted by the `sra` command until they are once more declared bootable:

```
# sra shutdown -nodes 'atlas[4-8]' -bootable no
```

In the following example, the specified nodes are marked as bootable and then booted:

```
# sra boot -nodes 'atlas[4-8]' -bootable yes
```

There is no default value for the `-bootable` option; if it is not explicitly specified by the user, no change is made to the bootable state.

2.8 Managing Boot Disks

The information in this section is organized as follows:

- The Alternate Boot Disk (see Section 2.8.1 on page 2–8)
- Configuring and Using the Alternate Boot Disk (see Section 2.8.2 on page 2–9)
- Booting from the Alternate Boot Disk (see Section 2.8.3 on page 2–12)
- The server_only Mount Option (see Section 2.8.4 on page 2–13)
- Creating a New Boot Disk from the Alternate Boot Disk (see Section 2.8.5 on page 2–14)

Note:

On an HP AlphaServer SC system composed of HP AlphaServer DS20L nodes:

- There is only one local disk for swap, local and temporary storage space.
- The Tru64 UNIX operating system disk is within the external storage configuration, and is visible only to member 1 through the use of multipathing.
- There is no alternate boot disk.

Therefore, most of this section does not apply to an HP AlphaServer SC system composed of HP AlphaServer DS20L nodes — the exception is Section 2.8.4.

2.8.1 The Alternate Boot Disk

Setting up an alternate boot disk is an optional task. If you choose to *configure* an alternate boot disk, you can then choose whether to *use* the alternate boot disk:

- Configuring allows you to boot from the alternate boot disk if the primary boot disk fails.
- Using allows you to mount the `tmp` and `local` partitions from the alternate boot disk, and to use its swap space.

Before you can use an alternate boot disk, you must first configure it.

Configuring an alternate boot disk does not affect the swap space or mount partitions. However, when using an alternate boot disk, the swap space from the alternate boot disk is added to the swap space from the primary boot disk, thus spreading the available swap space over two disks.

If booting from the primary boot disk, the `tmp` and `local` partitions on the alternate boot disk are mounted on `/tmp1` and `/local1` respectively.

If booting from the alternate boot disk, the `tmp` and `local` partitions on the alternate boot disk are mounted on `/tmp` and `/local` respectively — no `tmp` or `local` partitions are mounted on the primary boot disk.

All four mount points (`/tmp`, `/local`, `/tmp1`, and `/local1`) are CDSLs (Context-Dependent Symbolic Links) to member-specific files.

Table 2–1 shows how using an alternate boot disk affects the `tmp` and `local` partitions, and the swap space.

Table 2–1 Effect of Using an Alternate Boot Disk

Disks in Use	Booting from	tmp	local	swap
Primary boot disk only	Primary boot disk	The `tmp` partition on the primary boot disk is mounted on `/tmp`	The `local` partition on the primary boot disk is mounted on `/local`	Swap space of primary boot disk only
Primary boot disk and alternate boot disk	Primary boot disk	The `tmp` partition on the primary boot disk is mounted on `/tmp`; the `tmp` partition on the alternate boot disk is mounted on `/tmp1`	The `local` partition on the primary boot disk is mounted on `/local`; the `local` partition on the alternate boot disk is mounted on `/local1`	Swap space of both boot disks combined
Primary boot disk and alternate boot disk	Alternate boot disk	The `tmp` partition on the alternate boot disk is mounted on `/tmp`	The `local` partition on the alternate boot disk is mounted on `/local`	Swap space of alternate boot disk only

Table 2–1 Effect of Using an Alternate Boot Disk

Disks in Use	Booting from	tmp	local	swap
Alternate boot disk only	Alternate boot disk	The `tmp` partition on the alternate boot disk is mounted on `/tmp`	The `local` partition on the alternate boot disk is mounted on `/local`	Swap space of alternate boot disk only

2.8.2 Configuring and Using the Alternate Boot Disk

If you wish to configure and use the alternate boot disk, answer **`yes`** to the two relevant questions asked by the `sra setup` command during the installation process, as described in Chapter 5 or Chapter 6 (depending on the configuration) of the *HP AlphaServer SC Installation Guide*.

The information in this section is organized as follows:

- How to Use an Already-Configured Alternate Boot Disk (see Section 2.8.2.1 on page 2–9)
- How to Configure and Use an Alternate Boot Disk After Installation (see Section 2.8.2.2 on page 2–10)
- How to Stop Using the Alternate Boot Disk (see Section 2.8.2.3 on page 2–12)

2.8.2.1 How to Use an Already-Configured Alternate Boot Disk

If you configured the alternate boot disk during the installation process but did not use it, you can later decide to use the alternate boot disk by performing the following steps:

1. Set SC_USE_ALT_BOOT to 1 in each node's `/etc/rc.config` file, as follows:
   ```
   # scrun -n all '/usr/sbin/rcmgr set SC_USE_ALT_BOOT 1'
   ```
2. Run the `shutdown` command to reboot the nodes, as follows:
   ```
   # sra shutdown -nodes all
   ```

If you had not configured an alternate boot disk, setting SC_USE_ALT_BOOT in the `/etc/rc.config` file will have no effect.

2.8.2.2 How to Configure and Use an Alternate Boot Disk After Installation

If you chose not to configure the alternate boot disk during the installation process, you can do so later using either the `sra setup` command or the `sra edit` command, as described in this section.

Method 1: Using sra setup

To use the `sra setup` command to configure the alternate boot disk, perform the following steps:

1. Run the `sra setup` command, as described in Chapter 5 or Chapter 6 (depending on the configuration) of the *HP AlphaServer SC Installation Guide*:
   a. When asked if you would like to configure an alternate boot device, enter **`yes`**.
   b. When asked if you would like to use an alternate boot device, enter **`yes`**.
2. Build the new boot disk, as follows:
   ```
   # sra copy_boot_disk -nodes all
   ```

Method 2: Using sra edit

To use the `sra edit` command to configure the alternate boot disk, perform the following steps:

1. Use the `sra edit` command to add an alternate boot disk (that is, a second image) to the SC database:
   ```
   # sra edit
   sra> sys
   sys> show im
   valid images are [unix-first cluster-first cluster-second boot-first
   gen_boot-first]
   sys> add image boot-second
   sys> show im
   valid images are [unix-first cluster-first cluster-second boot-first
   boot-second gen_boot-first]
   sys>
   ```

Note:

Setting the `use alternate boot` value in the SC database has no effect; this value is used only when building the domain.

2. Edit the second image entry to set the SRM boot device and UNIX disk name.

```
sys> edit image boot-second
Id      Description                                          Value
-----------------------------------------------------------------
[0  ]   Image role                                           boot
[1  ]   Image name                                           second
[2  ]   UNIX device name                                     dsk1
[3  ]   SRM device name                                      #
[4  ]   Disk Location (Identifier)
[5  ]   default or not                                       no
[6  ]   swap partition size (%)                              30
[7  ]   tmp partition size (%)                               35
[8  ]   local partition size (%)                             35

-----------------------------------------------------------------
Select attributes to edit, q to quit
eg. 1-5 10 15

probe = probe for value

edit? 3
SRM device name                     [#]
new value? dka100

SRM device name                     [dka100]
Correct? [y|n] y
sys>
```

Note:

If you configure an alternate boot disk during the installation process, the swap space is set to 15% for the primary boot disk and 15% for the alternate boot disk.

However, if you use `sra edit` to configure an alternate boot disk after installation as described in this section, the swap space is set to 30% for each boot disk. You may consider this to be too much; if so, see Section 21.15.1 on page 21–38 for more information on how to change the swap space.

---

3. If you wish to use the alternate disk, update the `/etc/rc.config` file on each member to set the variable SC_USE_ALT_BOOT to 1, as follows:

```
# scrun -n all '/usr/sbin/rcmgr set SC_USE_ALT_BOOT 1'
```

   If you do not wish to use the alternate disk, skip this step.

4. Build the new boot disk, as follows:

```
# sra copy_boot_disk -nodes all
```

2.8.2.3 How to Stop Using the Alternate Boot Disk

If you no longer wish to use the alternate boot disk on a particular node (for example, `atlas2`), perform the following steps:

1. Set SC_USE_ALT_BOOT to zero in the node's `/etc/rc.config` file, as follows:
   # scrun -n atlas2 '/usr/sbin/rcmgr set SC_USE_ALT_BOOT 0'
2. Run the `sra shutdown` command to reboot the node, as follows:
   # sra shutdown -nodes atlas2 -reboot yes

Note:

Do not simply reboot the node. Use the `sra shutdown` command as shown above, to ensure that the `sra_clu_min stop` script is run. This script ensures that the alternate disk is removed from the `swapdevice` entry in the member's `/etc/sysconfigtab` file.

If you had not configured an alternate boot disk, setting SC_USE_ALT_BOOT in the `/etc/rc.config` file will have no effect.

2.8.3 Booting from the Alternate Boot Disk

If a node's boot disk fails, you may boot the alternate boot disk. However, it is not simply a matter of booting the alternate disk from the console. You must perform the following steps:

1. Ensure that the node whose boot disk is being switched (for example, `atlas5`) is at the SRM console prompt.
2. Run the following command from another node in the domain:
   # sra switch_boot_disk -nodes atlas5

Note:

The `sra switch_boot_disk` command will not work if run from a management server. You must run the command from another node that is in the same domain as the node whose boot disk has failed.

You can use the `sra switch_boot_disk` command repeatedly to toggle between primary and alternate boot disks. The `sra switch_boot_disk` command will do the following:

a. Ensure that the file domains `root`*N*`_domain`, `root`*N*`_tmp`, `root`*N*`_local`, `root`*N*`_tmp1`, and `root`*N*`_local1` point to the correct boot disk, where *N* is the member ID of the node (in the above example, *N* = 6).

b. Change the default boot disk for the node. This setting is stored in the SC database. The SC database refers to the boot disk as an **image**, where image 0 (the first image) is the primary boot disk and image 1 (the second image) is the alternate boot disk. The default image is used by the `sra boot` command to determine which disk to boot.

   You can use the `sra edit` command to view the current default image, as follows:

```
# sra edit
sra> node
node> show atlas5
```

   This displays a list of node-specific settings, including the default image:

```
[9 ] Node specific image_default              1
```

3. Boot the node — the `sra boot` command will automatically use the alternate boot disk:

```
# sra boot -nodes atlas5
```

Note:

If the node's local disks were not originally mounted as `server_only`, this step may fail — see Section 2.8.4 on page 2–13 for more information.

2.8.4 The server_only Mount Option

By default, local disks are mounted as **server only**, by using the `-o server_only` option. Specifying this mount option means that if a node panics or is reset, the local file systems are automatically unmounted.

However, mounting these disks as `server_only` also means that these file systems will not be accessible from other members in the domain. If you wish to remove the `server_only` mount option, run the following command:

```
# scrun -d all '/usr/sbin/rcmgr -c delete SC_MOUNT_OPTIONS'
```

If you do not specify this mount option, the local file systems (for example, `root`*N*`_domain`, `root`*N*`_tmp`, `root`*N*`_local`, `root`*N*`_tmp1`, or `root`*N*`_local1`) will remain mounted if a node panics or is reset. This can make it difficult to delete a member, or to switch to the alternate boot disk — if the node's boot disk fails, any attempt to boot the alternate boot disk will fail until the local file systems are unmounted.

If a node cannot be booted, the only way to unmount its local disks is to shut down the entire domain. Once shut down, the domain may be booted. The node with the failed boot disk can then be booted from its alternate boot disk.

If you wish to reapply the `server_only` mount option, run the following command:

```
# scrun -d all '/usr/sbin/rcmgr -c set SC_MOUNT_OPTIONS -o server_only'
```

2.8.5 Creating a New Boot Disk from the Alternate Boot Disk

If a boot disk fails on an HP AlphaServer ES45 or HP AlphaServer ES40 node, use the `sra copy_boot_disk` command to build a new boot disk, as described in this section.

Note:

If a boot disk fails on an HP AlphaServer DS20L node, you must delete and then re-add the affected member (see Section 21.6 on page 21–15, and Section 21.7 on page 21–17 respectively). If the boot-disk failure occurs on the first member of a domain, you must rebuild the entire domain (Section 21.8.2 on page 21–18).

To rebuild a boot disk on an HP AlphaServer ES45 or HP AlphaServer ES40 node, perform the following steps:

1. Ensure that the node whose boot disk has failed (for example, `atlas5`) is at the SRM console prompt.
2. Switch to the alternate boot disk by running the following command from another node in the domain:
   ```
   # sra switch_boot_disk -nodes atlas5
   ```
3. Replace the failed disk.
4. Boot the node from the alternate boot disk, as follows:
   ```
   # sra boot -nodes atlas5
   ```
   When the node is booted from the alternate boot disk, the swap space from the primary boot disk is not used.
5. If no graphics console is attached to the node, build the new boot disk as follows:
   ```
   # sra copy_boot_disk -nodes atlas5
   ```
   If a graphics console is attached to the node, perform the following steps instead of the above command:
   a. Enable root telnet access by placing a `ptys` entry in the `/etc/securettys` file.
   b. Specify the `-telnet` option in the `sra copy_boot_disk` command, so that you connect to the node using `telnet` instead of the console, as follows:
      ```
      # sra copy_boot_disk -nodes atlas5 -telnet yes
      ```
   c. Disable root telnet access by removing the `ptys` entry from the `/etc/securettys` file.
6. Shut down the node, as follows:
   ```
   # sra shutdown -nodes atlas5
   ```
7. Switch back to using the primary boot disk, as follows:
   ```
   # sra switch_boot_disk -nodes atlas5
   ```

8. Boot from the primary boot disk, as follows:
   ```
   # sra boot -nodes atlas5
   ```

Note:

For the `sra copy_boot_disk` command to work, the primary and alternate boot disks must be the first and second local disks on the system.

When the node is booted from the alternate boot disk, the swap space from the primary boot disk is not used.

The `sra copy_boot_disk` command may be used to update an alternate boot disk if changes have been made to the primary disk; for example, after building a new `vmunix` or changing the `sysconfigtab` file.

2.9 Shutting Down the Entire HP AlphaServer SC System

To shut down an entire HP AlphaServer SC system, use the `sra shutdown` command.

The `sra shutdown` command requires access to the console network *via* the console manager. The console manager runs on the management server (if used) or on Node 0 (if not using a management server).

The `sra shutdown` command fails if a `clu_quorum` or `sra delete_member` command is in progress, or if members are being added to the domain.

Before shutting down nodes, you should stop all jobs running on the nodes. See Section 5.8.3 on page 5–63 for more information on how to do this.

2.9.1 Shutting Down an HP AlphaServer SC System That Has a Single Management Server

To shut down an HP AlphaServer SC system that has a single management server, perform the following steps:

1. Shut down the domains, as follows:
   ```
   atlasms# sra shutdown -domains all
   ```
2. Shut down the management server, as follows:
   ```
   atlasms# shutdown -h now
   ```

2.9.2 Shutting Down an HP AlphaServer SC System That Has a Clustered Management Server

To shut down an HP AlphaServer SC system that has a clustered management server, perform the following steps:

1. Shut down the domains, as follows:
   ```
   atlasms0# sra shutdown -domains all
   ```
2. Shut down the clustered management server, as follows:
   ```
   atlasms0# shutdown -ch now
   ```

2.9.3 Shutting Down an HP AlphaServer SC System That Has No Management Server

To shut down an HP AlphaServer SC system that does not have a management server, perform the following steps:

1. Shut down all domains except the first domain, as follows:
   ```
   atlas0# sra shutdown -domains 'atlasD[1-31]'
   ```
   In an HP AlphaServer SC system with multiple domains and no management server, the console manager runs on the first domain (`atlasD0`). Therefore, shut down the other domains first, as shutting down `atlasD0` will remove access to the other nodes' consoles.
2. When these nodes have shut down, shut down the first domain, as follows:
   ```
   atlas0# sra shutdown -domains atlasD0
   ```

2.10 The Shutdown Grace Period

The **shutdown grace period** is the time between when the `shutdown` command is issued and when actual shutdown occurs. During this time, the `sra install` command is disabled and new members cannot be added to the domain.

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

1. Identify the PIDs associated with the `shutdown` command. For example:
   ```
   # ps ax | grep -v grep | grep 'shutdown'
   14680 ttyp5 I < 0:00.01 /usr/bin/shutdown -ch
   ```
   Depending on how far along `shutdown` is in the grace period, the `ps` command might show either `/usr/bin/shutdown` or `/usr/sbin/clu_shutdown`.
2. Terminate all shutdown processes by specifying their PIDs in a `kill` command on the originating member. For example:
   ```
   # kill 14680
   ```

If you kill the shutdown processes during the grace period, the shutdown is cancelled — you should then manually delete the `/etc/nologin` and `/cluster/admin/.clu_shutdown` files.

For more information, see the `shutdown(8)` reference page.

2.11 Shutting Down One or More Domain Members

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

In an HP AlphaServer SC system, the first three nodes in each domain are voting members. If any of these nodes is currently down, each of the other two nodes is a critical voting member.

2.11.1 Shutting Down One or More Non-Voting Members

Use the `sra shutdown` command to shut down one or more non-voting domain members; that is, any node other than domain member 1, 2, or 3. You can run this command from any node, as shown in the following examples:

- To shut down a single non-voting member:
  ```
  # sra shutdown -nodes atlas5
  ```
- To shut down a number of non-voting members:
  ```
  # sra shutdown -nodes 'atlas[5-10,12-15]'
  ```

2.11.2 Shutting Down Voting Members

When using the recommended configuration — that is, three voting members— you can shut down one of the voting members using the standard `sra shutdown` command, and quorum will be maintained; the domain will continue to be operational. To shut down two of the voting members, perform the steps shown in the following example (where `atlas` is an example system name):

1. Ensure that all domain members are up.
2. Use the following command to set the node votes to 0 (zero) on the two members to be shut down:
   ```
   # clu_quorum -m atlas2 0
   # clu_quorum -m atlas1 0
   ```

Note:

This command modifies member `/etc/sysconfigtab` attributes and not the member kernels. As a result, the domain is still running with the old attribute values.

3. Shut down the entire domain:
   ```
   # shutdown -ch
   ```

Note:

Step 2 does not affect expected votes in the running kernels; therefore, if you halt two voting members, the other member or members will lose quorum and hang.

4. Boot the members that you wish to remain in the domain.
   - If using a management server, issue the following command:
     ```
     # sra boot -nodes 'atlas[0,3-31]'
     ```
   - If not using a management server, issue the following command from the SRM console of Node 0:
     ```
     P00>>> boot boot_device
     ```
     Once Node 0 has booted, boot the rest of the domain, as follows:
     ```
     # sra boot -nodes 'atlas[3-31]'
     ```

Note:

Once `atlas1` and `atlas2` have booted, you should assign a vote to each, as described in Chapter 8 of the *HP AlphaServer SC Installation Guide*.

2.12 Shutting Down a Domain Member to Single-User Mode

If you need to shut down a domain 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 domain and that the running domain has a minimal reliance on the member running in single-user mode. In particular, halting the member satisfies services that require the domain member to have a status of `DOWN` before completing a service failover. If you do not first halt the domain member, the services do not fail over as expected.

To take a domain member to single-user mode, perform the following steps from the management server (if used) or Node 0 (if not using a management server):

```
atlasms# sra shutdown -nodes atlas2
atlasms# sra boot -nodes atlas2 -single yes
```

A domain 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 domain 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.

2.13 Resetting Members

Using the `sra shutdown` command is the recommended way to shut down a member. However, if a member is unresponsive to console commands, you may reset it using the `sra reset` command, as shown in the following example:

```
# sra reset -nodes atlas9
```

This command resets the member by entering the RMC mode and issuing a reset command. For HP AlphaServer DS20L nodes, reset is simulated by using Wakeup On LAN (WOL) functionality.

2.14 Halting Members

Use the `sra halt_in` command to halt one or more nodes, as shown in the following example:

```
# sra halt_in -nodes atlas9
```

This command enters the console RMC mode and issues a halt-in command.

Release the halt button with the following command:

```
# sra halt_out -nodes atlas9
```

Note:

In HP AlphaServer SC Version 2.6, the `sra halt_in` and `sra halt_out` commands are not available on HP AlphaServer DS20L nodes.

2.15 Powering Off or On a Member

Use the `sra power_off` command to power off one or more nodes, as shown in the following example:

```
# sra power_off -nodes all
```

This command enters the console RMC mode and issues a power-off command.

Restore the power with the following command:

```
# sra power_on -nodes all
```

Note:

In HP AlphaServer SC Version 2.6, the `sra power_on` and `sra power_off` commands are only available on an HP AlphaServer DS20L node when the node is already at the SRM prompt.

2.16 Configuring Nodes In or Out When Booting or Shutting Down

In HP AlphaServer SC Version 2.6, you can configure nodes in or out of an RMS partition when running the `sra boot` or `sra shutdown` command, by specifying the `-configure` option.

For the `sra boot` command, the default value is `-configure none`. If the node had been configured out, it will remain configured out unless you specify the `-configure in` option. If the node had been configured in, it remains configured in.

For the `sra shutdown` command, the default value is `-configure out`. This configures the node out of the partition before shutting it down. If you specify the `-configure none` option, the node remains "as is", but RMS will automatically configure it out once it is down.

Specify the `-configure in` option to reboot a configured-out node and configure it back into the partition.

For more information about the `sra` commands and related options, see Chapter 16.

3

Managing the SC Database

The HP AlphaServer SC database contains both static configuration information and dynamic data. The SC database is a shared resource used by many critical components in an HP AlphaServer SC system.

The `sra setup` command creates the SC database during the installation process. The database mechanisms are based on a simplified Structured Query Language (SQL) system. The `msql3d` daemon acts as a server for the SC database — it responds to requests from the various utilities and daemons belonging to the HP AlphaServer SC system.

The SC database supersedes both the RMS database (a SQL database) and the SRA database (stored in `/var/sra/sra-database.dat`, a flat file).

This information in this chapter is arranged as follows:

- Backing Up the SC Database (see Section 3.1 on page 3–2)
- Reducing the Size of the SC Database by Archiving (see Section 3.2 on page 3–4)
- Restoring the SC Database (see Section 3.3 on page 3–7)
- Deleting the SC Database (see Section 3.4 on page 3–10)
- Monitoring /var (see Section 3.5 on page 3–12)
- Cookie Security Mechanism (see Section 3.6 on page 3–12)
- SC Database Integrity Check Utility (see Section 3.7 on page 3–13)

3.1 Backing Up the SC Database

There are three ways to back up the SC database:

- Back Up the Complete SC Database Using the rmsbackup Command (see Section 3.1.1)
- Back Up the SC Database, or a Table, Using the rmstbladm Command (see Section 3.1.2)
- Back Up the SC Database Directory (see Section 3.1.3)

The first two backup methods only back up the standard tables in the SC database. They do not back up any user-created tables. To back up user-created tables, you must back up the entire SC database directory, as described in Section 3.1.3 on page 3–4.

3.1.1 Back Up the Complete SC Database Using the rmsbackup Command

To back up the complete SC database, run the `rmsbackup` command as the `root` user on the management server (if used) or on Node 0 (if not using a management server), as follows:

```
# rmsbackup
```

The `rmsbackup` command backs up all of the tables (structure and content) as a set of SQL statements, which you can later restore using the `rmstbladm` command. The SQL statements are written to a backup file called *`system_date`*`.sql`, where

- *`system`* is the name of the HP AlphaServer SC system
- *`date`* is the date on which the file was created, specified in the following format: *`YYYY-MM-DD-HH:mm`*

The `rmsbackup` command then compresses the backup file using the `gzip(1)` command, and stores it in the `/var/rms/backup` directory.

For example, if the SC database of the `atlas` system was backed up at 1:10 a.m. on 15th February, 2003, the resulting backup file would be as follows:

```
/var/rms/backup/atlas_2003-02-15-01:10.sql.gz
```

By default, the `rmsbackup` command will first archive the database, to remove any redundant entries. To omit the archive operation, specify the `-b` flag, as follows:

```
# rmsbackup -b
```

The SC database does not support a transaction mechanism, so it is possible to create a backup file whose contents are not fully consistent. Generally, the consistency of dynamic information (for example, RMS resources and jobs) does not matter, because the system adjusts the data if this backup is subsequently restored. However, you should ensure that no configuration changes are made during the backup operation. Configuration changes are made by the following commands — do not use any of these commands while the database is being backed up:

- `rcontrol create`

- rcontrol remove
- rcontrol start
- scmonmgr add
- scmonmgr server
- sra copy_boot_disk
- sra delete_member
- sra edit
- sra install
- sra setup
- sra update_firmware
- sysman pfsmgr
- sysman sc_cabinet
- sysman scfsmgr
- sysman sra_user

You can use the cron(8) command to schedule regular database backups. To minimize problems, choose a time when the above commands will not be used. For example, to use the cron command to run the rmsbackup command daily at 1:10 a.m., add the following line to the crontab file on the rmshost system:

```
10 1 * * * /usr/bin/rmsbackup
```

Note that you must specify the site-specific path for the rmsbackup command; in the above example, the rmsbackup command is located in the /usr/bin directory.

3.1.2 Back Up the SC Database, or a Table, Using the rmstbladm Command

You can use the rmstbladm command to save a copy of the database, as follows:

```
# rmstbladm -d > filename
```

where *filename* is the name of the file to which you are saving the database.

To save the contents of a specific table, use the rmstbladm command with the -t option, as shown in the following example:

```
# rmstbladm -d -t events > filename
```

In the above example, all records from the events table are saved (as SQL statements) in the *filename* file.

The *filename* files may be backed up using any standard backup facility.

3.1.3 Back Up the SC Database Directory

The database is contained in files in the `/var/rms/msqldb-3/rms_system` directory, where *system* is the name of the HP AlphaServer SC system. You cannot back up this directory while the `msql3d` daemon is running. To back up the database as complete files, perform the following steps in the specified order:

1. Stop mSQL as described in Section 5.9.2 on page 5–69.
2. Back up the files in the `/var/rms/msqldb-3/rms_system` directory.
3. Start mSQL as described in Section 5.9.3 on page 5–71.

3.2 Reducing the Size of the SC Database by Archiving

The information in this section is organized as follows:

- Deciding What Data to Archive (see Section 3.2.1)
- Data Archived by Default (see Section 3.2.2)
- The archive_tables Table (see Section 3.2.3)
- The rmsarchive Command (see Section 3.2.4)

3.2.1 Deciding What Data to Archive

If you take no action, the SC database will increase in size and will eventually affect RMS performance. Periodically, you should remove redundant records from certain tables that hold operational or transactional records, to ensure the efficient operation of the database.

Any operational records required for subsequent analysis should be archived first. For example, the records in the `resources` table can be analyzed to monitor usage patterns.

Transactional records, on the other hand, are of little interest once the transaction has completed, and can simply be deleted.

Table 3–1 lists the tables in which RMS stores operational or transactional records.

Table 3–1 Tables In Which RMS Stores Operational or Transactional Records

Table Name	Description
`acctstats`	RMS creates a record in the `acctstats` table each time it creates a record in the `resources` table.
`disk_stats`	Reserved for future use.

Table 3–1 Tables In Which RMS Stores Operational or Transactional Records

Table Name	Description
events	RMS creates a record in the events table each time a change occurs in node or partition status or environment.
jobs	RMS creates a record in the jobs table each time a job is started by the prun command.
link_errors	RMS creates a record in the link_errors table each time the switch manager detects a link error.
resources	RMS creates a record in the resources table each time a resource is requested by the allocate or prun command.
transactions	RMS creates a record in the transactions table each time the rmstbladm command is run, and each time the database is modified by the rmsquery command.

3.2.2 Data Archived by Default

The rmsarchive command automates the process of archiving old data. It archives certain records by default, as described in Table 3–2.

Table 3–2 Records Archived by Default by the rmsarchive Command

Table Name	Archive a Record from This Table If...
acctstats	The record is older than 48 hours, and all CPUs have been deallocated.
disk_stats	Reserved for future use.
events	The record is older than 48 hours, and the event has been handled.
jobs	The record is older than 48 hours, and the status is not blocked, reconnect, running, or suspended.
link_errors	The record is older than 168 hours.
resources	The record is older than 48 hours, and the status is not allocated, blocked, queued, reconnect, or suspended.

You can change the default period for which data is kept (that is, not archived) by modifying the lifetime field in the archive_tables table, as described in Section 3.2.3.4.

3.2.3 The archive_tables Table

The information in this section is organized as follows:

- Description of the archive_tables Table (see Section 3.2.3.1 on page 3–6)

- Adding Entries to the archive_tables Table (see Section 3.2.3.2 on page 3–6)
- Deleting Entries from the archive_tables Table (see Section 3.2.3.3 on page 3–6)
- Changing Entries in the archive_tables Table (see Section 3.2.3.4 on page 3–7)

3.2.3.1 Description of the archive_tables Table

The criteria used by the `rmsarchive` command to archive and delete records are held in the `archive_tables` table in the SC database. Each record in this table has the following fields:

`name`	The name of a table to archive.
`lifetime`	The maximum time in hours for the data to remain in the table.
`timefield`	The name of the field in table `name` that records the data's lifetime.
`selectstr`	A SQL select string, based on fields in table `name`, that determines which records to archive.

3.2.3.2 Adding Entries to the archive_tables Table

Use the `rmsquery` command to add entries to the `archive_tables` table.

For example, records from the `transactions` table are not archived by default. If you wish to archive these records, use the `rmsquery` command to create a suitable entry in the `archive_tables` table, as shown in the following examples:

- To set up `rmsarchive` to delete from the `transactions` table all records that are more than 15 days old, run the following command:

```
$ rmsquery "insert into archive_tables (name,lifetime,timefield) \
values ('transactions',360,'mtime')"
```

- To set up `rmsarchive` to delete from the `transactions` table all records for completed transactions that are more than 15 days (360 hours) old, run the following command:

```
$ rmsquery "insert into archive_tables (name,lifetime,timefield,selectstr) \
values ('transactions',360,'mtime','status = \'complete\'')"
```

  You must use a backslash before each single quote in the `selectstr` text, so that the SQL statement will be parsed correctly.

3.2.3.3 Deleting Entries from the archive_tables Table

Use the `rmsquery` command to delete entries from the `archive_tables` table.

For example, to delete from the `archive_tables` table all records related to the `transactions` table, run the following command:

```
$ rmsquery "delete from archive_tables where name='transactions'"
```

Note:

Do not delete the default entries from the `archive_tables` table (see Table 3–2).

3.2.3.4 Changing Entries in the archive_tables Table

Use the `rmsquery` command to change existing entries in the `archive_tables` table.

For example, to change the lifetime of the `events` table to 10 days, run the following command:

```
$ rmsquery "update archive_tables set lifetime=240 where name='events'"
```

Note:

In the default entries (see Table 3–2), do not change any field except `lifetime`.

3.2.4 The rmsarchive Command

To archive the database, run the `rmsarchive` command as the `root` user on the management server (if used) or on Node 0 (if not using a management server), as follows:

```
# rmsarchive
```

The `rmsarchive` command archives the records as a set of SQL statements in a file called *system_date*`.sql`, where

- *system* is the name of the HP AlphaServer SC system
- *date* is the date on which the `rmsarchive` command was run, specified in the following format:
  YYYY-MM-DD-HH`:`*mm*

The `rmsarchive` command then compresses the archive file using the `gzip(1)` command, and stores it in the `/var/rms/archive` directory.

For example, if the `rmsarchive` command was run on the `atlas` system at 1:10 a.m. on 15th February, 2003, the resulting archive file would be as follows:

```
/var/rms/archive/atlas_2003-02-15-01:10.sql.gz
```

3.3 Restoring the SC Database

The information in this section is organized as follows:

- Restore the Complete SC Database (see Section 3.3.1)
- Restore a Specific Table (see Section 3.3.2)
- Restore the SC Database Directory (see Section 3.3.3)

- Restore Archived Data (see Section 3.3.4)

3.3.1 Restore the Complete SC Database

If you used the `rmsbackup` command or `rmstbladm -d` command to back up the database (see Section 3.1.1 on page 3–2 or Section 3.1.2 on page 3–3 respectively), you can restore the database as follows:

1. If the `SC20rms` CAA application has been enabled and is running, stop the `SC20rms` application.

   You can determine the current status of the `SC20rms` application by running the `caa_stat` command on the first domain in the system (that is, `atlasD0`, where `atlas` is an example system name) as follows:
   ```
   # caa_stat SC20rms
   ```
   To stop the `SC20rms` application on all domains, use the `caa_stop` command as follows:
   ```
   # scrun -d all 'caa_stop SC20rms'
   ```
2. Stop the RMS daemons on every node, by running the following command once on any node:
   ```
   # scrun -n all '/sbin/init.d/rms stop'
   ```
   If your system has a management server, log into the management server, and stop its RMS daemons as follows:
   ```
   atlasms# /sbin/init.d/rms stop
   ```
3. Stop the HP AlphaServer SC SRA daemon on each domain by running the following command once on any node:
   ```
   # scrun -d all 'caa_stop SC15srad'
   ```
   If your system has a management server, stop its SRA daemon as follows:
   ```
   atlasms# /sbin/init.d/sra stop
   ```
4. Stop the SC Monitor daemon on every node by running the following command once on any node:
   ```
   # scrun -n all '/sbin/init.d/scmon stop'
   ```
   If your system has a management server, stop its SC Monitor daemon as follows:
   ```
   atlasms# /sbin/init.d/scmon stop
   ```
5. Now that you have stopped all of the daemons that use the database, you can restore the database. Use one of the files in the `/var/rms/backup` directory, as shown in the following example:
   ```
   # rmstbladm -r /var/rms/backup/atlas_2003-02-15-01:10.sql.gz
   ```
   It is not necessary to `gunzip` the file first.

To restart all of the daemons, perform the following steps:

1. If the `SC20rms` CAA application has been enabled, start the `SC20rms` application using the `caa_start` command as follows:
   ```
   # scrun -d all 'caa_start SC20rms'
   ```

2. Start the RMS daemons on the remaining nodes, by running the following command once on any node:
   ```
   # scrun -d all 'CluCmd /sbin/init.d/rms start'
   ```
   If your system has a management server, log into the management server and start its RMS daemons as follows:
   ```
   atlasms# /sbin/init.d/rms start
   ```
3. Start the HP AlphaServer SC SRA daemon on each domain by running the following command once on any node:
   ```
   # scrun -d all 'caa_start SC15srad'
   ```
   If your system has a management server, start its SRA daemon as follows:
   ```
   atlasms# /sbin/init.d/sra start
   ```
4. Start the SC Monitor daemon on every node by running the following command once on any node:
   ```
   # scrun -d all 'CluCmd /sbin/init.d/scmon start'
   ```
   If your system has a management server, start its SC Monitor daemon as follows:
   ```
   atlasms# /sbin/init.d/scmon start
   ```

3.3.2 Restore a Specific Table

To restore a specific table from a backup file that was created by the `rmsbackup` or `rmstbladm` command, use the `rmstbladm` command as follows:

```
# rmstbladm -r filename -t table_name
```

where *filename* is the name of the backup file (which may be a compressed file), and *table_name* is the name of a table that has been backed up.

For example, to restore the `jobs` table from the backup file created in Section 3.1.1 on page 3–2, run the following command:

```
# rmstbladm -r /var/rms/backup/atlas_2003-02-15-01:10.sql.gz -t jobs
```

This command will restore the backup entries into the existing `jobs` table — it does not replace the existing table. The restore operation automatically deletes duplicate entries.

3.3.3 Restore the SC Database Directory

If you backed up the SC database directory (see Section 3.1.3 on page 3–4), you can restore the database as follows:

1. Stop the HP AlphaServer SC SRA daemon on each domain by running the following command once on any node:
   ```
   # scrun -d all 'caa_stop SC15srad'
   ```
   If your system has a management server, stop its SRA daemon as follows:
   ```
   atlasms# /sbin/init.d/sra stop
   ```
2. Stop the SC Monitor daemon on every node by running the following command once on any node:

```
# scrun -n all '/sbin/init.d/scmon stop'
```

If your system has a management server, stop its SC Monitor daemon as follows:

```
atlasms# /sbin/init.d/scmon stop
```

3. Stop RMS and mSQL as described in Section 5.9.2 on page 5–69.
4. Restore the database files to the `/var/rms/msqldb-3/rms_system` directory.
5. Start RMS and mSQL as described in Section 5.9.3 on page 5–71.
6. Start the HP AlphaServer SC SRA daemon on each domain by running the following command once on any node:

```
# scrun -d all 'caa_start SC15srad'
```

If your system has a management server, start its SRA daemon as follows:

```
atlasms# /sbin/init.d/sra start
```

7. Start the SC Monitor daemon on every node by running the following command once on any node:

```
# scrun -n all '/sbin/init.d/scmon start'
```

If your system has a management server, start its SC Monitor daemon as follows:

```
atlasms# /sbin/init.d/scmon start
```

3.3.4 Restore Archived Data

To restore a specific table from an archive file that was created by the `rmsarchive` command, use the `rmstbladm` command as follows:

```
# rmstbladm -r filename -t table_name
```

where *filename* is the name of the archive file (which may be a compressed file), and *table_name* is the name of a table that has been archived.

For example, to restore the `jobs` table from the archive file created in Section 3.2.4 on page 3–7, run the following command:

```
# rmstbladm -r /var/rms/archive/atlas_2003-02-15-01:10.sql.gz -t jobs
```

This command will restore the archived entries into the existing `jobs` table — it does not replace the existing table. The restore operation automatically deletes duplicate entries.

3.4 Deleting the SC Database

To delete the SC database, perform the following steps:

1. Ensure that there are no allocated resources. One way to do this is to stop each partition using the `kill` option, as shown in the following example:

```
# rcontrol stop partition=big option kill
```

2. If the `SC20rms` CAA application has been enabled and is running, stop the `SC20rms` application.

You can determine the current status of the SC20rms application by running the caa_stat command on the first domain in the system (that is, atlasD0, where atlas is an example system name) as follows:

```
# caa_stat SC20rms
```

To stop the SC20rms application, use the caa_stop command as follows:

```
# caa_stop SC20rms
```

3. Stop the RMS daemons on every node, by running the following command once on any node:

```
# scrun -n all '/sbin/init.d/rms stop'
```

If your system has a management server, log into the management server, and stop its RMS daemons as follows:

```
atlasms# /sbin/init.d/rms stop
```

4. Stop the HP AlphaServer SC SRA daemon on each domain by running the following command once on any node:

```
# scrun -d all 'caa_stop SC15srad'
```

If your system has a management server, stop its SRA daemon as follows:

```
atlasms# /sbin/init.d/sra stop
```

5. Stop the SC Monitor daemon on every node by running the following command once on any node:

```
# scrun -n all '/sbin/init.d/scmon stop'
```

If your system has a management server, stop its SC Monitor daemon as follows:

```
atlasms# /sbin/init.d/scmon stop
```

6. Delete the current database as follows:

```
# msqladmin drop rms_system
```

where *system* is the name of the HP AlphaServer SC system.

To create a new SC database, run the sra setup command as described in Chapter 5 or Chapter 6 of the *HP AlphaServer SC Installation Guide*.

Note:

If you drop the SC database, you must re-create it by restoring a backup copy that was made after all of the nodes in the system were installed.

If any nodes were installed after the backup was created, you must re-install these nodes after restoring the database from the backup.

If you do not restore a database, but instead re-create it by using the sra setup command, you must completely redo the whole installation process.

3.5 Monitoring /var

The SC database is stored in the `/var` file system. You should regularly monitor the size of `/var` to check that it is not becoming full.

The `msql3d` daemon periodically checks the remaining storage in the `/var` file system, as follows:

- If the amount of storage available in the `/var` file system falls below 50MB, `msqld` prints a warning message in `/var/rms/adm/log/msqld.log` and also to the `syslog` subsystem.
- If the amount of storage available in the `/var` file system falls below 10MB, `msqld` prints an out of space message to `/var/rms/adm/log/msqld.log` and to the `syslog` subsystem, and exits.

3.6 Cookie Security Mechanism

The SC database is protected by security mechanisms that ensure that only the `root` user on nodes within the HP AlphaServer SC system can modify the database. One of these mechanisms uses a distributed cookie scheme. Normally, this cookie mechanism is enabled and managed by the `gxmgmtd` and `gxclusterd` daemons. However, you may need to disable the cookie mechanism in certain situations, such as the following:

- When upgrading a system that pre-dates the cookie security mechanism.
  Normally, the `sra` command automatically disables the cookie mechanism during an upgrade. For more information about upgrades, see Chapter 4 of the *HP AlphaServer SC Installation Guide*.
- When the cookie distribution mechanism is broken.
  For more information on how to identify when the cookie mechanism is broken, see Chapter 11 of the *HP AlphaServer SC Installation Guide*.

You can disable the cookie mechanism by using the `sra cookie` command. You must log into the `rmshost` system to use the `sra cookie` command. The `sra cookie` command provides the following functionality:

- To disable the cookie mechanism, run the following command:
  ```
  # sra cookie -enable no
  ```
- To enable the cookie mechanism, run the following command:
  ```
  # sra cookie -enable yes
  ```
- To check whether cookies are enabled or not, run the following command:
  ```
  # sra cookie
  MSQL cookies are currently enabled
  ```

While the cookie mechanism is disabled, the database is still protected from modification by non-`root` users. However, it is less secure.

3.7 SC Database Integrity Check Utility

This section explains a utility that can be used to check the internal integrity of the SC database. The information in this section is organized as follows:

- Overview of the Database Integrity Check Utility (see Section 3.7.1 on page 3–13)
- Possible Causes of Inconsistency in the SC Database (see Section 3.7.2 on page 3–14)
- Consistency Checks Performed by the SC Database Integrity Check (see Section 3.7.3 on page 3–15)
- Running the SC Database Integrity Check (see Section 3.7.4 on page 3–23)
- Output from the SC Database Integrity Check (see Section 3.7.5 on page 3–25)
- Database Integrity Check Events and Automatic E-mail Alerts (see Section 3.7.6 on page 3–27)

3.7.1 Overview of the Database Integrity Check Utility

The `sra dbcheck` command is a database integrity check utility that identifies internal inconsistencies in the SC database. You can use this utility to help you to identify the cause of problems that might be occurring in your HP AlphaServer SC system, and to help you to prevent potential problems arising from database inconsistency. See Section 3.7.2 on page 3–14 for information on possible causes of inconsistency in the SC database.

The database integrity check runs a series of tests on the database, and, depending on the options selected when the job is run, may attempt to fix inconsistencies that are found. For more information on the tests run by the utility, see Section 3.7.3 on page 3–15.

A number of options can be specified for the `sra dbcheck` command. For more information, see Section 3.7.4 on page 3–23.

The format of the output from the database integrity check depends on the options specified when the `sra dbcheck` command is invoked. For more information on understanding the output from the `sra dbcheck` command, see Section 3.7.5 on page 3–25.

You can specify that events are to be created when the database integrity check is run, and you can configure the system to send e-mail to operators to indicate the outcome of the integrity check. For more information, see Section 3.7.6 on page 3–27.

Note:

The `sra dbcheck` command in HP AlphaServer SC Version 2.6 does not perform any checks to ensure that the system state is accurately reflected in the SC database (external checks).

3.7.2 Possible Causes of Inconsistency in the SC Database

The SC database is updated by a number of commands. If any of these commands is interrupted, or terminates improperly, the SC database can be left in an inconsistent state.

The following are some of the commands that update the SC database:

- The `sra setup` command creates the SC database during the installation process. For more information on the installation process, see the *HP AlphaServer SC Installation Guide*.
- Upgrading the HP AlphaServer SC system can involve upgrading the operating system, or the software running on the system, or both. As part of an upgrade, the database upgrade script `sc_upgrade_db` is run, and this makes changes to the SC database. For more information on installing an upgrade, see the *HP AlphaServer SC Installation Guide*.
- The sysman `sc_cabinet` command is used to manage information on which cabinets hold nodes, terminal servers, and other pieces of hardware. This information is displayed on the Physical tab of the SC Viewer, and is used to determine a particular component's physical location. For more information, see Section 10.6 on page 10–25.
- The `sra edit` command can be used to modify the SC database in a controlled way, and can also be used to update configuration files that are dependent on the SC database. This command can be used to add or delete nodes and their attributes, edit system attributes, and to add or delete terminal servers, boot images, and domains. For more information on the `sra edit` command, see Section 16.2 on page 16–22.
- The `scmonmgr` command can be used to manage SC Monitor, and to add or delete records of objects such as management appliances, terminal servers, HSV or HSG cells, and Extreme or ProCurve switches in the SC database. The corresponding `scmond` daemon updates the SC database with the current state of the objects. For more information on the `scmonmgr` command, see Section 27.4.1 on page 27–16.
- The `scfsmgr` command can be used to create, manage, and delete SCFS file systems. For more information on the `scfsmgr` command, see Section 7.4 on page 7–8.

- The `pfsmgr` command is used to create, check, and delete PFS file systems on an FS domain, and to manage the mounting and unmounting of the PFS file systems globally across the HP AlphaServer SC system. For more information on the `pfsmgr` command, see Section 8.4 on page 8–13.
- The `rcontrol` command is used to create and manage partitions and configurations and other system information to enable resource management for running parallel compute jobs in a multi-user environment. For more information on the `rcontrol` command, see Section 5.3.2 on page 5–8.
- The Interconnect tab on the SC Viewer can be used to update switch information in the SC database. For more information on SC Viewer, see Chapter 10.

3.7.3 Consistency Checks Performed by the SC Database Integrity Check

Each check performed by the SC database integrity check utility has a unique label, and these labels are shown in the output from the `sra dbcheck` command. Each label is made up of four parts as follows:

- The first part of the label signifies that the check belongs to the database integrity check utility.
- The second part of the label identifies the table that is being checked for consistency.
- The third part of the label identifies the table attribute that is being checked for consistency.
- The fourth and final part of the label indicates the type of problem that the check will identify.

An example would be:
`dbcheck.sc_nodes.count.invalid`

where:

- `dbcheck` indicates that the check belongs to the database integrity check utility
- `sc_nodes` indicates that the table being checked for consistency is the `sc_nodes` table
- `count` indicates that the table attribute being checked for consistency is the `count` attribute (in the `sc_nodes` table)
- `invalid` indicates that the validity of the field is being checked

In this case, the count field will be invalid if the value in the count field of the `sc_nodes` table is not the same as the value in the count field of the `sc_system` table.

In all checks, it is the table that is being checked against that is considered to be correct; if the database integrity check attempts to fix an inconsistency, it attempts to make the table that is being checked consistent with the table that it is being checked against.

For example, in the `dbcheck.attributes.name.invalid` check, the system name specified in the `attributes` table is checked against the system name in the `sc_system` table. If there is an inconsistency, the system name in the `sc_system` table is assumed to be correct. If an attempt is made to fix the inconsistency, the system name in the `attributes` table is changed to be consistent with the system name in the `sc_system` table.

The consistency checks that are performed by the database integrity check fall into a number of categories. These are described in the following sections:

- Consistency of System Attributes (see Section 3.7.3.1 on page 3–16)
- Consistency of the SRA Configuration (see Section 3.7.3.2 on page 3–17)
- Consistency of Terminal Servers/Ports/CMF Information (see Section 3.7.3.3 on page 3–18)
- Consistency of Disk and Storage Information (see Section 3.7.3.4 on page 3–19)
- Consistency of Information on AdvFS/PFS/SCFS File Systems (see Section 3.7.3.5 on page 3–20)
- Consistency of Information in the Monitoring Subsystem (see Section 3.7.3.6 on page 3–22)
- Consistency of HSV Information (see Section 3.7.3.7 on page 3–23)

3.7.3.1 Consistency of System Attributes

The first set of tests performed by the database integrity check examine the consistency of system attributes. The tables checked include the following:

- `attributes`
- `sc_system`
- `sc_nodes`

The checks performed include the following:

- The SC database is not currently being created. If the database is currently being created, no further checks are performed and the utility exits.
- Attribute information is consistent across the three tables and other tables based on any of these attributes. This includes checking that the system name information and the total number of nodes in the system match the corresponding entry in the `sc_system` table.

Table 3–3 shows the system attribute checks, listed in alphabetical order.

Table 3–3 System Attributes Consistency Checks

Check Label	Verifies that:
dbcheck.attributes.name.invalid	The system name specified in the attributes table is the same as the system name in the sc_system table.
dbcheck.sc_nodes.count.invalid	The total number of nodes listed in the sc_nodes table is equal to the number of nodes specified in the sc_system table.
dbcheck.sc_system.db_setup.incomplete	The value of db_setup in the sc_system table is set to 1, indicating that the database is not currently being created. If the database is currently being created, no further checks are performed, and the database integrity check exits immediately. If the database is not in the process of being created, the database integrity check continues with the remaining system attribute checks.

3.7.3.2 Consistency of the SRA Configuration

The second set of tests performed by the database integrity check examines the consistency of node information across the database. The tables checked include the following:

- sc_nodes
- nodes
- sc_domain

The checks performed include the following:

- Node information across the database is consistent as derived from the sc_nodes table.
- Domain information across the database is consistent against the sc_domain table.

Table 3–4 shows the SRA configuration consistency checks, listed in alphabetical order.

Table 3–4 SRA Configuration Consistency Checks

Check Label	Verifies that:
dbcheck.nodes.cluster.mismatch	For each host, the serving domain (if populated) is consistent in the nodes table as derived from the sc_nodes table.
dbcheck.nodes.domain.invalid	The domain names in the nodes table are valid as derived from the sc_domain table.

Table 3–4 SRA Configuration Consistency Checks

Check Label	Verifies that:
dbcheck.nodes.memberid.invalid	For each host, the member ID (if populated) specified in the nodes table is consistent with the member ID in the sc_nodes table.
dbcheck.nodes.name.invalid	Each hostname specified in the nodes table is valid as derived from the sc_nodes table.
dbcheck.sc_nodes.domain.invalid	The domain names in the sc_nodes table are valid as derived from the sc_domain table.

3.7.3.3 Consistency of Terminal Servers/Ports/CMF Information

The third set of tests performed by the database integrity check examines the consistency of terminal server information, port information, and CMF information across the database. The tables checked include the following:

- sc_cmf
- sc_tservers
- tservers

The checks performed include the following:

- The CMF host information is the same across the sc_system and sc_cmf tables.
- The terminal servers listed in the sc_cmf table all exist in the sc_tservers table.
- All nodes with a console have a valid entry in the sc_cmf table and have a terminal server/port entry.
- The terminal server/port information in the sc_cmf table is valid as defined in the sc_tservers table and corresponding tserver_types entries.
- The name and type information for each of the terminal servers in the system is the same across the tservers and sc_tservers tables.

Table 3–5 shows the terminal servers/ports/CMF consistency checks, listed in alphabetical order.

Table 3–5 Terminal Servers/Ports/CMF Information Consistency Checks

Check Label	Verifies that:
dbcheck.sc_cmf.cmf_host.invalid	The CMF host listed for the hosts in the sc_cmf table is listed in the cmf_host attribute of the sc_system table.

Table 3–5 Terminal Servers/Ports/CMF Information Consistency Checks

Check Label	Verifies that:
dbcheck.sc_cmf.host.incomplete	Each node that has a console (ES40, ES45, or DS20L) as derived from the nodes table has an entry in the sc_cmf table.
dbcheck.sc_cmf.ts_host.empty	Each host listed in the sc_cmf table has a value in the terminal server name (ts_host) field.
dbcheck.sc_cmf.ts_port.empty	Each host listed in the sc_cmf table has a value in the terminal server port (ts_port) field.
dbcheck.sc_cmf.ts_port.invalid	The port numbers for all terminal server listed in the sc_cmf table are within the range of base port and total ports allowed for the corresponding terminal type as derived from the sc_tservers and tserver_types tables.
dbcheck.sc_cmf.tserver.invalid	The sc_cmf table does not have any references to non-existent terminal servers as derived from the sc_tservers table.
dbcheck.tservers.count.invalid	The number of terminal servers listed in the tservers table is the same as that listed in the sc_tservers table.
dbcheck.tservers.name.mismatch	All terminal servers listed in the tservers table have been derived from the sc_tservers table.
dbcheck.tservers.type.mismatch	For each terminal server listed in the tservers table, the terminal server type is the same as the terminal server type listed in the sc_tservers table.

3.7.3.4 Consistency of Disk and Storage Information

The fourth set of tests performed by the database integrity check examines the consistency of disk and storage information across the database. The tables checked include the following:

- sc_disk
- sc_disk_server
- sc_lsm_vols

The checks performed include the following:

- All disks in the sc_disk table have a valid serving domain associated with them.
- Disk/domain information in the sc_disk_server and sc_lsm_vols tables is valid and is consistent with information in the sc_disk table.
- All LSM volumes have a valid serving domain.

Table 3–6 shows the disk and storage consistency checks, listed in alphabetical order.

Table 3–6 Disk and Storage Consistency Checks

Check Label	Verifies that:
dbcheck.sc_disk.domain.invalid	All disks listed in the sc_disk table have a valid serving domain.
dbcheck.sc_disk_server.disk.invalid	All disks listed in the sc_disk_server table are valid as derived from the sc_disk table.
dbcheck.sc_disk_server.domain.invalid	All domains listed in the sc_disk_server table are valid as derived from the sc_domains table.
dbcheck.sc_disk_server.domain.mismatch	For each disk listed in the sc_disk_server table, the serving domain name matches the serving domain for the same disk in the sc_disk table.
dbcheck.sc_lsm_vols.disk.invalid	All disks listed in the sc_lsm_vols table are valid as derived from the sc_disk table.
dbcheck.sc_lsm_vols.domain.invalid	All domains listed in the sc_lsm_vols table are valid as derived from the sc_domains table.
dbcheck.sc_lsm_vols.domain.mismatch	For each disk listed in the sc_lsm_vols table, the serving domain name matches the serving domain for the same disk in the sc_disk table.

3.7.3.5 Consistency of Information on AdvFS/PFS/SCFS File Systems

The fifth set of tests performed by the database integrity check examines the consistency of information on AdvFS, PFS, and SCFS file systems.

The tables checked include the following:

- sc_advfs_filesets
- sc_advfs_vols
- sc_pfs
- sc_pfs_components
- sc_pfs_filesystems
- sc_pfs_mount
- sc_scfs

The checks performed include the following:

- All AdvFS file systems belong to a valid domain.
- All AdvFS volumes are built on a valid disk and the corresponding partition is shown to be occupied in the `sc_disk` table.
- All SCFS file systems are built on valid AdvFS file sets.
- Mount information is provided for all existing SCFS filesystems for each domain in the SC system.
- All domains mentioned in the SCFS mount information are valid domains.
- PFS mount information only contains valid PFS sets and valid domain information.

Table 3–7 shows the AdvFS/PFS/SCFS file system consistency checks listed in alphabetical order.

Table 3–7 AdvFS/PFS/SCFS File Systems Consistency Checks

Check Label	Verifies that:
`dbcheck.sc_advfs_vols.disk.invalid`	All disks (across each domain) listed in the `sc_advfs_vols` table are valid disks as derived from the `sc_disk` table.
`dbcheck.sc_advfs_vols.domain.invalid`	All AdvFS volumes listed in the `sc_advfs_vols` table have a valid serving domain as derived from the `sc_domain` table.
`dbcheck.sc_advfs_vols.partition.mismatch`	For each AdvFS volume specified in the `sc_advfs_vols` table, the disk partition being used has the corresponding partition entry in the `sc_disk` table (for a given domain) marked as used (set to -1).
`dbcheck.sc_pfs.pfs_component.incomplete`	Each PFS set listed in the `sc_pfs` table has at least one component in the `sc_pfs_components` table.
`dbcheck.sc_pfs_components.component_fs_path.incomplete`	For each `component_fs_path` in the `sc_pfs_components` table, there is an `scfs_fs_name`, based on a `component_fs_path` in the `sc_pfs_components` table, in the `sc_pfs_filesystems` table.
`dbcheck.sc_pfs_components.pfs_set.invalid`	All PFS sets listed in the `sc_pfs_components` table are valid PFS sets as derived from the `sc_pfs` table.
`dbcheck.sc_pfs_filesystems.pfs_set.invalid`	All PFS sets in the `sc_pfs_filesystems` table are valid `pfs_sets` as derived from the `sc_pfs` table.
`dbcheck.sc_pfs_filesystems.sc_pfs_mount.mismatch`	The `sc_pfs_mount` table has mount information for every PFS set listed in the `sc_pfs_filesystems` table.

Table 3–7 AdvFS/PFS/SCFS File Systems Consistency Checks

Check Label	Verifies that:
dbcheck.sc_pfs_mount.domain.invalid	All domains listed in the sc_pfs_mount table are valid domains as listed in the sc_domain table.
dbcheck.sc_pfs_mount.pfs_set.invalid	All PFS sets listed in the sc_pfs_mount table are valid PFS sets as derived from the sc_pfs table.
dbcheck.sc_pfs_mount.row.missing	The sc_pfs_mount table lists mount information for every PFS set listed in the sc_pfs table, for each domain in the sc_domain table.
dbcheck.sc_scfs.advfs_domain_fset_name.invalid	For each SCFS file system listed in the sc_scfs table, each triple of advfs_domain, fset_name, and clu_domain is valid as derived from the sc_advfs_filesets table.
dbcheck.sc_scfs.domain.invalid	All domains listed in the sc_scfs table are valid domains as derived from the sc_domains table.
dbcheck.sc_scfs.mount_point.empty	All SCFS file systems listed in the sc_scfs table have a non-null value in the mount_point field.

3.7.3.6 Consistency of Information in the Monitoring Subsystem

The sixth set of tests performed by the database integrity check examines the consistency of information in the Monitoring subsystem.

The tables checked include the following:

- sc_classes
- sc_monitor_dev

The checks performed include the following:

- All entries in sc_classes refer to a valid class that is known to the scmond daemon.
- All monitored objects are valid objects known to the SC system and belong to a valid class.

Table 3–8 shows the monitoring subsystem consistency checks, listed in alphabetical order.

Table 3–8 Monitoring Subsystem Consistency Checks

Check Label	Verifies that:
dbcheck.sc_classes.class.invalid	All classes listed in the sc_classes table are valid (the status_table field in the sc_classes table has a valid value for an existing table in the SC database).
dbcheck.sc_monitor_dev.class.invalid	Each object listed in the sc_monitor_dev table has a valid entry for the class field as derived from the sc_classes table.

3.7.3.7 Consistency of HSV Information

The final test performed by the database integrity check examines the consistency of HSV appliance information. The sc_hsv_cell table is checked against the sc_appliance table to ensure that all appliance names are valid.

Table 3–9 shows details of the HSV consistency check.

Table 3–9 Miscellaneous Consistency Check

Check Label	Verifies that:
dbcheck.sc_hsv_cell.appliance.invalid	The appliance field in the sc_hsv_cell table contains valid appliance names as derived from the sc_appliance table.

3.7.4 Running the SC Database Integrity Check

To run the integrity check interactively, run the sra dbcheck command with the options you want. The options are described in Section 3.7.4.1 on page 3–24.

If you wish, you can use the cron command to set up a schedule for running the sra dbcheck command on a regular basis, at a specific time. See the cron(8) reference page for more details.

3.7.4.1 sra dbcheck Syntax

To use the `sra dbcheck` command, you must be the `root` user. The `sra dbcheck` command can be run from any node.

The syntax of the `sra dbcheck` command is as follows:

```
sra dbcheck [-tcl yes|no] [-postevent yes|no] [-backup yes|no] [-fix
none|safe|all] [-log filename] [-d database] [-edir expect_directory] [-tdir
test_directory] [-ddir description_directory]
```

Table 3–10 describes the `sra dbcheck` options in alphabetical order.

Table 3–10 sra dbcheck Options

Option	Description
`-backup`	Specifies whether a backup is taken of the entire database. By default, the `-backup` option is set to `yes`, but the database is only backed up if the `-fix` option is set to `safe`, or `all`, and inconsistencies are found during the check.This is because when the `-fix` option is set to `safe`, or `all`, the database integrity check attempts to fix inconsistencies and makes changes to the database. In this case, it is important to have a backup of the database so that you can roll it back after the integrity check, if necessary. You can override this behavior by setting the `-backup` option to `no`.
`-d`	Specifies the database that is to be checked. If the `-d` *`database`* option is not specified, the system database is checked by default.
`-ddir`	Specifies the directory where the description files are held. Description files are used to translate the output of the database integrity check into English descriptions and suggested solutions. The default directory where the description files are held is `/usr/opt/srasysman/dbcheck/desc`. If you wish, you can specify a different directory.
`-edir`	Specifies the directory for sourcing in the expect scripts that `sra dbcheck` needs from another directory. The default is `/usr/opt/srasysman/dbcheck/exp`. If you wish, you can specify a different directory.
`-fix`	Specifies whether the `sra dbcheck` command attempts to fix some, all, or none of the inconsistencies it finds. The valid values are `safe`, `all`, or `none`. If `safe` is specified, the database integrity check attempts to fix inconsistencies it determines are safe to fix, and the output shows whether the inconsistency has been fixed, remains unfixed (the attempt to fix the inconsistency was unsuccessful), or should be fixed manually. If `all` is specified, the database integrity check attempts to fix all identified inconsistencies, and the output shows whether the inconsistency has been fixed, remains unfixed (the attempt to fix the inconsistency was unsuccessful), or should be fixed manually. If `none` is specified, the database integrity check will not attempt to fix any inconsistencies, but will report each inconsistency, and whether it is `safe`, (can be safely fixed automatically) or `manual` (should be fixed manually). If the `-fix` option is not specified, the default is `none`.

Table 3–10 sra dbcheck Options

Option	Description
-log	Specifies the name of the log file and where the log file is placed. By default, a log file with the name *currentdate*.log is created and is placed in the /var/sra/dbcheck directory. You can specify a different directory and file name.
-postevent	Specifies whether an event is generated when the sra dbcheck command runs. Valid values are yes or no. If the -postevent option is not specified, the default is no.
-tcl	Specifies whether the output is a formatted output or a summary. Valid values are yes or no. If yes is specified, the sra dbcheck command produces a formatted output that can be parsed by another script. If no is specified, the sra dbcheck command produces a summary output of four or five lines for each inconsistency found. If the -tcl option is not specified, the default is no. For more information on the output format, see Section 3.7.5 on page 3–25.
-tdir	Specifies the directory where the test scripts are held. See/usr/opt/srasysman/dbcheck/check/README for test naming conventions. The default directory where the test scripts are held is /usr/opt/srasysman/dbcheck/check. If you wish, you can specify a different directory.

3.7.5 Output from the SC Database Integrity Check

When the sra dbcheck command runs, the output is sent to stdout. The format of the output depends on the -tcl option chosen when the sra dbcheck command is run. If the -tcl option is set to no, or is not specified, a summary output is produced. If the -tcl option is set to yes, a formatted output is produced.

3.7.5.1 Example 1: Summary Output

In this example, no options are specified, and the command takes all the defaults. This is equivalent to invoking the command with the following options:

```
sra dbcheck -tcl no -postevent no -backup yes -fix none -d system -tdir /usr/opt/
sra/sysman/dbcheck/check -ddir /usr/opt/sra/sysman/dbcheck/desc -edir /usr/opt/
sra/sysman/dbcheck/exp
```

Example:

```
# sra dbcheck
check: dbcheck.nodes.memberid.invalid
fix: Can be fixed automatically
description: This check verifies that the  member id specified in the NODES
table matches the member id in the SC_NODES table for each host.
solution: Update the NODES table member field so that the member ids match
those in the SC_NODES memberid field.
```

```
other: atlas5

check: dbcheck.sc_cmf.host.incomplete
fix: Needs to be fixed manually
description: This check verifies that all nodes with a console (nodes in NODES
with type field of ES40, ES45, DS20L, or DS20) have an entry in SC_CMF.
solution: Add an entry to SC_CMF for each record in NODES not already in SC_CMF
and whose type is ES40, ES45, DS20L, or DS20
other: atlasms

check: dbcheck.sc_advfs_vols.disk.invalid
fix: Needs to be fixed manually
description: This check verifies that all disks in SC_ADVFS_VOLS are valid disks
as derived from SC_DISK.
solution: Correct or remove records in SC_ADVFS_VOLS where name is not found in
SC_DISK.
other: {atlasD0 dsk9a}
```

The output shows either four or five lines for each inconsistency found, as follows:

- The first line specifies the check label.
- The second line specifies whether the inconsistency can be fixed automatically or needs to be fixed manually.
- The third line provides a description of the check.
- The fourth line suggests the solution for fixing the inconsistency.
- The optional fifth line gives additional information; usually this is a list of the bad data, or the indexes into the table for the row containing the bad data.

If the command was run with the `-fix` option set to `safe` or `all`, the output would be similar, except that the second line of output for each discrepancy would show if the inconsistency had been fixed, or needs to be fixed manually. In addition, the database would have been backed up, and a message would be shown at the start of the output as follows:

```
# sra dbcheck -fix all

Discrepancy Found.. Initiating backup..
Backup Completed..
```

3.7.5.2 Example 2: Formatted Output

This example shows the same database integrity check, but with the `-tcl` option set to yes, and the `-fix` option not specified (set by default to `none`).

When `-tcl yes` is specified, the `sra dbcheck` command produces a formatted output that can be parsed by another script. This output contains one line for each test that found an inconsistency. The first column shows the check label. The second column indicates whether the inconsistency can be fixed safely (`safe`), automatically but unsafely (`auto`), or manually (`manual`), and a list of the associated objects.

Example:

```
# sra dbcheck -tcl yes

dbcheck.nodes.memberid.invalid  safe {atlas5}
dbcheck.sc_cmf.host.incomplete manual {atlasms}
dbcheck.sc_advfs_vols.disk.invalid manual {{atlasD0 dsk9a}}
```

If the command was run with the `-fix` option set to `safe` or `all`, the output would be similar, but all the discrepancies that the utility was able to fix would be shown as `fixed` in the second column of the output. If only some of the discrepancies could be fixed, there will be two lines of output for the test, one stating `fixed` and listing the objects or rows that were fixed, and the other stating `unfixed` and listing the appropriate objects or rows.

3.7.6 Database Integrity Check Events and Automatic E-mail Alerts

You can use the `-postevent yes` option to specify that an event is created in the SC database when the sra `dbcheck command` is run. You can view these events using either the SC Viewer or the `scevent` command (for more information, see Section 9.3 on page 9–9). You can also create alerts that will result in e-mails being automatically sent to operators to indicate the outcome of the database integrity check.

For more information on creating alerts, see Section 9.5 on page 9–13.

There are four possible outcomes from a run of the `sra dbcheck` command. Table 3–11 describes these outcomes.

Table 3–11 sra dbcheck Events

Severity	Type	Message and Description
`Failed`	`dbcheck.failed`	`Critical inconsistencies found. Further testing aborted. See /var/sra/dbcheck/`*`currentdate`*`.log.` This event is posted when the `sra dbcheck` command exits with an exit code of 2, indicating that a critical inconsistency has been found. If a check finds an inconsistency that makes it impossible or unnecessary to continue checking the database, a signal is sent to the utility. In this case, a `dbcheck.failed` event is posted. This event is posted once for each run of the database integrity check utility where a critical inconsistency is found.
`Error`	`dbcheck.error`	*`Type of error`*`. See /var/sra/dbcheck/`*`currentdate`*`.log.` This event is posted when the `sra dbcheck` command (or one of the checks that it runs) exits with an exit code of -1, indicating that an error occurred when running a check. The text of the message describes the type of error found; possible errors include a command line syntax error, TCL/expect errors in a test script, test scripts not being found, and no main procedure being found in a test script. This event is posted only once for each run of the database integrity check utility if one or more errors are found. The event posted will reflect the most severe error encountered during the run.
`Warning`	`dbcheck.warning`	`Dbcheck finished - inconsistencies were found. See /var/sra/dbcheck/`*`currentdate`*`.log.` This event is posted when the `sra dbcheck` command exits with an exit code of 1, indicating that only non-critical inconsistencies have been found, in at least one check. An inconsistency is considered to be non-critical if it does not preclude running the remainder of the tests. This event is posted only once for each run of the database integrity check utility.
`Info`	`dbcheck.attention`	`Dbcheck finished - there were no inconsistencies.` This event is posted when the `sra dbcheck` command exits with an exit code of 0 (zero), meaning that no inconsistencies were found by any of the checks. This event is posted only once for each run of the database integrity check utility.

4

Managing the Load Sharing Facility (LSF)

The information in this chapter is organized as follows:

- Introduction to LSF (see Section 4.1 on page 4–2)
- Setting Up Virtual Hosts (see Section 4.2 on page 4–5)
- Starting the LSF Daemons (see Section 4.3 on page 4–5)
- Shutting Down the LSF Daemons (see Section 4.4 on page 4–7)
- Checking the LSF Configuration (see Section 4.5 on page 4–9)
- Setting Dedicated LSF Partitions (see Section 4.6 on page 4–9)
- Customizing Job Control Actions (Optional) (see Section 4.7 on page 4–9)
- Configuration Notes (see Section 4.8 on page 4–10)
- LSF Topology Scheduler (see Section 4.9 on page 4–11)
- Operating LSF for HP AlphaServer SC (see Section 4.10 on page 4–16)
- The lsf.conf File (see Section 4.11 on page 4–26)
- Known Problems or Limitations (see Section 4.12 on page 4–29)

Note:

This chapter provides a brief introduction to Platform Computing Corporation's LSF® software ("LSF"). For more information, see the LSF reference pages or the following LSF documents:

- *Running Jobs with HP AlphaServer SC Platform LSF®*
- *HP AlphaServer SC Platform LSF® Quick Reference*
- *HP AlphaServer SC Platform LSF® Reference*
- *Administering HP AlphaServer SC Platform LSF®*

4.1 Introduction to LSF

LSF for HP AlphaServer SC combines the strengths of LSF and HP AlphaServer SC software to provide a comprehensive Distributed Resource Management (DRM) solution. LSF acts primarily as the workload scheduler, providing policy and topology-based scheduling. RMS acts as a parallel subsystem, and other HP AlphaServer SC software provides enhanced fault tolerance.

The remainder of the information in this section is organized as follows:

- Installing LSF on an HP AlphaServer SC System (see Section 4.1.1)
- LSF Directory Structure on an HP AlphaServer SC System (see Section 4.1.2)
- Compatibility with Earlier Releases of LSF (see Section 4.1.3)
- Using NFS to Share LSF Configuration Information (see Section 4.1.4)
- Using LSF Commands (see Section 4.1.5)

Note:

The term "LSF cluster" is used to describe a group of machines controlled by LSF. An LSF cluster can consist of one or more HP AlphaServer SC systems, or part of an HP AlphaServer SC system; it can also include additional machines of arbitrary architectures.

4.1.1 Installing LSF on an HP AlphaServer SC System

LSF is not automatically installed during the HP AlphaServer SC installation process. You must install LSF separately, as described in the *HP AlphaServer SC Installation Guide*.

4.1.2 LSF Directory Structure on an HP AlphaServer SC System

You specify the location of LSF files in an HP AlphaServer SC system by setting the LSF_TOP variable in the `install.config` file during the installation procedure. The default value of this variable is `/usr/share/lsf`; the information in this document is based on the assumption that LSF_TOP is set to this default value. For more information about LSF installation, see the *HP AlphaServer SC Installation Guide*

The LSF directory structure is as follows:

```
/usr/share/lsf/6.0
/usr/share/lsf/6.0/install
/usr/share/lsf/6.0/alpha5-rms
/usr/share/lsf/6.0/alpha5-rms/bin
/usr/share/lsf/6.0/alpha5-rms/etc
/usr/share/lsf/6.0/alpha5-rms/lib
/usr/share/lsf/6.0/include
```

```
/usr/share/lsf/6.0/man
/usr/share/lsf/6.0/misc
/usr/share/lsf/work
/usr/share/lsf/conf
```

For compatibility with older versions of LSF, the installation procedure also creates the following symbolic links:

```
/var/lsf/conf@ -> /usr/share/lsf/conf/
/var/lsf/work@ -> /usr/share/lsf/work/
/usr/opt/lsf/bin@ -> /usr/share/lsf/6.0/alpha5-rms/bin/
/usr/opt/lsf/etc@ -> /usr/share/lsf/6.0/alpha5-rms/etc/
/usr/opt/lsf/lib@ -> /usr/share/lsf/6.0/alpha5-rms/lib/
```

4.1.3 Compatibility with Earlier Releases of LSF

HP AlphaServer SC Version 2.6 hosts and HP AlphaServer SC Version 2.5 hosts cannot coexist in the same LSF cluster. You must upgrade all HP AlphaServer SC hosts in the LSF cluster to HP AlphaServer SC Version 2.6.

There are a number of differences in the behavior of HP AlphaServer SC Version 2.6 as compared with earlier versions of LSF, as follows:

- The topology scheduler options (the `bsub -ext` option or queue-level DEFAULT_EXTSCHED and MANDATORY_EXTSCHED) no longer determine the host type for the job. Regular LSF resource requirements (`bsub -R` option or queue-level `RES_REQ` parameter) or host type selection (`bsub -m` option) dictate where the job will run. For more information, see Section 4.10.2 on page 4–17.
- The default RMS allocation type is RMS_SNODE.
- The LSB_RLA_POLICY and LSB_RMS_NODESIZE node-level allocation parameters in the `lsf.conf` file are no longer needed. Use the `RMS[ptile=cpus_per_node]` option to specify node-level allocation.
- For topology scheduler options defined in the obsolete syntax, the following conditions apply:
  - Only the RMS topology options are honoured.
  - Options from the queue and from the `bsub` command are parsed separately before they are merged. For example, if the following options are specified in the old syntax when the `bsub` command is run:

    ```
    bsub -extsched "ptile=1" ...
    ```

    and the default topology scheduler for the queue is RMS_SNODE, the `ptile=1` option is ignored because it is not an RMS topology option; the net result is that the RMS_SNODE allocation type is used.

In HP AlphaServer SC Version 2.5, the same options would have resulted in the following parameters being used:

```
RMS_SNODE;ptile=1
```

For more information on the new syntax for the queue-level topology scheduler, see Section 4.9 on page 4–11. For more information on the new syntax for the `bsub` command, see the *HP AlphaServer SC User Guide*.

4.1.4 Using NFS to Share LSF Configuration Information

The LSF_TOP directory (`/usr/share/lsf`) file system must be exported from one of the following:

- The management server (if using a management server)
- Node 0 (if not using a management server)

The LSF_TOP directory must be NFS-mounted to each domain using the standard UNIX mount procedure on the head node of each domain. Add an entry for the LSF_TOP directory to the `/etc/member_fstab` file on the head node of each domain. For more information on how to use NFS mount, see Section 22.6 on page 22–11.

Note:

Do not use AutoFS to mount LSF installation.

4.1.5 Using LSF Commands

Before executing any LSF command, you must update your environment using either the `/usr/share/lsf/conf/cshrc.lsf` file or the `/usr/share/lsf/conf/profile.lsf` file, depending on the shell used. This creates the necessary environment for running LSF.

Instead of sourcing these files each time you log in, you can incorporate them into your `.cshrc` or `.profile` files as follows:

- If using C shell (`csh` or `tcsh`), add the following lines to the `~/.cshrc` file:

```
if ( -f /usr/share/lsf/conf/cshrc.lsf ) then
  source /usr/share/lsf/conf/cshrc.lsf
endif
```

- If not using C shell, add the following lines to the `$HOME/.profile` file:

```
if [ -f /usr/share/lsf/conf/profile.lsf ] ; then
  . /usr/share/lsf/conf/profile.lsf
fi
```

You must place this `if` statement after the statements that set the default PATH and MANPATH environment variables in the `.cshrc` or `.profile` file.

4.2 Setting Up Virtual Hosts

LSF can treat each domain in an HP AlphaServer SC system as a single **virtual server host**. A single set of LSF daemons controls all members of a domain. This does not preclude running LSF daemons on every node. However, each domain should be configured either as a virtual host or as up to 32 real hosts — not as a combination of both. HP recommends that you set up each domain as a virtual host. Each virtual host has the same name as the domain.

Note:

When LSF is used in the Virtual Hosts mode (one set of LSF daemons controlling one domain), all nodes in one domain must be in the same RMS partition. It is possible to have multiple RMS partitions on an HP AlphaServer SC system controlled by LSF, but only if partition boundaries are aligned with the LSF virtual host boundaries.

When a partition layout is changed in an HP AlphaServer SC system (for example, switching between a daytime partition and a nighttime partition, LSF must be restarted on the affected hosts.

If a host is added to a partition dynamically, LSF must be restarted on the host. If a node is removed from a partition dynamically, LSF does not need to be restarted on the other hosts in the partition.

If an RMS partition is under LSF control, all domains and nodes belonging to that partition should be included in the LSF cluster.

4.3 Starting the LSF Daemons

The information in this section is organized as follows:

- Starting the LSF Daemons on a Management Server or Single Host (see Section 4.3.1)
- Starting the LSF Daemons on A Number of Real Hosts (see Section 4.3.2)
- Starting the LSF Daemons on a Virtual Host (see Section 4.3.3)
- Starting the LSF Daemons on a Number of Virtual Hosts (see Section 4.3.4)
- Checking that the LSF Daemons Are Running (see Section 4.3.5)

4.3.1 Starting the LSF Daemons on a Management Server or Single Host

To start the LSF daemons on a management server or single host, perform the following steps:

1. Log on to the management server or single host as the `root` user.
2. Check that the LSF environment is set (for information on how to set the LSF environment, see Section 4.1.5 on page 4–4).
3. Start the LSF daemons by running the following command:
   ```
   # lsf_daemons start
   ```
   (The `lsf_daemons` command is located in the `/usr/share/lsf/6.0/alpha5-rms/etc` directory.)

4.3.2 Starting the LSF Daemons on A Number of Real Hosts

To start the LSF daemons on a number of real hosts, perform the following steps:

1. Log on to any host as the `root` user.
2. Start the LSF daemons by running the following command:
   ```
   # scrun -n LSF_hosts 'lsf_daemons start'
   ```
   where *LSF_hosts* specifies the list of LSF real hosts. For more information about the syntax of the `scrun` command, see Section 12.1 on page 12–2.

   (The `lsf_daemons` command is located in the `/usr/share/lsf/6.0/alpha5-rms/etc` directory.)

Note:

To use LSF commands *via* the `scrun` command, the `.cshrc` or `.profile` file should be modified, as described in Section 4.1.5 on page 4–4, to set up the LSF environment at shell initialization stage. Otherwise, the full path to the command must be provided.

4.3.3 Starting the LSF Daemons on a Virtual Host

To start the LSF daemons on a virtual host, perform the following steps:

1. Log on to the first node of the virtual host as the `root` user.
2. Run the following command:
   ```
   # caa_start lsf
   ```
   The `caa_start` command is located in the `/usr/sbin` directory.

4.3.4 Starting the LSF Daemons on a Number of Virtual Hosts

To start the LSF daemons on a number of virtual hosts, perform the following steps:

1. Log on to any host as the `root` user.
2. Start the LSF daemons on all of the specified hosts by running the following command:
   `# scrun -n LSF_hosts 'caa_start lsf'`

   where *`LSF_hosts`* specifies the first node of each virtual host. For more information about the syntax of the `scrun` command, see Section 12.1 on page 12–2.

   The `caa_start` command is located in the `/usr/sbin` directory.

4.3.5 Checking that the LSF Daemons Are Running

Use the `scps` command to check that the LSF daemons are running (for more information on the `scps` command, see the `scps(8)` reference page). Search for processes that are similar to the following:

```
root 17426 1 0 Oct 15 ? 2:04 /usr/share/lsf/6.0/alpha5-rms/etc/lim
root 17436 1 0 Oct 15 ? 0:00 /usr/share/lsf/6.0/alpha5-rms/etc/sbatchd
root 17429 1 0 Oct 15 ? 0:00 /usr/share/lsf/6.0/alpha5-rms/etc/res
```

4.4 Shutting Down the LSF Daemons

The information in this section is organized as follows:

- Shutting Down the LSF Daemons on a Management Server or Single Host (see Section 4.4.1)
- Shutting Down the LSF Daemons on a Number of Real Hosts (see Section 4.4.2)
- Shutting Down the LSF Daemons on a Virtual Host (see Section 4.4.3)
- Shutting Down the LSF Daemons on A Number of Virtual Hosts (see Section 4.4.4)

4.4.1 Shutting Down the LSF Daemons on a Management Server or Single Host

To shut down the LSF daemons on a management server or single host, perform the following steps:

1. Log onto the management server or single host as the `root` user.
2. Check that the LSF environment is set (for information on how to set the LSF environment, see Section 4.1.5 on page 4–4).
3. Stop the LSF daemons by running the following command:
   `# lsf_daemons stop`

   (The `lsf_daemons` command is located in the `/usr/share/lsf/6.0/alpha5-rms/etc` directory.)

4.4.2 Shutting Down the LSF Daemons on a Number of Real Hosts

To shut down the LSF daemons on a number of real hosts, perform the following steps:

1. Log on to any host as the `root` user.
2. Stop the LSF daemons by running the following command:
   `# scrun -n LSF_hosts 'lsf_daemons stop'`

   where *`LSF_hosts`* specifies the list of LSF real hosts. For more information about the syntax of the `scrun` command, see Section 12.1 on page 12–2.

   (The `lsf_daemons` command is located in the `/usr/share/lsf/6.0/alpha5-rms/etc` directory.)

Note:

To use LSF commands *via* the `scrun` command, the `.cshrc` or `.profile` file should be modified, as described in Section 4.1.5 on page 4–4, to set up the LSF environment at shell initialization stage. Otherwise, the full path to the command must be provided.

4.4.3 Shutting Down the LSF Daemons on a Virtual Host

To shut down the LSF daemons on a virtual host, perform the following steps:

1. Log on to the first node of the virtual host as the `root` user.
2. Run the following command:
   `# caa_stop lsf`

   The `caa_stop` command is located in the `/usr/sbin` directory.

4.4.4 Shutting Down the LSF Daemons on A Number of Virtual Hosts

To shut down the LSF daemons on a number of virtual hosts, perform the following steps:

1. Log on to any host as the `root` user.
2. Shut down the LSF daemons on all of the specified hosts by running the following command:
   `# scrun -n LSF_hosts 'caa_stop lsf'`

   where *`LSF_hosts`* specifies the first node of each virtual host. For more information about the syntax of the `scrun` command, see Section 12.1 on page 12–2.

   The `caa_stop` command is located in the `/usr/sbin` directory.

4.5 Checking the LSF Configuration

To check the LSF configuration, use the following commands:

- `lsload`

  Displays load information for hosts.

- `lshosts`

  Displays static resource information about hosts.

- `bhosts`

  Displays static and dynamic resource information about hosts.

Note:

If you remove a host permanently from an RMS partition, you should also remove the `rms` resource keyword entry for that host in the `lsf.cluster.`*`cluster_name`* file.

For more information about these and other LSF commands, see the LSF documentation or the LSF reference pages.

4.6 Setting Dedicated LSF Partitions

Use the RMS `rcontrol` command to prevent `prun` jobs from running directly on partitions dedicated to LSF, as shown in the following example:

```
# rcontrol set partition=parallel configuration=day type=restricted
```

See Chapter 5 for more information about the `rcontrol` command.

4.7 Customizing Job Control Actions (Optional)

By default, LSF carries out job control actions by sending the appropriate signal to suspend, terminate, or resume a job. If your jobs need special job control actions, use the RMS `rcontrol` command in the queue configuration to change the default job controls.

Use the JOB_CONTROLS parameter in the `lsb.queues` file to configure suspend, terminate, or resume job controls for the queue, as follows:

```
JOB_CONTROLS =    SUSPEND[command] |
                  RESUME[command] |
                  TERMINATE[command]
```

where *`command`* is an `rcontrol` command in the following form:

```
rcontrol [suspend | kill | resume] batchid=$LSB_JOBID
```

See the *HP AlphaServer SC Platform LSF® Reference* manual for more information about the JOB_CONTROLS parameter in the `lsb.queues` file.

See Chapter 5 for more information about the `rcontrol` command.

The following example shows how to create a TERMINATE job control action:

```
Begin Queue
QUEUE_NAME=queue1
.
.
.
JOB_CONTROLS = TERMINATE[rcontrol kill batchid=$LSB_JOBID]
.
.
.
End Queue
```

4.8 Configuration Notes

The information in this section is organized as follows:

- Maximum Job Slot Limit (see Section 4.8.1 on page 4–10)
- Per-Processor Job Slot Limit (see Section 4.8.2 on page 4–10)
- Management Servers (see Section 4.8.3 on page 4–10)
- Default Queue (see Section 4.8.4 on page 4–11)
- Maximum Number of sbatchd Connections (see Section 4.8.5 on page 4–11)

4.8.1 Maximum Job Slot Limit

By default, the maximum job slot limit is set to the number of CPUs that the Load Information Manager (LIM) reports, specified by MXJ=! in the `Host` section of the `lsb.hosts` file. Do not change this default.

4.8.2 Per-Processor Job Slot Limit

The `rms` queue in the `lsb.queues` file has the per-processor job slot limit set to 1, specified by PJOB_LIMIT=1. Do not change this value.

4.8.3 Management Servers

LIM is locked on management servers running LSF; therefore, LSF will not schedule jobs to management servers.

4.8.4 Default Queue

The default LSF queue is named `normal`. This is the standard default LSF queue, and has no topology scheduler settings. However, in HP AlphaServer SC Version 2.6, the default scheduling policy is RMS_SNODE, therefore the `normal` queue performs RMS_SNODE scheduling by default.

The `rms` queue is also configured in the `lsb.queue` file during installation; this queue is customized to run jobs on HP AlphaServer SC hosts only, and by default performs RMS_SNODE scheduling.

To show the queue configuration details, run the following command:

```
# bqueues -l
```

HP recommends that you use the `rms` queue as the default queue. To configure the `rms` queue as the default queue, set DEFAULT_QUEUE=rms in the `lsb.params` file.

For information on how to create and modify queue parameters, and how to create additional queues, see the *Administering HP AlphaServer SC Platform LSF®* manual and the *HP AlphaServer SC Platform LSF® Reference* manual.

4.8.5 Maximum Number of sbatchd Connections

If LSF operates on a large system (for example, a system with more than 32 nodes), you may need to configure the parameter MAX_SBD_CONNS in the `lsb.params` file. This parameter controls the maximum number of files `mbatchd` can have open and connected to `sbatchd`. The default value of MAX_SBD_CONNS is 32.

In a very busy system with many jobs being dispatched, running, and finishing at the same time, you may see that it takes a very long time for `mbatchd` to update the status change of a job, and to dispatch new jobs. If your system shows this behavior, set MAX_SBD_CONNS=300 or MAX_SBD_CONNS=*number_of_nodes**2, whichever is less. Setting MAX_SBD_CONNS too high may slow down the speed of `mbatchd` dispatching new jobs.

4.9 LSF Topology Scheduler

The topology scheduler for RMS jobs determines which hosts will execute the RMS job. As system administrator, you specify a queue-level topology scheduler. If you specify that the scheduler is mandatory, users cannot overwrite the queue-level specification when submitting jobs. If you specify a non-mandatory scheduler, users can overwrite the queue-level specification with job-level options, by using the `-ext` option of the `bsub` command, as described in the *HP AlphaServer SC User Guide*.

4.9.1 Syntax

To specify a queue-level topology scheduler, set the appropriate parameter in the `lsb.queues` file, with the following specification:
parameter=RMS[[*allocation_type*][;*topology*][;*flags*]]

where *parameter* is DEFAULT_EXTSCHED for a non-mandatory topology scheduler (see Section 4.9.2), or MANDATORY_EXTSCHED for a mandatory topology scheduler (see Section 4.9.3).

Note that the first and last square brackets — [ ] — around the group of options must be entered as part of the parameter specification; the brackets around the *allocation_type*, *topology*, and *flags* items indicate that these items are optional. These brackets should not be entered.

For example:
DEFAULT_EXTSCHED=RMS[RMS_SNODE;rails=2]

Note:

The old syntax *parameter=extsched_options* (for example, DEFAULT_EXTSCHED=RMS_SNODE) used in earlier versions of LSF is obsolete.

You must use the new syntax, as follows:

parameter=RMS[[*allocation_type*][;*topology*][;*flags*]]

for example, DEFAULT_EXTSCHED=RMS[RMS_SNODE].

The default allocation type for the DEFAULT_EXTSCHED parameter on the `rms` queue is RMS_SNODE. There is no default value for the MANDATORY_EXTSCHED parameter. If no value is defined for either the MANDATORY_EXTSCHED parameter or the DEFAULT_EXTSCHED parameter on a queue, the queue performs RMS_SNODE scheduling by default.

4.9.1.1 Allocation Type

The *allocation_type* option specifies the type of node allocation, and can have one of the following values:

- RMS_SNODE

  RMS_SNODE specifies *sorted node allocation*. Nodes do not need to be contiguous: gaps are allowed between the leftmost and rightmost nodes of the allocation map. This is the default allocation type for RMS jobs.

  LSF sorts nodes according to RMS topology (numbering of nodes and domains), which takes precedence over LSF sorting order.

The allocation is more compact than in RMS_SLOAD; allocation starts from the leftmost node allowed by the LSF host list, and continues rightward until the allocation specification is satisfied.

Use RMS_SNODE on larger LSF clusters where the only factor that matters for job placement decisions is the number of available job slots.

- RMS_SLOAD

  RMS_SLOAD specifies *sorted load allocation*. Nodes do not need to be contiguous: gaps are allowed between the leftmost and rightmost nodes of the allocation map.

  LSF sorts nodes based on host preference and load information, which takes precedence over RMS topology (numbering of nodes and domains).

  The allocation starts from the first host specified in the list of LSF hosts, and continues until the allocation specification is satisfied.

  Use RMS_SLOAD on smaller LSF clusters, where the job placement decision should be influenced by host load, or where you want to keep a specific host preference.

- RMS_MCONT

  RMS_MCONT specifies *mandatory contiguous node allocation*. The allocation must be contiguous: between the leftmost and rightmost nodes of the allocation map, each node must either have at least one CPU that belongs to this allocation, or this node must be configured out completely.

  The sorting order for RMS_MCONT is RMS topological order; LSF preferences are not taken into account.

  The allocation is more compact than in RMS_SLOAD, but requires contiguous nodes. Allocation starts from the leftmost node that allows contiguous allocation. Nodes that are out of service are not considered as gaps.

In HP AlphaServer SC Version 2.6, the LSF preemption, backfill, and job slot reservation features are supported for each allocation policy.

4.9.1.2 Topology

The *topology* option specifies the topology of the allocation, and can have the following values:

- nodes=*nodes* | ptile=*cpus_per_node*

  nodes specifies the number of nodes that the allocation requires, or the number of CPUs per node.

The ptile topology option is different from the LSF ptile keyword used in the span section of the resource requirement string (bsub -R "span[ptile=n]"). If the ptile topology option is specified in the -ext option of the bsub command, the value of bsub -n must be an exact multiple of the ptile value.

The following example is valid, because 12 (-n) is exactly divisible by 4 (ptile):

```
$ bsub -n 12 -ext "RMS[ptile=4]"
```

- base=*base_node_name*

  If base is specified with the RMS_SNODE or RMS_MCONT allocation, the starting node for the allocation is the base node name, instead of the leftmost node allowed by the LSF host list.

  If base is specified with the RMS_SLOAD allocation, RMS_SNODE allocation is used.

4.9.1.3 Flags

The *flags* option specifies other allocation options. The only supported flags are rails=*number* and railmask=*bitmask*. See Section 5.12 on page 5–76 for more information about these options.

4.9.1.4 LSF Configuration Parameters

The topology options nodes and ptile, and the rails flag, are limited by the values of the corresponding parameters in the lsf.conf file, as follows:

- nodes is limited by LSB_RMS_MAXNUMNODES
- ptile is limited by LSB_RMS_MAXPTILE
- rails is limited by LSB_RMS_MAXNUMRAILS

4.9.2 DEFAULT_EXTSCHED

The DEFAULT_EXTSCHED parameter in the lsb.queues file specifies default external scheduling options for the queue.

The -ext options from the bsub command are merged with the DEFAULT_EXTSCHED options, and the -ext options override any conflicting queue-level options set by DEFAULT_EXTSCHED, as shown in Example 4–1.

The DEFAULT_EXTSCHED parameter can be used in combination with the MANDATORY_EXTSCHED parameter in the same queue, as shown in Example 4–2.

If any topology options (nodes, ptile, or base) or flags (rails or railmask) are set by the DEFAULT_EXTSCHED parameter, and you want to override the default setting so that you specify a blank value for these options, use the appropriate keyword with no value, in the -ext option of the bsub command, as shown in Example 4–3.

The default topology scheduler — that is, the default value of the DEFAULT_EXTSCHED parameter — is RMS_SNODE. If no value is defined for either the MANDATORY_EXTSCHED parameter or the DEFAULT_EXTSCHED parameter, the queue will perform RMS_SNODE scheduling.

Example 4–1

A job is submitted with the following options:

```
-ext "RMS[base=atlas0;rails=1;ptile=2]"
```

The `lsf.queues` file contains the following entry:

```
DEFAULT_EXTSCHED=RMS[RMS_SNODE;rails=2]
```

Result: LSF uses the following topology scheduler options for scheduling:

```
RMS_SNODE;rails=1;base=atlas0;ptile=2
```

Example 4–2

A job is submitted with the following options:

```
-ext "RMS[base=atlas0;ptile=2]"
```

The `lsf.queues` file contains the following entries:

```
DEFAULT_EXTSCHED=RMS[rails=2]
MANDATORY_EXTSCHED=RMS[RMS_SNODE;ptile=4]
```

Result: LSF uses the following topology scheduler options for scheduling:

```
RMS_SNODE;rails=2;base=atlas0;ptile=4
```

Example 4–3

A job is submitted with the following options:

```
-ext "RMS[RMS_SNODE;nodes=]"
```

The `lsb.queues` file contains the following entry:

```
DEFAULT_EXTSCHED=RMS[nodes=2]
```

Result: LSF uses the following topology scheduler options for scheduling:

```
RMS_SNODE
```

4.9.3 MANDATORY_EXTSCHED

The MANDATORY_EXTSCHED parameter in the `lsb.queues` file specifies mandatory external scheduling options for the queue.

The `-ext` options from the `bsub` command are merged with the MANDATORY_EXTSCHED options, and the MANDATORY_EXTSCHED options override any conflicting job-level options set by `-ext`, as shown in Example 4–4.

The MANDATORY_EXTSCHED parameter can be used in combination with the DEFAULT_EXTSCHED parameter in the same queue, as shown in Example 4–5.

To prevent users from setting the topology options (`nodes`, `ptile`, or `base`) or flags (`rails` or `railmask`) by using the `-ext` option of the `bsub` command, you can use the MANDATORY_EXTSCHED option to set the appropriate keyword with no value, as shown in Example 4–6.

There is no default value for the MANDATORY_EXTSCHED parameter.

Example 4–4

A job is submitted with the following options:
```
-ext "RMS[base=atlas0;rails=1;ptile=2]"
```

The `lsf.queues` file contains the following entry:
```
MANDATORY_EXTSCHED=RMS[RMS_SNODE;rails=2]
```

Result: LSF uses the following topology scheduler options for scheduling:
```
RMS_SNODE;rails=2;base=atlas0;ptile=2
```

Example 4–5

A job is submitted with the following options:
```
-ext "RMS[base=atlas0;ptile=2]"
```

The `lsf.queues` file contains the following entries:
```
DEFAULT_EXTSCHED=RMS[rails=2]
MANDATORY_EXTSCHED=RMS[RMS_SNODE;ptile=4]
```

Result: LSF uses the following topology scheduler options for scheduling:
```
RMS_SNODE;rails=2;base=atlas0;ptile=4
```

Example 4–6

A job is submitted with the following options:
```
-ext "RMS[RMS_SNODE;nodes=4]"
```

The `lsf.queues` file contains the following entry:
```
MANDATORY_EXTSCHED=RMS[nodes=]
```

Result: LSF overrides both `-ext` settings.

4.10 Operating LSF for HP AlphaServer SC

The information in this section is organized as follows:

- Remote LSF Adapter for RMS (RLA) (see Section 4.10.1 on page 4–17)
- Running Jobs on Any Host Type (see Section 4.10.2 on page 4–17)
- Using the brun Command to Run a Job on an RMS Host (see Section 4.10.3 on page 4–18)
- LSF Licensing (see Section 4.10.4 on page 4–18)

- Maintaining a Pool of Nodes in Reserve (see Section 4.10.5 on page 4–19)
- Running Pre-Execution Programs (see Section 4.10.6 on page 4–19)
- LSF Job Termination Reasons (see Section 4.10.7 on page 4–20)
- User Information for Interactive Batch Jobs (see Section 4.10.8 on page 4–24)

4.10.1 Remote LSF Adapter for RMS (RLA)

The remote LSF adapter (RLA) runs on each LSF host within an RMS partition. RLA is started by the `sbatchd` daemon, and handles all communication between the LSF topology scheduler and RMS.

4.10.2 Running Jobs on Any Host Type

You can specify several types of topology scheduler options at job submission, and LSF will schedule jobs appropriately. Use the LSF resource requirements specification (the `-R` option of the `bsub` command, or the RES_REQ parameter in the queue definition in the `lsb.queues` file) to identify the host types required for your job. Jobs that do not specify RMS-related options (`"type==any"`) can be scheduled on HP AlphaServer SC RMS hosts, and jobs with RMS-related topology scheduler options can run on non-RMS hosts.

For example, SGI IRIX hosts and HP AlphaServer SC hosts running RMS can exist in the same LSF cluster. You can use the `-R` option to define host-type resource requirements for either IRIX hosts or HP AlphaServer SC hosts, as shown in the following examples. Your job will run either on an IRIX host or an HP AlphaServer SC host, but not on both.

Example 4–7

```
$ bsub -R "IRIX || ALPHASC" -ext "RMS[ptile=2]" jobname
```

In the above example, the job runs on an IRIX host or an HP AlphaServer SC host if one is available, but not on both. If the job runs on an HP AlphaServer SC host, the RMS `ptile` topology scheduler option is enforced.

Example 4–8

```
$ bsub -R "type==any" -ext "RMS[ptile=2];SGI_CPUSET[CPU_LIST=1]" jobname
```

In this example, if the job runs on an HP AlphaServer SC host, the RMS `ptile` topology scheduler option is enforced. If the job runs on an IRIX host with cpusets, the SGI `cpuset` topology scheduler option is enforced. If the job runs on any other host type, the topology scheduler options are ignored.

4.10.3 Using the brun Command to Run a Job on an RMS Host

A user who has LSF administrator permissions can force a job to run on an RMS host by using the `brun -c` command. If the RMS allocation cannot be satisfied for any reason, the job is dispatched, but is requeued and returned to a pending state. The user can run the `brun -c` command again to start the job.

4.10.4 LSF Licensing

LSF licenses are managed by the HP AlphaServer SC licensing mechanism, which determines whether LSF is correctly licensed for the appropriate number of CPUs on the LSF host. This new licensing method does not interfere with existing file-based FLEXlm licensing (using `license.dat` or OEM license file).

The license is not transferable to any other hosts in the LSF cluster. The following LSF features are enabled:

- `lsf_base`
- `lsf_manager`
- `lsf_parallel`
- `lsf_sched_fairshare`
- `lsf_sched_parallel`
- `lsf_sched_preemption`
- `lsf_sched_resource_reservation`
- `lsf_sched_advance_reservation`
- `platform_hpc`

To get the status of the license, use the following command:

```
$ lmf list for ASC
```

If the LSF master host becomes unlicensed, the whole LSF cluster is unavailable. LSF commands will not run on an unlicensed host, and the following message is displayed in the `lim` daemon log file:

```
ls_gethostinfo(): Host does not have a software license
```

If a server host becomes licensed or unlicensed at run time, LSF automatically licenses or unlicenses the host.

4.10.4.1 How to Get Additional LSF Licenses

To get licenses for additional LSF features, contact Platform at `license@platform.com`. For example, to enable LSF floating client licenses in your LSF cluster, you will need a license key for the `lsf_float_client` feature.

For more information about LSF features and licensing, see the *Administering HP AlphaServer SC Platform LSF*® manual.

4.10.5 Maintaining a Pool of Nodes in Reserve

Use the RLA_RMS_ONLINENODES parameter in the `lsf.conf` file to define how many nodes to maintain online at any given time out of the total node count in any partition that LSF can schedule jobs to. The remaining nodes are held in reserve and are available to add to the pool of online nodes to maintain the specified maximum when an online node becomes unavailable. You can define the online nodes as a number or a percentage. You cannot define which nodes are to be online and which nodes are to be spare.

Note:

LSF assumes that it controls all available nodes in an RMS partition, and calculates the spare node pool based on the static configuration. LSF cannot determine whether a given node is online or in the spare node pool.

For example:

On a partition with 1024 nodes and with RLA_RMS_ONLINENODES=:98%, LSF can schedule jobs to 1024 X 98% = 1000 online nodes; the remaining 24 nodes are held in reserve. If any of the 1000 online nodes goes out of service, RLA adds as many of the spare nodes to the pool of online nodes as are needed to maintain 1000 online nodes. When the node comes back in service, RLA withdraws the spare node from the online pool. For better performance, spare nodes are drawn from the end of a partition.

4.10.6 Running Pre-Execution Programs

Although LSF daemons only run on the management node of a virtual host, batch jobs can run on any nodes within a virtual host that satisfy the scheduling and topological requirements. By default, pre-execution commands can only run on the node where the `sbatchd` daemon runs.

To enable pre-execution on other nodes in an RMS allocation, the `prun` job run in the pre-execution program picks up the following environment variables set by LSF:

- RMS_RESOURCEID=*partition.resource_id*
- RMS_MACHINE

These variables give LSF the information it needs to run the job on the nodes required by the pre-execution program.

Note:

Post-execution commands will only run on the daemon node, and will only run after the RMS allocation has been destroyed.

You can control where to run a pre-execution program so that it runs on one of the following:

- The daemon node of a virtual host
- The first execution node in an allocation
- All nodes in an allocation, one instance per node

To run a pre-execution program on the daemon node of a virtual host, run the pre-execution program normally. The pre-execution program does not need to make use of the RMS_MACHINE and RMS_RESOURCEID environment variables.

To run a pre-execution program on the first execution node in an allocation, use the `prun -n 1` command as follows:

```
$ prun -n 1 my_pre_exec
```

where *my_pre_exec* is the name of the pre-execution program.

To run a pre-execution program on all nodes in an allocation, one instance per node, use the `prun -m cyclic` option to start the pre-execution program on all processors of each node. The pre-execution program that `prun` invokes must compare the RMS_RANK and RMS_NODEID prun environment variables so that when RMS_RANK=RMS_NODEID, the pre-execution process on the first processor of each node should keep running, while allowing the remaining pre-execution processes on a node to exit normally.

See the RMS `prun` command in the *HP AlphaServer SC RMS Reference Manual* for more information about the `-m` and `-n` options.

4.10.7 LSF Job Termination Reasons

When a job finishes, LSF reports the last job termination action it took against the job and logs it to the `lsb.acct` file. Use the `bacct -l` command to view the job exit information logged to the `lsb.acct` file, as follows:

```
% bacct -l 3088
Accounting information about jobs that are:
- submitted by all users.
- accounted on all projects.
- completed normally or exited
- executed on all hosts.
- submitted to all queues.
------------------------------------------------------------------------------
Job <3088>, User <user1>, Project <default>, Status <EXIT>, Queue <rms>,
```

```
Command
                    <prun hostname>
Wed Aug 6 17:09:44: Submitted from host <atlasD0>, CWD <$HOME>;
Wed Aug 6 17:09:58: Dispatched to 7 Hosts/Processors <3*atlasD0> <4*atlasD1>;
Wed Aug 6 17:09:58: rms_rid=parallel.3759;rms_alloc=3*atlas1 2*atlas[2-3];
Wed Aug 6 17:09:58: Completed ,exit>; TERM_RUNLIMIT: job was killed after LSF
run limit expired
Accounting information about this job:
CPU_T     WAIT     TURNAROUND   STATUS     HOG_FACTOR    MEM    SWAP
0.00      0        3224         exit       0.0000        0K     0K

SUMMARY: ( time unit: second )
Total number of done jobs:        0      Total number of exited jobs: 1
Total CPU time consumed:        0.0      Average CPU time consumed: 0.0
Maximum CPU time of a job:      0.0      Minimum CPU time of a job: 0.0
Total wait time in queues:      0.0
Average wait time in queue:     0.0
Maximum wait time in queue:     0.0      Minimum wait time in queue: 0.0
Average turnaround time:         43 (seconds/job)
Maximum turnaround time:       3224      Minimum turnaround time: 3224
Average hog factor of a job:   0.00 ( cpu time / turnaround time )
Maximum hog factor of a job:   0.00      Minimum hog factor of a job: 0.00
```

4.10.7.1 Termination Reasons Displayed by the bacct Command

When LSF detects that a job is terminated, the `bacct -l` command displays a termination reason. Table 4–1 lists the LSF job termination reasons.

Table 4–1 LSF Job Termination Reasons

Keyword	Reason
TERM_ADMIN	Job killed by root or LSF administrator
TERM_CHKPNT:	Job killed after checkpointing
TERM_CPULIMIT	Job killed after reaching LSF CPU usage limit
TERM_DEADLINE	Job killed after deadline expires
TERM_EXTERNAL_SIGNAL	Job killed by a signal external to LSF
TERM_FORCE_ADMIN	Job killed by root or LSF administrator without time for cleanup
TERM_FORCE_OWNER	Job killed by owner without time for cleanup
TERM_LOAD	Job killed after load exceeds threshold
TERM_MEMLIMIT	Job killed after reaching LSF memory usage limit
TERM_OWNER	Job killed by owner
TERM_PREEMPT	Job killed after preemption

Table 4–1 LSF Job Termination Reasons

Keyword	Reason
TERM_PROCESSLIMIT	Job killed after reaching LSF process limit
TERM_REQUEUE_ADMIN	Job killed and requeued by root or LSF administrator
TERM_REQUEUE_OWNER	Job killed and requeued by owner
TERM_RMS	Job terminated abnormally in RMS
TERM_RUNLIMIT	Job killed after reaching LSF run time limit
TERM_SWAP	Job killed after reaching LSF swap usage limit
TERM_THREADLIMIT	Job killed after reaching LSF thread limit
TERM_WINDOW	Job killed after queue run window closed
TERM_ZOMBIE	Job exited while LSF is not available

For more information about the mapping between the integer value logged to the `lsb.acct` file and termination reason keyword, see the `lsbatch.h` file.

Table 4–2 shows examples of the output from the `bhist` and `bacct -l` commands for termination reasons.

Table 4–2 Example Output from bacct and bhist Commands

Termination Cause	Termination Reason from bacct Command	Example Output from bhist Command
bkill -s KILL bkill job_id	Completed <exit>; TERM_OWNER or TERM_ADMIN	Thu Mar 13 17:32:05: Signal <KILL> requested by user or administrator <user2>; Thu Mar 13 17:32:06: Exited by signal 2. The CPU time used is 0.1 seconds;
bkill -r	Completed <exit>; TERM_FORCE_ADMIN or TERM_FORCE_OWNER when SBD is not reachable. Otherwise, TERM_USER or TERM_ADMIN	Thu Mar 13 17:32:05: Signal <KILL> requested by user or administrator <user2>; Thu Mar 13 17:32:06: Exited by signal 2. The CPU time used is 0.1 seconds;

Table 4–2 Example Output from bacct and bhist Commands

Termination Cause	Termination Reason from bacct Command	Example Output from bhist Command
TERMINATE_WHEN	Completed <exit>; TERM_LOAD/ TERM_WINDOWS/ TERM_PREEMPT	Thu Mar 13 17:33:16: Signal <KILL> requested by user or administrator <user2>; Thu Mar 13 17:33:18: Exited by signal 2. The CPU time used is 0.1 seconds;
Memory limit reached	Completed <exit>; TERM_MEMLIMIT	Thu Mar 13 19:31:13: Exited by signal 2. The CPU time used is 0.1 seconds;
Run limit reached	Completed <exit>; TERM_RUNLIMIT	Thu Mar 13 20:18:32: Exited by signal 2. The CPU time used is 0.1 seconds.
CPU limit	Completed <exit>; TERM_CPULIMIT	Thu Mar 13 18:47:13: Exited by signal 24. The CPU time used is 62.0 seconds;
Swap limit	Completed <exit>; TERM_SWAPLIMIT	Thu Mar 13 18:47:13: Exited by signal 24. The CPU time used is 62.0 seconds;
Regular job exits when host crashes	Rusage 0, Completed <exit>; TERM_ZOMBIE	Thu Jun 12 15:49:02: Unknown; unable to reach the execution host; Thu Jun 12 16:10:32: Running; Thu Jun 12 16:10:38: Exited with exit code 143. The CPU time used is 0.0 seconds;
brequeue –r	For each requeue, Completed <exit>; TERM_REQUEUE_ADMIN or TERM_REQUEUE_OWNER	Thu Mar 13 17:46:39: Signal <REQUEUE_PEND> requested by user or administrator <user2>; Thu Mar 13 17:46:56: Exited by signal 2. The CPU time used is 0.1 seconds;
bchkpnt -k	On the first run: Completed <exit>; TERM_CHKPNT	Wed Apr 16 16:00:48: Checkpoint succeeded (actpid 931249); Wed Apr 16 16:01:03: Exited with exit code 137. The CPU time used is 0.0 seconds;
Kill –9 <RES> and job	Completed <exit>; TERM_EXTERNAL_SIGNAL	Thu Mar 13 17:30:43: Exited by signal 15. The CPU time used is 0.1 seconds;

Table 4–2 Example Output from bacct and bhist Commands

Termination Cause	Termination Reason from bacct Command	Example Output from bhist Command
Job terminates abnormally in RMS	Completed <exit>; TERM_RMS	Thu Mar 13 17:30:43: Exited with 125;
Others	Completed <exit>;	Thu Mar 13 17:30:43: Exited with 3; The CPU time used is 0.1 seconds;

The following behavior applies when LSF is determining and interpreting termination reasons:

- If a queue-level JOB_CONTROL is configured, LSF cannot determine the result of the action. The termination reason only reflects what the termination reason could be in LSF.
- LSF does not always catch external signals sent directly to a job.
- If LSF cannot deallocate a resource when a job is done, the RMS resource status may remain allocated, and the TERM_RMS termination reason is not set.

4.10.8 User Information for Interactive Batch Jobs

The LSF cluster automatically tracks user and account information for interactive batch jobs that are submitted with the `bsub -Ip` command or the `bsub -Is` command. User and account information is registered as entries in the `utmp` file, which holds information for commands such as `who`. Registering user information for interactive batch jobs in `utmp` allows more accurate job accounting. For more information, see the `utmp(4)` reference page.

4.10.9 Using LSF/HPC Extensions

To enable LSF extensions for compressed host name list in the records in the `lsb.events` and `lsb.acct` files, and shortened PID list in output from the `bjobs` command, set the LSF_HPC_EXTENSIONS parameter in the `lsf.conf` file, as follows:
`LSF_HPC_EXTENSIONS="`*`extension_name ...`*`"`

The following extension name is supported:

- SHORT_EVENTFILE

  Compresses long host name lists when event records are written to the `lsb.events` and `lsb.acct` files for large parallel jobs. The short host string has the format:
  `number_of_hosts*real_host_name`

Note:

When the SHORT_EVENTFILE extension is enabled, daemons and commands from LSF versions earlier than Version 6.0 cannot recognize the `lsb.events` and `lsb.acct` file format.

For example, if the original host list record is:

```
6 "hostA" "hostA" "hostA" "hostA" "hostB" "hostC"
```

redundant host names are removed and the short host list record becomes:

```
3 "4*hostA" "hostB" "hostC"
```

The SHORT_EVENTFILE extension applies to the following events:

– JOB_START when a normal job is dispatched

– JOB_FORCE when a job is forced with the `brun` command

– JOB_CHUNK when a job is inserted into a job chunk

– JOB_FINISH in the `lsb.acct` file

For example, when the SHORT_EVENTFILE extension is not enabled, a JOB_START event such as the following would be logged in the `lsb.events` file:

```
"JOB_START" "6.0" 1058989891 710 4 0 0 10.3 64 "hostA" "hostA" "hostA" "hostA"
"hostB" "hostB" "hostB" "" "" 0 "" 0
```

When the SHORT_EVENTFILE extension is enabled, the same JOB_START event shortens the host list to "`4*hostA`" and "`3*hostB`", as follows:

```
"JOB_START" "6.0" 1058998174 812 4 0 0 10.3 2 "4*hostA" "3*hostB" "" "" 0 "" 0
""
```

When the SHORT_EVENTFILE extension is not enabled, a JOB_FINISH event such as the following would be logged in the `lsb.acct` file:

```
"JOB_FINISH" "6.0" 1058990001 710 33054 33816578 64 1058989880 0 0 1058989891
"user1" "normal" "span[ptile=32]" "" "" "hostA"
"/scratch/remotecluster/work/remote/logdir" "" "" "" "1058989880.710" 0 64
"hostA" "hostA" "hostA" "hostA" "hostB" "hostB" "hostB" 64 10.3 "" "sleep 100"
0.079999 0.270000 0 0 -1 0 0 0 0 0 0 0 -1 0 0 0 0 0 -1 "" "default" 0 64 "" ""
0 4304 6024 "" "" ""
```

When the SHORT_EVENTFILE extension is enabled, the same JOB_FINISH record shortens the host list to "`4*hostA`" and "`3*hostB`" as follows:

```
"JOB_FINISH" "6.0" 1058998282 812 33054 33816578 64 1058998163 0 0 1058998174
"user1" "normal" "span[ptile=32]" "" "" "hostA"
"/scratch/remotecluster/work/remote/logdir" "" "" "" "1058998163.812" 0 2
"4*hostA"
"3*hostB" 64 10.3 "" "sleep 100" 0.039999 0.259999 0 0 -1 0 0 0 0 0 0 0 -1 0 0
0 0 0 -1 "" "default" 0 64 "" "" 0 4304 6024 "" "" "" "" 0
```

4.10.10 Limitations on User-Level Checkpointing

The following limitations apply on user-level checkpointing:

- Remote Shell (RSH) or Secure Shell (SSH) software must be enabled on each host where checkpoint and restart take place. You must disable the password prompt in RSH and SSH.

  Note:

  Use the LSF_RSH parameter in the `lsf.conf` file to select the remote access shell. The `echkpnt` and `erestart` programs use the command you specify to start the LSF default checkpoint and restart programs `echkpnt.default` and `erstart.default`. You must specify a full path.

- The `bsub -i` command or jobs reading from standard input (`stdin`) are not supported. As LSF handles the original job and restarted job as two jobs, you should use the `bsub -o` and `bsub -e` commands so that LSF will append `stdout` and `stderr` to files correctly.
- Only sequential jobs with one process per job are supported; jobs submitted by the `prun` command are not supported.
- Checkpointing and restart must be done within LSF. Because they rely on environment variables exported from the sbatchd daemon, the `echkpnt` and `erestart` programs are not supported as external tools; you must use the `bchkpnt` and `brestart` commands to invoke `echkpnt` and `erestart` programs.

4.11 The lsf.conf File

This section describes the following parameters in the `lsf.conf` file:

- LSB_RLA_UPDATE (see Section 4.11.1 on page 4–27)
- LSB_RMSACCT_DELAY (see Section 4.11.2 on page 4–27)
- LSB_RMS_MAXNUMNODES (see Section 4.11.3 on page 4–27)
- LSB_RMS_MAXNUMRAILS (see Section 4.11.4 on page 4–27)
- LSB_RMS_MAXPTILE (see Section 4.11.5 on page 4–28)
- LSB_SHORT_HOSTLIST (see Section 4.11.6 on page 4–28)
- LSF_ENABLE_EXTSCHEDULER (see Section 4.11.7 on page 4–28)
- LSF_HPC_EXTENSIONS (see Section 4.11.8 on page 4–28)

- RLA_RMS_ONLINENODES (see Section 4.11.9 on page 4–29)

4.11.1 LSB_RLA_UPDATE

Syntax: `LSB_RLA_UPDATE=`*`seconds`*

Description: Specifies how often RLA should refresh its RMS map.

Default: 600 seconds

4.11.2 LSB_RMSACCT_DELAY

Syntax: `LSB_RMSACCT_DELAY=`*`time_seconds`*

Description: If set, the RES daemon waits the specified number of seconds before exiting to allow LSF and RMS job statistics to synchronize.

If `LSB_RMSACCT_DELAY`=0, the RES daemon waits indefinitely until the database is up to date.

Default: Undefined, the RES daemon does not wait at all

4.11.3 LSB_RMS_MAXNUMNODES

Syntax: `LSB_RMS_MAXNUMNODES=`*`integer`*

Description: Specifies the maximum number of nodes in a system. Specifies a maximum value for the `nodes` argument to the topology scheduler options. The `nodes` argument can be specified in the following ways:

- The `-ext` option of the `bsub` command
- The DEFAULT_EXTSCHED and MANDATORY_EXTSCHED parameters in the `lsb.queues` file.

Default: 1024

4.11.4 LSB_RMS_MAXNUMRAILS

Syntax: `LSB_RMS_MAXNUMRAILS=`*`integer`*

Description: Specifies the maximum number of rails in a system. Specifies a maximum value for the `rails` argument to the topology scheduler options. The `rails` argument can be specified in the following ways:

- The `-ext` option of the `bsub` command
- The DEFAULT_EXTSCHED and MANDATORY_EXTSCHED parameters in the `lsb.queues` file.

Default: 32

4.11.5 LSB_RMS_MAXPTILE

Syntax: `LSB_RMS_MAXPTILE=`*`integer`*

Description: Specifies the maximum number of CPUs per node in a system. Specifies a maximum value for the `ptile` argument to the topology scheduler options. The `ptile` argument can be specified in the following ways:

- The `-ext` option of the `bsub` command
- The DEFAULT_EXTSCHED and MANDATORY_EXTSCHED parameters in the `lsb.queues` file.

Default: 32

4.11.6 LSB_SHORT_HOSTLIST

Syntax: `LSB_SHORT_HOSTLIST=1`

Description: Displays an abbreviated list of hosts in `bjobs` and `bhist`, for a parallel job where multiple processes of a job are running on a host. Multiple processes are displayed in the following format: *`processes*host`*.

For example, if a parallel job is running 64 processes on `atlasd2`, the information is displayed in the following manner: `64*atlasd2`.

Default: Undefined (reports hosts in the default long format)

4.11.7 LSF_ENABLE_EXTSCHEDULER

Syntax: `LSF_ENABLE_EXTSCHEDULER=y|Y`

Description: Enables `mbatchd` external scheduling.

Default: Undefined

4.11.8 LSF_HPC_EXTENSIONS

Syntax: `LSF_HPC_EXTENSIONS="extension_name ..."`

Description: Enables LSF extensions for compressed host name list in the records in the `lsb.events` and `lsb.acct` files, and shortened PID list in output from the `bjobs` command.

Valid Values: SHORT_EVENTFILE

For more information on LSF HP AlphaServer SC Extensions, see Section 4.10.9 on page 4–24.

Default: Undefined

4.11.9 RLA_RMS_ONLINENODES

Syntax: `RLA_RMS_ONLINENODES=[[`*partition@*`]`*machine*`]:integer[%]...`

Description: Defines the maximum number or percentage of nodes to maintain online at any given time out of the total node count in any partition that LSF can schedule jobs to. The remaining nodes are held in reserve and available to maintain the specified maximum when a node becomes unavailable.

You cannot define which nodes are to be online and which nodes are to be spare.

LSF assumes that it controls all available nodes in an RMS partition, and calculates the spare node pool based on the static configuration. LSF cannot determine whether a given node is online or in the spare node pool.

Default: Undefined

4.12 Known Problems or Limitations

The following LSF-related problems or limitations are known in HP AlphaServer SC Version 2.6:

- The `bsub` and `esub` commands cannot apply the default queue-level PROCLIMIT

  If a queue-level default PROCLIMIT exists, and users submit a job without specifying the `-n` option, the job will use four CPUs instead of the default PROCLIMIT.

- Incorrect rails option may cause the job to remain pending forever

  LSF checks the validity of rail options at job submission against the LSB_RMS_MAXNUMRAILS parameter if it is set in the `lsf.conf` file. If incorrect rail option values pass this check, the job pends forever.

- LSF uses its own access control, usage limits, and accounting mechanism. You should not change the default RMS configuration for these features. Configuration changes may interfere with the correct operation of LSF. Do not use the commands, or configure any of the following features, described in Chapter 5:
  - Idle timeout
  - Memory limits
  - Maximum and minimum number of CPUs
  - Time limits
  - Time-sliced gang scheduling
  - Partition queue depth

Known Problems or Limitations

- When a partition is blocked or down, the status of `prun` jobs becomes `UNKNOWN`, and `bjobs` shows the jobs as still running. If the job is killed, `bjobs` reflects the change.
- The LSF log directory LSF_LOGDIR, which is specified in the `lsf.conf` file, must be a local directory. Do not use an NFS-mounted directory as LSF_LOGDIR. The default value for LSF_LOGDIR is `/var/lsf_logs`. This mount point (`/var`) is on CFS, not NFS, and is shared among domain members.

5

Managing the Resource Management System (RMS)

An HP AlphaServer SC system is a distributed-memory parallel machine in which each node is a distinct system with its own operating system controlling processes and memory. To run a parallel job, each component of the job must be started as individual processes on the various nodes that are used for the parallel job.

Resource Management System (RMS) is the system component that manages parallel operation of jobs. While the operating system on each node manages its own processes, RMS manages processes across the HP AlphaServer SC system.

For information about RMS commands, see Appendix A of the *HP AlphaServer SC User Guide*.

The information in this chapter is organized as follows:

- RMS Overview (see Section 5.1 on page 5–2)
- RMS Accounting (see Section 5.2 on page 5–3)
- Monitoring RMS (see Section 5.3 on page 5–6)
- Basic Partition Management (see Section 5.4 on page 5–8)
- Resource and Job Management (see Section 5.5 on page 5–15)
- Advanced Partition Management (see Section 5.6 on page 5–35)
- Controlling Resource Usage (see Section 5.7 on page 5–45)
- Node Management (see Section 5.8 on page 5–61)
- RMS Servers and Daemons (see Section 5.9 on page 5–67)
- Site-Specific Modifications to RMS: the pstartup Script (see Section 5.10 on page 5–74)
- RMS and CAA Failover Capability (see Section 5.11 on page 5–74)
- Using Dual Rail (see Section 5.12 on page 5–76)
- Useful SQL Commands (see Section 5.13 on page 5–76)

5.1 RMS Overview

This section introduces you to several RMS concepts, and describes the tasks performed by RMS.

5.1.1 RMS Concepts

As a system administrator, managing RMS involves managing the following:

- SC database

  RMS uses a Structured Query Language (SQL) database to coordinate its activities. The database contains configuration information, dynamic status data, and historical data. For more information about the SC database, see Chapter 3.

- Partitions

  A partition is a logical division of the nodes of the HP AlphaServer SC system into an organizational unit. For more information about RMS partitions, see Section 5.4 on page 5–8.

- Users

  To control access to partitions, RMS maintains a record of each user that is allowed use of the RMS system. User records are optional. However, without user records, any user can use any resource in the HP AlphaServer SC system. For more information about RMS users, see Section 5.6.2 on page 5–37.

- Projects

  A project comprises a set of users. A user may be in more than one project. Projects provide a convenient way of controlling the resource access of a group of users. In addition, projects can reflect the organizational affiliations of users. For more information about RMS projects, see Section 5.6.2 on page 5–37.

- Access controls

  Access control records associate users or projects with partitions. The access control record determines whether a given user can use the resources of a partition. In addition, you can impose limits to these resources. For more information about RMS access controls, see Section 5.6.2 on page 5–37.

- Accounting

  When users use resources, a record is stored in the SC database. Individual users or projects can use these records to determine resource usage. For more information about RMS accounting, see Section 5.2 on page 5–3.

- RMS servers and daemons

  RMS servers and daemons generally start and run automatically. However, a few operations require you to manually start or stop the RMS system. For more information about RMS servers and daemons, see Section 5.9 on page 5–67.

5.1.2 RMS Tasks

RMS is responsible for the following tasks:

- Allocates resources

  RMS has knowledge about the state of the system and is able to match user requests for resources (for example, CPUs) against the available resources.

- Schedules resources

  RMS is responsible for deciding when it will allocate resources to user jobs so that it most effectively uses the HP AlphaServer SC system.

- Invokes processes

  A parallel job comprises processes that run on different nodes. RMS is responsible for starting processes on nodes. It also takes care of important housekeeping duties associated with this, such as redirection of standard I/O and signals.

- Monitors node state

  RMS must know the status of nodes so that it can use node resources effectively.

- Handles events

  RMS responds to changes in node state by running scripts. The scripts perform automated actions to either report the new state or correct the error conditions.

- Monitors the HP AlphaServer SC Interconnect

  RMS monitors the state of the HP AlphaServer SC Interconnect switch. In addition, HP AlphaServer SC Interconnect diagnostics can be run through RMS.

5.2 RMS Accounting

When RMS allocates a resource, an entry for that resource is added to the `acctstats` table in the SC database. Entries in the `acctstats` table are updated periodically, as defined by the relevant `poll-interval` attribute in the `attributes` table.

The `rms-poll-interval` attribute sets the polling interval for all servers; the default value is 30 (seconds).

Each `acctstats` entry contains the information described in Table 5–1.

Table 5–1 Fields in acctstats Table

Field	Description
name	The name (number) of the resource as stored in the resources table.
uid	The UID of the user to whom the resource was allocated.
project	The project name under which the user was allocated the resource. Users who are members of multiple projects can select which project is recorded in the acctstats table by setting the RMS_PROJECT environment variable (to the project name), or by using the -P option, before allocating a resource.
started	The date and time at which the CPUs were allocated.
ctime	The date and time at which the statistics in this record were last collected.
etime	The elapsed time (in seconds) since CPUs were first allocated to the resource, including any time during which the resource was suspended.
atime	The total elapsed time (in seconds) that CPUs have been actually allocated — excludes time during which the resource was suspended. This time is a total for all CPUs used by the resource; for example: if the resource was allocated for 100 seconds and the resource had 4 CPUs allocated to it, this field would show 400 seconds.
utime	The total CPU time charged while executing user instructions for all processes executed within this resource. This total can include processes executed by several prun instances executed within a single allocate.
stime	The total CPU time charged during system execution on behalf of all processes executed within this resource. This total can include processes executed by several prun instances executed within a single allocate.
cpus	The number of CPUs used by the resource.
mem	The maximum memory extent of the program (in megabytes).
pageflts	The number of page faults requiring I/O summed over processes.
memint	Reserved for future use.
running	1 (one) in this field indicates that a resource is running or suspended. 0 (zero) indicates that the resource has been deallocated.

You can use the name field in the acctstats table to access the corresponding record in the resources table. This provides more information about the resource.

Each resources entry contains the information described in Table 5–2.

Table 5–2 Fields in resources Table

Field	Description
name	The name (number) of the resource.
partition	The name of the partition in which the resource is allocated.
username	The name of the user to which the resource has been allocated.
hostnames	The list of hostnames allocated to the resource. The list comprises a number of node specifications (the CPUs used by the nodes are specified in the cpus field). For resource requests that have not been allocated, the value of this field is Null.
status	The status of the resource. This can be one of the following: • queued or blocked — the resource has not yet been allocated any CPUs • allocated or suspended — the resource has been allocated CPUs, and is still running • finished — all jobs ran normally to completion • killed — one or more processes were killed by a signal (the resource is finished) • expired — the resource exceeded its time limit • aborted — the user killed the resource (the user used rcontrol kill resource or killed the prun or allocate commands) • syskill — the root user used rcontrol kill resource to kill the resource • failed — the jobs failed to start or a system failure (for example, node failure) killed the resource
cpus	The list of CPUs allocated on the nodes specified in the hostnames field.
nodes	The list of node numbers corresponding to the hostnames field. This field shows node numbers relative to the start of the partition (that is, the first node in the partition is Node 0). This field does not include (that is, skips) configured-out nodes.
startTime	While the resource is waiting to be allocated, this field specifies the time at which the request was made. When the resource has been allocated, this field specifies the time at which the resource was allocated.
endTime	If the resource is still allocated, this field is normally Null. However, if a timelimit applies to the resource, this field contains that time at which the resource will reach its timelimit. When the resource is finished (freed), this field contains the time at which the resource was deallocated.
priority	The current priority of the resource.
flags	State information used by the partition manager.
ncpus	The number of CPUs allocated to the resource.
batchid	If the resource is allocated by a batch system, this field contains an ID assigned by the batch system to the resource. This value of this field is -2 if no batchid has been assigned.
memlimit	The memory limit that applies to the resource. This value of this field is -2 if no memory limit applies.
project	The name of the project to which the resource has been allocated.
pid	The process ID.
allocated	Whether CPUs have been allocated to a resource or not. The value of this field is 0 (zero) initially, and changes to 1 (one) when CPUs are allocated to the resource. When resources are deallocated for the final time, the value of this field changes to 0 (zero).

5.2.1 Accessing Accounting Data

An example accounting summary script is provided in `/usr/opt/rms/examples/scripts/accounting_summary`. You can address site-specific needs by retrieving data from the SC database.

You can retrieve data using the `rmsquery` command. For example, the following command retrieves the total allocated time for each user of the system, sorted by `username`:

```
# rmsquery "select resources.username,acctstats.atime
   from resources,acctstats
   where resources.name=acctstats.name
   and (resources.status='finished'
     or resources.status='aborted'
     or resources.status='killed'
     or resources.status='expired')
   order by resources.username"
```

The above example considers only resources where the user's job ran to completion (that is, it ignores jobs where the `root` user killed the job or the system failed while running the job).

5.3 Monitoring RMS

This section describes the commands and tools that are available to monitor the status of RMS, including the following:

- rinfo (see Section 5.3.1 on page 5–6)
- rcontrol (see Section 5.3.2 on page 5–8)
- rmsquery (see Section 5.3.3 on page 5–8)

5.3.1 rinfo

The `rinfo` command shows the status of nodes, partitions, resources, servers, and jobs. It can be run as the `root` user or as an ordinary user. The data displayed by the `rinfo` command is taken from the SC database.

The `rinfo -h` command displays the available options. For example, `rinfo` without any options produces output similar to the following:

```
# rinfo
MACHINE        CONFIGURATION
atlas          day
PARTITION        CPUS     STATUS        TIME        TIMELIMIT        NODES
root             128                                                 atlas[0-31]
left             0/12     running       04:36:45                      atlas[0-2]
```

where

- MACHINE (`atlas`) shows the name of the HP AlphaServer SC system.
- CONFIGURATION (`day`) shows the currently active configuration (see Section 5.4 on page 5–8).

- PARTITION (`root`, `left`) shows the names of partitions in the configuration, the number of CPUs (allocated and available) in each partition, the partition status, the start time, the timelimit, and the nodes in each partition. The `root` partition is a special partition comprising all nodes in the HP AlphaServer SC system.

  Note:

  While a partition is in the `running` or `closing` state, RMS correctly displays the current status of the resources and jobs.

  However, if the partition status changes to `blocked` or `down`, RMS displays the following:

  - Resources status = status of resources at the time that the partition status changed to `blocked` or `down`
  - Jobs status = set to the `unknown` state

  RMS is unable to determine the real state of resources and jobs until the partition runs normally.

If a job is running, the `rinfo` command also displays the active resources and jobs, as shown in the following example:

```
# rinfo
MACHINE         CONFIGURATION
atlas           day
PARTITION         CPUS     STATUS        TIME          TIMELIMIT      NODES
root              128                                                 atlas[0-31]
left              4/12     running       04:41:44                     atlas[0-2]
RESOURCE          CPUS     STATUS        TIME          USERNAME       NODES
left.855          2        allocated     05:22         root           atlas0
JOB               CPUS     STATUS        TIME          USERNAME       NODES
left.849          2        running       00:02         root           atlas0
```

In this example, one resource is allocated (`855`) and that resource is running one job (`849`).

From time to time, some nodes may have failed or may be configured out. You can show the status of all nodes as follows:

```
# rinfo -n
running          atlas[0-2]
configured out   atlas3
```

This shows that `atlas0`, `atlas1`, and `atlas2` are running and that `atlas3` is configured out.

The `-nl` option shows more details about nodes. It shows how many CPUs, how many rails, how much memory, how much swap, and how much `/tmp` space is available on each node. This option also shows why nodes are configured out.

5.3.2 rcontrol

The `rcontrol` command shows more detailed information than the `rinfo` command. The `rcontrol help` command displays the various `rcontrol` options. For example, you can examine a partition as follows:

```
# rcontrol show partition=left
active           1
configuration    day
configured_nodes atlas[0-2]
cpus             12
free_cpus        8
memlimit         192
mincpus
name             left
nodes            atlas[0-2]
startTime        935486112
status           running
timelimit
timeslice
type             parallel
```

5.3.3 rmsquery

The `rinfo` and `rcontrol` commands do not show all of the information about partitions, resources, and jobs. You may query the database to display all of the available information. For example, the following command shows all partition attributes for all configurations:

```
# rmsquery -v "select * from partitions order by configuration"
```

5.4 Basic Partition Management

Partitions are used to organize nodes in the HP AlphaServer SC system into groups for organizational, management, and policy reasons. Nodes need not be members of a partition; however, you cannot run parallel jobs on such nodes. A node may be a member of only one active partition at a time — a node cannot be in two partitions at the same time.

Although the members of a partition do not have to be a contiguous set of nodes, overlapping of partitions is not allowed — for example, you cannot create the following:

- Partition `X` with members `atlas0` and `atlas3`
- Partition `Y` with members `atlas1`, `atlas2`, and `atlas4`

Partitions in turn are organized into configurations. A configuration comprises a number of partitions. Only one configuration can be active at a time. Configurations are used to manage alternate policy configurations of the system. For example, there could be three configurations: `day`, `night`, and `weekend`. A partition may exist in one configuration but not in another.

Generally, you create and manage partitions of the same name in different configurations as though each partition was unrelated. However, partitions of the same name in different configurations are related in the following respects:

- Access policies for users and projects apply to a given partition name in all configurations (see Section 5.6.2 on page 5–37).
- Jobs that are running on a partition when a configuration changes can continue to run in the new configuration, provided that the jobs are running on nodes that are part of the new configuration (see Section 5.5 on page 5–15).

This section describes the following basic partition management activities:

- Creating Partitions (see Section 5.4.1 on page 5–9)
- Specifying Configurations (see Section 5.4.2 on page 5–10)
- Starting Partitions (see Section 5.4.3 on page 5–12)
- Reloading Partitions (see Section 5.4.4 on page 5–13)
- Stopping Partitions (see Section 5.4.5 on page 5–13)
- Deleting Partitions (see Section 5.4.6 on page 5–15)

5.4.1 Creating Partitions

Partitions are created using the `rcontrol` command. For example, the following set of commands will create a number of partitions:

```
# rcontrol create partition=fs configuration=day nodes='atlas[0-1]'
# rcontrol create partition=big configuration=day nodes='atlas[2-29]'
# rcontrol create partition=small configuration=day nodes='atlas[30-31]'
```

This creates partitions and configurations of the layout specified in Table 5–3.

Table 5–3 Example Partition Layout 1

Partition	Nodes
`fs`	0–1
`big`	2–29
`small`	30–31

Note:

As mentioned earlier, a node cannot be in two partitions at the same time. You must ensure that you do not create illegal configurations.

Management servers must not be members of any partition.

In addition to nodes (described above), there are other partition attributes. These are described in Section 5.6 on page 5–35.

If a user does not specify a partition (using the `-p` option to the `allocate` and `prun` commands), the value of the `default-partition` attribute is used. When the SC database is created, this attribute is set to the value `parallel`. If you would like the `default-partition` attribute to have a different value, you can modify it as shown in the following example:

```
# rcontrol set attribute name=default-partition val=small
```

5.4.2 Specifying Configurations

There is no command to explicitly create a configuration — a configuration exists as soon as a partition is created, and ceases to exist as soon as the last partition that refers to it is deleted. See Section 5.4.6 on page 5–15 for more information about deleting partitions. When you create a partition, you also specify the name of the configuration.

In the following examples, two configurations are specified:

```
# rcontrol create partition=fs configuration=day nodes='atlas[0-1]'
# rcontrol create partition=big configuration=day nodes='atlas[2-29]'
# rcontrol create partition=small configuration=day nodes='atlas[30-31]'
# rcontrol create partition=fs configuration=night nodes='atlas[0-1]'
# rcontrol create partition=big configuration=night nodes='atlas[2-20]'
# rcontrol create partition=small configuration=night nodes='atlas[21-30]'
# rcontrol create partition=serial configuration=night nodes=atlas31
```

This creates partitions and configurations of the layout specified in Table 5–4.

Note:

For a given configuration:

- A node cannot be in more than one partition.
- The node range of a partition cannot overlap with that of another partition.

Table 5–4 Example Partition Layout 2

Configuration	Partition	Nodes
day	fs	0–1
	big	2–29
	small	30–31
night	fs	0–1
	big	2–20
	small	21–30
	serial	31

Switching between configurations involves stopping one set of partitions and starting another set. The process of starting and stopping partitions is described in the next few sections. In principal, configurations allows you to quickly change attributes of partitions. However, if jobs are running on the partitions, there are a number of significant restrictions that may prevent you from switching between configurations. These restrictions are due to the interaction between jobs that were originally started with one set of partition attributes but are now running with a new set of partition attributes.

When jobs are running on a partition, the following attributes cannot be changed because changing them will affect RMS operation:

- The nodes in the partition

  If a job is running on a node and the partition in its new configuration does not include that node, the job will continue to execute. However, the status of the job does not update (even when the job finishes) and you may be unable to remove the job. If you inadvertently create such a situation, the only way to correct it is to switch back to the original configuration. As soon as the original partition is started, the status of the job will update correctly.

  If you start a partition with a different name on the same set of nodes, a similar situation applies — in effect, you are changing the nodes in a partition.

- Memory limit

  If you reduce the memory limits, jobs that started with a higher memory limit may block.

If jobs are running when a partition is restarted, changes made to the following attribute will affect the job:

- Idle timeout (see Section 5.5.8 on page 5–24)

  The timer starts again — in effect, the timeout is extended by the partition restart.

If jobs are running when a partition is restarted, changes made to the following attributes do not apply to the jobs:

- Minimum number of CPUs

  A job with fewer CPUs continues to run.

- Timelimit

  The original timelimit applies.

- Partition Queue Depth

  This only applies to new resource requests.

5.4.3 Starting Partitions

When a partition is first created, it is created in the `down` state. A partition in the `down` state will not allocate resources or run jobs. To use the partition, it must be started. You can start a partition in two ways:

- Start a configuration.

  When you start a configuration, all partitions in the configuration are started. If you have several configurations, you must first start a configuration (before starting partitions) to designate the configuration that is to be activated.

  To start a configuration, use the `rcontrol` command as shown in the following example:

  ```
  # rcontrol start configuration=day
  ```

- Start the individual partition.

  This starts just this partition; other partitions that are members of the same configuration are unaffected.

  To start a partition, use the `rcontrol` command as shown in the following example:

  ```
  # rcontrol start partition=big
  ```

Before starting a partition, RMS checks that all configured-in nodes in the partition are in the running state. If any node in the partition is not running, the partition will fail to start. If the partition is started successfully, the partition status changes from `down` to `running` (as shown by the `rinfo` command).

When starting a partition, `rcontrol` runs a script called `pstartup`, as indicated in the `rcontrol` message output. See Section 5.6.1 on page 5–35 for more information about `pstartup`.

5.4.4 Reloading Partitions

When a partition is started, RMS reads the required information from the SC database. RMS does not automatically reread this data during operation, so it is not aware of any configuration changes that you make. If you make any of the following configuration changes, you must reload the partition:

- Any changes made by the *RMS Projects and Access Controls* menu (`sra_user`)
- A change to the `pmanager-idletimeout` attribute
- A change to the `pmanager-queuedepth` attribute

If you use `rcontrol` to modify the `users`, `projects`, or `access_controls` table, RMS automatically reloads the partition.

To manually reload the partition, use the `rcontrol reload` command as shown in the following example:

```
# rcontrol reload partition=big
```

Not all attributes of a partition are reread by doing a partition reload — some only take effect when a partition is started, as described in Section 5.4.5.

Note:

The `rcontrol reload partition` command has an optional `debug` feature. You should only use the `debug` feature if directly requested by HP to do so. If you are requested to enable debugging, do so in a separate command — do not use the `debug` feature when reloading a partition to reload the partition attributes.

5.4.5 Stopping Partitions

A partition must be in the `down` state before you can delete a partition.

In addition, some partition attributes only take effect when a partition is started, so you must stop and then restart the partition if you make configuration changes. For example, you must restart the partition to apply changes to the following partition attributes:

- Memory limit
- Minimum CPUs
- Timeslice quota
- Timelimit
- Type
- Elan use

Partitions are used to allocate resources and execute jobs associated with the resources (see Section 5.5.1 on page 5–16 for a definition of resource and job). Simply stopping a partition does not have an immediate affect on a user's `allocate` or `prun` command or the user's processes. These continue to execute, performing computations, doing I/O, writing text to `stdout`. However, since the partition is stopped, RMS is not actively managing the resources and jobs (for more information, see Section 5.5.10 on page 5–28).

While the partition is stopped, `rinfo` continues to show resources and jobs as follows:

- Resources: `rinfo` shows the state (`allocated`, `suspended`, and so on) that the resource was in when the partition was stopped.
- Jobs: `rinfo` shows the `unknown` state. The `jobs` table in the SC database stores the state that the job was in when the partition was stopped.

As described in Section 5.5.10 on page 5–28, it is possible to stop and restart a partition while jobs continue to execute. However, if you plan to change any of the partition's attributes, you should review Section 5.4.2 on page 5–10 before restarting the partition.

You can stop a partition in any of the following ways:

- A simple stop

  In this mode, the partition stops. Jobs continue to run, and the resources associated with these jobs remain allocated.

- Kill the jobs

  In this mode, the partition manager kills all jobs. While killing the jobs, the partition is in the `closing` state. When all jobs are killed, the resources associated with these jobs are freed and the partition state changes to `down`.

- Wait until the jobs exit

  In this mode, the partition manager changes the state of the partition to `closing`. In this state, it will not accept new requests from the `allocate` or `prun` commands. When all currently running jobs finish, the resources associated with these jobs are freed and the partition state changes to `down`.

To stop the partition, use the `rcontrol stop partition` command as shown in the following example:

```
# rcontrol stop partition=big
```

To kill all jobs and stop the partition, use the `rcontrol stop partition` command with the `kill` option, as shown in the following example:

```
# rcontrol stop partition=big option kill
```

To wait for jobs to terminate normally and then stop the partition, use the `wait` option, as shown in the following example:

```
# rcontrol stop partition=big option wait
```

As when starting a partition, you can stop either a given partition or a configuration.

- Stop the individual partition.

  To stop a partition, use the `rcontrol` command as shown in the following example:
  ```
  # rcontrol stop partition=big option kill
  ```

- Stop a configuration.

  To stop a configuration, use the `rcontrol` command as shown in the following example:
  ```
  # rcontrol stop configuration=day option kill
  ```

When you stop a partition, its status changes from `running` or `blocked` to `closing` and then to `down` (as shown by the `rinfo` command). RMS preserves resource numbers when the partition is restarted.

To switch between two configurations, the currently active configuration must first be stopped and then the partitions in the new configuration started. However, you do not need to explicitly stop the original partitions — `rcontrol` will automatically stop partitions in the currently active configuration if a different configuration is being started.

5.4.6 Deleting Partitions

Note:

Stop the partition (see Section 5.4.5 on page 5–13) before you delete the partition.

Partitions are deleted using the `rcontrol` command. For example, the following set of commands will delete the partitions created in Section 5.4.1 on page 5–9:

```
# rcontrol remove partition=fs configuration=day
# rcontrol remove partition=big configuration=day
# rcontrol remove partition=small configuration=day
# rcontrol remove partition=fs configuration=night
# rcontrol remove partition=big configuration=night
# rcontrol remove partition=small configuration=night
# rcontrol remove partition=serial configuration=night
```

5.5 Resource and Job Management

This section describes the following resource and job management activities:

- Resource and Job Concepts (see Section 5.5.1 on page 5–16)
- Viewing Resources and Jobs (see Section 5.5.2 on page 5–17)
- Suspending Resources (see Section 5.5.3 on page 5–20)
- Killing and Signalling Resources (see Section 5.5.4 on page 5–21)
- Resource Cleanup (see Section 5.5.5 on page 5–22)

- Running Jobs as Root (see Section 5.5.6 on page 5–23)
- Managing Exit Timeouts (see Section 5.5.7 on page 5–24)
- Idle Timeout (see Section 5.5.8 on page 5–24)
- Managing Core Files (see Section 5.5.9 on page 5–25)
- Resources and Jobs during Node and Partition Transitions (see Section 5.5.10 on page 5–28)
- Running User-Defined Scripts (see Section 5.5.11 on page 5–34)

5.5.1 Resource and Job Concepts

There are two phases to running a parallel program in RMS:

- Allocate the CPUs to the user.

  RMS allocates CPUs in response to the `allocate` or `prun` commands. When `allocate` or `prun` request CPUs, RMS creates a resource. A unique number identifies resources. The `rinfo` command shows resources information.

- Execute the parallel program.

  RMS executes the parallel program using the `prun` command. Each instance of `prun` creates a new job that executes the parallel program. A unique number identifies each job. The `rinfo` command shows jobs information.

There are two ways in which users can use `prun` to execute programs:

- Simply run `prun`.
  In this mode, `prun` first creates the resource and then runs the program.
- Use `allocate` to create the resource.
  When the resource is allocated, `allocate` creates a shell. The user then uses `prun` to execute the program within the previously allocated resource. While the resource is allocated, the user can run several jobs one after the other. You can also run several jobs at the same time by running `prun` in the background.

When a user requests a resource, the resource goes through several states:

1. If the HP AlphaServer SC system does not have enough CPUs, nodes, or memory to satisfy the request, the resource is placed into the `queued` state. The resource stays in the `queued` state until the request can be satisfied.
2. If the resource request would cause the user (or project associated with the user) to exceed their quota of CPUs or memory, the request is not queued; instead, it is placed into the `blocked` state. The resource stays in the `blocked` state until the user or project has freed other resources (so that the request can be satisfied within quota) and the HP AlphaServer SC system has enough CPUs, nodes, or memory to satisfy the request.

3. When the request can be satisfied, the CPUs and nodes are allocated to this resource. The resource is placed into the `allocated` state.
4. While the resource is in the `allocated` state, the user may start jobs.
5. After a resource reaches the `allocated` state, RMS may suspend the resource. This may be because a user explicitly suspended it (see Section 5.5.3 on page 5–20) or because a higher priority request preempts the resource. In addition, when timeslice is enabled on the partition, RMS suspends resources to implement the timeslice mechanism. If RMS suspends the resource, the resource status is set to `suspended`. When RMS resumes the resource, the state is set to `allocated`.
6. When the user is finished with the resource, the resource is set to the `finished` or `killed` state. The `finished` state is used when the resource request (either `allocate` or `prun` command) terminates normally. The `killed` state is used if a user kills the resource. Once a resource is `finished` or `killed`, the `rinfo` command no longer shows the resource (unless a resource request has been killed but has not yet had its CPUs deallocated — for such a request, the `rinfo` command displays the status of `zombie`); however, the state is updated in the `resources` table in the SC database.

Once a resource is allocated, the CPUs that have been allocated to the resource remain associated with the resource; that is, a resource does not migrate to different nodes or CPUs.

5.5.2 Viewing Resources and Jobs

The `rinfo` command allows you to view the status of all active resources and jobs. It does not show finished or killed jobs, with the exception of resource requests that have been killed but have not yet had their CPUs deallocated; for these requests, the `rinfo` command displays the status of `zombie`.

Note:

The status of resources and jobs only has meaning when the partition is in the `running` state. At other times, the status of resources and jobs reflects their state at the time when the partition left the `running` state. This means that while a partition is not in the `running` state, the `allocate` and `prun` commands may have actually exited. In addition, the processes associated with a job may also have exited.

The state of resources and jobs can only be updated by starting the partition. When the partition starts, it determines the actual state so that `rinfo` shows the correct data. During this phase, a resource may have a `reconnect` status indicating that RMS is attempting to verify the true state of the resource.

To view the status of partitions, resources, and jobs, simply run `rinfo` without any arguments, as shown in the following example:

```
# rinfo
MACHINE         CONFIGURATION
atlas           day
PARTITION          CPUS      STATUS        TIME   TIMELIMIT     NODES
root                 16                                       atlas[0-3]
parallel           8/12     running       17:31                atlas[1-3]
RESOURCE           CPUS      STATUS        TIME    USERNAME     NODES
parallel.254          4   allocated       17:31        fred     atlas[1-2]
parallel.255          4   allocated       01:30         joe     atlas3
JOB                CPUS      STATUS        TIME    USERNAME     NODES
parallel.240          4     running       00:15        fred     atlas[1-2]
parallel.241          4     running       00:02        fred     atlas[1-2]
parallel.242          4     running       01:30         joe     atlas3
```

You may also use `rinfo` with either the `-rl` or the `-jl` option, to view resources or jobs respectively.

The resource list tells you which user is using which system resource. For example, user `joe` is using 4 CPUs on `atlas3`. In addition, it shows how many jobs are running. However, when run without any flags, the `rinfo` command does not relate jobs to the resources in which they are running.

In HP AlphaServer SC Version 2.6, you can use the `-r` and `-rl` options with the `rinfo` command to view information on the jobs and processes running on the CPUs allocated to a given resource request, as shown in the following examples:

```
atlas0# rinfo -r 56
RESOURCE     PARTITION    CPUS   PRI   USERNAME     PID          PROJECT
56           parallel     4      50    duncan       12831        default

ETIME        ATIME        UTIME        STIME        PAGEFLTS     MEM
00:03        00:12        00:00        00:00        2171         8

STATUS       HOSTNAMES
allocated    atlas[0-3]

JOBS
52

PROCESSES PER NODE
2 atlas[0-3]
```

When run with the `-rl` option, the `-rinfo` command also lists each job associated with a resource, the ID numbers of the job's processes, and the name of the node the processes are running on, as shown in the following example:

```
atlas0# rinfo -rl 56
RESOURCE     PARTITION    CPUS   PRI   USERNAME     PID          PROJECT
56           parallel     4      50    duncan       12831        default
```

```
ETIME        ATIME        UTIME        STIME        PAGEFLTS     MEM
00:03        00:12        00:00        00:00        2171         8

STATUS       HOSTNAMES
allocated    atlas[0-3]

JOBS
52

HOSTNAME     JOB       PROCESSES
atlas0       52       12833    12832
atlas1       52        1931     1930
atlas2       52        2028     2027
atlas3       52        2028     2027
```

You can also determine other information from the output of the `rinfo -r` and `rinfo -rl` commands, including accounting statistics and the number of processes per node. Note that the number of processes per node will include the loader process that starts the execution of the jobs on the node.

You can also use the RMS database to relate jobs to resources, as shown in the following examples:

- To find out which jobs are associated with the resource `parallel.254`, use `rmsquery` to find the associated job records in the `jobs` table, as follows:

  ```
  # rmsquery -v "select name,status,cmd from jobs where resource='254'"
  name       status        cmd
  ---------------------------------
  240        running       a.out 200
  241        running       a.out 300
  ```

  The job numbers are 240 and 241. You can relate these to the `rinfo` display. In addition, using `rmsquery`, you can also determine other information not shown by `rinfo`. In the above example, the name of the command is shown.

- To find out which resource is associated with the job `parallel.242`, use `rmsquery` to find the associated job records in the `jobs` table, as follows:

  ```
  # rmsquery -v "select resource from jobs where name='242'"
  resource
  --------
  255
  ```

  Note:

  When displaying resource and job numbers, `rinfo` shows the name of the partition that the resource or job is associated with (for example, `parallel.254`). However, `rinfo` uses this convention only for your convenience — job and resource numbers are unique across all partitions. When using database queries, just use the numbers.

Note also that while job and resource numbers are unique, they are not necessarily consecutive. Although resource IDs are allocated in sequence, a `select` statement does not, by default, order the results by resource ID. You can use `rmsquery` to show results in a specific sequence. For example, to order resources by start time, use the following command:

```
# rmsquery "select * from resources order by startTime"
```

Note also that resource numbers are different to job numbers. A resource and job with the same number are not necessarily related.

5.5.3 Suspending Resources

You may suspend resources. When a resource is suspended, it has a status of `suspended`. Suspending a resource allows the CPUs, nodes, and memory that the resource was previously using to be used by other resource requests. If the resource has any jobs associated with it, RMS sends a SIGSTOP to all the underlying processes associated with the job.

When you resume a resource, RMS schedules the request in much the same way as when requests are normally scheduled. Therefore, the resource may not start running until other resources are freed up. Although the scheduling is done using the normal rules, there is one difference: the resumed request uses the same nodes and CPUs as it was using when originally allocated. When the resource is placed into the `allocated` state, RMS sends a SIGCONT to all the underlying processes associated with the resource's jobs.

To suspend a resource, use the `rcontrol` command.

There are two ways to specify the resource being suspended:

- By specifying a specific resource number
- By specifying combinations of partition name, user name, project name, and status

To suspend a resource using its resource number, use `rcontrol` as shown in the following example:

```
# rinfo -rl
RESOURCE         CPUS     STATUS   TIME     USERNAME NODES
small.870        4        allocated00:04    fred     atlas30
# rcontrol suspend resource=870
# rinfo -rl
RESOURCE         CPUS     STATUS   TIME     USERNAME NODES
small.870        4        suspended00:04    fred     atlas30
```

Note:

Although the `rinfo` command shows the resource as `small.870`, the resource is uniquely identified to `rcontrol` by the number, `870`, not by `small.870`.

To suspend a resource using the partition name, user name, project name, or status, use `rcontrol` as shown in the following examples:

```
# rcontrol suspend resource partition=parallel user=fred
# rcontrol suspend resource project=sc
# rcontrol suspend resource partition=big project=proj1 project=proj2
status=queued status=blocked
```

When different values of the same criteria are specified, a resource matching either value is selected. Where several different criteria are used, the resource must match each of the criteria. For example, the last example is parsed like this:

SUSPEND RESOURCES
IN
    `big` partition
WHERE
    `project` is `proj1` OR `proj2`
AND
    `status` is `queued` OR `blocked`

The entity that RMS allocates and schedules is the resource — jobs are managed as part of the resource in which they were started. So when you suspend a resource, you are suspending all the jobs belonging to the resource and all the processes associated with each job.

You resume a resource as shown in the following examples:

```
# rcontrol resume resource=870
# rcontrol resume resource partition=big status=suspended
# rcontrol resume resource user=fred
```

The `rcontrol suspend resource` command can be run by either the `root` user or the user who owns the resource. If the `root` user has suspended a resource, the user who owns the resource cannot resume the resource.

RMS may suspend resources either as part of timeslice scheduling or because another resource request has a higher priority. If this happens, the `rinfo` command also shows that the resource is suspended. However, any attempt to resume the resource using the `rcontrol` command will fail.

5.5.4 Killing and Signalling Resources

You can force a resource to terminate by killing it, as shown in the following example:

```
# rcontrol kill resource=870
```

You can also specify combinations of partition, user, project, and status to kill resources, as shown in the following example:

```
# rcontrol kill resource partition=big project=proj1 project=proj2 status=queued
status=blocked
```

Note:

Be careful to make sure you are using the resource number, not a job number. If you use a job number instead of a resource number, `rcontrol` will attempt to find a resource with that number and you may kill an unintended resource.

When a resource is killed, all jobs associated with the resource are terminated. Processes associated with the jobs are terminated by being sent a SIGKILL signal.

The `root` user can use the `rcontrol` command to kill any resource. A non-`root` user may only use the `rcontrol` command to kill their own jobs.

Note:

When a resource is killed, little feedback is given to the user. However, if the user specifies the `-v` option, `prun` will print messages similar to the following:

```
prun: connection to server pmanager-big lost
prun: loaders exited without returning status
```

You can use the `rcontrol kill resource` command to send other signals to the process in a job. The signal number or the signal name can be used in the command.

The following example command shows how to send the USR1 signal:

```
# rcontrol kill resource=870 signal=USR1
```

In the above example, a USR1 signal is sent to each process in all jobs associated with the resource.

5.5.5 Resource Cleanup

When a parallel job completes, RMS executes its cleanup process before allocating the CPUs to another resource request. The cleanup process consists of instructing the `rmsd` daemon on each node that was in use to kill any remaining processes, remove shared memory segments, (optionally) delete core files, or execute an `epilogue` script. This cleanup normally completes quickly and the resource may be displayed briefly by `rinfo` as being in a `zombie` state.

The cleanup process will be delayed if any remaining processes associated with a node's resource cannot be removed. In such cases, `rinfo` continues to display the resource as being in the `zombie` state. A long delay in cleaning up after a parallel job could cause the partition to block if the `rmsd` daemon becomes unresponsive. If the cleanup after a parallel job exceeds the value specified in the `cleanup-timeout` attribute (default 15 minutes) and the `ignore-cleanup-failure` attribute is set, RMS deallocates the zombie resource. The resource is then available for new resource requests. However, you may wish to examine the

processes running on node to determine if the hung process eventually finished up normally. If processes remain, you should report this problem to your HP support engineer, who may ask you to gather more information from logs or to run the `dumpsys` command.

5.5.6 Running Jobs as Root

Normally, resources are allocated from a running partition; that is, you specify a partition to use with the `allocate` or `prun` commands. The `root` user can allocate resources and run jobs this way. However, the `root` user may also allocate resources from the `root` partition, as shown in the following example:

```
# prun -n 2 -N 2 -p root hostname
atlas0
atlas1
```

The `root` partition differs from other partitions in the following respects:

- It may only be used by the `root` user.
- It is neither started nor stopped. Consequently, it does not have a partition manager daemon.
- It always contains all nodes.
- Although you can use the `-n`, `-N`, and `-c` options in the same way as you would normally allocate a resource, a resource is not created. This means that the `root` user can run programs on CPUs and nodes that are already in use by other users (that is, you can run programs on CPUs and nodes that are already allocated to other resources). In effect, using the `root` partition bypasses the resource allocation phase and proceeds directly to the execution phase.
- The status of nodes is ignored. The `prun` command will attempt to run the program on nodes that are not responding or that have been configured out.

The `root` user can also allocate resources and run jobs on normal partitions. The same constraints to granting the resource request (available CPUs and memory) are applied to the `root` user as to ordinary users — with one exception: the `root` user has higher priority and can preempt non-`root` users. This forces other resources into the `suspended` state to allow the `root` user's resource to be allocated.

Note:

Do not use the `allocate` or `prun` command to allocate (as `root`) all the CPUs of any given node in the partition. If the partition is stopped while the resource remains allocated and later started, the `pstartup` script (described in Section 5.6.1 on page 5–35) will not run.

5.5.7 Managing Exit Timeouts

RMS determines that a job has finished when all of the processes that make up the job have exited, or when one of the processes has been killed by a signal. However, it is also possible for one or more processes to exit, leaving one or more processes running. In some cases, this may be normal behavior. However, in many cases the early exit of a process may be due to a fault in the program. In parallel programs, such a fault will probably cause the program as a whole to hang. You can use an exit timeout to control RMS behavior in such situations.

When the first process in a job exits, RMS starts a timer. If all remaining processes have not exited when this time period expires — that is, within the exit timeout period — RMS determines that these processes are hung and kills them. When the exit timeout expires, the `prun` command exits with a message such as the following:

```
prun: Error: program did not complete within 5000 seconds of the first exit
```

By default, the exit timeout is infinite (that is, the exit timeout does not apply and a job is allowed to run forever). There are two mechanisms for changing this, as follows:

- You can set the `exit-timeout` attribute in the `attributes` table. The value is in seconds.
- You can set the RMS_EXITTIMEOUT environment variable before running `prun`. The value is in seconds.

You can create the `exit-timeout` attribute as shown in the following example:

```
# rcontrol create attribute name=exit-timeout val=3200
```

You can modify the `exit-timeout` attribute as shown in the following example:

```
# rcontrol set attribute name=exit-timeout val=1200
```

You should choose a value for the exit timeout in consultation with users of your system. If you choose a small value, it is possible that correctly behaving programs may be killed prematurely. Alternatively, a long timeout allows hung programs to consume system resources needlessly.

The RMS_EXITTIMEOUT environment variable overrides any value that is specified by the `exit-timeout` attribute. This is useful when the `exit-timeout` attribute is too short to allow a program to finish normally (for example, process 0 in a parallel program may do some post-processing after the parallel portion of the program has finished and the remaining processes have exited).

5.5.8 Idle Timeout

The `allocate` command is used to allocate a resource. Generally, the user then uses `prun` to run a job within the previously allocated resource. The resource remains allocated until the user exits `allocate`. It is possible that users may use `allocate` and forget to exit — this

consumes system resources wastefully. You can prevent this by using the `pmanager-idletimeout` attribute. This defines a time (in seconds) that a resource is allowed to be idle (that is, without running a job) before it is deallocated by RMS.

By default, the `pmanager-idletimeout` attribute is set to `0` or `Null`, indicating that no timeout should apply. You can set the attribute as shown in the following example:

```
# rcontrol set attribute name=pmanager-idletimeout val=300
```

You must reload all partitions (see Section 5.4.4 on page 5–13) for the change to take place.

Once a resource has been idle for longer than the value specified by `pmanager-idletimeout`, `allocate` will exit with the following message:

```
allocate: Error: idle timeout expired for resource allocation
```

The exit status from the `allocate` command is set to 125.

5.5.9 Managing Core Files

Core-file management in RMS has the following aspects:

- Location of Core Files (see Section 5.5.9.1 on page 5–25)
  By default, core files are place in `/local/core/rms`. You can change this location.
- Backtrace Printing (see Section 5.5.9.2 on page 5–26)
  When RMS detects a core file, it runs a core analysis script that prints a backtrace of the core. You can change this behavior.
- Preservation and Cleanup of Core Files (see Section 5.5.9.3 on page 5–27)
  By default, RMS does not delete core files. You can configure RMS so that it automatically deletes core files when a resource finishes.

5.5.9.1 Location of Core Files

By default, RMS system places core files of parallel programs into the `/local/core/rms` directory. The `/local` path is a CDSL to a local file system on the internal disk of each node. This means that if a large parallel program exits abnormally, the core files are written to a local file system instead of across the network.

If you wish, you can change the location of core files by setting the `local-corepath` attribute. However, to make the change take effect, you will need to restart RMS.

Use the `rcontrol` command to create the attribute, as shown in the following example:

```
# rcontrol create attribute name=local-corepath val=/local/apps/cores
```

Then use the `rcontrol` command to change the attribute, as shown in the following example:

```
# rcontrol set attribute name=local-corepath val=/apps/cores
```

In the first example, the /local file system is specified. This means that core files are written to a file system that is physically served by the local node. In the second example, an SCFS file system is specified. This means that all core files are written to a single file system. In general, writing core files to an SCFS file system will take longer than writing them to multiple local file systems. Whether this has a significant impact depends on the number of processes that core dump in a single job.

The core files are not written directly to the local-corepath directory. Instead, when a resource is allocated, a subdirectory is created in local-corepath. For example, for resource 123, the default location of core files is /local/core/rms/123.

A change to the local-corepath attribute only takes effect when RMS is started. To restart RMS, enter the following command:

```
# rmsctl restart
```

5.5.9.2 Backtrace Printing

There are two aspects to backtrace printing:

- Determining that a process has exited abnormally
- Analyzing the core file

RMS uses two mechanisms to determine that a process has been killed by a signal, as follows:

- RMS detects that the process it has started has been killed by a signal. It is possible for a process to fork child processes. However, if the child processes are killed by a signal, RMS will not detect this; RMS only monitors the process that it directly started.
- The process that RMS started exits with an exit code of 128 or greater. This handles the case where the process started by RMS is not the real program but is instead a shell or wrapper script.

If users run their programs inside a shell (for example, prun sh -c 'a.out'), no special action is needed when a.out is killed. In this example, sh exits with an exit code of 127 plus the signal number. However, if users run their program within a wrapper script (for example, prun wrapper.sh), they must write the wrapper script so that it returns a suitable exit code. For example, the following fragment shows how to return an exit code from a script written in the Bourne shell (see the sh(1b) reference page for more information about the Bourne shell):

```
#!/bin/sh
.
.
a.out
retcode=$?
.
.
exit $retcode
```

When RMS determines that a program has been killed by a signal, it runs an analysis script that prints a backtrace of the process that has failed. The analysis script looks at any core files and uses the `ladebug(1)` debugger to print a backtrace. The analysis script also runs an RMS program, `edb`, which searches for errors that may be due to failures in the HP AlphaServer SC Interconnect (Elan exceptions).

Note:

You must have installed the HP AlphaServer SC Developer's Software License (OSF-DEV) if you would like `ladebug` to print backtraces.

Note:

The analysis script runs within the same resource context as the program being analyzed. Specifically, it has the same memory limit. If the memory limit is lower than 200 MB, `ladebug` may fail to start.

You may also replace the core analysis script with a script of your own. If you create a file called `/usr/local/rms/etc/core_analysis`, this file is run instead of the standard core analysis script.

5.5.9.3 Preservation and Cleanup of Core Files

The default behavior is for RMS to leave the core files in place when the resource request finishes. You can change this behavior so that RMS cleans up the core file directory when the resource finishes. You can specify this in either of the following ways:

- The user can define the RMS_KEEP_CORE environment variable
- You can set the `rms-keep-core` attribute

To specify that RMS should delete (that is, not keep) the core file directory, set the value of the `rms-keep-core` attribute to 0, as follows:

```
# rcontrol set attribute name=rms-keep-core val=0
```

To specify that RMS should not delete the core files, set the value of the `rms-keep-core` attribute to 1, as follows:

```
# rcontrol set attribute name=rms-keep-core val=1
```

You must restart RMS for these attribute changes to take effect.

Setting the `rms-keep-core` attribute to 0 means that you do not need to manage the cleanup of core files. If you opt to keep core files (by setting the `rms-keep-core` attribute to 1), you may need to introduce some mechanism to delete old unused core files; otherwise, they may eventually fill the file system.

As mentioned in Section 5.5.9.1 on page 5–25, the core file directory is specific to the resource. If you allocate a resource, each of the jobs that run under that resource use the same core file directory. If you opt to have RMS delete core files, it does not do so until the resource is finished. Therefore, if you use `allocate` to allocate a resource and then use `prun` to run the job, the core files from the job are not deleted until you exit from the shell created by `allocate`. If the user knows in advance that a program may be killed, an `allocate`-followed-by-`prun` sequence allows the user to analyze the core files even if `rms-keep-core` is set to 0.

The RMS_KEEP_CORE environment variable allows a user to override the value of the `rms-keep-core` attribute on a per-resource basis. The user sets the environment variable to either 0 or 1 and then allocates the resource.

Caution:

If the `rmsd` daemon on a node is restarted, it uses the `rms-keep-core` attribute to determine whether or not it should delete the core file directory — it ignores the value of the RMS_KEEP_CORE environment variable. This means that it is possible that RMS will delete core files even though the user has set the RMS_KEEP_CORE environment variable to 1.

5.5.10 Resources and Jobs during Node and Partition Transitions

This section describes what happens to resources and jobs when a partition stops running normally, either because a node has died or because the partition has been manually stopped. It also discusses what happens when the partition is started again.

5.5.10.1 Partition Transition

In this section, we assume that you are stopping a partition without using the `option wait` or `option kill` arguments. The behavior of resources and jobs when the `option` argument is used is described in Section 5.4.5 on page 5–13.

Stopping a partition (omitting the `option wait` or `option kill` arguments) does not have an immediate effect on running jobs; the processes that were running as part of the job continue to run normally. The `prun` command associated with the job will continue to print standard output (`stdout`) and standard error (`stderr`) from the processes. However, if you have used the `-v` option, it will print a message similar to the following:

```
prun: connection to server pmanager-big lost
```

When you stop a partition, the RMS daemon associated with the partition (the partition manager) exits. Since the daemon is not active, no further scheduling actions will take place — jobs that were running continue to run; jobs that were suspended remain suspended. New

resource requests cannot be granted. The last status of resources and jobs remains frozen in the database. `rinfo` shows this status for resources. However, for jobs, `rinfo` shows a status of `unknown`. This is because, while resource status cannot change, the actual state of the processes belonging to jobs is not known.

In normal operation, once RMS has started the processes belonging to a job, the processes run without any supervision from RMS (assuming that the job is not preempted by a higher-priority resource). The main supervision of the processes occurs when the job finishes. A job finishes in the following situations:

- All processes exit
- One or more processes are killed (for example, they exit abnormally)
- The `prun` command itself is killed
- `rcontrol kill resource` is used to kill the resource

If a job continues to run while the partition is stopped and the partition is later restarted, the job and associated processes are unaffected. If the job finishes while the partition is stopped, RMS is unable to handle this event in the normal way. However, the next time the partition is started, RMS is able to detect that changes have occurred in the job and take appropriate action.

Table 5–5 shows what happens to a resource, job, and associated processes while the partition is stopped and when the partition is later restarted.

Table 5–5 Effect on Active Resources of Partition Stop/Start

Job Behavior	Aspect	While Partition Down	When Partition Started
Continues to run	Resource status	Unchanged	Shows `reconnect` until `prun` and `pmanager` make contact; then shows status as determined by scheduler.
	Job status	`rinfo` shows `unknown`	As determined by scheduler.
	`prun`	Continues to run; with `-v`, prints `Connection to server pmanager-parallel lost`	With `-v`, prints message that `pmanager` is ok.
	Processes	Continues to run	Continues to run.

Table 5–5 Effect on Active Resources of Partition Stop/Start

Job Behavior	Aspect	While Partition Down	When Partition Started
Was in suspended state when partition stopped	Resource status	Unchanged	Has `suspended` state. RMS treats this resource as though the `root` user had used `rcontrol suspend resource`. The `root` user must resume the resource.
	Job status	`rinfo` shows `unknown`	Shows `suspended`.
	`prun`	Continues to run; with `-v`, prints `Connection to server pmanager-parallel lost`	With `-v`, prints message that `pmanager` is ok.
	Processes	Remain in stopped state	Remain in stopped state until `root` resumes the resource.
Was queued or blocked	Resource status	Unchanged	Unchanged
	Job status	Not applicable (there is no job)	Not applicable.
	`prun`	Continues to run; with `-v`, prints `Connection to server pmanager-parallel lost`	With `-v`, prints message that `pmanager` is ok.
	Processes	Not applicable (there is no process)	Not applicable.
`prun` killed (typically when user enters Ctrl/C)	Resource status	Unchanged	Shows `reconnect` for a while. After a timeout, RMS determines that `prun` has exited. Status in database is marked `failed`. EndTime in database is set to current time.
	Job status	`rinfo` shows `unknown`	Status is marked `failed`.
	`prun`	Exits when killed	Not applicable.
	Processes	Processes are killed	Not applicable.
All processes exit	Resource status	Unchanged	Shows `reconnect` for a while. When RMS determines (from `prun`) that the processes have exited, status is changed to `finished`. EndTime in database is set to current time.

Table 5–5 Effect on Active Resources of Partition Stop/Start

Job Behavior	Aspect	While Partition Down	When Partition Started
	Job status	`rinfo` shows `unknown`	Status marked `failed`.
	`prun`	Exits. With `-v`, prints final status	Not applicable.
	Processes	Exit	Not applicable.
One or more processes are killed	Resource status	`prun` exits	Shows `reconnect` for a while. When RMS determines (from `prun`) that the processes have exited, status is changed to `finished`. EndTime in database is set to current time.
	Job status	`rinfo` shows `unknown`	Status marked `failed`.
	`prun`	Prints backtrace from killed processes	Exits.
	Processes	The remaining processes are killed	Not applicable.
`rcontrol kill resource`		Not possible while partition stopped.	Not applicable.
Process killed in suspended resource	Resource status	Unchanged (that is, `suspended`)	Remains `suspended`. When `rcontrol resume resource` is used, becomes marked `killed`.
	Job status	Remains `unknown`	Remains `suspended` until resumed; then `killed`.
	`prun`	Does not exit	Does not exit until resumed, then exits.
	Processes	The remaining processes remain in the stopped state	Remaining processes are killed when the resource is resumed.
`prun` killed while resource suspended	Resource status	Remains `suspended`	Remains `suspended`. When `rcontrol resume resource` is used, becomes marked `failed`.
	Job status	Remains `unknown`	Remains `suspended` until resumed; then `failed`.
	`prun`	Exits	Not applicable.

Table 5–5 Effect on Active Resources of Partition Stop/Start

Job Behavior	Aspect	While Partition Down	When Partition Started
	Processes	The remaining processes remain in the stopped state	Remaining processes are killed when the resource is resumed.

Table 5–5 shows that it is possible to stop a partition and restart it later without impacting the normal operation of processes. However, this has three effects, as follows:

- If the resource was suspended when the partition was stopped, the `root` user must resume the resource after the partition is started again. This applies even if the resource was suspended by the scheduler (preempted by a higher priority resource or by timesliced gang scheduling).
- If `prun` or any of the processes exit, the end time recorded in the database is the time at which the partition is next started, not the time at which the `prun` or processes exited.

5.5.10.2 Node Transition

In the above discussion, there were no changes in the nodes that formed part of the partition. However, a node can fail, or you might want to configure a node out of a partition. This section describes what happens to the resources, jobs, and associated processes when a node fails or is configured out. (Section 5.8 on page 5–61 describes what happens to a node when it fails or is configured out.)

Although there are many possible sequences of events, there are two basic situations, as follows:

- The node fails.

  The node is marked as not responding. The partition briefly goes into the `blocked` state while the node is automatically configured out. The partition then returns to the `running` state.

- The node is configured out using the `rcontrol configure out` command.

  The sequence of events for this can be any of the following:
  - Node is configured out while partition is still running. The partition briefly blocks and then resumes running without the node.
  - Partition is stopped using `rcontrol`. Node is configured out. Partition is started.
  - Partition is stopped using `rcontrol`. Partition is removed (that is, deleted). Partition of same name is created, omitting a node. Partition is started.

Table 5–6 discusses the effect on resources, jobs, and processes for these two basic situations. The table only addresses resources to which the affected node was allocated. Other resources are handled as described in Table 5–5.

Table 5–6 Effect on Active Resources of Node Failure or Node Configured Out

Situation	Aspect	While Partition Blocked/Stopped	When Partition Next Returns to Running State
Node fails	Resource status	`rinfo` (briefly) shows `blocked`	Resource is marked `failed`. End time is time at which partition is started.
	Job status	`rinfo` (briefly) shows `blocked`	Resource is marked `failed`.
	`prun`	Continues to run until "configure out" completes.	Exits.
	Processes	Continue to run (or remain stopped if the resource was suspended). Processes on the failed node are, of course, lost.	Processes on other nodes are killed.
Node is configured out	Resource status	`rinfo` shows `blocked`	Resource is marked `failed`. End time is time at which "configure out" occurred, or (for stopped partition) partition is started.
	Job status	`rinfo` shows `blocked`	Resource is marked `failed`.
	`prun`	Continues to run until "configure out" completes or partition is restarted.	Exits.
	Processes	Continue to run (or remain stopped if the resource was suspended). Processes on the configured-out node were lost when the node failed.	Processes on other nodes are killed.

5.5.10.3 Orphan Job Cleanup

Throughout the various node and partition transitions, RMS attempts to keep the SC database and the true state of jobs and processes synchronized. However, from time to time, RMS fails to remove processes belonging to resources that are no longer allocated. You can use the `sra_orphans` script to detect and optionally clean up these processes.

You can run this script in a number of different ways, as shown by the following examples:

```
# scrun -n all '/usr/opt/srasysman/bin/sra_orphans -kill rms -kill nontty'
# prun -N a -p big /usr/opt/srasysman/bin/sra_orphans -kill rms -kill nontty
```

The sra_orphans script takes the following optional arguments:

- -nowarn

  Prints a warning each time it find a process that was started by RMS that should not be active. In addition, it prints a warning about any process whose parent does not have a tty device (that is, a process not started by an interactive user). With the -nowarn option, the sra_orphans script ignores such processes.

- -kill rms

  Kills any processes that RMS has started that should not be active. Unless this option is specified, sra_orphans only prints a warning (if the -nowarn option is not specified).

- -kill nontty

  Kills any processes whose parent does not have a tty device. Unless this option is specified, sra_orphans only prints a warning (if the -nowarn option is not specified).

5.5.11 Running User-Defined Scripts

RMS allows you to run user-defined scripts as part of the process of allocating and deallocating CPUs to a user. In the process of allocating CPUs to a user, RMS optionally executes a script; this script must be named prologue. In the process of deallocating CPUs, RMS optionally executes a script; this script must be named epilogue. The scripts execute as root.

RMS checks for prologue and epilogue scripts in the default site-specific directory as specified by the local-rmsdir attribute. The default value for this attribute is /usr/local/rms. The existence of the executable scripts prologue and epilogue (or symbolic links with the same names) in this directory is sufficient to trigger their execution.

If such scripts exist, CPU allocation and deallocation will not complete until after the scripts have run. This may introduce delays. For example, when a user's prun command completes, the CPUs are not deallocated until after the epilogue scripts have completed on all nodes.

Where multiple parallel jobs are run on the same CPUs as part of a single resource request, the prologue script is executed prior to the first job and the epilogue script is executed after the last job.

The prologue and epilogue scripts should return 0 (zero) on success. A non-zero exit status on any node is interpreted as an error and causes the parallel job to fail with the following message:

```
prun: Error: prologue failed on
```

and the exit status 122. Failure of an epilogue script is not reported to the user — in general it is executed after the prun command has finished.

Caution

If a `prologue` or `epilogue` script returns a non-zero exit status, the RMS daemon automatically configures the node out. In doing so, it sets the `reason` field for the node to `prologue failed` or `epilogue failed`, as appropriate.

When using `prologue` and `epilogue` scripts, users should be aware that there may be other parallel jobs running on the node at the same time as the script. Failure of the script, followed by automatic configuring out of the node, will cause such jobs to be killed.

5.6 Advanced Partition Management

This section describes the following advanced partition management topics:

- Partition Types (see Section 5.6.1 on page 5–35)
- Controlling User Access to Partitions (see Section 5.6.2 on page 5–37)
- Controlling Whether the Partition Uses HP AlphaServer SC Elan Adapter Cards (see Section 5.6.3 on page 5–45)

5.6.1 Partition Types

Partitions have a `type` attribute that determines how the partition is to be used. The partition types are as follows:

- `parallel`
  This type of partition is used to run programs that are started by the `prun` command. Users are not allowed to login to the nodes directly — instead, the nodes in the partition are reserved for use by RMS.
- `restricted`
  This partition type must be used for partitions where resources are allocated under the control of LSF. This provides the necessary environment for running LSF, by allowing only the root user to create allocations and submit jobs, and by specifying the owners of the resources in the application interface. The `restricted` partition type can also be used for other batch scheduling systems, provided that they are organized in such a way that a root-owned process (daemon) allocates resources and submits jobs.
- `batch`
  This partition type is used by certain batch systems and is retained for backward compatibility. Do not use this partition type for partitions controlled by LSF.

- login
  RMS will not allow users to allocate resources in this type of partition. This partition type is intended to allow you to reserve nodes for normal interactive use.
- general
  This partition type combines aspects of parallel and login partitions: users can login for normal interactive use and they may also allocate resources. Although a user may allocate a resource, the user does not have exclusive use of the CPUs and nodes allocated to the resource, because other users may log in to those nodes and run programs.

By default when you create a partition, the type is set to general. You can specify a different partition type using the rcontrol command, as shown in the following example:

```
# rcontrol create partition=fs configuration=day type=login nodes='atlas[0-1]'
# rcontrol create partition=big configuration=day type=parallel nodes='atlas[2-29]'
# rcontrol create partition=small configuration=day nodes='atlas[30-31]'
```

This creates the fs partition of type login, the big partition of type parallel, and the small partition of type general.

You may also change the partition type by using the rcontrol command, as shown in the following example:

```
# rcontrol set partition=small configuration=day type=restricted
```

You must restart the partition for this change to take effect.

The only way to run programs on nodes in a parallel partition is to run them through RMS. To enforce this, the HP AlphaServer SC system uses the /etc/nologin_*hostname* file. By removing or creating the /etc/nologin_*hostname* file, you can allow or prevent interactive logins to the node. For example, the file /etc/nologin_atlas2 controls access to atlas2.

The /etc/nologin_*hostname* files must be created and deleted so that they reflect the configuration of the various partitions. This is done automatically by rcontrol when you start partitions. On the HP AlphaServer SC system, rcontrol runs a script called pstartup.OSF1 when you start a partition. This script creates and deletes /etc/nologin_*hostname* files as described in Table 5–7.

Table 5–7 Actions Taken by pstartup.OSF1 Script

Partition Type	Action
parallel	Creates an /etc/nologin_*hostname* file for each node in the partition.
restricted	Creates an /etc/nologin_*hostname* file for each node in the partition
batch	Creates an /etc/nologin_*hostname* file for each node in the partition.
login	Deletes the /etc/nologin_*hostname* file for each node in the partition.
general	Deletes the /etc/nologin_*hostname* file for each node in the partition.

The pstartup.OSF1 script is only run when you start a partition. No action is taken when you stop a partition. Therefore, the /etc/nologin_*hostname* files remain in the same state as that which they had before the partition was stopped. If you are switching between configurations, then as the partitions of the new configuration are started, the /etc/nologin_*hostname* files are created and deleted to correspond to the new configuration.

You should not need to manually create or delete /etc/nologin_*hostname* files, unless you remove a node from a parallel partition and then attempt to log into this node. Since the node is not in any partition, the pstartup.OSF1 script will not process that node. As the node was previously in a parallel partition, an /etc/nologin_*hostname* file will exist. If you would like users to login to the node, you must manually delete the /etc/nologin_*hostname* file.

If you wish to implement a different mechanism to control access to partitions, you can write a site-specific pstartup script. If rcontrol finds a script called /usr/local/rms/etc/pstartup, it will run this script instead of the pstartup.OSF1 script.

5.6.2 Controlling User Access to Partitions

The information in this section is organized as follows:

- Concepts (see Section 5.6.2.1 on page 5–37)
- RMS Projects and Access Controls Menu (see Section 5.6.2.2 on page 5–38)
- Using the rcontrol Command (see Section 5.6.2.3 on page 5–43)

5.6.2.1 Concepts

RMS recognizes users by their standard UNIX accounts (UID in the /etc/passwd file). Unless you specify otherwise:

- All users are members of a project called default.
- The default project allows unlimited access to the HP AlphaServer SC system resources.

To control user access to resources, you can add the following information to the SC database:

- Users

  A user is identified to RMS by the same name as their UNIX account name (that is, the name associated with a UID). You must create user records in the SC database if you plan to create projects or apply access controls to individual users.

- Projects

  A project is a set of users. A user can be a member of several projects at the same time. Projects have several uses:

- A project is a convenient way to specify access controls for a large number of users — instead of specifying the same controls for each user in turn, you can add the users as members of a project and specify access controls on the project.
- Resource limits affect all members of the project as a group. For example, if one member of a group is using all of the resources assigned to the project, other members of the project will have to wait until the first user is finished.
- Accounting information is gathered on a project basis. This allows you to charge resource usage on a project basis.

Users specify the project they want to use by setting the RMS_PROJECT environment variable before using `allocate` or `prun`.

If a user is not a member of a project, by default they become a member of the `default` project.

- Access controls

  You can associate an access control record with either a user or a project. The access control record specifies the following:

  - The name of the project or user.
  - The partition to which it applies. The specified access control record applies to the partition of a given name in all configurations.
  - The priority the user or project should have in this partition.
  - The maximum number of CPUs the user or project can have in this partition.
  - The maximum memory the user or project can use in this partition.

There are two ways to create/modify/delete users, projects, and access controls:

- Use the *RMS Projects and Access Controls* menu (see Section 5.6.2.2 on page 5–38)

  This provides a menu interface similar to other Tru64 UNIX Sysman interfaces.

- Use the `rcontrol` command (see Section 5.6.2.3 on page 5–43)

  This provides a command line interface.

5.6.2.2 RMS Projects and Access Controls Menu

To create users and projects and to specify access controls, use the *RMS Projects and Access Controls* menu. You can access the *RMS Projects and Access Controls* menu by running the following command:

```
# sysman sra_user
```

You may also use the `sysman -menu` command. This presents a menu of Tru64 UNIX system nonmanagement tasks. You can access the *RMS Projects and Access Controls* menu in both the *Accounts* and AlphaServer SC *Configuration* menus.

Note:

When you use `sra_user` to change users, projects, or access controls, the changes only take effect when you reload (see Section 5.4.4 on page 5–13) or restart (see Section 5.4.5 on page 5–13 and Section 5.4.3 on page 5–12) the partition.

The *RMS Projects and Access Controls* menu contains the following buttons:

- Manage RMS Users...
  This allows you to add, modify, or delete users. You can assign access controls to users. You can specify the projects of which the user is a member.
- Manage RMS projects...
  This allows you to add, modify, or delete projects — including the `default` project. You can assign access controls to projects. You can also add and remove users from the project.
- Synchronize Users...
  This allows you to synchronize the UNIX user accounts with the SC database. Specifically, it identifies UNIX users who are not present in the SC database. It also identifies users whose UNIX accounts have been deleted. It then offers to add or delete these users in the SC database.

  The Synchronize Users menu is typically used to load many users into the SC database after the system is first installed. When the users have been added, you would typically add users to projects, and assign access controls, using the Manage RMS Users and Manage RMS projects menus.

Figure 5–1 shows an example RMS User dialog box.

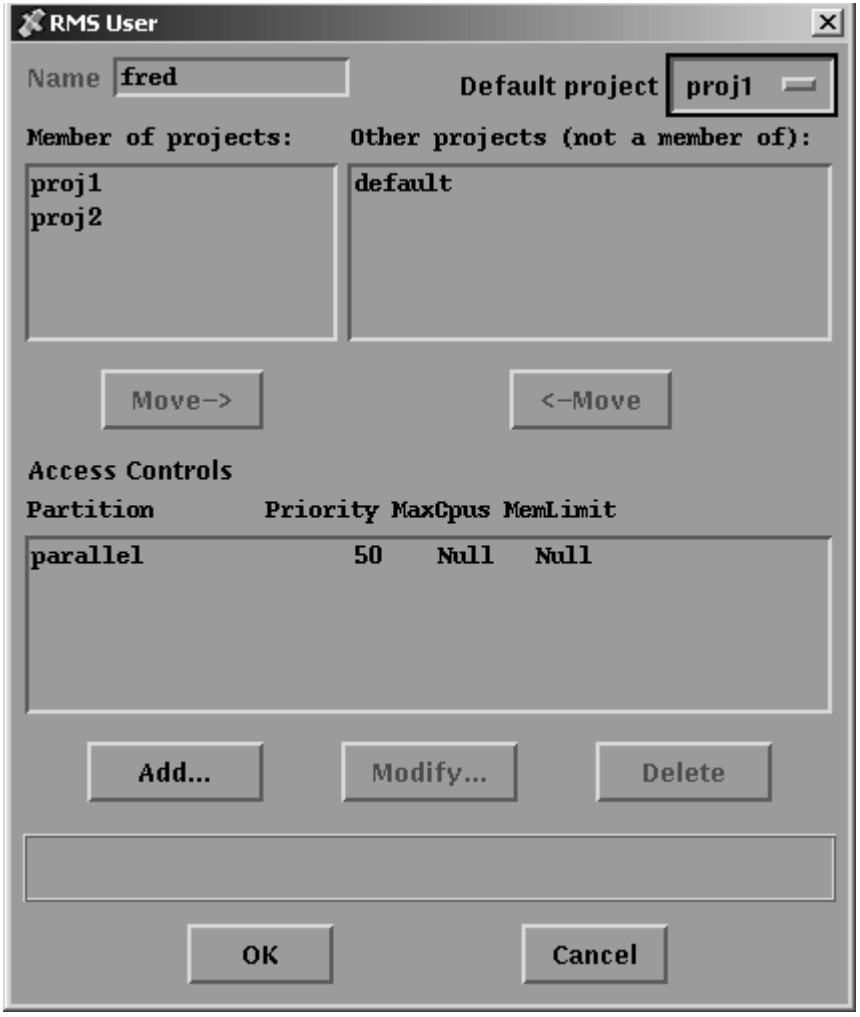

Figure 5–1 RMS User Dialog Box

Figure 5–2 shows an example Manage Partition Access and Limits dialog box.

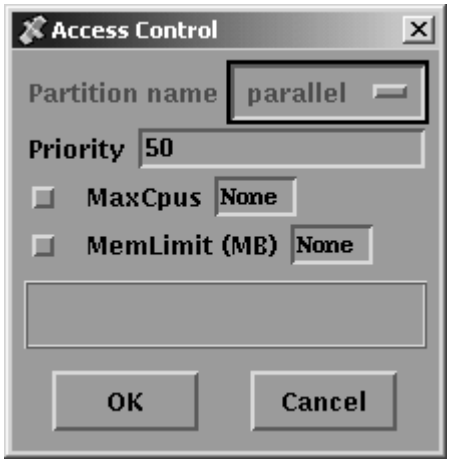

Figure 5–2 Manage Partition Access and Limits Dialog Box

This section describes how to use the *RMS Projects and Access Controls* menu to perform the following tasks:

- Apply memory limits (see Section 5.6.2.2.1 on page 5–41)
- Define the maximum number of CPUs (see Section 5.6.2.2.2 on page 5–42)

When you have used the *RMS Projects and Access Controls* menu, you should reload all partitions in the current configuration to apply your changes. See Section 5.4.4 on page 5–13 for information on how to reload partitions. Stopping and restarting partitions or restarting the configuration (see Section 5.4.5 on page 5–13 and Section 5.4.3 on page 5–12) will also apply your changes.

5.6.2.2.1 RMS Projects and Access Controls Menu: Applying Memory Limits

You can specify the memory limit of members of a project, or of an individual user for a given partition. To define memory limits by using the *RMS Projects and Access Controls* menu, perform the following steps:

1. Start the *RMS Projects and Access Controls* menu as follows:
   ```
   # sysman sra_user
   ```
2. Click on either the Manage RMS Users... button or the Manage RMS Projects... button.
3. Select the user or project to which you wish to apply the memory limit.
4. Click on the Modify… button. For a user, this displays the RMS User dialog box, as shown in Figure 5–1 on page 5–40.

5. Click on the Add... button. This displays the Access Control dialog box, as shown in Figure 5–2 on page 5–41.
6. Select the partition to which you wish to apply the limit.
7. Click on the MemLimit checkbox and enter the memory limit (in units of MB) in the field beside it.
8. Click on the OK button.
9. Click on the OK button to confirm the changes on the RMS User display.
10. This updates the SC database. To propagate your changes, reload the partition as described in Section 5.4.4 on page 5–13.

For more information about memory limits, see Section 5.7.2 on page 5–47.

5.6.2.2.2 RMS Projects and Access Controls Menu: Defining the Maximum Number of CPUs

For a given project or user, you can define the maximum number of CPUs that the user or project can use in a given partition. To do this using the *RMS Projects and Access Controls* menu, perform the following steps:

1. Start the *RMS Projects and Access Controls* menu as follows:
   ```
   # sysman sra_user
   ```
2. Click on either the Manage RMS Users... button or the Manage RMS Projects... button.
3. Select the user or project to which you wish to apply the limit.
4. Click on the Modify… button. For a user, this displays the RMS User dialog box as shown in Figure 5–1 on page 5–40.
5. Click on the Add... button. This displays the Access Control dialog box, as shown in Figure 5–2 on page 5–41.
6. Select the partition to which you wish to apply the limit.
7. Click on the MaxCpus checkbox and enter the maximum number of CPUs in the field beside it.
8. Click on the OK button.
9. Click on the OK button to confirm the changes on the RMS User display.
10. This updates the SC database. To propagate your changes, reload the partition as described in Section 5.4.4 on page 5–13.

For more information about maximum number of CPUs, see Section 5.7.4 on page 5–53.

5.6.2.3 Using the rcontrol Command

Use the `rcontrol` command to create users, projects, and access controls, as shown in the following examples:

```
# rcontrol create project=proj1 description='example 1'
# rcontrol create project=proj2 description='others'
# rcontrol create user=fred projects='proj1 proj2'
# rcontrol create access_control=proj1 class=project partition=big priority=60
# rcontrol create access_control=proj2 class=project partition=big priority=70
maxcpus=100 memlimit=null
```

When creating an object, some attributes must be specified while others are optional. If you omit an attribute, its value is set to `Null`. Table 5–8 shows the required and optional attributes for each type of object.

Table 5–8 Specifying Attributes When Creating Objects

Object Type	Attribute	Required/Optional	Description
`user`	`name`	Required	The name of the user
	`projects`	Optional	A list of projects that the user is a member of — the first project in the list is the user's default project
`project`	`name`	Required	The name of the project
	`description`	Optional	A description of the project
`access_controls`	`name`	Required	The name of a user or a project
	`class`	Required	Specifies whether the name attribute is a user or project name
	`partition`	Required	The name of the partition to which the access controls apply
	`priority`	Optional	The priority assigned to resources allocated by this user or project — a value of 0–100 should be specified
	`maxcpus`	Optional	The maximum number of CPUs that the user or project can allocate from the partition
	`memlimit`	Optional	The memory limit that applies to resources allocated by the user or partition

Use the `rcontrol` command to change existing users, projects, or access controls, as shown in the following examples:

```
# rcontrol set user=fred projects=proj1
# rcontrol set project=proj1 description='a different description'
# rcontrol set access_control=proj1 class=project priority=80 partition=big
maxcpus=null
```

Use the `rcontrol` command to delete a user, project, or access control, as shown in the following examples:

```
# rcontrol remove user=fred
# rcontrol remove project=proj1
# rcontrol remove access_control=proj1 class=project partition=big
```

Each time you use `rcontrol` to manage a user, project, or access control, all running partitions are reloaded so that the change takes effect.

This section describes the following tasks:

- Using the rcontrol Command to Apply Memory Limits (see Section 5.6.2.3.1 on page 5–44)
- Using the rcontrol Command to Define the Maximum Number of CPUs (see Section 5.6.2.3.2 on page 5–44)

When you have used the *RMS Projects and Access Controls* menu, you should reload all partitions in the current configuration to apply your changes. See Section 5.4.4 on page 5–13 for information on how to reload partitions. Stopping and restarting partitions or restarting the configuration (see Section 5.4.5 on page 5–13 and Section 5.4.3 on page 5–12) will also apply your changes.

5.6.2.3.1 Using the rcontrol Command to Apply Memory Limits

You can specify the memory limit of members of a project, or of an individual user for a given partition. The following example shows how to use the `rcontrol` command to set a memory limit of 1000MB on the `big` partition:

```
# rcontrol set partition=big configuration=day memlimit=1000
```

For more information about memory limits, see Section 5.7.2 on page 5–47.

5.6.2.3.2 Using the rcontrol Command to Define the Maximum Number of CPUs

For a given project or user, you can define the maximum number of CPUs that the user or project can use in a given partition. The following example shows how to use the `rcontrol` command to set the maximum number of CPUs for the `proj1` project to 4:

```
# rcontrol set access_control=proj1 class=project priority=80 partition=big
maxcpus=4
```

For more information about maximum number of CPUs, see Section 5.7.4 on page 5–53.

5.6.3 Controlling Whether the Partition Uses HP AlphaServer SC Elan Adapter Cards

In HP AlphaServer SC Version 2.6, you can control whether individual partitions use HP AlphaServer SC Elan adapter cards. If the `use_elan` field is set to 0 (zero) for a partition, RMS does not attempt to initialize HP AlphaServer SC Elan adapter cards on the nodes in the partition, nor does it create Elan adapter card capabilities for parallel applications. This control replaces the use of the `ignore-elan` attribute in earlier releases. This attribute can be set as follows:

```
# rcontrol set partition parallel use_elan 0
```

You must restart RMS on all nodes in the partition for this change to take effect.

To remove the control on partitions using HP AlphaServer SC Elan adapter cards, stop the partition and use the `rcontrol` command to set the value of the `use_elan` field to 1, as shown in the following example:

```
# rcontrol set partition parallel configuration day use_elan 1
```

You must restart RMS for this change to take effect.

It is also no longer necessary for all nodes in an RMS system to have HP AlphaServer SC Elan adapter cards. Individual (management) nodes that are not part of a partition should be declared using the `-I` option with the `rmsbuild` command. The nodes in a partition must either all have HP AlphaServer SC Elan adapter cards or all not have them. Set the `use_elan` field in the `partitions` table to 0 for nodes that do not have HP AlphaServer SC Elan adapter cards, as shown in the following example:

```
# rcontrol create partition p1 nodes 'atlas[20-23]' use_elan 0
```

5.7 Controlling Resource Usage

The information in this section is organized as follows:

- Resource Priorities (see Section 5.7.1 on page 5–46)
- Memory Limits (see Section 5.7.2 on page 5–47)
- Minimum Number of CPUs (see Section 5.7.3 on page 5–53)
- Maximum Number of CPUs (see Section 5.7.4 on page 5–53)
- Time Limits (see Section 5.7.5 on page 5–55)
- Enabling Timesliced Gang Scheduling (see Section 5.7.6 on page 5–56)
- Partition Queue Depth (see Section 5.7.7 on page 5–59)
- Processor Sets (see Section 5.7.8 on page 5–60)

Note:

If you make configuration changes, they will only take effect when a partition is started, so you must stop and then restart the partition (see Section 5.4.5 on page 5–13).

5.7.1 Resource Priorities

When a resource request is made, RMS assigns a priority to it. The priority is represented by an integer number in the range 0–100. Unless you determine otherwise, the default priority is 50. Priorities are used when allocating resources — higher priority requests are granted before lower priority requests.

Priority is determined by access controls associated with either a user or a project. You specify access controls by using either the *RMS Projects and Access Controls* menu (see Section 5.6.2 on page 5–37) or the `rcontrol` command (Section on page 5–42).

The priority of a resource is initially determined by the user's priority. This is determined by the following precedence rules:

1. If the user has an access record for the partition being used, the access record sets the priority. The remaining precedence rules are not used.
2. If the user is a member of a project and the project has an access record, the project access record determines the priority. The remaining precedence rules are not used.
3. If the user is not a member of a project, the `default` project access record (if it exists) determines the priority.
4. If the `default` project has no access record for the partition, the priority is set to the value of the `default-priority` attribute, which is initially set to 50.

You can change the value of the `default-priority` attribute by using the `rcontrol` command, as shown in the following example:

```
# rcontrol set attribute name=default-priority val=0
```

After a resource has been assigned its initial priority, you can change the priority by using the `rcontrol` command, as shown in the following example:

```
# rinfo -rl
RESOURCE          CPUS    STATUS          TIME            USERNAME      NODES
big.916              1    allocated       00:14           fred          atlas0
# rcontrol set resource=916 priority=5
```

Priorities are associated with resources, not with jobs, so do not use a job number in the `rcontrol set resource` command. If you do so, you may affect an unintended resource. You cannot change the priorities of resources that have finished.

The `root` user can change the priority of any resource. Non-`root` users can change the priority of their own resources. A non-`root` user cannot increase the initial priority of a resource.

When scheduling resource requests, RMS first considers resource requests of higher priority. Although CPUs may already be assigned to low priority resources, the same CPUs may be assigned to a higher priority resource request — that is, the higher priority resource request preempts the lower priority allocated resource. When this happens, the higher priority resource has an `allocated` status and the lower priority resource has a `suspended` status. When RMS suspends a resource (that is, puts a resource into a `suspended` state), all jobs associated with the resource are also suspended. A SIGSTOP signal is sent to the processes associated with the jobs.

Resources of higher priority do not always preempt lower priority resources. They do not preempt if allocating the resource would cause the user to exceed a resource limit (for example, maximum number of CPUs or memory limits). In this case, the resource is blocked.

5.7.2 Memory Limits

This section provides the following information about memory limits:

- Memory Limits Overview (see Section 5.7.2.1 on page 5–47)
- Setting Memory Limits (see Section 5.7.2.2 on page 5–49)
- Memory Limits Precedence Order (see Section 5.7.2.3 on page 5–49)
- How Memory Limits Affect Resource and Job Scheduling (see Section 5.7.2.4 on page 5–50)
- Memory Limits Applied to Processes (see Section 5.7.2.5 on page 5–51)

5.7.2.1 Memory Limits Overview

Parallel programs need exclusive access to CPUs. Much of the focus on RMS elsewhere in this chapter is about how RMS manages the CPUs in nodes as a resource that is available for allocation to programs. However, programs also need another resource: memory. RMS can manage the memory resource of nodes. You specify how memory is to be managed using memory limits (`memlimit`).

The following aspects of RMS are involved with memory limits:

- When RMS receives a resource request, a memory limit can be assigned to the resource request. This is done by configuring the partition or setting access controls, or the user can use the RMS_MEMLIMIT environment variable.

- The memory limit forces RMS to consider the memory requirements of the resource request when scheduling the resource. This means that RMS must find nodes with enough memory and swap space to satisfy the resource request.
- RMS starts the processes associated with the resource with memory limits. The processes cannot exceed these memory limits. If the processes exceed their memory limit, the process is terminated or their memory request is denied.

Memory limits are useful for the following purposes:

- Memory limits allow fair sharing of resources between different users. By specifying memory limits, you can prevent one user's processes from taking all of memory. This means that memory is shared fairly among users on the same node.
- You can control the degree of swapping. By setting a low memory limit, you can ensure that processes cannot consume so much memory that they need to swap. If some processes swap, the overhead impacts all users of the node — even if they are not swapping.

If you are running applications with different characteristics, you should consider dividing your nodes into different partitions so that you can have different memory limits on each partition. For example, you might want to have a partition with high memory limits for processes that consume large amounts of memory and where you allow swapping, and a different partition with a smaller memory limit for processes that do not normally swap.

Note:

Do not set the memory limit to a value less than 200MB. If you do so, you can prevent the core file analysis process from performing normally. See Section 5.5.9 on page 5–25 for a description of core file analysis.

Memory limits can be enabled in several ways, as follows:

- You can specify a memory limit for the partition.
- You can specify a memory limit in the access controls for a user or project for a given partition.
- A user can use the RMS_MEMLIMIT environment variable.

The memory limit that RMS uses is associated with a node's CPU. The memory limit that applies to a process is indirectly derived from the number of CPUs that are associated with the process. For example, if a node has 4GB of memory and 4 CPUs, an appropriate memory limit might be 1GB. This allows various combinations of resource allocations where each

CPU has 1GB of memory associated with it. For example, four users could each be using 1 CPU (and 1GB of memory assigned to each) or two users could have 2 CPUs (with 2GB of memory assigned to each).

Since the memory limit applies to CPUs, users must consider this when allocating resources so that their processes have appropriate memory. In the example just described, if a user has one process per CPU, each process has a memory limit of 1GB. If a user wants to run a process that needs 4GB of memory, they should assign 4 CPUs to the process — even if the process only runs on one CPU. Although it appears that three CPUs are idle, in fact they cannot do useful work anyway since all of the node's memory is devoted to the user's process. Allowing other processes to run would in fact overload the node because their memory usage could cause the node to swap — adversely affecting performance.

A user can determine if memory limits apply by using the `rinfo -q` command and by examining the RMS_MEMLIMIT environment variable.

5.7.2.2 Setting Memory Limits

You can specify a memory limit for a partition by setting a value for the `memlimit` field in the partition's record in the `partitions` table. The value is specified in units of MB. Once set, this sets a default limit for all users of the partition.

You may also specify a memory limit using access controls. This allows you to specify the memory limit of members of a project, or of an individual user for a given partition. To create access controls that specify memory limits, use either the *RMS Projects and Access Controls* menu (see Section 5.6.2.2 on page 5–38) or the `rcontrol` command (Section 5.6.2.3 on page 5–43).

A user can also specify a memory limit by setting the RMS_MEMLIMIT environment variable. If memory limits were not otherwise enabled, this sets a memory limit for subsequent `allocate` or `prun` commands. If memory limits apply (either through partition limit or through access controls), the value of RMS_MEMLIMIT must be less than the memory limit. If a user attempts to set their memory limit to a larger value, an error message is displayed, similar to the following:

```
prun: Error: can't allocate 1 cpus: exceeds usage limit
```

5.7.2.3 Memory Limits Precedence Order

The memory limit that applies to a resource request is derived as follows (in precedence order):

1. The RMS_MEMLIMIT environment variable (if present) specifies the memory limit, and overrides other ways of specifying the memory limit. However, the RMS_MEMLIMIT environment variable can only be used to lower the limit that would otherwise apply — you cannot raise your memory limit. If you attempt to raise your

memory limit by setting RMS_MEMLIMIT to a value greater than your allocated memory, all jobs that you attempt to run will fail (due to insufficient memory limits) until the RMS_MEMLIMIT value is lowered or removed.

2. If the user has an access control record, the memory limit in the access control record applies. If the value is `Null`, no memory limits apply to the resource. If the value is not `Null` (that is, if the value is a number), the memory limit for the resource uses this value (unless overridden as described in rule 1).
3. If the user is a member of a project, and the project has an access control record, the memory limit in the access control record applies. If the value is `Null`, no memory limits apply to the resource. If the value is not `Null` (that is, if the value is a number), the memory limit for the resource uses this value (unless overridden as described in rule 1).
4. If the user's project has no access control record, the access control record of the default project applies.
5. If the default project has an access control record, the memory limit in the access control record applies. If the value is `Null`, no memory limits apply to the resource. If the value is not `Null` (that is, if the value is a number), the memory limit for the resource uses this value (unless overridden as described in rule 1).
6. If no access control records apply, the `memlimit` field in the partition's record in the `partitions` table applies. If this value is `Null`, no memory limits apply to the resource. If the value is not `Null` (that is, if the value is a number), the memory limit for the resource uses this value (unless overridden as described in rule 1).

5.7.2.4 How Memory Limits Affect Resource and Job Scheduling

When a node is allocated to a resource, the processes associated with the resource consume memory. For resources with an associated memory limit, RMS knows how much memory that resource can consume. RMS can thus determine whether allocating more resources on that node would overload the available memory. The available memory of a node is composed of physical memory and swap space. However, RMS does not use this data — instead, it uses the `max-alloc` attribute. When finding nodes to allocate to a resource, RMS totals the memory limits of all of the resources already assigned to the node; if the new request causes the memory usage to exceed the `max-alloc` value, RMS will not use the node.

In scheduling terms, this becomes noticeable when different priorities or timesliced gang scheduling is used. Normally, you would expect a high-priority or timesliced resource to force the suspension of lower priority or older resources. However, if allocating the higher priority or timesliced resource would cause `max-alloc` to be exceeded, RMS will not allocate these resources. Instead, the resource is put into the `queued` state. The resources will

remain queued until other resources are deallocated, thus freeing memory. Resource priorities are described in Section 5.7.1 on page 5–46; time limits are described in Section 5.7.5 on page 5–55.

RMS uses the `max-alloc` attribute instead of examining the size of physical memory or swap space because this allows you to tune your system for the type and characteristics of the applications that you run. For example, if `max-alloc` is close to the physical memory size, RMS will not schedule jobs that would cause the system to swap. Conversely, if `max-alloc` is closer to the swap space size, more processes can run on the system.

You can change the value of `max-alloc` by using the `rcontrol` command, as shown in the following example:

```
# rcontrol set attribute name=max-alloc val=3800
```

If all of the nodes in the system have the same memory size, a single `max-alloc` attribute can be used. However, if different nodes have different sized memory, you can create several `max-alloc` attributes. The name of the attribute is `max-alloc-`*`size`*, where *`size`* is the amount of memory in gigabytes. For example, `max-alloc-8` and `max-alloc-16` specify the `max-alloc` for nodes of 8GB and 16GB memory respectively.

You can create the `max-alloc-8` attribute by using the `rcontrol` command, as shown in the following example:

```
# rcontrol create attribute name=max-alloc-8 val=5132
```

You can determine the amount of memory and swap space of each node by using the `rinfo -nl` command.

5.7.2.5 Memory Limits Applied to Processes

In HP AlphaServer SC Version 2.6, the `set-vm-limit` and `inherit-user-limits` attributes are supported. These attributes are not added automatically when HP AlphaServer SC is installed. You can create these attributes and set their values as in the following example:

```
# rcontrol create attribute=attribute-name val=value
```

If the `set-vm-limit` attribute exists and its value is set to 0 (zero), then all application processes run with their virtual memory limit set to the maximum for the node they are running on. Lower limits (such as those imposed by the RMS_MEMLIMIT environment variable, or inherited from `prun`) are not applied.

If the `set-vm-limit` attribute does not exist, or if its value is set to 1, then all application processes run with their virtual memory limit set to the maximum for the node they are running on. However, lower limits imposed by the RMS_MEMLIMIT environment variable can be applied. If the `inherit-user-limits` attribute is set to 1 (one) and the `set-vm-limit` attribute is set to 1 (one), then a lower limit can be inherited from the `prun` process.

If the `inherit-user-limits` attribute exists and its value is set to 1 (one), then all application processes run with the virtual memory, data, stack, and maximum memory size and limits inherited from the `prun` process, unless RMS has imposed a value.

If the `inherit-user-limits` attribute does not exist, or if its value is set to 0 (zero), then all application processes run with the virtual memory, data, stack, and maximum memory size and limits set to the maximum values for the node they are running on, unless RMS has imposed a value.

If neither of these attributes exist, the virtual memory size will take RMS_MEMLIMIT values, as in HP AlphaServer SC Version 2.5.

Table 5–9 shows how the `set-vm-limit` attribute, the `inherit-user-limits` attribute, and the RMS_MEMLIMIT environment variable work together.

Table 5–9 RMS Limits

RMS_MEMLIMIT set?	inherit-user-limits	set-vm-limit	Data, Stack, Max. Memory Size	Virtual Memory Size
No	0	0	Max. for Node[1]	Max. for Node
No	0	1	Max. for Node	Max. for Node
No	1	0	Inherit from `prun`[2]	Max. for Node
No	1	1	Inherit from `prun`	Inherit from `prun`
Yes	0	0	RMS-MEMLIMIT	Max. for Node
Yes	0	1	RMS-MEMLIMIT	RMS-MEMLIMIT
Yes	1	0	RMS-MEMLIMIT	Max. for Node
Yes	1	1	RMS-MEMLIMIT	RMS-MEMLIMIT

[1]The maximum limit on the node where RMS starts the user's program.

[2]The limit as currently set in the user shell where `prun` was executed.

When memory limits apply, RMS uses `setrlimit(2)` to specify the memory limits so that the stack, data, and resident set sizes are set to the memory limit. You can see the effect of this by setting RMS_MEMLIMIT, and running `ulimit(1)` as a job, as follows:

```
# setenv RMS_MEMLIMIT 200
# prun -n 1 /usr/ucb/ulimit -a
time(seconds)        unlimited
file(blocks)         unlimited
data(kbytes)         204800
```

```
stack(kbytes)         204800
memory(kbytes)        204800
coredump(blocks)      unlimited
nofiles(descriptors) 4096
vmemory(kbytes)       262144
```

With memory limits, a process may fail to start, may fail to allocate memory, or may overrun its stack. The exact behavior depends on whether the program exceeds the data or stack limits. The following behavior may be observed:

- `prun` reports that all processes have exited with a status of 1. This does not always indicate that the program has exceeded its data segment limit. An exit status of 1 can also indicate that a shared library cannot be found.
- A program can get an ENOMEM error from `malloc(3)` or similar function. The reaction of the program is program-specific but if the error is not handled correctly could result in a segment violation.
- A program may attempt to exceed its stack limit. If this happens, a SIGSEGV is generated for the process.

5.7.3 Minimum Number of CPUs

You can specify the minimum number of CPUs that should apply to resource allocations with a given partition. An attempt to allocate fewer CPUs would be rejected with a message such as the following:

```
prun: Error: can't allocate 1 cpus on 1 nodes: min cpus per request is 2
```

This feature is useful in the following cases:

- Reserves the partition for genuine parallel jobs.
- Excludes small-scale parallel jobs.
- Reduces possible fragmentation of the partition where many small jobs prevent large jobs from running.

To enable this feature, set the `mincpus` attribute of the partition, as shown in the following example:

```
# rcontrol set partition=big configuration=day mincpus=16
```

5.7.4 Maximum Number of CPUs

You can control the maximum number of CPUs that a user or project can allocate at any given time. Once the user or project attempts to allocate more than the permitted number of CPUs, the allocate request is blocked.

The maximum number of CPUs that a user or project can use is specified using access controls. To create access controls, use either the *RMS Projects and Access Controls* menu (see Section 5.6.2.2 on page 5–38) or the `rcontrol` command (see Section 5.6.2.3 on page 5–43).

The maximum number of CPUs for a given user can be specified in access control records for a specific individual user or a project or both. The maximum numbers are determined as follows:

- If the individual user has an access control record that has a non-Null maximum number of CPUs, that limit applies to the individual user.
- The user is always a member of a project: either a specific project or the default project. If that project has an access control record with a non-Null maximum number of CPUs, that limit applies to the project.
- If no access controls apply (or they have a Null value), the maximum number of CPUs for the user's project is set to the number of CPUs in the partition; that is, the total number of CPUs on all configured-in nodes in the partition.

The maximum number of CPUs limit can be set to a higher value than the actual number of CPUs available in the partition. This is only useful if you are using timesliced gang scheduling — in effect, it allows several resources and jobs of a given partition to timeslice with each other. See Section 5.7.6 on page 5–56 for more details.

When applying the maximum number of CPUs limit to an individual user, RMS counts the number of CPUs that the user is using on the partition. If the user attempts to allocate more CPUs than the limit, the request is blocked. The request remains blocked until the user frees enough CPUs to allow the request to be granted. The CPUs can be freed either because the resource allocation finishes or because the user suspends their resource (see Section 5.5.3 on page 5–20).

When applying the maximum number of CPUs limit to a project, RMS counts the number of CPUs that all members of the project are using on the partition. If any individual member of the project attempts to allocate more CPUs than the limit, the request is blocked. The request remains blocked until some members of the project free enough CPUs to allow the request to be granted. The CPUs can be freed either because the resource allocations finish or because the user or users suspend their resources (see Section 5.5.3 on page 5–20).

When a user has both an individual access control and is a member of a project that has an access control that specifies the maximum number of CPUs, RMS will not allow the user to exceed either limit. In practical terms, this means that it is not useful to set the maximum number of CPUs for an individual user higher than the project's limits. This is because the user will reach the project limit before exceeding the individual limit.

Setting an individual limit can be useful in the following situations:

- You want to prevent a user from using all of the CPUs that are available to a project. This allows other project members to access the CPUs.

- A user is a member of several projects. In this situation, the user can make several resource requests, each using a different project. An individual limit can control the total number of CPUs that the user has access to, regardless of what project they use.

When counting CPU usage, RMS does not count resources that have been suspended using `rcontrol`. However, resources that are suspended by RMS itself (that is, suspended by the scheduler) are counted. A resource is suspended by RMS when a high-priority resource request preempts a lower-priority resource and when timesliced gang scheduling is enabled.

Note:

`rinfo -q` shows the number of CPUs in use by individuals and projects. However, it assumes that suspended resources have been suspended by `rcontrol`. It does not correctly count CPU usage when resources are suspended by preemption or timesliced gang scheduling.

5.7.5 Time Limits

You can impose a time limit on resources by setting the `timelimit` attribute of the partition. The `timelimit` is specified in units of seconds and is elapsed time.

To set the `timelimit` field for a partition, use the `rcontrol` command as shown in the following example:

```
# rcontrol set partition=small configuration=day timelimit=30
```

If a resource is still allocated, its `timelimit` field is normally `Null`. However, if a timelimit applies to the resource, this field contains that time at which the resource will reach its timelimit. When the resource is finished (freed), this field contains the time at which the resource was deallocated.

The time limit is applied against the time that a resource spends in the `allocated` state. If the resource is suspended, the effective time limit on the resource is extended to account for the time while the resource is suspended.

When the time limit on a resource expires, RMS sends a SIGXCPU to all processes belonging to the resource, and `prun` may or may not print the following message to indicate that the time limit has expired (where `a.out` is an example command):

```
prun: timelimit expired for a.out
```

If the user has used `allocate` and then uses `prun`, the time limit may have already expired when an attempt is made to use `prun`. In this case, `prun` prints a message similar to the following:

```
prun: Error: failed to start job: cputime exceeded
```

5.7.6 Enabling Timesliced Gang Scheduling

Normally, when RMS allocates a resource, the resource remains in control of the nodes and CPUs allocated to the resource until the resource finishes. If many nodes and CPUs are already allocated to resources, subsequent resource requests (at the same priority) enter a queue and wait there until nodes and CPUs become free.

With timesliced gang scheduling, RMS re-examines the currently allocated and queued resources on a periodic basis. The `timeslice` field of a partition determines the period. When RMS examines the currently allocated and queued resources, it can suspend the resources that are currently allocated and allocate their nodes and CPUs to some of the queued resource requests. At the next timeslice period, RMS again examines the allocated, suspended, and queued resources. It will unsuspend (that is, allocate) resources that it had previously suspended, and suspend currently allocated resources. During each timeslice, resources are alternately allocated and suspended.

When RMS suspends a resource, all jobs associated with the resource are suspended. Therefore, while resources alternate between `allocated` and `suspended`, their associated jobs alternate between `running` and `suspended`.

RMS uses timesliced gang scheduling for resources of the same priority. Resources of different priorities are not timesliced — instead, the higher priority resource is allocated, forcing currently allocated lower-priority resources to be suspended.

To enable timesliced gang scheduling, set the `timeslice` attribute of the partition to the number of seconds in the desired timeslice period, as shown in the following example:

```
# rcontrol set partition=big configuration=day timeslice=100
```

To disable timesliced gang scheduling, set the `timeslice` attribute of the partition to `Null`, as shown in the following example:

```
# rcontrol set partition=big configuration=day timeslice=Null
```

To propagate the changes, (re)start the partition.

As mentioned above, RMS will allocate (that is, start timeslicing) requests that are queued. However, it will not allocate requests that are blocked. The effect is that a user can submit requests that are started in timeslice mode until that user reaches the maximum number of CPUs limit, at which point the user's requests are blocked. The requests remains blocked until one or more of the allocated requests finishes. As the allocated requests finish, the blocked requests will unblock, allowing them to start timeslicing.

To make effective use of timesliced gang scheduling, organize your users into appropriate project groupings and use access controls, so that you can determine how many requests can timeslice before requests become blocked. The factors in this organization are as follows:

- Maximum number of CPUs
  As explained in Section 5.7.4 on page 5–53, this is specified by access controls. Once a user reaches this limit, allocation requests become blocked and hence do not timeslice. You can set this limit to a number that is larger than the number of actual CPUs in the partition. For example, if the limit is set to twice the number of CPUs in the partition, a user can run two jobs at the same time where each job has allocated all CPUs. (As each job must alternate with the other job, the overall execution time is roughly the same as if one job ran serially after the other). If you do not specify the maximum number of CPUs, the effective limit is set to the number of actual CPUs in the partition.
- Memory limits
  As explained in Section 5.7.2 on page 5–47, this is specified either for the partition or by access controls. If a user's `allocate` request would cause the memory limit to be exceeded, the request is blocked and hence not timesliced. Since memory limits are closely associated with CPUs, the memory limit and maximum number of CPUs must be coordinated. For example, if the maximum number of CPUs is set to twice the number of actual CPUs in the partition, an appropriate memory limit is half of what you would have used if timeslice was not enabled. That is, if the `max-alloc` attribute is 4GB, a limit of 512MB is appropriate. If you do not reduce the memory limit (for example, if you use 1GB), `allocate` requests queue because of memory limits before they block because of the maximum number of CPUs limit. There is more information about why you should use memory limits in conjunction with timesliced gang scheduling later in this section.
- Projects
  RMS counts CPU usage by all users in a given project. When a given user makes a resource request, it is possible that other members of the project are already using so many CPUs that the request is blocked (by the maximum number of CPUs limit). Since RMS does not timeslice gang-schedule blocked resources, timeslicing will not allow resource requests by members of the same project to alternate in timesliced gang-schedule mode unless the maximum number of CPUs limit is larger than the actual number of CPUs in the partition. If the maximum number of CPUs limit is equal to or smaller than the actual number of CPUs in a partition, requests from the users of the same project will not timeslice but will timeslice with requests from users in other projects. By default, unless you specify otherwise, all users belong to the default project. This means that you must put all users into different projects. Your grouping of users depends on your local policy and management situation.

Use the *RMS Projects and Access Controls* menu (see Section 5.6.2.2 on page 5–38) or the `rcontrol` command (see Section 5.6.2.3 on page 5–43) to assign users to projects and to specify access controls for partitions.

In timesliced gang-schedule mode, priorities still have an effect. Resources of high priority are scheduled before resources of lower priority. In effect, a resource timeslices with resources of the same priority: if a lower-priority resource exits, it will not be allocated during a given timeslice while higher-priority requests are using its CPUs. It is possible for high-priority and low-priority resources to be allocated and timeslice at the same time. However, this only happens if the high-priority resources are using a different set of CPUs to the lower priority resources.

The combined effect of different priorities and timeslice can produce very complex situations. Certain combinations of resource requests (and job duration) can cause all requests from an individual user to be allocated to the same set of CPUs. This means that this user's resources timeslice among themselves instead of several different users' resources timeslicing among each other. A similar effect can occur for projects of different priorities (where members of the same project timeslice among themselves instead of among users from different projects). For this reason, you should observe the following recommendations when you are using timesliced gang scheduling:

- Reserve high priorities for exceptions.
- Carefully consider a user's request pattern before using access controls to grant the user a maximum number of CPUs greater than the partition size. In most situations, this may only be justified if the user typically requests most of the CPUs in a partition per resource request.
- A similar situation exists for access controls for projects. Only grant a high maximum number of CPUs for a project if you do not have many projects (that is, if almost all of your users are in one project).

In timesliced gang-schedule mode, RMS will allocate the same CPUs and nodes to several resource requests and their associated jobs and processes. Of course, at any given time, only one resource is in the `allocated` state; the others are in the `suspended` state.

However, while this ensures that different processes are not competing for a CPU, it does not prevent the processes from competing for memory and swap space.

At each timeslice period, processes that were running are suspended (sent a SIGSTOP signal) and other processes are resumed (sent a SIGCONT signal). The resumed processes start running. As they start running, they may force the previously running processes to swap — that is, the previously running processes swap out, and the resumed processes swap in. Clearly, this has an impact on the overall performance of the system. You can control this using memory limits. In effect, memory limits allow you to control the degree of concurrency; that is, the number of jobs that can operate on a node or CPU at a time.

The degree of control of concurrency provided by memory limits also has a significant impact on how resource allocations are distributed across the system. Unless you use memory limits to control concurrency, it is possible that many resources will end up timeslicing on one node. This is more noticeable in the following cases:

- When resource request sizes are small compared to the maximum number of CPUs limit (which defaults to the number of CPUs in the partition). This causes problems because the users making the requests will not reach their maximum number of CPUs limit; hence, each request is eligible for timeslicing.
- When resources are long-running and many requests are queued while few resources finish. This is because, as resources are not finishing, there are no free CPUs. If there are free CPUs, RMS uses them in preference to CPUs already in use. However, when all CPUs are in use, RMS allocates each request starting on the first node in the partition.

For these reasons, you are strongly recommended to use memory limits in conjunction with timesliced gang scheduling.

You are also recommended to use the `memsplit-enabled` attribute (new in HP AlphaServer SC Version 2.6) when using memory limits in conjunction with timesliced gang scheduling. If the `memsplit-enabled` attribute is created and its value is set to 1 (one), RMS categorizes free CPUs and nodes into groups where the nodes all have the same amount of free memory. A request for new CPUs is then restricted to only those nodes that have enough free memory.

To create and set the `memsplit-enabled` attribute, use the `rcontrol` command as follows:

```
# rcontrol create attribute name=memsplit-enabled val=1
```

Memory limits are described in Section 5.7.2 on page 5–47.

5.7.7 Partition Queue Depth

You can constrain the maximum number of resource requests that a partition manager will handle at any given point in time. The limit is controlled by the `pmanager-queuedepth` attribute. If not present, or if set to `Null` or `0` (zero), no queue depth constraint applies. By default, the `pmanager-queuedepth` attribute is set to zero. You can modify the value of the `pmanager-queuedepth` attribute as shown in the following example:

```
# rcontrol set attribute name=pmanager-queuedepth val=20
```

You must reload all partitions (see Section 5.4.4 on page 5–13) for the change to take effect.

Note the following points about using the pmanager-queuedepth attribute:

- Once the pmanager-queuedepth attribute is set, resources that would otherwise appear in the blocked state no longer appear in the database and are not shown by the rinfo command.
- Once the maximum number of resources — as specified by pmanager-queuedepth — exist (that is, resources in the queued, blocked, allocated, or suspended states), subsequent prun or allocate requests will either fail (if the immediate option is specified), or back off (that is, prun will ask pmanager to allocate the request, but will be rejected; the user's prun continues to run, but rinfo does not show a corresponding resource — not even blocked or queued):
  - If prun is run with the -v option, the following message is printed:
    ```
    prun: Warning: waiting for queue space
    ```
  - If prun is not run with the -v option, the user will not understand why their request is not shown by the rinfo command.
- As a resource finishes, it allows a request to be accepted (because the finishing resource brings the system below the pmanager-queuedepth value). However, the request that is accepted is selected randomly from the backed-off requests — there is no "queue" of requests.
- There is no way to see how many users have requests that have exceeded the queue depth.

5.7.8 Processor Sets

Processor sets allow processes to be bound to groups of CPUs. Support for the Tru64 UNIX processor set has been added to RMS in HP AlphaServer SC Version 2.6.

Processor set support ensures that applications are restricted to the allocated CPUs. This is important when a node is being shared between multiple users, especially when the users are running multi-threaded applications. If the processor-sets attribute is set to 1, RMS creates processor sets when a node is shared between two or more resource requests. The CPUs allocated by RMS are assigned to these processor sets and jobs are bound to them.

Processor set creation is minimized using the following methods:

- Where possible, CPUs are assigned to an existing processor set on allocation of the CPUs to a resource request. If necessary, a new processor set is created.
- CPUs are removed from the processor set if the resource request is suspended, and restored if it is resumed.

- CPUs are removed from the processor set when the resource request is deallocated. The processor set is added to a free list at this point, thus minimizing processor set create requests.

Processor set support is enabled when the `processor-sets` attribute exists and is set to 1, as follows:

```
# rcontrol create attribute=processor-sets val=1
```

You must restart RMS for this change to take effect.

5.8 Node Management

This section describes the following node management topics:

- Configure Nodes In or Out (see Section 5.8.1 on page 5–61)
- Booting Nodes (see Section 5.8.2 on page 5–63)
- Shutting Down Nodes (see Section 5.8.3 on page 5–63)
- Node Failure (see Section 5.8.4 on page 5–64)
- Draining Nodes (see Section 5.8.5 on page 5–66)

5.8.1 Configure Nodes In or Out

When you create the SC database, by default RMS assumes that the nodes in the database are available for use. However, from time to time, a node may not be in a usable state or you may wish to prevent RMS from using the node. You indicate the latter by configuring the node out. Configure nodes out using the `rcontrol configure out` command, as shown in the following example:

```
# rcontrol configure out nodes='atlas[0-1]' reason='to replace fan6'
```

When you start partitions that have configured-out nodes, those nodes will not be used to run parallel jobs. When you configure out a node, `rinfo` no longer shows that node in its partition, and no longer counts the node's CPUs in the total number of CPUs for the partition. (For the `root` partition, `rinfo` displays the total number of CPUs, including those of configured-out nodes.) You can determine which nodes are configured out by using the `rinfo -nl` command, as shown in the following example:

```
# rinfo -nl
running                 atlas[2-31]
configured out          atlas[0-1]
.
.
.
REASON CONFIGURED OUT
'to replace fan6'                                   atlas[0-1]
```

When you configure out a node, RMS in effect ignores the node. This has implications for various RMS functions, as follows:

- The status of the node is `configured out` — you cannot tell from the status whether the node is running or halted.
- The `configured out` status applies to all configurations (that is, a node may be a member of partitions in different configurations — it is configured out of all of these partitions).
- As described in Section 5.6.1 on page 5–35, RMS runs the `pstartup.OSF1` script to control interactive login to partitions. When a node is configured out, no actions are taken on the node. This means that the `/etc/nologin_`*`hostname`* file is untouched by starting a partition and remains in the same state as it was before the node was configured out.

To start using the node again, configure the node in using the `rcontrol configure in` command, as shown in the following example:

```
# rcontrol configure in nodes='atlas[0-1]'
```

It is not necessary to stop a partition before configuring out a node. Instead, when you configure out the node, the partition will briefly block and then resume running without the node. As explained in Section 5.5.10.2 on page 5–32, any resources or jobs running on that node will be cleaned up, and their status will be set to `failed`. You will be reminded of this when you issue the command to configure out the node, as follows:

```
# rcontrol configure out atlas0

rcontrol: will kill user jobs. Continue (y/n) :n
```

When you configure in a node, the partition status briefly changes to `blocked` and the node status is `unknown`. Seconds later, the node status should change to `running` and the partition returns to the `running` state. The node is now included in the partition and is available to run jobs. If RMS is unable to operate on the node (for example, if the node is not actually running), the node status changes from `unknown` to `not responding`, and then to `configured out` as the node is automatically configured out. The partition then returns to the `running` state.

If the node is running but RMS is not running, the node enters an `active` state. By default, the node status changes from `active` to `configured out` as the node is automatically configured out. The behavior is controlled by the `configure-out-active` attribute in the RMS database. The node is also automatically configured out if a blocked process causes the partition as a whole to be blocked for more than the time specified by the `partition-blocked-timeout` attribute.

5.8.2 Booting Nodes

Chapter 2 describes how to use the `sra boot` command to boot nodes. Usually, a halted node is in the `configured out` state. This is either because the node was configured out before it was shut down, or because the node was automatically configured out by the partition. When you boot a node, you can specify whether the `sra` command should automatically configure in the node or not.

If you do not specify that `sra` should automatically configure in the nodes, the booted node will be running but will remain in the `configured out` state. RMS will not use this node to run jobs, and the `rinfo -pl` command will not show this node as a member of the partition. To manually configure in the node, you should check that the node is actually running (for example, use the `sra info` command) before using the `rcontrol configure in` command. However, as explained in Section 5.8.1, if the node is not responding normally it will be automatically configured out.

5.8.3 Shutting Down Nodes

Chapter 2 describes how to shut down a node. When you shut down a node, you can specify whether the `sra` command should automatically configure out the node before halting the node.

The node state (as shown by the `rinfo -n` command) after a node is shut down depends on the situation, as follows:

- The node is configured out before shutdown and the partition is running.

  The node state is `configured out`. All resources and jobs running on the node are cleaned up and their status is set to `failed`.

- The node is not configured out before shutdown and the partition is running.

  The node is automatically configured out and its state is set to `configured out`. All resources and jobs running on the node are cleaned up and their status is set to `failed`.

- The node is configured out before shutdown and the partition is down.

  The node state is `configured out`. Since the partition is down, the status of resources is not updated. As displayed by the `rinfo` command, the status of jobs is `unknown`. When the partition is next started, all resources and jobs running on the node are cleaned up and their status is set to `failed`.

- The node is not configured out before shutdown and the partition is down.

  The node state is set to `not responding`. Since the partition is down, the status of resources is not updated. As displayed by the `rinfo` command, the status of jobs is `unknown`. You will be unable to start the partition — you must configure the node out (or reboot the node) before the `rcontrol` command will start the partition.

5.8.4 Node Failure

This section describes how RMS reacts to failure of node hardware or software. The information in this section is organized as follows:

- Node Status (see Section 5.8.4.1 on page 5–64)
- Partition Status (see Section 5.8.4.2 on page 5–65)

5.8.4.1 Node Status

The status of a node (as shown by `rinfo -n`) can be one of the following:

- `running`
  The node is running normally — the `rmsd` daemon on the node is responding.
- `active`
  The node is a member of its domain, but the `rmsd` daemon on the node is not responding. This can indicate one of the following:
  – The `rmsd` daemon has exited and is unable to restart. This could be due to a failure of the RMS software, but is probably caused by a failure of the node's system software.
  – RMS has been manually stopped on the node (see Section 5.9.2 on page 5–69).
  – The `rmsd` daemon is unable to communicate.
  – The node is hung — it continues to be a member of a domain, but is not responsive.

  Use the `sra info` command to determine the true state of the node. If you are able to log into the node and the `rmsd` daemon appears to be active, restart RMS on that node. You should report such problems to your HP support engineer, who may ask you to gather more information before restarting RMS on the node.
- `not responding`
  The node is not a member of its domain, and the `rmsd` daemon on the node is not responding. This can indicate one of the following:
  – The management network has a failure that prevents communications.
  – The node is halted (or in the process of halting).
  – The node is hung in some other way.

  Use the `sra info` command to determine the true state of the node. If the node is halted, check to see if the node has crashed, as described in Section 29.5.2 on page 29–17.
- `configured out`

  The node has been configured out of all partitions of which it is a member. See Section 5.8.1 on page 5–61 for further information on nodes in a configured out state.

5.8.4.2 Partition Status

The status of a partition can be one of the following:

- `running`
  The partition has been started, is running normally, and can be used to allocate resources and run jobs. The partition manager is active, and all nodes in the partition are in the `running` state.
- `closing`
  The partition is being stopped using the `wait` option in the `rcontrol stop partition` command. Users cannot allocate resources or run jobs. The partition stays in this state until all jobs belonging to all currently allocated resources are finished. At that point, the state changes to `down`.
- `down`
  The partition has been stopped. The partition manager exits when a partition is stopped. While in this state, users cannot allocate resources or run jobs.
- `blocked`
  The partition was running, but one or more nodes are not responding. The partition does not stay in this state for long — as soon as the node status of the non-responsive nodes is set to `not responding` or `active`, the partition manager automatically configures out the nodes. The partition should then return to the `running` state. While in the `blocked` state, the partition manager stops allocation and scheduling operations. Resources cannot be allocated. All resources in the `queued` or `blocked` state remain in that state. If `allocate` or `prun` exits (either normally or because a user sent a signal), the state of the resource and associated jobs remain unchanged.

Note:

While a partition is in the `running` or `closing` state, RMS correctly displays the current status of the resources and jobs.

However, if the partition status changes to `blocked` or `down`, RMS displays the following:

- Resources status = status of resources at the time that the partition status changed to `blocked` or `down`
- Jobs status = set to the `unknown` state

RMS is unable to determine the real state of resources and jobs until the partition runs normally.

5.8.5 Draining Nodes

The RMS partition manager provides support for draining jobs from a set of CPUs, and then configuring out the CPUs. The syntax of the command for draining nodes is as follows:

```
rcontrol drain node[=]nodelist [option[=]cancel]
```

where:

- `nodelist` is a list of nodes in the same partition.

The nodes are configured out when the current CPU allocations are completed. New or queued resource requests are not allocated on these nodes. Requests for specific sets of nodes that span a node currently being drained fail if one of the requested nodes is being drained, and a message is displayed as shown in the following example:

```
atlas0# prun -B0 -N6 -R hwbcast=1 hostname
prun: Error: can't allocate 6 cpus on nodes 0-5: atlas4 is being drained
```

New requests that do not specify a precise set of nodes succeed by avoiding nodes that are being drained. If a request is made to configure out a node that is in the process of being drained, the jobs on that node are killed and the node is configured out immediately.

Information on nodes being drained is maintained in the nodes table, and is reported by the rinfo command as shown in the following example:

```
atlas0# rinfo -nl
STATUS              NODES
running             atlas[0-4,6-7]
drain               atlas4
configured out      atlas5
...
REASON CONFIGURED OUT           NODES
drain complete                  atlas5
```

The state on nodes being drained is preserved over a partition restart.

The `cancel` option allows you to cancel a drain operation on one or more nodes.

5.9 RMS Servers and Daemons

The information in this section is organized as follows:

- Overview (see Section 5.9.1 on page 5–67)
- Stopping the RMS System and mSQL (see Section 5.9.2 on page 5–69)
- Manually Starting RMS (see Section 5.9.3 on page 5–71)
- Stopping and Starting RMS Servers (see Section 5.9.4 on page 5–72)
- Running the Switch Manager (see Section 5.9.5 on page 5–72)
- Log Files (see Section 5.9.6 on page 5–73)

5.9.1 Overview

In a normal running system, the following RMS daemons run on each node:

- `rmsmhd`
  This daemon is responsible for starting other RMS daemons. It monitors their status and if they exit, it restarts them.
- `rmsd`
  This daemon is responsible for gathering data about the local node, and it is involved in the creation of parallel programs. The `rmsd` daemon does not exit when a partition is stopped.

One node acts as the "master" node. It is designated as `rmshost`, and is aliased as such in the `/etc/hosts` file. The following daemons exist on the RMS master node:

- `msql3d`
  This daemon is responsible for managing the SC database. It responds to SQL commands to update and read the database.
- `mmanager`
  This daemon is the machine manager. It is responsible for monitoring the `rmsd` daemons on any nodes that are not members of an active partition, or nodes in a partition that is down or blocked.
- `pmanager`
  This daemon is the partition manager — there is one `pmanager` daemon for each active partition. A partition manager daemon is started in response to a start partition request from `rcontrol`. Once started, it is responsible for resource allocation and scheduling for that partition. It is responsible for monitoring the `rmsd` daemons on nodes that are members of the active partition. When the partition is stopped, the partition manager changes the status of the partition to `down` and exits.
- `eventmgr`
  This daemon is the event manager. It is responsible for dispatching events to the event handler scripts.

- tlogmgr
  This daemon is the transaction logger.
- swmgr
  This daemon is the Network Switch Manager. It is responsible for monitoring the HP AlphaServer SC Interconnect switch.

The daemons are started and stopped using scripts in /sbin/init.d with appropriate links in /sbin/rc0.d and /sbin/rc3.d, as described in Table 5–10 on page 5–68.

However, when SC20rms and SC05msql are registered as CAA applications, the startup scripts are modified as follows:

- The /sbin/init.d/msqld script does not start msql3d — instead, CAA is used to start and stop msql3d. Generally, once SC05msql is a registered CAA application, you should use caa_start and caa_stop. However, you can also use /sbin/init.d/msqld with the force_start and force_stop arguments. If the SC05msql CAA application is in the running state, a force_stop will cause msql3d to exit. However, CAA will restart it a short time later.
- The /sbin/init.d/rms script starts rmsmhd on all nodes except the node running the SC20rms application. On that node, CAA will have already started rmsmhd, so /sbin/init.d/rms does nothing.

Table 5–10 Scripts that Start RMS Daemons

Script	Action
msqld	Starts the msql3d daemon on rmshost.
rms	Starts the rmsmhd daemon. This in turn starts the other daemons as appropriate.

In the RMS system, the daemons are known as servers. You can view the status of all running servers as follows:

```
# rmsquery -v "select * from servers"
name                hostname   port   pid      rms   startTime            autostart    status     args
-------------------------------------------------------------------------------------------------------
tlogmgr             rmshost    6200   239278   1     05/17/00 11:13:56    1            ok
eventmgr            rmshost    6201   239283   1     05/17/00 11:13:57    1            ok
mmanager            rmshost    6202   269971   1     05/17/00 16:53:38    1            ok
swmgr               rmshost    6203   239286   1     05/17/00 11:14:01    1            ok
jtdw                rmshost    6204   -1       0     --/--/-- --:--:--    0            error
pepw                rmshost    6205   -1       0     --/--/-- --:--:--    0            error
pmanager-parallel   rmshost    6212   395175   1     05/22/00 10:35:48    1            ok         Null
rmsd                atlasms    6211   369292   1     05/18/00 10:06:26    1            ok         Null
rmsmhd              atlasms    6210   239292   1     05/17/00 11:13:55    0            ok         Null
rmsd                atlas4     6211   2622210  1     05/22/00 10:55:52    1            ok         Null
rmsmhd              atlas4     6210   2622200  1     05/22/00 10:55:52    0            ok         Null
rmsd                atlas1     6211   1321303  1     05/22/00 10:35:41    1            ok         Null
```

```
rmsd            atlas2    6211   1676662  1    05/22/00 10:35:38    1          ok        Null
rmsmhd          atlas1    6210   1056783  1    05/18/00 13:50:02    0          ok        Null
rmsmhd          atlas2    6210   1574228  1    05/17/00 22:17:08    0          ok        Null
rmsd            atlas3    6211   2291581  1    05/22/00 10:35:40    1          ok        Null
rmsmhd          atlas3    6210   2098007  1    05/17/00 22:13:03    0          ok        Null
rmsd            atlas0    6211   580099   1    05/17/00 11:51:18    1          ok        Null
rmsmhd          atlas0    6210   578952   1    05/17/00 10:05:43    0          ok        Null
```

Note that the `jtdw` and `pepw` servers are reserved for future use; their errors can be ignored.

You can check that a server is actually running as follows:

```
$ rinfo -s rmsd atlas1
rmsd                atlas0        running         580099
```

5.9.2 Stopping the RMS System and mSQL

RMS usually operates normally, as it is started when you boot nodes. However, it is sometimes useful to stop the RMS system, such as in the following cases:

- When you want to install a new version of the RMS software.
- When you want to rebuild the database from scratch.

To stop the RMS system, perform the following steps:

1. Ensure that there are no allocated resources. One way to do this is to stop each partition using the `kill` option, as shown in the following example:
   ```
   # rcontrol stop partition=big option kill
   ```
2. **Note:**

   If the `SC20rms` CAA application has not been enabled, skip this step.

   If the `SC20rms` CAA application has been enabled and is running, stop the `SC20rms` application.

   You can determine the current status of the `SC20rms` application by running the `caa_stat` command on the first domain in the system (that is, `atlasD0`, where `atlas` is an example system name) as follows:
   ```
   # caa_stat SC20rms
   ```
   To stop the `SC20rms` application, use the `caa_stop` command, as follows:
   ```
   # caa_stop SC20rms
   ```
3. Stop the RMS daemons on every node, by running the following command once on any node:
   ```
   # rmsctl stop
   ```

Note:

If the SC20rms CAA application has been enabled and you did not stop the SC20rms application as described in step 2, then you will not be able to stop the RMS daemons in this step — CAA will automatically restart RMS daemons on the node where the SC20rms application was last located.

4. Stop the msql3d daemon in one of the following ways, depending on whether you have registered SC05msql as a CAA service:

   - Case 1: SC05msql is registered with CAA

     If the SC05msql CAA application has been enabled and is running, stop the SC05msql application.

     You can determine the current status of the SC05msql application by running the caa_stat command on the first domain in the system (that is, atlasD0, where atlas is an example system name), as follows:
     ```
     # caa_stat SC05msql
     ```

     To stop the SC05msql application, use the caa_stop command, as follows:
     ```
     # caa_stop SC05msql
     ```

   - Case 2: SC05msql is not registered with CAA

     If the SC05msql CAA application has not been enabled, stop the msql3d daemon by running the following command on the RMS master node (rmshost):
     ```
     # /sbin/init.d/msqld stop
     ```

This process stops the RMS system.

At this stage, any attempt to use an RMS command will result in an error similar to the following:
```
rinfo: Warning: Can't connect to mSQL server on rmshost: retrying ...
```

This is because the msql3d daemon was stopped in step 4 above. If you skip step 4 (so that the msql3d daemon is running, but RMS daemons are stopped), you will be able to access the database but unable to execute commands that require the RMS daemons. Different commands need different RMS daemons, so the resulting error messages will differ. A typical message is similar to the following:
```
rcontrol: Warning: RMS server pmanager-parallel (rmshost) not responding
```

Note:

If you did not perform step 1 above, this process will not stop any jobs that are running when the RMS system is stopped.

5.9.3 Manually Starting RMS

If you stopped RMS as described in Section 5.9.2 on page 5–69, you can restart RMS by performing the following steps:

1. Start the `msql3d` daemon in one of the following ways, depending on whether you have registered `SC05msql` as a CAA service:

   - Case 1: `SC05msql` is registered with CAA

     If the `SC05msql` CAA application has been enabled and is stopped, start the `SC05msql` application.

     You can determine the current status of the `SC05msql` application by running the `caa_stat` command on the first domain in the system (that is, `atlasD0`, where `atlas` is an example system name) as follows:

     ```
     # caa_stat SC05msql
     ```

     To start the `SC05msql` application, use the `caa_start` command as follows:

     ```
     # caa_start SC05msql
     ```

   - Case 2: `SC05msql` is not registered with CAA

     If the `SC05msql` CAA application has not been enabled, start the `msql3d` daemon by running the following command on the RMS master node (`rmshost`):

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

2. **Note:**

   If the `SC20rms` CAA application has not been enabled, skip this step.

   If the `SC20rms` CAA application has been enabled and is stopped, start the `SC20rms` application.

   You can determine the current status of the `SC20rms` application by running the `caa_stat` command on the first domain in the system (that is, `atlasD0`, where `atlas` is an example system name) as follows:

   ```
   # caa_stat SC20rms
   ```

   To start the `SC20rms` application, use the `caa_start` command as follows:

   ```
   # caa_start SC20rms
   ```

3. Start the RMS daemons on the remaining nodes, by running the following command once on any node:

   ```
   # rmsctl start
   ```

5.9.4 Stopping and Starting RMS Servers

Sometimes you need to stop and start RMS servers (daemons), such as in the following cases:

- The `swmgr` daemon must be stopped if you want to use the `jtest` program with the `-r` (`raw`) option.
- You must stop and restart the `eventmgr` daemon if you change the `event_handlers` table.

You stop servers using the `rcontrol` command, as shown in the following example:

```
# rcontrol stop server=swmgr
```

You start servers using the `rcontrol` command, as shown in the following example:

```
# rcontrol start server=swmgr
```

Note:

Do not simply kill the daemon process; the `rmsmhd` daemon will restart it.

If you stop a server with `rcontrol`, and shut down and boot the node, the server is automatically started when the node boots.

Do not start or stop partitions using `rcontrol start/stop server`; use `rcontrol start/stop partition` instead.

5.9.5 Running the Switch Manager

The switch manager (`swmgr`) daemon runs on `rmshost` — that is, either on the management server (if your HP AlphaServer SC system has a management server), or on Node 0 (if your HP AlphaServer SC system does not have a management server).

The `swmgr` daemon polls the switch for errors every 30 seconds, which can result in high CPU usage. This does not adversely affect the management server, which is lightly loaded.

However, if your HP AlphaServer SC system does not have a management server, you should reduce the CPU usage by setting the `swmgr-poll-interval` attribute to a value that is higher than 30 seconds.

For example, to poll the switch every 5 minutes, perform the following steps:

1. Check the value of the `swmgr-poll-interval` attribute, as follows:

   ```
   # rmsquery "select val from attributes where name='swmgr-poll-interval'"
   120
   ```

   - If the attribute has been defined, a numerical value is returned. In this example, the attribute has been set to 120, indicating that the switch is polled for errors every 2 minutes.
   - If the attribute has not been defined, no value is returned.

2. Set the swmgr-poll-interval attribute in one of the following ways:
   - If the swmgr-poll-interval attribute has not been defined, create a record for this attribute, as follows:
     # **rcontrol create attribute name=swmgr-poll-interval val=300**
   - If the swmgr-poll-interval attribute has been defined, update the value using the set command, as follows:
     # **rcontrol set attribute name=swmgr-poll-interval val=300**

5.9.6 Log Files

RMS creates log files. These can be useful in diagnosing problems with RMS. Log files are stored in two locations:

- /var/rms/adm/log

  On each system (management server or domain), this is a clusterwide directory. When you install RMS, this directory is created on all systems. However, you generally only need to look at this directory on rmshost. The exception is if you have configured swmgr to run on a different node (see Section 5.9.5 on page 5–72).

- /var/log

  This is a node-local directory.

The RMS log files are described in Table 5–11.

Table 5–11 RMS Log Files

Log File	Description
/var/rms/adm/log/mmanager.log	The mmanager daemon writes debug messages to this file.
/var/rms/adm/log/pmanager-*name*.log	This is the pmanager debug file, where *name* is the name of the partition.
/var/rms/adm/log/event.log	The rmsevent_node script writes a brief message to this file when it runs.
/var/rms/adm/log/eventmgr.log	This file records eventmgr handling of events. It also contains messages from the event handling scripts. This file can be very large.
/var/rms/adm/log/swmgr.log	This file contains debug messages from the swmgr daemon.
/var/rms/adm/log/tlogmgr.log	This file contains debug messages from the tlogmgr daemon.
/var/rms/adm/log/msqld.log	This file contains messages from the msql3d daemon. This file is overwritten each time the daemon starts.
/var/log/rmsmhd.log	This file contains operational messages from the node's rmsmhd and rmsd daemons.

5.10 Site-Specific Modifications to RMS: the pstartup Script

When a partition is started, `rcontrol` runs a `pstartup` script. The startup script is located as follows:

1. `rcontrol` runs the `/usr/opt/rms/etc/pstartup` script.
2. `pstartup` looks for a script called `/usr/opt/rms/etc/pstartup.OSF1` and executes it.
3. If `pstartup.OSF1` finds a file called `/usr/local/rms/etc/pstartup`, it executes it and takes no further action. Otherwise, it implements the policy described in Section 5.6.1 on page 5–35.

Therefore, if you wish to implement your own policy, you may write your own script and place it in a file called `/usr/local/rms/etc/pstartup`.

5.11 RMS and CAA Failover Capability

If your HP AlphaServer SC system does not have a management server, or if your HP AlphaServer SC system has a clustered management server, you can configure RMS to failover between nodes, as described in Chapter 8 of the *HP AlphaServer SC Installation Guide*. You cannot configure RMS to failover if your HP AlphaServer SC system has a single management server. When failover is enabled, the `msql3d` daemon is started by CAA, not by `/sbin/init.d/msqld`.

This section describes the following tasks:

- Determining Whether RMS is Set Up for Failover (see Section 5.11.1 on page 5–74)
- Removing CAA Failover Capability from RMS (see Section 5.11.2 on page 5–75)

See Chapter 23 for more information on how to manage highly available applications — for example, how to monitor and manually relocate CAA applications.

5.11.1 Determining Whether RMS is Set Up for Failover

To determine whether RMS is set up for failover, run the `caa_stat` command, as follows:

```
# /usr/sbin/caa_stat SC20rms
# /usr/sbin/caa_stat SC05msql
```

If RMS is not set up for failover, the following message appears:

```
Could not find resource SC20rms
Could not find resource SC05msql
```

If failover is enabled, the command prints status information, including the name of the host that is currently the RMS master node (`rmshost`).

5.11.2 Removing CAA Failover Capability from RMS

To remove CAA failover capability from RMS, perform the following steps on any node in the first domain (that is, `atlasD0`, where `atlas` is an example system name):

1. Identify the current RMS master node (`rmshost`), as follows:
   `# /usr/sbin/caa_stat SC20rms`
2. Stop the RMS daemons on `rmshost`, as follows:
   `# /usr/sbin/caa_stop SC20rms`
3. Stop the `msql3d` daemon on `rmshost`, as follows:
   `# /usr/sbin/caa_stop SC05msql`
4. Unregister the `SC20rms` and `SC05msql` resource profiles from CAA, as follows:
   ```
   # /usr/sbin/caa_unregister SC20rms
   # /usr/sbin/caa_unregister SC05msql
   ```
5. Delete the `SC20rms` and `SC05msql` CAA application resource profiles, as follows:
   ```
   # /usr/sbin/caa_profile -delete SC20rms
   # /usr/sbin/caa_profile -delete SC05msql
   ```

   Note:

   The `caa_profile delete` command will delete the profile scripts associated with the available service.

   These scripts are normally held in the `/var/cluster/caa/script` directory. If you accidentally delete the `SC20rms` or `SC05msql` profile scripts, you can restore them by copying them from the `/usr/opt/rms/examples/scripts` directory.

6. Edit the `/etc/hosts` file on the management server, and on each domain, to set `rmshost` as Node 0 (that is, `atlas0`).
7. Log on to `atlas0` and start the `msql3d` daemon, as follows:
   `atlas0# /sbin/init.d/msqld start`
8. Update the `attributes` table in the SC database, as follows:
   `atlas0# rcontrol set attribute name=rmshost val=atlas0`
9. Log on to the node identified in step 1, and start the RMS daemons there, as follows:
   `# /sbin/init.d/rms start`
10. If the node identified in step 1 is not `atlas0`, log on to `atlas0` and restart the RMS daemons there, as follows:
    `atlas0# /sbin/init.d/rms restart`

5.12 Using Dual Rail

To control how an application uses the rails, use the `-R` option with either the `allocate` or `prun` command. The syntax of the `-R` option is as follows:

```
[-R rails=numrails | railmask=mask]]
```

where

- The *numrails* value can be 1 (use one rail) or 2 (use both rails).

  By default, RMS will automatically assign one rail to the application.
- The `railmask` argument takes a bit field:
  - A *mask* of 1 indicates that the application should use the first rail (rail 0).
  - A *mask* of 2 indicates that the application should use the second rail (rail 1).
  - A *mask* of 3 indicates that the application should use both rails (rail 0 and rail 1).

You do not have to rewrite the application to use multiple rails — the MPI and Shmem libraries automatically use whichever rails RMS has assigned to the application.

See the *HP AlphaServer SC User Guide* for more information.

5.13 Useful SQL Commands

Note:

In HP AlphaServer SC Version 2.6, HP recommends that you use the `rcontrol` command instead of the `rmsquery` command to insert or update SC database entries — the only `rmsquery` commands supported are those documented in this guide.

This section provides the SQL commands that are most often used by an HP AlphaServer SC system administrator.

To find the names of all tables, enter the following command:

```
# rmsquery
sql> tables
```

To find the datatypes of fields in all tables, enter the following command:

```
# rmstbladm -d | grep create
```

Note:

The `rmstbladm` command does not support user-defined tables or fields.

Generally, all fields are either strings or numbers; the above command is only needed if you need to know whether the string has a fixed size, or whether a `Null` value is allowed. An easier way to display the names of fields is to use the `rmsquery -v "select..."` command, as follows:

```
# rmsquery -v "select * from access_controls"
name  class  partition  priority  maxcpus  memlimit
-----------------------------------------------------
```

The above command is an example of a query. The syntax is as follows:

`select`	(that is, identify and print records)
`*`	(all fields)
`from access_controls`	(from the `access_controls` table)

Note:

You must enclose the SQL statement in double quotes to ensure that the `*` is passed directly to the database without being further processed by the shell.

You can narrow the search by specifying certain criteria, as shown in the following example:

```
# rmsquery -v "select * from access_controls where partition='bonnie'"
```

The `where` clause allows you to select only those records that match a condition. In the above example, you select only those records that contain the text `bonnie` in the `partition` field.

Note:

In the above example, `bonnie` is enclosed in single quotes. This is because the `partition` field is a string field. If the specified field is a number field, you must omit the quotes.

If you forget to include the single quotes, you get an error, as follows:

```
# rmsquery -v "select * from access_controls where partition=bonnie"
rmsquery: failed to connect: Unknown field "access_controls.bonnie"
```

You can select specific fields, as shown in the following example:

```
# rmsquery -v "select name,status from resources where status='allocated'"
```

You can select from several tables. The following query selects records from the `resources` and `acctstats` tables. The query matches records from the `resources` and `acctstats` tables where the `name` field has the same value in both tables. (In SQL terms, this is known as a `join` operation). Then it selects only records where the status is `finished`. It then prints the `name` and `status` values from the `resources` table, and the `etime` value from the corresponding `acctstats` table:

```
# rmsquery -v "select resources.name,resources.status,acctstats.etime
   from resources,acctstats
   where resources.name=acctstats.name
   and resources.status='finished'"
```

To create records, use the `rmsquery "insert..."` command, as follows:

```
# rmsquery "insert into access_controls values ('joe','user','bonnie',1,20,200)"
```

You can specify `Null` values, as follows:

```
# rmsquery "insert into access_controls values
('dev','project','bonnie',1,20,Null)"
```

To update records, use the `rmsquery "update..."` command, as follows:

```
# rmsquery "update access_controls set maxcpus=14 where name='joe'"
# rmsquery "update access_controls set maxcpus=14,memlimit=Null where name='joe'"
```

To delete records, use the `rmsquery "delete..."` command, as follows:

```
# rmsquery "delete from access_controls where name='joe'"
```

6

Overview of File Systems and Storage

This chapter provides an overview of the file system and storage components of the HP AlphaServer SC system.

The information in this chapter is structured as follows:

- Introduction (see Section 6.1 on page 6–2)
- Changes in HP AlphaServer SC File Systems in Version 2.6 (see Section 6.2 on page 6–2)
- SCFS (see Section 6.3 on page 6–3)
- PFS (see Section 6.4 on page 6–5)
- Preferred File Server Nodes and Failover (see Section 6.5 on page 6–8)
- Storage Overview (see Section 6.6 on page 6–9)
- External Data Storage Configuration (see Section 6.7 on page 6–13)

6.1 Introduction

This section provides an overview of the HP AlphaServer SC Version 2.6 storage and file system capabilities. Subsequent sections provide more detail on administering the specific components.

The HP AlphaServer SC system is comprised of multiple domains. There are two types of domains: File-Serving (FS) domains and Compute-Serving (CS) domains. HP AlphaServer SC Version 2.6 supports a maximum of four FS domains.

The nodes in the FS domains serve their file systems, *via* an HP AlphaServer SC high-speed proprietary protocol (SCFS), to the other domains. File system management utilities ensure that the served file systems are mounted at the same point in the name space on all domains.

The result is a data file system (or systems) that is globally visible and performs at high speed. PFS uses the SCFS component file systems to aggregate the performance of multiple file servers, so that users can have access to a single file system with a bandwidth and throughput capability that is greater than a single file server.

6.2 Changes in HP AlphaServer SC File Systems in Version 2.6

The major changes in HP AlphaServer SC file-system capability that have been introduced in Version 2.6 include the following:

- SCFS file systems exported from an FS domain are now mounted from preferred server cluster aliases. In HP AlphaServer SC Version 2.5, the default cluster alias was used to mount SCFS file systems. In HP AlphaServer SC Version 2.6, each member of an FS domain has a cluster alias associated with it. This new preferred server cluster alias is now used to mount SCFS file systems, thus reducing the number of network message hops inside the FS domain.

  The new preferred server cluster aliases are in the format:
  `atlasD`*`n`*`-atlas`*`m`*

  where:

  – *n* is the domain number

  – *m* is the member number

  For example, `atlasD0-atlas0`.

  The new preferred server cluster aliases are associated with the management LAN (10.128).

- Quota support has been added for SCFS file systems. For more information, see Section 7.4.15 on page 7–22.

- Additional mount option arguments can be used with the `scfsmgr create` command. For more information, see Section 7.4.1.1 on page 7–11.
- The O_UBCIO option (defined in <`sys/scfs.h`>) and the CFS subsystem `io_env` attribute have been added to allow for more flexibility in choosing whether a file is opened in FAST mode or UBC (Unified Buffer Cache) mode. For more information, see Section 7.2 on page 7–3.
- Three new mount-related sub-commands have been added to the `scfsmgr` command:
  - The `scfsmgr sync_domain` command checks the mount status of all SCFS and PFS file systems in the specified domain and either unmounts or mounts them as appropriate.
  - The `scfsmgr try_mount` and `scfsmgr try_unmount` commands can be used on FS domains and CS domains to troubleshoot specific client nodes or client domains that are giving problems.

    The `try_mount` and `try_unmount` commands are also available with the `pfsmgr` command.
- In HP AlphaServer SC Version 2.6, Access Control Lists (ACLs) can be used in SCFS file systems. For more information, see Section 7.5 on page 7–27.
- The SCFS kernel subsystem can make use of multiple Elan rails, if they are present on the system, to perform SCFS I/O data transfers. This allows SCFS clients to load-balance their requests over all the rails that they are permitted to use and can see the server nodes on. For more information on using multiple rails, see Section 7.8.2.3 on page 7–34.

6.3 SCFS

With SCFS, a number of nodes in up to four domains are designated as file servers, and these domains are referred to as FS domains. The file server nodes are normally connected to external high-speed storage subsystems (RAID arrays). These nodes serve the associated file systems to the remainder of the system (the other FS domains and the CS domains) *via* the HP AlphaServer SC Interconnect. In HP AlphaServer SC Version 2.6, SCFS file systems exported from an FS domain are mounted from preferred server cluster aliases. The preferred server cluster aliases are associated with the management LAN (10.128).

The default mode (known as FAST mode) of operation for SCFS is to ship data transfer requests directly to the node serving the file system. On the server node, there is a per-file-system SCFS server thread in the kernel. For a write transfer, this thread transfers the data directly from the user's buffer *via* the HP AlphaServer SC Interconnect to the serving node UBC, and writes it to disk.

Data transfers are done in blocks, and disk transfers are scheduled once the block has arrived. This allows large transfers to achieve an overlap between the disk and the HP AlphaServer SC Interconnect. Note that the transfers bypass the client systems' UBC. Bypassing the UBC allows the system to operate on data sizes larger than the system page size (8KB) up to 128KB.

Although bypassing the UBC is efficient for large sequential writes and reads, the data is read by the client multiple times when multiple processes read the same file. While this will still be fast, it is less efficient; therefore, it may be worth setting the mode so that UBC is used (see Section 6.3.1).

A data integrity mechanism in SCFS ensures that data written to a file is valid, even if file server nodes crash. This mechanism replays client write requests that have not been committed on the server.

6.3.1 Selection of FAST Mode

The default mode of operation for an SCFS file system is set when the system administrator sets up the file system using the `scfsmgr` command (see Chapter 7).

The default mode can be set to FAST (that is, bypasses the UBC) or UBC (that is, uses the UBC). The default mode applies to all files in the file system.

The default mode can be overridden either by setting environment variables or by providing extra flags at open system call time. For more information, see Section 7.2 on page 7–3.

Note:

If the default mode is set to UBC, the file system performance and characteristics are equivalent to that expected of an NFS-mounted file system.

6.3.2 Getting the Most Out of SCFS

SCFS is designed to deliver high bandwidth transfers for applications performing large serial I/O. Disk transfers are performed by a kernel subsystem on the file server node using the HP AlphaServer SC Interconnect kernel-to-kernel message transport.

The HP AlphaServer SC Interconnect reaches its optimum bandwidth at message sizes of 64KB and above. Because of this, optimal SCFS performance will be attained by applications performing transfers that are in excess of this figure. An application performing a single 8MB write is just as efficient as an application performing eight 1MB writes or sixty-four 128KB writes — in fact, a single 8MB write is slightly more efficient, due to the decreased number of system calls.

Because the SCFS system overlaps HP AlphaServer SC Interconnect transfers with storage transfers, optimal user performance will be seen at user transfer sizes of 128KB or greater. Double buffering occurs when a chunk of data (`io_block`, default 128KB) is transferred and is then written to disk while the next 128KB is being transferred from the client system *via* the HP AlphaServer SC Elan adapter card.

This allows overlap of HP AlphaServer SC Interconnect transfers and I/O operations. The `sysconfig` parameter `io_block` in the SCFS stanza allows you to tune the amount of data transferred by the SCFS server (see Section 7.8 on page 7–32). The default value is 128KB. If the typical transfer at your site is smaller than 128KB, you can decrease this value to allow double buffering to take effect.

HP recommends UBC mode for applications that use short file system transfers — performance will not be optimal if FAST mode is used. This is because FAST mode trades the overhead of mapping the user buffer into the HP AlphaServer SC Interconnect against the efficiency of HP AlphaServer SC Interconnect transfers. Where an application does many short transfers (less than 16KB), this trade-off results in a performance drop. In such cases, UBC mode should be used.

6.4 PFS

Using SCFS, a single FS node can serve a file system or multiple file systems to all of the nodes in the other domains. When normally configured, an FS node will have multiple storage sets (see Section 6.6 on page 6–9), in one of the following configurations:

- There is a file system per storage set — multiple file systems are exported.
- The storage sets are aggregated into a single logical volume using LSM — a single file system is exported.

Where multiple file server nodes are used, multiple file systems will always be exported. This solution can work for installations that wish to scale file system bandwidth by balancing I/O load over multiple file systems. However, it is more generally the case that installations require a single file system, or a small number of file systems, with scalable performance.

PFS provides this capability. A PFS file system is constructed from multiple component file systems. Files in the PFS file system are striped over the underlying component file systems.

When a file is created in a PFS file system, its mapping to component file systems is controlled by a number of parameters, as follows:

- The component file system for the initial stripe

  This is selected at random from the set of components. Using a random selection ensures that the load of multiple concurrent file accesses is distributed.

- The stride size

  This parameter is set at file system creation. It controls how much data is written per file to a component before the next component is used.

- The number of components used in striping

  This parameter is set at file system creation. It specifies the number of components file systems over which an individual file will be striped. The default is all components. In file systems with very large numbers of components, it can be more efficient to use only a subset of components per file (see discussion below).

- The block size

  This number should be less than or equal to the stride size. The stride size must be an even multiple of the block size. The default block size is the same value as the stride size. This parameter specifies how much data the PFS file system will issue (in a read or write command) to the underlying file system. Generally, there is not a lot of benefit in changing the default value. SCFS (which is used for the underlying PFS components) is more efficient at bigger transfers, so leaving the block size equal to the stride size maximizes SCFS efficiency.

These parameters are specified at file system creation. They can be modified by a PFS-aware application or library using a set of PFS specific ioctls.

In a configuration with a large number of component file systems and a large client population, it can be more efficient to restrict the number of stripe components. With a large client population writing to every file server, the file servers experience a higher rate of interrupts. By restricting the number of stripe components, individual file server nodes will serve a smaller number of clients, but the aggregate throughput of all servers remains the same. Each client will still get a degree of parallel I/O activity, due to its file being striped over a number of components. This is true where each client is writing to a different file. If each client process is writing to the same file, it is obviously optimal to stripe over all components.

6.4.1 PFS and SCFS

PFS is a layered file system. It reads and writes data by striping it over component file systems. SCFS is used to serve the component file systems to the CS nodes.

Figure 6–1 shows a system with a single FS domain comprised of four nodes, and two CS domains identified as single clients. The FS domain serves the component file systems to the CS domains. A single PFS is built from the component file systems. The first member (member 1) of each CS domain imports the SCFS volumes from the file server nodes. This member is also known as the importing node. Other CS nodes access the SCFS file system through CFS.

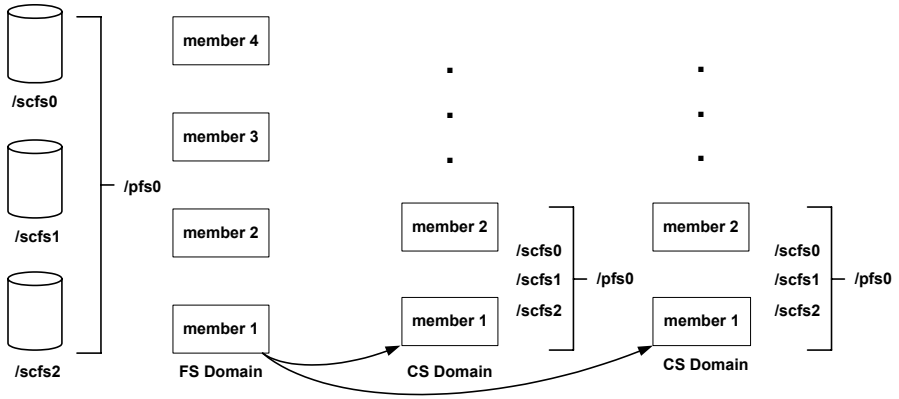

Figure 6–1 Example PFS/SCFS Configuration

6.4.1.1 User Process Operation

Processes running in either (or both) of the CS domains act on files in the PFS file system. Depending on the offset within the file, PFS will map the transaction onto one of the underlying SCFS components and pass the call down to SCFS. The SCFS client code passes the I/O request, this time for the SCFS file system, *via* the HP AlphaServer SC Interconnect to the appropriate file server node. At this node, the SCFS thread will transfer the data between the client's buffer and the file system. Multiple processes can be active on the PFS file system at the same time, and can be served by different file server nodes.

6.4.1.2 System Administrator Operation

The file systems in an FS domain are created using the `scfsmgr` command. This command allows the system administrator to specify all of the parameters needed to create and export the file system. The `scfsmgr` command performs the following tasks:

- Creates the AdvFS file domain and file set
- Creates the mount point
- Populates the requisite configuration information in the `sc_scfs` table in the SC database, and in the `/etc/exports` file
- Nominates the preferred file server node
- Synchronizes the other domains, causing the file systems to be imported and mounted at the same mount point

To create the PFS file system, the system administrator uses the `pfsmgr` command to specify the operational parameters for the PFS file system and identify the component file systems. The `pfsmgr` command performs the following tasks:

- Builds the PFS file system by creating on-disk data structures

- Creates the mount point for the PFS file system
- Synchronizes the client systems
- Populates the requisite configuration information in the `sc_pfs` table in the SC database

The following extract shows example contents from the `sc_scfs` table in the SC database:

```
clu_domain      advfs_domain       fset_name     preferred_server  rw   speed    status     mount_point
-------------------------------------------------------------------------------------------------------
atlasD0         scfs0_domain       scfs0         atlas0            rw   FAST     ONLINE     /scfs0
atlasD0         scfs1_domain       scfs1         atlas1            rw   FAST     ONLINE     /scfs1
atlasD0         scfs2_domain       scfs2         atlas2            rw   FAST     ONLINE     /scfs2
atlasD0         scfs3_domain       scfs3         atlas3            rw   FAST     ONLINE     /scfs3
```

In this example, the system administrator created the four component file systems nominating the respective nodes as the preferred file server (see Section 6.5 on page 6–8). This caused each of the CS domains to import the four file systems and mount them at the same point in their respective name spaces. The PFS file system was built on the FS domain using the four component file systems; the resultant PFS file system was mounted on the FS domain. Each of the CS domains also mounted the PFS file system at the same mount point.

The end result is that each domain sees the same PFS file system at the same mount point. Client PFS accesses are translated into client SCFS accesses and are served by the appropriate SCFS file server node. The PFS file system can also be accessed within the FS domain. In this case, PFS accesses are translated into CFS accesses.

When building a PFS, the system administrator has the following choice:

- Use the set of complete component file systems; for example:
  `/pfs/comps/fs1`; `/pfs/comps/fs2`; `/pfs/comps/fs3`; `/pfs/comps/fs4`
- Use a set of subdirectories within the component file systems; for example:
  `/pfs/comps/fs1/x`; `/pfs/comps/fs2/x`; `/pfs/comps/fs3/x`; `/pfs/comps/fs4/x`

Using the second method allows the system administrator to create different PFS file systems (for instance, with different operational parameters), using the same set of underlying components. This can be useful for experimentation. For production-oriented PFS file systems, the first method is preferred.

6.5 Preferred File Server Nodes and Failover

In HP AlphaServer SC Version 2.6, you can configure up to four FS domains. Although the FS domains can be located anywhere in the HP AlphaServer SC system, HP recommends that you configure either the first domain(s) or the last domain(s) as FS domains — this provides a contiguous range of CS nodes for MPI jobs.

Because file server nodes are part of CFS, any member of an FS domain is capable of serving the file system. When an SCFS file system is being configured, one of the configuration parameters specifies the preferred server node. This should be one of the nodes with a direct physical connection to the storage for the file system.

If the node serving a particular component fails, the service will automatically migrate to another node that has connectivity to the storage.

6.6 Storage Overview

There are two types of storage in an HP AlphaServer SC system:

- Local or Internal Storage (see Section 6.6.1 on page 6–10)
- Global or External Storage (see Section 6.6.2 on page 6–11)

Figure 6–2 shows the HP AlphaServer SC storage configuration.

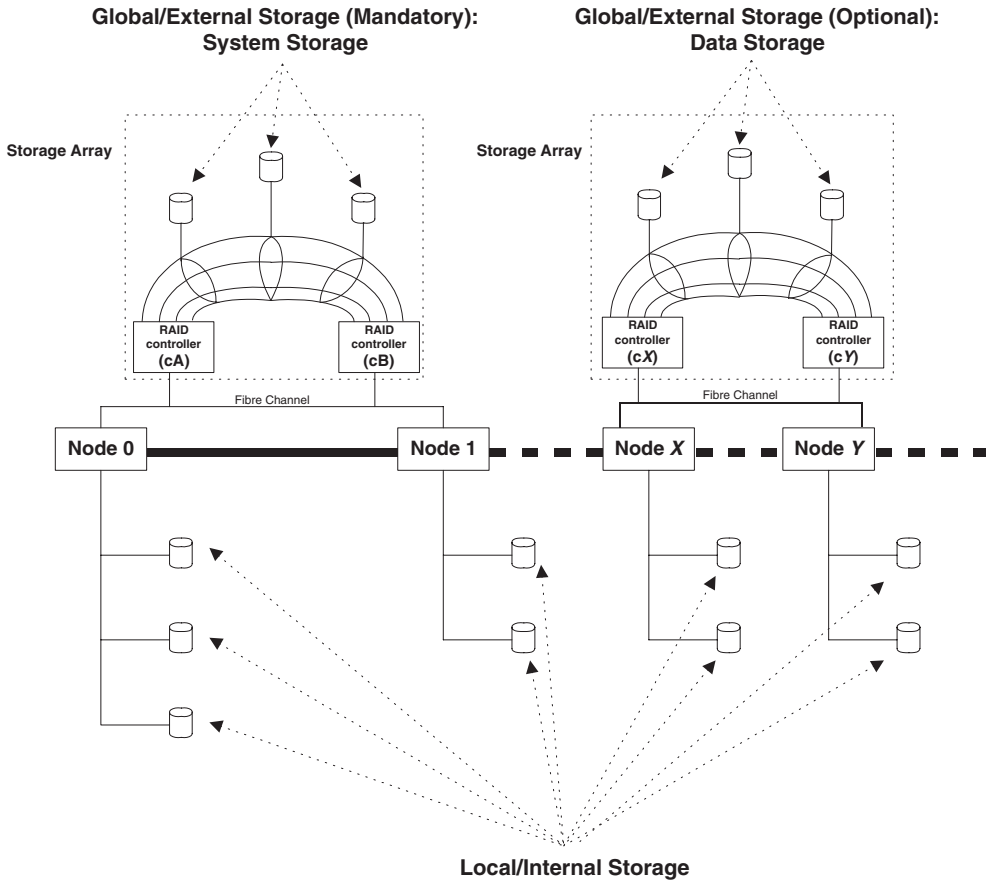

Figure 6–2 HP AlphaServer SC Storage Configuration

6.6.1 Local or Internal Storage

Local or internal storage is provided by disks that are internal to the node cabinet and not RAID-based. Local storage is not highly available. Local disks are intended to store volatile data, not permanent data.

Local storage improves performance by storing copies of node-specific temporary files (for example, swap and core) and frequently used files (for example, the operating system kernel) on locally attached disks.

The SRA utility can automatically regenerate a copy of the operating system and other node-specific files, in the case of disk failure.

Each HP AlphaServer ES45 or HP AlphaServer ES40 node requires at least two local disks. The first node of each domain requires a third local disk to hold the base Tru64 UNIX operating system.

The first disk is used to hold the following:

- The node's boot partition
- Swap space
- `tmp` and `local` partitions (mounted on `/tmp` and `/local` respectively)
- `cnx h` partition (used by the TruCluster `cnx` module to record membership and `cluster_root` device information)

The second disk (alternate boot disk or backup boot disk) on each node is just a copy of the first disk. In the case of primary disk failure, the system can boot the alternate disk. For more information about the alternate boot disk, see Section 2.5 on page 2–5.

6.6.1.1 Using Local Storage for Application I/O

PFS provides applications with scalable file bandwidth. Some applications have processes that need to write temporary files or data that will be local to that process — for such processes, you can write the temporary data to any local storage that is not used for boot, swap, and core files. If multiple processes in the application are writing data to their own local file system, the available bandwidth is the aggregate of each local file system that is being used. In HP AlphaServer SC Version 2.6, node-local file systems mounted with the `server_only` option provide a significantly faster file creation time than that provided by a regular mounted file system.

6.6.2 Global or External Storage

Global or external storage is provided by RAID arrays located in external storage cabinets, connected to a subset of nodes (minimum of two nodes) for availability and throughput.

In HP AlphaServer SC Version 2.6, three types of global or external storage are supported: Modular SAN Array 1000 (MSA1000), HSG-based storage arrays, and Enterprise Virtual Array storage systems (HSV-based).

An MSA1000-based storage array consists of the following:

- A pair of MSA1000 controllers
- Cache modules
- Redundant power supplies
- An array of physical disks

An HSG-based storage array contains the following in system cabinets with space for disk storage:

- A pair of HSG80 RAID controllers
- Cache modules
- Redundant power supplies

An HSV-based storage array consists of the following:

- A pair of HSV110 RAID controllers.
- An array of physical disk drives that the controller pair controls. The disk drives are located in drive enclosures that house the support systems for the disk drives.
- Associated physical, electrical, and environmental systems.
- The SANworks HSV Element Manager, which is the graphical interface to the storage system. The element manager software resides on the SANworks Management Appliance and is accessed through a browser.
- SANworks Management Appliance, switches, and cabling.

External storage is fully redundant in that each storage array is connected to two RAID controllers, and each RAID controller is connected to at least a pair of host nodes. To provide additional redundancy, a second Fibre Channel switch may be used, but this is not obligatory.

HP uses the following terms to describe RAID configurations:

- Stripeset (RAID 0)
- Mirrorset (RAID 1)

- RAIDset (RAID 3/5)
- Striped Mirrorset (RAID 0+1)
- JBOD (Just a Bunch Of Disks)
- ADG (Advanced Data Guarding)

Organizing external storage as Mirrorsets or RAIDsets ensures that the system continues to function in the event of physical media failure.

External storage is further subdivided as follows:

- System Storage (see Section 6.6.2.1 on page 6–12)
- Data Storage (see Section 6.6.2.2 on page 6–13)

6.6.2.1 System Storage

System storage is mandatory and is served by the first node in each domain. The second node in each domain is also connected to the system storage, for failover. These node pairs each require at least three additional disks — in a domain composed of HP AlphaServer DS20L nodes, the node pair requires a fourth additional disk — which they will share from the RAID subsystems (Mirrorsets). These disks are required as follows:

- One disk to hold the `/`, `/usr`, and `/var` directories of the domain AdvFS file system
- One disk to be used for generic boot partitions when adding new domain members
- One disk to be used as a backup during upgrades
- In a domain composed of HP AlphaServer DS20L nodes, one disk to hold the base Tru64 UNIX operating system

Note:

Do not configure a quorum disk in HP AlphaServer SC Version 2.6.

The remaining storage capacity of the external storage subsystem can be configured for user data storage and may be served by any connected node.

System storage can be provided by MSA1000-based storage arrays (see Section 6.7.1 on page 6–13), HSG-based storage arrays (see Section 6.7.2 on page 6–15), or HSV-based storage arrays (see Section 6.7.3 on page 6–17).

See Chapters 2 and 3 of the *HP AlphaServer SC Installation Guide* for more information on how to configure external storage.

6.6.2.2 Data Storage

Data storage is optional, but all data storage must be connected to the FS domains. HP recommends that you use MSA1000-based storage arrays (see Section 6.7.1 on page 6–13) and/or HSV-based storage arrays (see Section 6.7.3 on page 6–17) for data storage. Although you can use a combination of these storage array types in your SC system, only one type of storage array should be used within an individual PFS file system, to ensure consistent performance.

See Chapters 2 and 3 of the *HP AlphaServer SC Installation Guide* for more information on how to configure external storage.

6.6.2.3 External Storage Hardware Products

The HP AlphaServer SC system supports Switched Fibre Channel solutions *via* the StorageWorks products that are described in Table 6–1.

Table 6–1 Supported RAID Products

Products	Configuration	Host Adapters	Controllers[1]
ESA and EMA Families	Switched Fibre Channel	KGPSA-CA/DA	2 x HSG80
EVA Family	Switched Fibre Channel	KGPSA-CA/DA	2 x HSV110
Modular SAN Array 1000	Switched Fibre Channel	KGPSA-DA	2 x MSA1000

[1]Controllers and nodes are connected to single or dual 8-port or 16-port Fibre Channel switches.

6.7 External Data Storage Configuration

The information in this section is organized as follows:

- MSA1000 Controllers — Active/Standby Mode (see Section 6.7.1 on page 6–13)
- HSG-Based Storage Arrays — Multiple-Bus Failover Mode (see Section 6.7.2 on page 6–15)
- HSV-Based Storage Arrays — Multipathing Support (see Section 6.7.3 on page 6–17)

6.7.1 MSA1000 Controllers — Active/Standby Mode

Active/Standby mode has the following characteristics:

- Only one MSA1000 controller is active; the other is on standby
- Each controller has one port

- Units are presented by the MSA1000 on the active controller
- Each host with two host bus adapters has one active path to the units at any given time
- Hosts control the failover process by moving unit(s) from one controller to the other.

A typical active/standby configuration based on the MSA100 is shown in Figure 6–3.

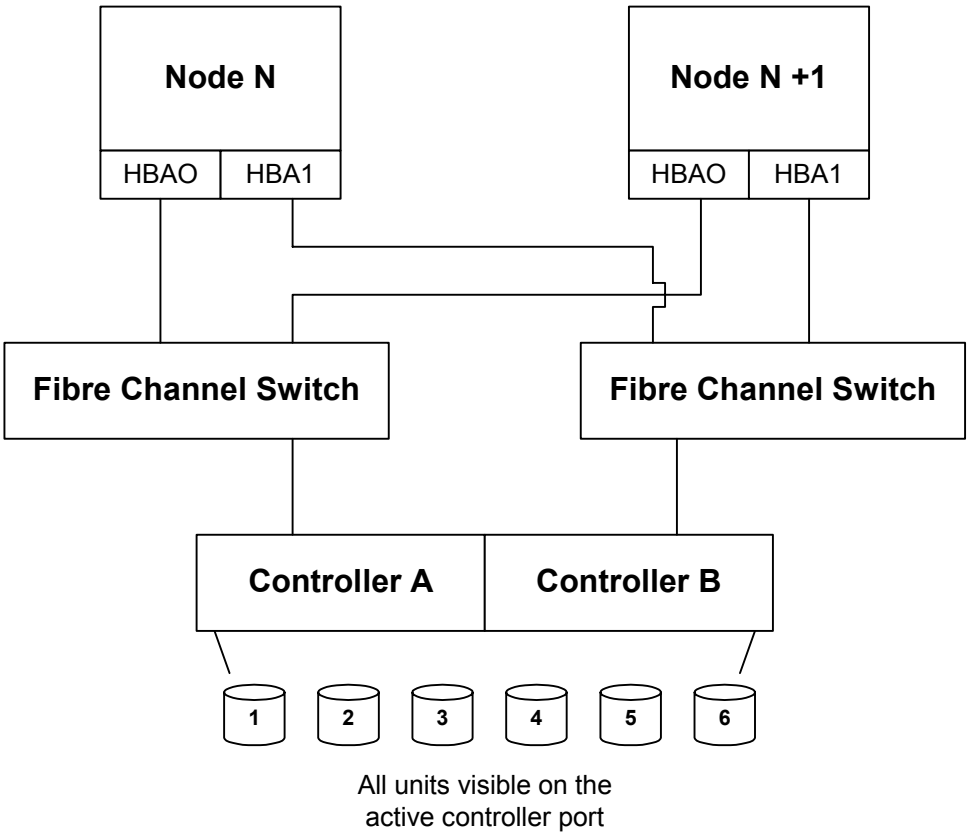

Figure 6–3 Typical Active/Standby Mode Configuration on MSA1000

The configuration shows two file server nodes connected to the same fabric, and storage for both nodes. At least two nodes must be connected to the storage fabric in order to ensure availability in the case of failure.

Each node has two connections to the storage from two different host bus adapters (HBAs). Each MSA1000 controller has one port connected to one switch, and the two controllers are connected to separate switches.

6.7.2 HSG-Based Storage Arrays — Multiple-Bus Failover Mode

Multiple-bus failover mode has the following characteristics:

- Hosts control the failover process by moving the unit(s) from one controller to another
- All units (0 through 199) are visible at all host ports, but accessible only through one controller at any specific time
- Each host has two or more paths to the units

Each host must have special software to control failover. With this software, the host sees the same units visible through two (or more) paths. When one path fails, the host can issue commands to move the units from one path to another.

In multiple-bus failover mode, you can specify which units are normally serviced by a specific controller of a controller pair. This process is called preferring or preferment.

Units can be preferred to one controller or the other by using the PREFERRED_PATH switch of the ADD UNIT or SET UNIT command. For example, use the following command to prefer unit D1 to 'this controller':

```
HSG80> SET D1 PREFERRED_PATH=THIS
```

Note:

This is a temporary, initial preference, which can be overridden by the host(s).

Remember the following points when configuring controllers for multiple-bus failover:

- Multiple-bus failover can compensate for a failure in any of the following:
  - Controller
  - Switch or hub
  - Fibre Channel link
  - Fibre Channel Host Bus Adapter cards (HBAs)
- A host can re-distribute the I/O load between the controllers

A typical multiple-bus failover configuration is shown in Figure 6–4.

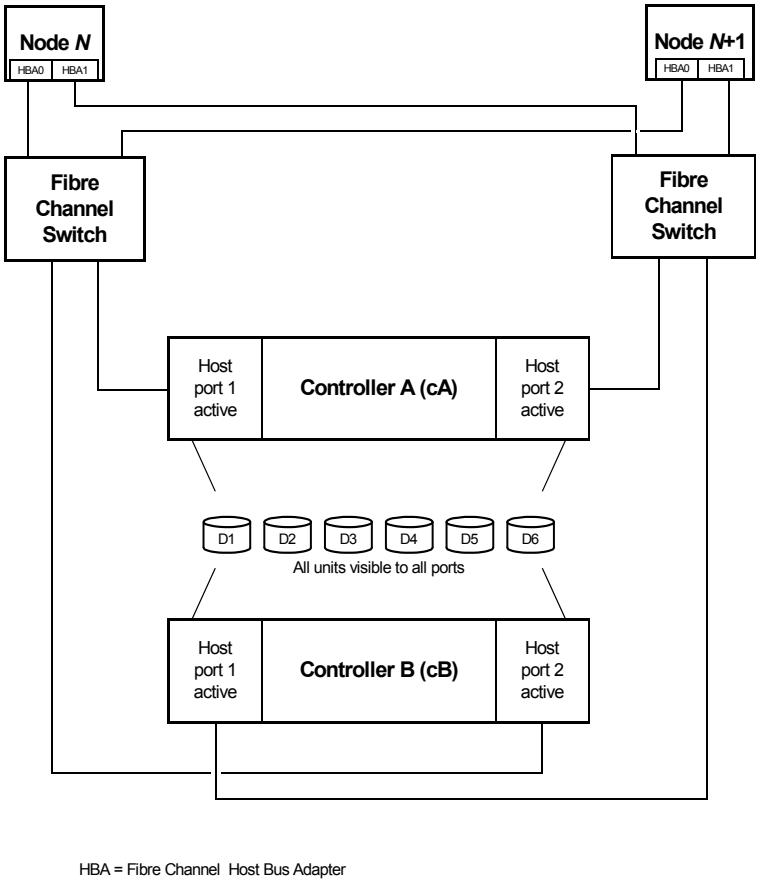

Figure 6–4 Typical Multiple-Bus Failover Configuration

The configuration shows two file server nodes connected to the same storage fabric, and storage for both nodes. At least two nodes must be connected to the storage fabric to ensure availability in the case of failure.

Each node has two connections to the storage from two different host bus adapters (HBAs). Two HBAs are used for bandwidth and resilience. This is optimal, but is not mandatory — if one adapter is used, access to the storage can be maintained, in the case of adapter loss, *via* DRD.

Each HBA is connected to two different switches, again for failure resilience.

Each switch has two connections to each RAID array. The RAID array has two controllers (A and B), each of which has two ports. For fully redundant configuration, the cabling from the switch to the controller should be as shown in Figure 6–4.

In multibus failover mode, this configuration provides the best bandwidth and resilience.

6.7.3 HSV-Based Storage Arrays — Multipathing Support

The Enterprise Virtual Array supports a multipathing environment (high availability). For Tru64 UNIX, the multipathing is native to the operating system. No additional software installation is required.

Figure 6–5 shows a block diagram of how the whole storage system works:

- The HSV controller pair connects to two Fibre Channel fabrics, to which the hosts also connect.
- The HSV Element Manager is the software that controls the storage system. It resides on the SANworks Management Appliance. The SANworks Management Appliance connects into the Fibre Channel fabric.
- The controller pair connects to the physical disk array through Fibre Channel arbitrated loops. There are two separate loop pairs: loop pair 1 and loop pair 2. Each loop pair consists of 2 loops, each of which runs independently, but which can take over for the other loop in case of failure. The actual cabling of each loop is shown in Appendix A of the HP *StorageWorks Enterprise Virtual Array Initial Setup User Guide*.

For more information about setting up external data storage on HSV disks, see the HP *SANworks Tru64 UNIX Kit for Enterprise Virtual Array Installation and Configuration Guide*.

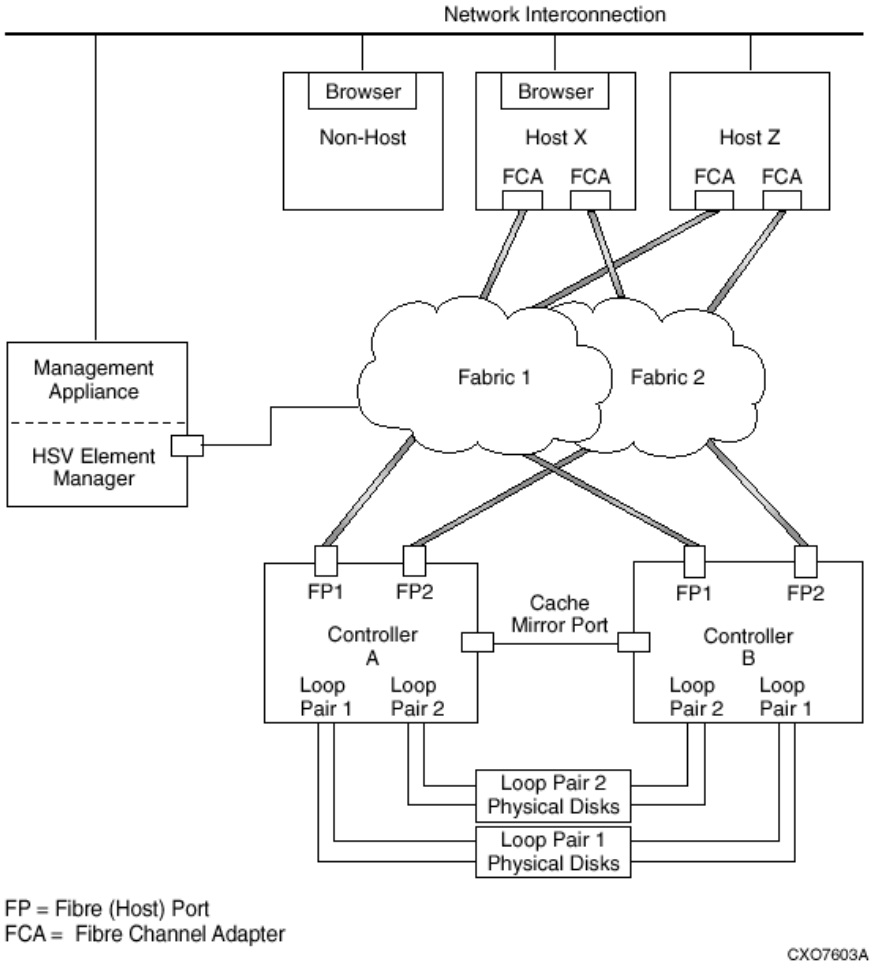

Figure 6–5 Overview of Enterprise Virtual Array Component Connections

7

Managing the SC File System (SCFS)

The SC file system (SCFS) provides a global file system for the HP AlphaServer SC system.

This information in this chapter is arranged as follows:

- SCFS Overview (see Section 7.1 on page 7–2)
- SCFS Configuration Attributes (see Section 7.2 on page 7–3)
- Creating SCFS File Systems (see Section 7.3 on page 7–7)
- The scfsmgr Command (see Section 7.4 on page 7–8)
- Access Control Lists (see Section 7.5 on page 7–27)
- SysMan Menu (see Section 7.6 on page 7–27)
- Monitoring and Correcting File-System Failures (see Section 7.7 on page 7–28)
- Tuning SCFS (see Section 7.8 on page 7–32)
- SC Database Tables Supporting SCFS File Systems (see Section 7.9 on page 7–35)

7.1 SCFS Overview

The HP AlphaServer SC system is comprised of multiple domains. There are two types of domains: File-Serving (FS) domains and Compute-Serving (CS) domains. HP AlphaServer SC Version 2.6 supports a maximum of four FS domains. The number of FS domains is specified when the `sra setup` command is first run during the installation process. If you wish, you can later convert a CS domain to an FS domain (subject to the supported maximum of four FS domains); for more information, see Section 21.5 on page 21–12.

The SCFS file system exports file systems from an FS domain to the other domains. Therefore, it provides a global file system across all nodes of the HP AlphaServer SC system. The SCFS file system is a high-performance file system that is optimized for large I/O transfers. When accessed *via* the FAST mode, data is transferred between the client and server nodes using the HP AlphaServer SC Interconnect network for efficiency.

SCFS file systems may be configured by using the `scfsmgr` command (see Section 7.4 on page 7–8) or by using SysMan Menu (see Section 7.6 on page 7–27). You can use the `scfsmgr` command or SysMan Menu, on any node or on a management server (if present), to manage all SCFS file systems. The system automatically reflects all configuration changes on all domains. For example, when you place an SCFS file system on line, it is mounted on all domains.

The underlying storage of an SCFS file system is an AdvFS fileset on an FS domain. Within an FS domain, access to the file system from any node is managed by the CFS file system and has the usual attributes of CFS file systems (common mount point, coherency, and so on). An FS domain serves the SCFS file system to nodes in the other domains. In effect, an FS domain exports the file system, and the other domains import the file system.

This is similar to — and, in fact, uses features of — the NFS system. For example, `/etc/exports` is used for SCFS file systems. The mount point of an SCFS file system uses the same name throughout the HP AlphaServer SC system so there is a coherent file name space. Coherency issues related to data and metadata are discussed later.

The SCFS kernel subsystem can make use of multiple Elan rails, if they are present on the system, to perform SCFS I/O data transfers. This allows SCFS clients to load-balance their requests over all the rails that they are permitted to use and can see the server nodes on. For more information on using multiple rails, see Section 7.8.2.3 on page 7–34.

See the `scfs(4)` reference page for more information on SCFS.

7.2 SCFS Configuration Attributes

The SC database contains SCFS configuration data. The `/etc/fstab` file is not used to manage the mounting of SCFS file systems. However, the `/etc/exports` is used for this purpose. Use SysMan Menu or the `scfsmgr` command to edit this configuration data — do not update the contents of the SC database directly. Do not add entries to, or remove entries from, the `/etc/exports` file. Once entries have been created, you can edit the `/etc/exports` file in the usual way.

An SCFS file system is described by the following attributes:

- AdvFS domain and fileset name

  This is the name of the AdvFS domain and fileset that contains the underlying data storage of an SCFS file system. This information is only used by the FS domain that serves the SCFS file system. However, although AdvFS domain and fileset names generally need only be unique within a given domain, the SCFS system uses unique names. Therefore, the AdvFS domain and fileset name must be unique across the HP AlphaServer SC system.

  In addition, HP recommends the following conventions:

  – You should use only one AdvFS fileset in an AdvFS domain.

  – The domain and fileset names should use a common root name. For example, an appropriate name would be `data_domain#data`.

  SysMan Menu uses these conventions. The `scfsmgr` command allows more flexibility.

- Mountpoint

  This is the pathname of the mountpoint for the SCFS file system. This is the same on all domains in the HP AlphaServer SC system.

- Preferred Server

  This specifies the node that normally serves the file system. When an FS domain is booted, the first node that has access to the storage will mount the file system. When the preferred server boots, it takes over the serving of that storage. For best performance, the preferred server should have direct access to the storage. The `cfsmgr` command controls which node serves the storage.

- Mount Options

  In HP AlphaServer SC Version 2.5 and lower, the only available mount options for SCFS mounts were `rw` (read/write) or `ro` (read only). In HP AlphaServer SC Version 2.6, the mount option arguments available for SCFS mounts are similar to those available for the Tru64 UNIX `mount -o` command, with the exception of the Tru64 UNIX `server_only` option, which is not supported for SCFS mounts. The mount option information is stored in the `rw` field of the `sc_scfs` table.

For more information, see Section 7.4.1.1 on page 7–11 and the `mount(8)` reference page.

- FAST or UBC

  SCFS network packets are either pure data packets or metadata packets. Metadata packets are standard NFS packets and always follow the standard NFS path — over either UDP or TCP — between the NFS importing node on the CS domain and the file-serving nodes. Within a CS domain, CFS redirects metadata-related operations from any other CS node to the importing node.

  Pure data packets may take two distinct routes; the route taken depends on the SCFS mount mode, which can be FAST or UBC. If mounted as FAST, the data packets will be sent directly between the CS node and the file server node through the HP AlphaServer SC Interconnect, bypassing the standard NFS path. If mounted as UBC, the data packets will follow the same route as the metadata packets.

  The FAST and UBC modes are each suitable for different types of data transfer, as follows:

  – SCFS FAST

    The FAST mode is suited to large data transfers where bypassing the UBC provides better performance. In addition, since accesses are made directly to the serving node, writes from multiple client nodes to distinct areas of a file are serialized so that they are seen correctly at the server, and data coherency is preserved. (In UBC mode, multiple writes to the same page may not be written as intended, due to the read/modify/write characteristics of NFS).

    In FAST mode, multiple readers of the same data will all have to obtain the data individually from the server node since the UBC is bypassed on the client nodes.

    When FAST mode is used, a data integrity mechanism in SCFS ensures that data written to a file is valid, even if file server nodes crash. This mechanism replays write requests from clients that have not been committed on the server.

    While a file is opened *via* the FAST mode, all subsequent file `open()` calls on that domain will inherit the FAST attribute even if not explicitly specified.

  – SCFS UBC

    The UBC mode is suited to small data transfers, such as those produced by formatted writes in Fortran. Data coherency has the same characteristics as NFS.

    If a file is currently opened *via* the UBC mode, and a user attempts to open the same file *via* the FAST mode, an error (EINVAL) is returned to the user.

  Whether the SCFS file system is mounted FAST or UBC, in HP AlphaServer SC Version 2.6 the access for individual files can be overridden in a number of ways, as follows:

  – If the file is opened with the O_SCFSIO option (defined in <`sys/scfs.h`>), access is *via* the FAST path.

- If the file is opened with the O_UBCIO option (defined in <sys/scfs.h>), access is *via* the UBC path.
- If the CFS subsystem io_env attribute on the client is set to 1, and the SCFS_IO environment variable is set to FAST, access is *via* the FAST path.
- If the CFS subsystem io_env attribute on the client is set to 1, and the SCFS_IO environment variable is set to UBC, access is *via* the UBC path.
- If the file has an executable bit set[1], access is *via* the UBC path.

The first of these conditions that is found to be true determines the access path for the file. If none of these conditions is found to be true, the file is opened according to the default FAST/UBC mode.

It is possible to change the default FAST/UBC mode for an existing SCFS file system without rebooting the entire system. For more information, see Section 7.2.1 on page 7–7.

- ONLINE or OFFLINE

  You do not directly mount or unmount SCFS file systems. Instead, you mark the SCFS file system as ONLINE or OFFLINE. When you mark an SCFS file system as ONLINE, the system will mount the SCFS file system on all domains. When you mark the SCFS file system as OFFLINE, the system will unmount the file system on all domains.

  The state is persistent. For example, if an SCFS file system is marked ONLINE and a node or domain is restarted, the SCFS file system will be mounted as soon as the system has completed booting.

- Mount Status

  This indicates whether an SCFS file system is mounted or not. This attribute is specific to a domain (that is, each domain has a mount status). The mount status values are listed in Table 7–1.

Table 7–1 SCFS Mount Status Values

Mount Status	Description
mounted	The SCFS file system is mounted on the domain.
not-mounted	The SCFS file system is not mounted on the domain.

1. Note that mmap() operations are not supported for FAST files. This is because mmap() requires the use of UBC. Executable binaries are normally mmap'd by the loader. The exclusion of executable files from the default mode of operation allows binary executables to be used in an SCFS FAST file system.

Table 7–1 SCFS Mount Status Values

Mount Status	Description
`mounted-busy`	The SCFS file system is mounted, but an attempt to unmount it has failed because the SCFS file system is in use. When a PFS file system uses an SCFS file system as a component, the SCFS file system is in use and cannot be unmounted until the PFS file system is also unmounted. In addition, if a CS domain fails to unmount the SCFS, the FS domain does not attempt to unmount the SCFS, but instead marks it as `mounted-busy`.
`mounted-stale`	The SCFS file system is mounted, but the FS domain that serves the file system is no longer serving it. Generally, this is because the FS domain has been rebooted — for a period of time, the CS domain sees `mounted-stale` until the FS domain has finished mounting the AdvFS file systems underlying the SCFS file system. The `mounted-stale` status only applies to CS domains.
`mount-not-served`	The SCFS file system was mounted, but all nodes of the FS domain that can serve the underlying AdvFS domain have left the domain.
`mount-failed`	An attempt was made to mount the file system on the domain, but the mount command failed. When a mount fails, the reason for the failure is reported as an event of class `scfs` and type `mount.failed`. See Chapter 9 for details of how to access this event type.
`mount-noresponse`	The file system is mounted; however, the FS domain is not responding to client requests. Usually, this is because the FS domain is shut down.
`mounted-io-err`	The file system is mounted, but when you attempt to access it, programs get an I/O Error. This can happen on a CS domain when the file system is in the `mount-not-served` state on the FS domain.
`unknown`	Usually, this indicates that the FS domain or CS domain is shut down. However, a failure of an FS or CS domain to respond can also cause this state.

Note:

In HP AlphaServer SC Version 2.6, SCFS file systems exported from an FS domain are mounted from preferred server cluster aliases (for example, `atlasD0-atlas2`). In HP AlphaServer SC Version 2.5, the default cluster alias (for example, `atlasD0`) was used to mount SCFS file systems.

The attributes of SCFS file systems can be viewed using the `scfsmgr show` command, as described in Section 7.4.8 on page 7–17.

7.2.1 Changing SCFS File System Mode (FAST/UBC)

Normally, when you change the default FAST/UBC mode for an existing SCFS file system in HP AlphaServer SC Version 2.6, the change does not come into effect until you reboot the entire system. However, there is an alternative method of bringing the change into effect; using this method, you will not need to reboot the entire system.

To change the default mode without rebooting the entire system, perform the following steps:

1. Take the file system offline by running the `scfsmgr offline` command, as shown in the following example:
   ```
   # scfsmgr offline /scfs0
   ```
2. When the file system is offline on all nodes, flush the kernel cache for the file system on all nodes, as shown in the following example:
   ```
   # scrun -n all /usr/sbin/scfsctrl unmount -s atlasD0 -p /scfs0
   ```
3. Change the value in the `speed` field of the `sc_scfs` table to `UBC` or `FAST`, as shown in the following example:
   ```
   # rmsquery "update sc_scfs set speed='UBC' where mount_point='/scfs0'"
   ```
4. Bring the file system online by running the `scfsmgr online` command, as shown in the following example:
   ```
   # scfsmgr online /scfs0
   ```

For more information on using the `scfsmgr offline` and the `scfsmgr online` commands, see Section 7.4.4 on page 7–15 and Section 7.4.5 on page 7–15.

7.3 Creating SCFS File Systems

To create an SCFS file system, use the `scfsmgr` command (see Section 7.4 on page 7–8) or SysMan Menu (see Section 7.6 on page 7–27). This creates the AdvFS domain and fileset, and updates the `/etc/exports` file and the SC database (see Section 7.2 on page 7–3).

The general steps to create an SCFS file system are as follows:

1. Configure a unit (virtual disk) on one of the RAID systems attached to an FS domain. You should configure the unit so that it is appropriate to your needs (for example, RAIDset, Mirrorset). How you do this is transparent to SCFS. You must also designate the unit to its primary controller and set access paths so that only nodes on the FS domain can “see” the unit. See Chapter 6 for more information about storage. For information on how to convert a CS domain to an FS domain, see Section 21.5 on page 21–12.
2. Ensure that all nodes in the FS domain that have access to this storage are booted. This allows you to confirm that your access paths are correct.

3. At this stage you are ready to create the SCFS file system. You have two options:
   - Use the GUI `sysman scfsmgr` command and select the Create... option. This guides you through a series of steps where you pick the appropriate disk and various options.
   - Use the CLI `scfsmgr create` command. The syntax of the `scfsmgr` command is described below. The `scfsmgr create` command creates the AdvFS fileset, updates the `/etc/exports` file, updates the SC database, and mounts the file system on all available domains.

7.4 The scfsmgr Command

The `scfsmgr` command is a tool that allows you to create, manage, and delete SCFS file systems. The `scfsmgr` command can be used on any node, including a management server (if present), to manage SCFS file systems across all domains in the HP AlphaServer SC system.

It is important to note that all mounting and unmounting of SCFS and PFS file systems is done by the `scmountd` daemon as part of a `sync` operation. The `scfsmgr online/offline` and `pfsmgr online/offline` commands do not actually mount or unmount file systems; these commands update the desired state of the file systems in the database, trigger a `sync` operation, and return to the user. The mounting or unmounting of the file systems happens some time later when the `sync` operation finds that the actual state of a file system does not match its desired state.

A `sync` operation is triggered in the following circumstances:

- When an administrator changes the desired state of a file system by using the `online` and `offline` commands (see Section 7.4.4 on page 7–15 and Section 7.4.5 on page 7–15)
- When certain events occur, such as when a domain is booted or an importing node in a CS domain crashes
- When the `scfsmgr sync` command is run (see Section 7.4.10 on page 7–19)

During the `sync` operation, the `scmountd` daemon examines all SCFS and PFS file systems on all domains and nodes of the HP AlphaServer SC system, and mounts or unmounts the file systems as required to achieve their desired states as defined in the database. Note that the `sync` operation itself is asynchronous and is carried out in the background by the `scmountd` daemon, which runs on the management server (if present) or domain 0. You can use the `scfsmgr status` command to monitor the progress of the `sync` operation.

Only one `sync` operation can be in progress at any time. Issuing further `online` or `offline` commands during a `sync` operation simply updates the desired states in the database and queues more `sync` requests.

When the sync operation has completed, all file systems should be in their desired state on all nodes and domains. However, sometimes a sync operation cannot change the state of a file system. For example, a file system that is in use cannot be unmounted, and a sync operation will not be able to take a disk offline if user jobs have files open on the disk. In such a case, the sync operation marks the file system as mounted-busy, and proceeds to the next step in the operation. If some file systems are not in their desired state when a sync operation has completed, you must investigate what caused the mount or unmount operation to fail, and manually resolve the problem that is preventing the sync operation from succeeding (for example, by killing the user jobs that are preventing a disk from being taken offline). When you have resolved the problem, trigger the sync operation again by running the scfsmgr sync command (see Section 7.4.10 on page 7–19).

The syntax of the scfsmgr command is as follows:

```
scfsmgr <command> <command arguments>
```

This section describes the following scfsmgr commands:

- scfsmgr create (see Section 7.4.1 on page 7–9)
- scfsmgr destroy (see Section 7.4.2 on page 7–13)
- scfsmgr export (see Section 7.4.3 on page 7–14)
- scfsmgr offline (see Section 7.4.4 on page 7–15)
- scfsmgr online (see Section 7.4.5 on page 7–15)
- scfsmgr scan (see Section 7.4.6 on page 7–16)
- scfsmgr server (see Section 7.4.7 on page 7–16)
- scfsmgr show (see Section 7.4.8 on page 7–17)
- scfsmgr status (see Section 7.4.9 on page 7–18)
- scfsmgr sync (see Section 7.4.10 on page 7–19)
- scfsmgr sync_domain (see Section 7.4.11 on page 7–19)
- scfsmgr upgrade (see Section 7.4.12 on page 7–20)
- scfsmgr try_mount (see Section 7.4.13 on page 7–20)
- scfsmgr try_unmount (see Section 7.4.14 on page 7–21)
- scfsmgr Quota Commands (see Section 7.4.15 on page 7–22)

The scfsmgr command is also described in the scfsmgr(8) reference page.

7.4.1 scfsmgr create

The scfsmgr create command creates an SCFS file system.

This command:

- Creates the AdvFS fileset and file domain that will be used by SCFS.

- Updates the SC database and the /etc/exports file (on the serving FS domain) to reflect the addition of a new SCFS file system to the domain.
- Creates the mount point associated with the file system, and sets the permissions.

The syntax of this command is as follows:

```
scfsmgr create name mountpoint domain
common_options:[advfs_options]:[nfs_options]
server FAST|UBC owner group permissions volume
```

where:

- *name* is the name of the AdvFS file system to be created (Example: data_domain#data)
- *mountpoint* is the mount point — scfsmgr creates this if it does not exist; the pathname of the mount point cannot be within another SCFS file system (Example: /data)
- *domain* is the name of the FS domain (Example: atlasD0)
- *common_options:advfs_options:nfs_options* specifies the mount options for the file system (for more information on specifying mount options, see Section 7.4.1.1 on page 7–11)
- *server* is the name of the preferred server on the FS domain (Example: atlas3)
- FAST|UBC specifies how the file system is mounted on the other domains
- *owner* is the owner of the mount point (Example: root)
- *group* is group of the mount point (Example: system)
- *permissions* is the permissions of the mount point (Example: 755)
- *volume* is the name of a disk partition or LSM volume used for the creation of the AdvFS domain (for example, /dev/disk/dsk10c). This is the first volume of the AdvFS domain. Additional volumes can be added to the AdvFS domain by using the addvol command.

When the scfsmgr create command is complete, the AdvFS domain and fileset exist and the mountpoint is created. However, the SCFS file system is in the OFFLINE state; therefore, it is not mounted. To mount the SCFS file system, use the scfsmgr online command (see Section 7.4.5 on page 7–15).

Examples:

```
# scfsmgr create advfs_domain_1#s_test_mnt_1 /s_test_mnt_1 atlasD0 \
rw atlas0 FAST root system 755 /dev/disk/dks1c
```

```
# scfsmgr create advfs_domain_2#s_test_mnt_2 /s_test_mnt_2 atlasD0 \
rw atlas1 FAST root system 755 /dev/disk/dsk8c
```

7.4.1.1 Specifying Mount Options

An SCFS file system is mounted as AdvFS on the serving node and imported as NFS on other nodes. In HP AlphaServer SC Version 2.6, you can specify options for mounts on the serving node (`advfs_options`), options for mounts on the importing nodes (`nfs_options`), and options that are common to both (`common_options`). The mount option information is stored in the `rw` field of the `sc_scfs` table.

The mount option arguments available for SCFS mounts are similar to those available for the Tru64 UNIX `mount -o` command, with the exception of the `server_only` argument, which is not supported by the `scfsmgr` command. For more information on mount options, please see the `mount(8)` reference page.

Note:

The `server_only` AdvFS argument is not supported by the `scfsmgr` command, and should not be used with `scfsmgr`. The `server_only` argument makes the mounted file system accessible only to the node that is performing the mount. Other nodes in the domain cannot read or write to the file system.

The format of the field is as follows:
common_options:advfs_options:nfs_options

where:

- *common_options* are options that apply to both AdvFS and NFS mounts.

Note:

In the `common_options`, you must specify whether the mount is to be read/write (`rw`) or read only (`ro`). This is the only mandatory option; all other mount options are optional.

- *advfs_options* are options that apply to SCFS mounts on the serving node.
- *nfs_options* are options that apply to SCFS mounts on the importing nodes.
- A colon (`:`) is used to separate the option types.
- A comma (`,`) is used to separate the arguments within the option types.

Examples are given below. In these examples, the name of the AdvFS mount is `file#system` and the mount point is considered to be /`mount_point`. The serving domain is `atlasD0`.

Example 7–1 Common Options Only Specified

```
rw
```

In this case, the `rw` option is applied to both AdvFS and NFS mounts. The commands used at mount time are as follows:

For AdvFS (serving node):
```
/usr/sbin/mount -o rw file#system /mount_point
```
For NFS (importing nodes):
```
/usr/sbin/mount -o rw atlasD0-atlas0:/mount_point /mount_point
```
Equivalent forms of the field are:
```
rw::
rw:
```

Example 7–2 Common Options and AdvFS Options Specified

```
rw:noexec
```

In this case the `rw` option is applied to both types of mounts and the `noexec` option is applied only to AdvFS mounts. The commands used at mount time are as follows:

For AdvFS (serving node):
```
/usr/sbin/mount -o rw,noexec file#system /mount_point
```
For NFS (importing nodes):
```
/usr/sbin/mount -o rw atlasD0-atlas0:/mount_point /mount_point
```
Equivalent forms of the options are:
```
rw:noexec:
```

Example 7–3 Common Options and NFS Options Specified

```
rw::proplist
```

In this case, the `rw` option is applied to both the AdvFS mounts (serving node) and the NFS mounts (importing nodes), and the `proplist` option is applied only to NFS mounts.

Example 7–4 Common Options, AdvFS Options, and NSF Options Specified

```
rw,sync:noexec:proplist
```

In this case, the commands used at mount time are as follows:

For AdvFS (serving node):

```
/usr/sbin/mount -o rw,sync,noexec file#system /mount_point
```

For NFS (importing nodes):

```
/usr/sbin/mount -o rw,sync,proplist atlasD0-atlas1:/mount_point /
mount_point
```

It is important that you do not specify NFS mount options in the `advfs_options` part of the field.

Changing the Contents of the rw Field

You can change the contents of the `rw` field using `rmsquery` as in the following examples:

```
# rmsquery "update sc_scfs set rw='rw::proplist'"
```

This sets the options for all SCFS mounts to the options described in Example 7–3.

```
# rmsquery "update sc_scfs set rw='rw::proplist' where mount_point='/fs/scfs_1'"
```

This sets the mount options for the file system mounted in `/fs/scfs_1` to the options described in Example 7–4.

Note:

When you make changes to the `rw` field of the `sc_scfs` table, the changes are not reflected automatically on file systems that are already online. To activate the changes, you must unmount the affected file systems by marking them as OFFLINE, and then mount them again by marking them as ONLINE.

For example, if you change the `rw` field for the `/fs/scfs_1` file system while the file system is mounted, you will need to unmount and remount the file system as follows after you have made the changes to the `rw` field:

```
# scfsmgr offline /fs/scfs_1
# scfsmgr online /fs/scfs_1
```

7.4.2 scfsmgr destroy

The `scfsmgr destroy` command deletes an entry from the SCFS configuration, and optionally deletes the associated AdvFS fileset and file domain.

The syntax of this command is as follows:

`scfsmgr destroy` *`mountpoint`* `[all]`

If the `all` keyword is specified after the mountpoint, the command attempts to delete the AdvFS fileset and file domain, using the `rmfset` and `rmfdmn` commands.

If you do not specify the `all` keyword, the command removes only the SCFS file system from the SC database, and the associated entry from the `/etc/exports` file on the serving domain.

To relocate an SCFS mountpoint, run the `scfsmgr destroy` command (without the `all` keyword) to delete the existing entry, and then run the `scfsmgr export` command to export the AdvFS fileset on a different path.

7.4.3 scfsmgr export

Use the `scfsmgr export` command to export an existing file system *via* SCFS. The operation of this command is equivalent to `scfsmgr create` except that the creation of the AdvFS filesystem is skipped.

This command:

- Updates the SC database and the `/etc/exports` file (on the serving FS domain) to reflect the addition of a new SCFS file system to the domain.
- Creates the mount point associated with the file system, and sets the permissions.

The syntax of this command is as follows:

```
scfsmgr export name mountpoint domain
common_options:[advfs_options]:[nfs_options]
server FAST|UBC owner group permissions
```

where:

- *name* is the name of the AdvFS file system to be exported
  (Example: `data_domain#data`)
- *mountpoint* is the mount point — `scfsmgr` creates this if it does not exist
  (Example: `/data`)
- *domain* is the name of the FS domain
  (Example: `atlasD0`)
- *common_options:advfs_options:nfs_options* specifies the mount options for the file system (for more information on specifying mount options, see Section 7.4.1.1 on page 7–11)
- *server* is the name of the preferred server on the FS domain
  (Example: `atlas3`)
- `FAST|UBC` specifies how the file system is mounted on the other domains
- *owner* is the owner of the mount point
  (Example: `root`)

- *group* is group of the mount point
  (Example: system)
- *permissions* is the permissions of the mount point
  (Example: 755)

When the scfsmgr export command is complete, the AdvFS domain and fileset exist and the mountpoint is created. However, the SCFS file system is in the OFFLINE state; therefore, it is not mounted. To mount the SCFS file system, use the scfsmgr online command (see Section 7.4.5 on page 7–15).

7.4.4 scfsmgr offline

The scfsmgr offline command changes the desired state of the specified SCFS file systems to OFFLINE in the database and triggers a sync operation. The scfsmgr offline command completes as soon as the sync request has been sent to the scmountd daemon; the sync operation itself is carried out in the background, and there may be a delay of a few minutes before the file systems are actually unmounted. During this time, you can monitor the progress of the sync operation using the scfsmgr status command.

During the sync operation, the scmountd daemon examines all SCFS and PFS file systems on all domains and nodes of the HP AlphaServer SC system, and mounts or unmounts the file systems as required to achieve their desired states as defined in the database. For more information on the sync operation, see Section 7.4 on page 7–8.

Only one sync operation can be in progress at a time; issuing further online or offline commands while a sync operation is in progress will only queue more sync requests.

If an SCFS file system is a component of a Parallel File System (PFS) (see Chapter 8), the scfsmgr command will mark the SCFS file system as OFFLINE. However, the SCFS file system will not be unmounted by an FS or CS domain until the PFS file system is marked as OFFLINE and the PFS file system is unmounted by all nodes in the domain.

The syntax of this command is as follows:
scfsmgr offline *mountpoint*|all

If the keyword all is specified, the command marks all SCFS file systems as OFFLINE.

7.4.5 scfsmgr online

The scfsmgr online command changes the desired state of the specified SCFS file systems to ONLINE in the database and triggers a sync operation. The scfsmgr online command completes as soon as the sync request has been sent to the scmountd daemon; the sync operation itself is carried out in the background, and there may be a delay of a few minutes before the file systems are actually mounted. During this time, you can monitor the progress of the sync operation using the scfsmgr status command

During the `sync` operation, the `scmountd` daemon examines all SCFS and PFS file systems on all domains and nodes of the HP AlphaServer SC system, and mounts or unmounts the file systems as required to achieve their desired states as defined in the database. For more information on the `sync` operation, see Section 7.4 on page 7–8.

Only one `sync` operation can be in progress at a time; issuing further `online` or `offline` commands while a `sync` operation is in progress will only queue more `sync` requests.

The syntax of this command is as follows:

```
scfsmgr online mountpoint|all
```

If the keyword `all` is specified, the command marks all SCFS file systems as ONLINE.

7.4.6 scfsmgr scan

SysMan Menu must know the names of available disks or LSM volumes so that it can guide you through the creation process for an SCFS file system. Use the `scfsmgr scan` command to load disk and LSM data into the SC database.

The syntax of this command is as follows:

```
scfsmgr scan
```

On a large system, this command may take a long time to run.

The data is only needed by SysMan Menu. The `scfsmgr` command does not use this data. If you add or remove storage, rerun the `scfsmgr scan` command.

7.4.7 scfsmgr server

The `scfsmgr server` command changes the preferred server of the file system on an FS domain.

This command:

- Updates the SC database to reflect the new preferred server for the file system.
- Relocates the serving of the currently served file system to the new preferred server, if the new preferred system is available, on the specified file domain. If the new preferred server is unable to locally access all disks, the file system is not actually migrated to the preferred server (that is, the `-f` option is not used in the `cfsmgr` command).

The `scfsmgr server` command completes as soon as the `scmountd` daemon is informed — while the new preferred server for the file system is recorded in the SC database, the actions to migrate the file system happen later.

The syntax of this command is as follows:

```
scfsmgr server mountpoint preferred-server
```

7.4.8 scfsmgr show

The `scfsmgr show` command allows you to view the status of SCFS file systems.

The syntax of this command is as follows:
`scfsmgr show [`*`mountpoint`*`]`

If *`mountpoint`* is not specified, the command shows a summary status of all SCFS file systems. If *`mountpoint`* is specified, the command shows detailed status of the specified SCFS file system.

The following example shows the command output when the mountpoint is not specified:

```
# scfsmgr show
State       Mountpoint     Server      Mount status
-----       ----------     ------      ------------
online      /data1         atlas2      mounted: atlasD[0-2] not-mounted: atlasD3
online      /data2         atlas3      mounted: atlasD[0-3]
online      /scr1          !atlas4!    mounted: atlasD[0-3]
offline     /scr2          atlas4      not-mounted: atlasD[0-3]
```

The mount status is shown in summary format. For example, `/data1` is mounted on `atlasD0`, `atlasD1`, and `atlasD2`, but not mounted on `atlasD3`.

The name of the node that is serving the underlying AdvFS file system is also shown. If the node is not the preferred file server, the name is enclosed within exclamation marks (!). For example, `/scr1` is served by `atlas4`, but `atlas4` is not the preferred server.

When the FS domain has not mounted a file system, the name of the preferred server is shown. For example, `atlasD0` has not mounted `/scr2` (because it is offline). There is no actual server; therefore, the preferred server (`atlas4`) is shown.

The following example shows the command output when a mountpoint is specified:

```
# scfsmgr show /data1
Mountpoint: /data1
Filesystem: data1_domain#data1
Preferred Server: atlas2
Attributes: FAST  rw
Fileserving Domain State:
          Domain            Server               State
          atlasD0           atlas2               mounted
Importing Domains:
          Domain            Mounted On           State
          atlasD1           atlas32              mounted
          atlasD2           atlas65              mounted
          atlasD3                                not-mounted
```

If `/data1` had been a component of a PFS file system (see Chapter 8), the name and state of the PFS file system would also have been shown.

7.4.9 scfsmgr status

The `scfsmgr status` command shows the status of operations in the `scmountd` daemon. This command is useful when you have just issued an `scfsmgr online` or `scfsmgr offline` command. Normally, shortly after you issue the command, the `scfsmgr show` command would reflect corresponding changes in the system. For example, if you mark an SCFS file system as OFFLINE, you should see the mount status change to `not-mounted` on all domains (SysMan Menu is useful for this as it periodically refreshes the data).

However, sometimes an action may appear to take a long time to complete. There are several reasons for this:

- A domain may be booting. If any node in a domain is booting, actions to mount or unmount file systems are postponed until the domain completes booting. To see whether a node in a domain is being booted, use the `sramon` command. If a domain is booting, the `scmountd` daemon discards any command; therefore, the `scfsmgr status` command will show no command in progress. When the boot completes, the `srad` daemon sends a message to the `scmountd` daemon to initiate the actions.
- A domain may be slow in completing mount or unmount operations. If this happens, the `scfsmgr status` command will show a command in progress and you will be able to identify which domain is active.

The following example shows the command output from an idle system (that is, the `scmountd` daemon is idle):

```
# scfsmgr status
No command in progress
Domain: atlasD0 (0) state: unknown   command state: idle   name: (39);  timer: not set
Domain: atlasD1 (0) state: unknown   command state: idle   name: (40);  timer: not set
Domain: atlasD2 (0) state: unknown   command state: idle   name: (41);  timer: not set
Domain: atlasD3 (0) state: unknown   command state: idle   name: (42);  timer: not set
```

The following example shows the command output when the `scmountd` daemon is actively processing a command:

```
# scfsmgr status
Command in progress: sync state: scfs_mount_remote
Domain: atlasD0 (0)  state: responding  command state: finished name: scfs_mount_remote
(42); timer: not set
Domain: atlasD1 (0)  state: responding  command state: running name: scfs_mount_remote
(43); timer: expires in 40 secs
Domain: atlasD2 (1)  state: timeout  command state: idle name: scfs_mount_remote (59);
timer: not set
Domain: atlasD3 (1)  state: not-responding  command state: not-responding name:
scfs_mount_remote (43); timer: not set
```

In this example, a command is executing. The command name (`sync`) is an internal command name — it does not necessarily correspond with the name of an `scfsmgr` command. Each line shows the state of each domain.

In this example, atlasD0 has just finished running the scfs_mount_remote script. The script names are provided for support purposes only.

However, the state and timer information is useful. If the script is still running, it periodically updates the scmountd daemon so the timer is restarted. For example, the script running on atlasD1 is running and responding (command state is running; timer is set).

However, if the script fails to update the scmountd daemon, a timeout occurs. For example, atlasD2 has timed out. This is an unusual situation and must be investigated.

In this example, atlasD3 is not responding. This is normal if atlasD3 is shut down. If atlasD3 is running, the situation must be investigated.

7.4.10 scfsmgr sync

If you suspect at any time that the state of one or more file systems (as shown by the df or scfsmgr show commands) is not as you expect, run the scfsmgr sync command. The scfsmgr sync command sends a request to the scmountd daemon to trigger a sync operation to correct the state of the file systems. The scfsmgr sync command completes as soon as the sync request has been sent to the scmountd daemon; the sync operation itself is carried out in the background, and there may be a delay of a few minutes before the file system states are corrected. During this time, you can monitor the progress of the sync operation using the scfsmgr status command.

During the sync operation, the scmountd daemon examines all SCFS and PFS file systems on all domains and nodes of the HP AlphaServer SC system, and mounts or unmounts the file systems as required to achieve their desired states as defined in the database. For more information on the sync operation, see Section 7.4 on page 7–8.

A common use of the scfsmgr sync command is in situations where a previous sync operation was unable to unmount a file system because user jobs had files open on it at the time. In such a situation, when the user jobs have finished on the file system, you can run the scfsmgr sync command to trigger another sync operation to correct the state of the file system. This time, when the sync operation sees that the desired state of the file system is OFFLINE and attempts to unmount the file system, the operation will succeed (because the problem has been resolved).

The syntax of this command is as follows:

```
scfsmgr sync
```

7.4.11 scfsmgr sync_domain

The scfsmgr sync_domain command is similar to the scfsmgr sync command, but operates only on a specified domain. It checks the mount status of all SCFS and PFS file systems in the specified domain and either unmounts or mounts them as appropriate.

The `scfsmgr sync_domain` command completes as soon as the `scmountd` daemon is informed — the actions to synchronize the file systems happen later.

The syntax of this command is as follows:

```
scfsmgr sync_domain domain
```

where *domain* is the name of the domain where the mount states are to be synchronized.

7.4.12 scfsmgr upgrade

The `scfsmgr upgrade` command is used by the `sra` command when upgrading an HP AlphaServer SC system. The `scfsmgr upgrade` command imports the SCFS and PFS file system definitions from the `/etc/scfs.conf` and `/etc/pfs.conf` files into the SC database. If the SCFS or PFS file system is already defined in the SC database, the `scfsmgr upgrade` command does not re-import the file system.

Before using the `scfsmgr upgrade` command, the primary FS domain must be running. If the root component file system of a PFS is served by another FS domain, that domain must also be running. This is so that the upgrade command can mount the root component file system. The primary FS domain is the first domain name in the SCFS_SRV_DOMS environment variable in the `/etc/rc.config.common` file.

The syntax of this command is as follows:

```
scfsmgr upgrade
```

7.4.13 scfsmgr try_mount

Caution:

The `scfsmgr try_mount` command is only intended for troubleshooting. It is not a substitute for the `scfsmgr online` command.

The `scfsmgr try_mount` command triggers an SCFS file system mount on a selected node or domain. You can use this command on FS domains and CS domains to troubleshoot specific SCFS client nodes or client domains that are giving problems. While the file system is being mounted, information about the operation is displayed. This allows you to monitor the commands and actions that are being executed during the mount operation.

The `scfsmgr try_mount` command and the `scfsmgr online` command have different purposes, and they behave in a different way internally (even though they may seem to accomplish a similar task). The effects of the `online` command are more persistent — when the command is executed, certain database tables are modified to ensure that if the system

fails or is rebooted, or a command such as `scfsmgr sync` is executed, the state of the mounts can be preserved or restored. When the `try_mount` command is executed, only values that are used for informational purposes are updated.

The syntax of the `scfsmgr try_mount` command is as follows:
`scfsmgr try_mount` *`domain`*|*`node name`*|*`mountpoint`*

where:

- *`domain`* is the name of the domain where the mount is to be performed

  (Example: `atlasD0`)
- *`node`* is the name of the node where the mount is to be performed

  (Example: `atlas0`)
- *`name`* is the name of the file system in the format *`domain#filename`*

  (Example: `data_domain#data`)
- *`mountpoint`* is the pathname of the mount point for the file system

  (Example: `/data`)

Examples:

```
# scfsmgr try_mount atlas0 /s_test_mnt_1
# scfsmgr try_mount atlasD0 advfs_domain_2#s_test_mnt_2
```

7.4.14 scfsmgr try_unmount

Caution:

The `scfsmgr try_unmount` command is only intended for troubleshooting. It is not a substitute for the `scfsmgr offline` command.

The `scfsmgr try_unmount` command triggers an SCFS file system unmount on a selected node or domain. You can use this command on FS domains and CS domains to troubleshoot specific SCFS client nodes or client domains that are giving problems. While the file system is being unmounted, information about the operation is displayed. This allows you to monitor the commands and actions that are being executed during the unmount operation.

The `scfsmgr try_unmount` command and the `scfsmgr offline` command have different purposes, and they behave in a different way internally (even though they may seem to accomplish a similar task). The effects of the `offline` command are more persistent — when the command is executed, certain database tables are modified to ensure that if the

system fails or is rebooted, or a command such as `scfsmgr sync` is executed, the state of the mounts can be preserved or restored. When the `try_unmount` command is executed, only values that are used for informational purposes are updated.

The syntax of the `scfsmgr try_unmount` command is as follows:
`scfsmgr try_unmount [-f] domain|node name|mountpoint`

where:

- `-f` forces the unmount operation.

  If the command specifies an FS domain, or a node in an FS domain, the SCFS file system will be unmounted on the FS domain even if the file system is currently mounted by CS domains.

  If the command specifies a CS domain, or a node in a CS domain, the `-f` option has no impact on the outcome or execution of the command.

  Using the `-f` option does not result in user processes being killed, regardless of whether the command is used on an FS domain or a CS domain.

- *`domain`* is the name of the domain where the unmount is to be performed.

  (Example: `atlasD0`)

- *`node`* is the name of the node where the unmount is to be performed.

  (Example: `atlas0`)

- *`name`* is the name of the file system.

  (Example: `data_domain#data`)

- *`mountpoint`* is the pathname of the mount point for the file system.

  (Example: `/data`)

Examples:

```
# scfsmgr try_unmount atlas0 /s_test_mnt_1
# scfsmgr try_unmount atlasD0 advfs_domain_2#s_test_mnt_2
# scfsmgr try_unmount -f atlas0 /s_test_mnt_1
```

7.4.15 scfsmgr Quota Commands

HP AlphaServer SC Version 2.6 includes quota support for SCFS file systems. Quota support allows you to limit both the number of files and the total amount of disk space that are allocated in SCFS file systems on behalf of a given user or group.

Quota support in SCFS is similar to AdvFS quota support in the base Tru64 UNIX operating system, but there are some differences, as follows:

- The `scfsmgr setquota` command is specific to HP AlphaServer SC systems and replaces the `edquota` command in Tru64 UNIX.
- As `scfsmgr` commands can only be run by the `root` user, other users cannot check their own quota information for SCFS file systems.
- The grace period for SCFS file system quotas is specified in days, whereas the grace period for quotas in AdvFS file systems can be specified in days, hours, minutes, and seconds.

Note:

HP does not recommend enabling quotas for SCFS file systems that are part of a PFS file system, as the quotas will only be applied to the individual SCFS file systems and not to the entire PFS file system.

If quotas are enabled on an SCFS file system that is part of a PFS file system, messages indicating that the user quota has been exceeded may occur when the quota has been exceeded on the SCFS file system, even when there is space remaining on the PFS file system.

For more information on managing PFS file systems, see Chapter 8.

The commands and command options for quota support for SCFS file systems are described in this section. For general information about managing quotas, see the HP *Tru64 UNIX System Administration* manual.

7.4.15.1 scfsmgr quota

The `scfsmgr quota` command displays quota information for a user or group on SCFS file systems.

The syntax of this command is as follows:

`scfsmgr quota [-u|-g] user_name|group_name [name|mountpoint|all]`

where:

- `-u` specifies the quota type as a user quota
- `-g` specifies the quota type as a group quota
- *`user_name`* is the name of the user whose quota information is to be shown
- *`group_name`* is the name of the group whose quota information is to be shown

- *name* is the name of the file system

  (Example: data_domain#data)
- *mountpoint* is the pathname of the mount point for the file system

  (Example: /data)
- all specifies all SCFS file systems

Example:

```
# scfsmgr quota -u test1
atlasD0: - mount point = /scfs1
                [Quota  Limits  (MB)]            [Inode  Limits]
User           used    soft    hard    timeleft used    soft    hard    timeleft
-------------- ------- ------- ------- -------- ------- ------- ------- --------
test1          31      32      36      0        15      0       0       0
```

7.4.15.2 scfsmgr quotaon

The scfsmgr quotaon command enables quota enforcement for SCFS file systems.

The syntax of this command is as follows:

```
scfsmgr quotaon [-u|-g] name|mountpoint|all
```

where:

- -u specifies the quota type as a user quota
- -g specifies the quota type as a group quota
- *name* is the name of the file system

  (Example: data_domain#data)
- *mountpoint* is the pathname of the mount point for the file system

  (Example: /data)
- all specifies all SCFS file systems

Example:

```
# scfsmgr quotaon -u all
```

7.4.15.3 scfsmgr quotaoff

The scfsmgr quotaoff command disables quota enforcement for SCFS file systems.

The syntax of this command is as follows:

```
scfsmgr quotaoff [-u|-g] name|mountpoint|all
```

where:

- -u specifies the quota type as a user quota

- `-g` specifies the quota type as a group quota
- *`name`* is the name of the file system

  (Example: `data_domain#data`)
- *`mountpoint`* is the pathname of the mount point for the file system

  (Example: `/data`)
- `all` specifies all SCFS file systems

Example:

```
# scfsmgr quotaoff /scfs1
```

7.4.15.4 scfsmgr setquota

The `scfsmgr setquota` command sets quotas for a user or group on an SCFS file system.

The syntax of this command is as follows:

```
scfsmgr setquota [-u|-g] user_name|group_name name|mountpoint
[-b b_softlimit, b_hardlimit, b_grace_time]
[-i i_softlimit, i_hardlimit, i_grace_time]
```

where:

- `-u` specifies the quota type as a user quota
- `-g` specifies the quota type as a group quota
- *`user_name`* is the name of the user whose quota information is to be shown
- *`group_name`* is the name of the group whose quota information is to be shown
- *`name`* is the name of the file system

  (Example: `data_domain#data`)
- *`mountpoint`* is the pathname of the mount point for the file system

  (Example: `/data`)
- `-b` specifies the megabyte limits
  - `b_softlimit` specifies the preferred megabyte limit; when this limit is reached, a warning is given at login
  - `b_hardlimit` specifies the maximum number of megabytes a user or group is allowed to create on the file system
  - `b_grace_time` specifies the maximum number of days a user is allowed to continue working after the megabyte soft limit is reached
- `-i` specifies inode limits

- `i_softlimit` specifies the preferred inode limit; when this limit is reached, a warning is given at login
- `i_hardlimit` specifies the maximum number of inodes a user or group is allowed to create on the file system
- `i_grace_time` specifies the maximum number of days a user is allowed to continue working after the inode soft limit is reached

Example:

```
# scfsmgr setquota -u test1 /scfs1 -b 60,65,1
- setquota successful
```

7.4.15.5 scfsmgr repquota

The `scfsmgr repquota` command summarizes quota information for SCFS file systems.

The syntax of this command is as follows:

`scfsmgr repquota [-u|-g] name|mountpoint|all`

where:

- `-u` specifies the quota type as a user quota
- `-g` specifies the quota type as a group quota
- *`name`* is the name of the file system

  (Example: `data_domain#data`)
- *`mountpoint`* is the pathname of the mount point for the SCFS file system

  (Example: `/data`)
- `all` specifies all SCFS file systems

Example:

```
# scfsmgr repquota -u all
atlasD0: mount point /scfs1
                [Quota  Limits  (MB)]             [Inode  Limits]
User            used    soft    hard    timeleft used    soft    hard    timeleft
--------------- ------- ------- ------- -------- ------- ------- ------- --------
root            54      65      60      0        4       0       0       0
test1           31      65      60      0        15      0       0       0
```

7.5 Access Control Lists

In HP AlphaServer SC Version 2.6, Access Control Lists (ACLs) can be used in SCFS file systems. ACLs are an extension of Tru64 UNIX permissions on files and directories, and can be used to set permissions for specific users and groups on files and directories. In addition to the permissions normally associated with a file or directory, an ACL contains a list of users and groups with the specific permissions for each user and group.

In order to use ACLs in SCFS file systems, you must enable ACLs on each domain, and specify the `proplist` mount option for NFS (importing) nodes in the `rw` field of the `sc_scfs` table for each SCFS file system that ACLs are to be used on.

For more information, and for instructions on enabling ACLs, see the HP *Tru64 UNIX System Administration* manual. For more information on the `sc_scfs` table and specifying mount options, see Section 7.4.1.1 on page 7–11.

Note:

When you make changes to the `rw` field of the `sc_scfs` table, the changes are not reflected automatically for file systems that are already online. To activate the changes, you must unmount the file systems by marking them as OFFLINE, and then mount them again by marking them as ONLINE.

For example, if you change the `rw` field for the `/fs/scfs_1` file system while the file system is mounted, you will need to unmount and remount the file system as follows after you have made the changes to the `rw` field:

```
# scfsmgr offline /fs/scfs_1
# scfsmgr online /fs/scfs_1
```

7.6 SysMan Menu

SysMan Menu provides an alternate interface to the `scfsmgr` command. To directly invoke the SCFS option within SysMan Menu, enter the following command:

```
# sysman scfsmgr
```

If the DISPLAY environment variable is set, `sysman` provides a graphical user interface. If the DISPLAY environment variable is not set, `sysman` provides a command line interface.

If you invoke SysMan Menu without specifying the `scfsmgr` accelerator, `sysman` displays the list of all tasks that can be performed. To select the SCFS option, select the AlphaServer SC *Configuration* menu option, followed by the *Manage SCFS File Systems* menu option.

Quota commands and ACL commands are not available *via* SysMan Menu.

7.7 Monitoring and Correcting File-System Failures

This section describes how to monitor file systems and how to take corrective action when failures occur. The file-system management tools manage both SCFS and PFS file systems, because there is an interaction between SCFS and PFS file systems.

7.7.1 Overview of the File-System Management System

The File-System Management System is based on the following:

- The SC database

  The SC database contains configuration data (for example, the mount point pathnames for an SCFS file system) and dynamic data (for example, the mount state of an SCFS file system on a specific domain).

- The `scfsmgr` and `pfsmgr` commands

  These commands provide a user interface to the system. All of the state information shown by these commands is based on data in the SC database. When a user performs an action, such as placing a file system online, the command updates the SC database with this state change and sends a command to the `scmountd` daemon.

- The `scmountd` daemon

  This daemon runs on the management server (if present) or one of the nodes in domain 0. The `scmountd` daemon responds to the `scfsmgr` and `pfsmgr` commands. It also responds to the completion of the boot process.

  The `scmountd` daemon responds to commands and events by invoking scripts on FS and CS domains. These scripts perform the actions to mount or unmount file systems. The `scmountd` daemon coordinates activities — for example, it ensures that FS domains mount file systems before CS domains attempt to import the file systems.

  The `scmountd` daemon logs its actions in the `/var/sra/adm/log/scmountd/scmountd.log` file.

- The `srad` daemon

  The `srad` daemon is primarily responsible for booting and shutting down domains and nodes. The `srad` daemon is also the mechanism by which the `scmountd` daemon invokes scripts.

  A log of the scripts being invoked is stored in the `/var/sra/adm/log/scmountd/srad.log` file. Programming errors in the scripts are recorded in this log file.

  See the `srad(8)` reference page for more information about the `srad` daemon.

- The file system management scripts

  These scripts perform the mount and unmount actions. Some scripts are responsible for mounting, others for unmounting. Each script follows this general sequence:

  a. Reads the ONLINE/OFFLINE state of each file system from the SC database.
  b. Compares this state to the actual mount state.
  c. If these states differ, attempts to mount or unmount the file system as appropriate.
  d. Updates the actual mount state in the SC database.

  Each script records its actions in the `/var/sra/adm/log/scmountd/fsmgrScripts.log` file. Node-specific actions on PFS file systems are logged in the `/var/sra/adm/log/scmountd/pfsmgr.`*`nodename`*`.log` file.

7.7.2 Monitoring File-System State

The following tools monitor file-system state:

- The `scfsmgr show` command

  This command shows the state of all SCFS file systems, as described in Section 7.4.8 on page 7–17.

- The `pfsmgr show` command

  This command shows the state of all PFS file systems, as described in Section 8.4.2.5 on page 8–18.

- The HP AlphaServer SC event system

  The file-system management tools use the HP AlphaServer SC event system to report changes in file-system state. Use the `scevent` command to view these events. Use the `scalertmgr` command to send an e-mail when specific file-system failures occur. The HP AlphaServer SC event system is described in Chapter 9. The events that are specific to file-system management are described in Section 7.7.3.

- The `scfsmgr status` command

  Many of the `scfsmgr` and `pfsmgr` commands complete before the actions that underlie the command have completed or even started. Use the `scfsmgr status` command to determine whether the file-system management system has finished processing a command. Section 7.4.9 on page 7–18 describes how to interpret the `scfsmgr status` command output.

7.7.3 File-System Events

To display the complete set of file-system event types, run the following command:

```
# scevent -l -v -f '[category filesystem]'
```

Table 7–2 describes the event classes.

Table 7–2 File-System Event Classes

Class	Description
scfs	This class of event reports failures in mount or unmount operations on SCFS file systems. Successful mounts or unmounts are reported as either advfs (FS domain) or nfs (CS domain) events.
pfs	This class of event reports mounts, unmounts, and failed operations on PFS file systems.
nfs	This class of event reports mounts and unmounts of SCFS file systems on CS domains. This class also reports mounts and unmounts of standard NFS file systems, not just SCFS file systems.
cfs	This class of event reports events from the Cluster File System (CFS) subsystem. These events apply to all file systems — not just SCFS file systems. On an FS domain, these events report on the node performing file-serving operations for an SCFS file system. On a CS domain, these events report on the node that has mounted (and thus serves) an SCFS file system.
advfs	This class of event reports events from the Advanced File System (AdvFS) subsystem. These events apply to all file systems — not just SCFS file systems. Generally, the events record mounts and unmounts. However, they also report important failures, such as AdvFS domain panics.

7.7.4 Interpreting and Correcting File-System Failures

This section describes a typical scenario to explain how to find and correct failures. This section does not attempt to explain all failures, but provides a general technique for identifying problems.

We start with all SCFS file systems offline, as shown by the following command:

```
# scfsmgr show
State      Mountpoint      Server          Mount status
-----      ----------      ------          ------------
offline    /s/scr          atlas2          not-mounted: atlasD[0-1]
offline    /s/data         atlas3          not-mounted: atlasD[0-1]
```

We then place the file systems online, using the `scfsmgr online` command, as follows:

```
# scfsmgr online all
Marking all SCFS as online.
Actions to place filesystem(s) online have been initiated
```

Later, we observe that `/s/data` has not been mounted. To investigate, we run the `scfsmgr show` command, as follows:

```
# scfsmgr show
```

```
State      Mountpoint     Server      Mount status
-----      ----------     ------      ------------
online     /s/scr         atlas2      not-mounted: atlasD[0-1]
online     /s/data        !none!      mount-failed: atlasD0  not-mounted: atlasD1
```

The mount of `/s/data` has failed on `atlasD0` (the FS domain), so no attempt was made to mount it on `atlasD1`. Therefore, its status is `not-mounted`.

When mount attempts fail, the file-system management system reports the failure in an event. To view the event, use the `scevent` command as follows (to display events that have occurred in the previous 10 minutes):

```
# scevent -f '[age < 10m]'

08/02/02 15:30:48         atlasD0         advfs     fset.mount        AdvFS: AdvFS fileset
scr_domain#scr mounted on /s/scr

08/02/02 15:30:48         atlasD0         cfs       advfs.served      CFS: AdvFS domain
scr_domain is now served by node atlas2

08/02/02 15:30:49         atlas3          scfs      mount.failed      Mount of /s/data
failed: atlas3: data_domain#data on /s/data: No such domain, fileset or mount directory
atlas0: exited with status 1

08/02/02 15:30:50         atlasD1         nfs       mount             NFS: NFS filesystem
atlasD0-atlas0:/s/scr mounted on /s/scr

08/02/02 15:30:50         atlasD1         cfs       fs.served         CFS: Filesystem /s/
src is now served by node atlas32
```

The first two events show a successful mount of `/s/scr` on `atlasD0` (by node `atlas2`). The final two events show that `/s/scr` was successfully mounted on `atlasD1` (by node `atlas32`).

However, the third event shows that `atlas3` failed to mount `/s/data`. The reason given is that `data_domain#data` does not exist. A possible cause of this is that a link has inadvertently been manually deleted from the `/etc/fdmns` directory. See the AdvFS documentation for more information on how `/etc/fdmns` is used. If the underlying AdvFS domain has not also been deleted on disk (for example, by using the `disklabel` command), you can recover the AdvFS domain by recreating the link to `data_domain` in the `/etc/fdmns` directory.

If the `data_domain` is lost, you can create a new version by manually creating the AdvFS domain and recreating the link. Alternatively, you can delete the SCFS file system as follows:

```
# scfsmgr destroy /s/data
```

Before using the `scfsmgr` command to create the file system again, you must run the `disklabel` command so that the disk partition is marked as unused.

Because events are such a useful source of failure information, HP suggests that you monitor events whenever you use the `scfsmgr` or `pfsmgr` commands. On a large system, it is useful to monitor warning and failure events only. You can continuously monitor warning and failure events, by running the following `scevent` command:

```
# scevent -c -f '[severity ge warning]'
```

You can also use the `scfsmgr try_mount` and `scfsmgr try_unmount` commands to help you to identify file system problems. These commands force the mount or unmount of an SCFS file system, and display information about the operation as it is performed. See Section 7.4.13 on page 7–20 and Section 7.4.14 on page 7–21 for more information on using the `scfsmgr try_mount` and `scfsmgr try_unmount` commands.

7.8 Tuning SCFS

The information in this section is organized as follows:

- Tuning SCFS Kernel Subsystems (see Section 7.8.1 on page 7–32)
- Tuning SCFS Server Operations (see Section 7.8.2 on page 7–32)
- Tuning SCFS Client Operations (see Section 7.8.3 on page 7–34)
- Monitoring SCFS Activity (see Section 7.8.4 on page 7–35)

7.8.1 Tuning SCFS Kernel Subsystems

To tune any of the SCFS subsystem attributes permanently, you must add an entry to the appropriate subsystem stanza, either `scfs` or `scfs_client`, in the `/etc/sysconfigtab` file. Do not edit the `/etc/sysconfigtab` file directly — use the `sysconfigdb` command to view and update its contents. Changes made to the `/etc/sysconfigtab` file will take effect when the system is next booted. Some of the attributes can also be changed dynamically using the `sysconfig` command, but these settings will be lost after a reboot unless the changes are also added to the `/etc/sysconfigtab` file.

7.8.2 Tuning SCFS Server Operations

A number of configurable attributes in the `scfs` kernel subsystem affect SCFS serving. Some of these attributes can be dynamically configured, while others require a reboot before they take effect. For a detailed explanation of the `scfs` subsystem attributes, see the `sys_attrs_scfs(5)` reference page.

The default settings for the `scfs` subsystem attributes should work well for a mixed workload. However, performance may be improved by tuning some of the parameters.

7.8.2.1 SCFS I/O Transfers

SCFS I/O achieves best performance results when processing large I/O requests.

If a client generates a very large I/O request, such as writing 512MB of data to a file, this request will be performed as a number of smaller operations. The size of these smaller operations is dictated by the `io_size` attribute of the server node for the SCFS file system. The default value of the `io_size` attribute is 16MB.

This subrequest is then sent to the SCFS server, which in turn performs the request as a number of smaller operations. This time, the size of the smaller operations is specified by the `io_block` attribute. The default value of the `io_block` attribute is 128KB. This allows the SCFS server to implement a simple double-buffering scheme which overlaps I/O and interconnect transfers.

Performance for very large requests may be improved by increasing the `io_size` attribute, though this will increase the setup time for each request on the client. You must propagate this change to every node in the FS domain, and then reboot the FS domain.

Performance for smaller transfers (<256KB) may also be improved slightly by reducing the `io_block` size, to increase the effect of the double-buffering scheme. You must propagate this change to every node in the FS domain, and then reboot the FS domain.

7.8.2.2 SCFS Synchronization Management

The SCFS server will synchronize the dirty data associated with a file to disk, if one or more of the following criteria is true:

- The file has been dirty for longer than `sync_period` seconds. The default value of the `sync_period` attribute is 10.
- The amount of dirty data associated with the file exceeds `sync_dirty_size`. The default value of the `sync_dirty_size` attribute is 256MB.
- The number of write transactions since the last synchronization exceeds `sync_handle_trans`. The default value of the `sync_handle_trans` attribute is 204.

If an application generates a workload that causes one of these conditions to be reached very quickly, poor performance may result because I/O to a file regularly stalls waiting for the synchronize operation to complete. For example, if an application writes data in 128KB blocks, the default `sync_handle_trans` value would be exceeded after writing 25.5MB. Performance may be improved by increasing the `sync_handle_trans` value. You must propagate this change to every node in the FS domain, and then reboot the FS domain.

Conversely, an application may generate a workload that does not cause the `sync_dirty_size` and `sync_handle_trans` limits to be exceeded — for example, an application that writes 32MB in large blocks to a number of different files. In such cases, the data is not synchronized to disk until the `sync_period` has expired. This could result in poor performance as UBC resources are rapidly consumed, and the storage subsystems are left idle. Tuning the dynamically reconfigurable attribute `sync_period` to a lower value may improve performance in this case.

7.8.2.3 SCFS Multiple Rails

The SCFS kernel subsystem can make use of multiple Elan rails, if they are present on the system, to perform SCFS I/O data transfers. This allows SCFS clients to load-balance their requests over all the rails that they are permitted to use and can see the server nodes on.

SCFS treats each rail as an independent resource, and tracks the local node SCFS utilization of each rail. When an I/O request is being sent from a client to a server, the SCFS subsystem on the client determines the set of rails on which the target server node is visible. SCFS then selects the rail with the lightest load from the rail set, maps the buffers associated with the I/O request for use on this rail, and finally sends the I/O request to the server node.

If a request fails because a rail drops out, at either the server end or the client end, the system handles the rail failure in the same way as server node failure. The I/O request processing restarts, and repeats the process of determining the set of rails on which the target server node is visible. The failed rail will not now be included in the set of available rails.

By default, SCFS makes use of all available rails on the system, but you can set a subsystem attribute (the `try_rails` attribute) to force SCFS to use only a selected set of rails. For more information about tuning the `try_rails` attribute, see the `sys_attrs_scfs(5)` reference page.

7.8.3 Tuning SCFS Client Operations

In HP AlphaServer SC Version 2.6, the `scfs_client` kernel subsystem has two configurable attributes, the `max_buf` attribute and the `max_buf_total` attribute.

- The `max_buf` attribute is a file system limit that specifies the maximum amount of data that a client will allow to be shadow copied for one SCFS file system, before blocking new requests from being issued. The default value of the `max_buf` attribute is 256MB, and it can be dynamically modified.
- The `max_buf_total` attribute is similar to the `max_buf` attribute, but while the `max_buf` attribute sets an upper limit for memory usage on a single file system, the `max_buf_total` attribute can be used to set an upper limit for memory usage on all SCFS file systems on a node. It specifies the maximum amount of data that a client will allow to be shadow copied in total, before blocking new requests from being issued. The amount of memory used by each SCFS file system is aggregated, and if the total reaches the limiting value specified in the `max_buf_total` attribute, I/O is stalled until the current I/O is dealt with. The default value of the `max_buf_total` attribute is 512MB.

For example, if the `max_buf` attribute is set to 256MB, the client can only shadow copy 256MB of data at a time to any one file system. However, if the `max_buf_total` attribute is set to 512MB, then the sum total of data shadow copied by the client for all file systems cannot exceed 512MB, regardless of whether the amount of data shadow copied on each individual file system is less than 256MB.

The client keeps shadow copies of data written to an SCFS file system so that, in the event of a server crash, the requests can be re-issued.

The SCFS server notifies clients when requests have been synchronized to disk so that they can release the shadow copies, and allow new requests to be issued.

If a client node is accessing many SCFS file systems, for example *via* a PFS file system (see Chapter 8), it may be better to reduce the `max_buf` setting. This will minimize the impact of maintaining many shadow copies for the data written to the different file systems.

For a detailed explanation of the `max_buf` and `max_buf_total` subsystem attributes, see the `sys_attrs_scfs_client(5)` reference page.

7.8.4 Monitoring SCFS Activity

The activity of the `scfs` kernel subsystem, which implements the SCFS I/O serving and data transfer capabilities, can be monitored by using the `scfs_xfer_stats` command. You can use this command to determine what SCFS file systems a node is using, and report the SCFS usage statistics for the node as a whole, or for the individual file systems, in summary format or in full detail. This information can be reported for a node as an SCFS server, as an SCFS client, or both.

For details on how to use this command, see the `scfs_xfer_stats(8)` reference page.

7.9 SC Database Tables Supporting SCFS File Systems

Note:

This section is provided for informational purposes only, and is subject to change in future releases.

This section describes the SC database tables that are used by the SCFS file-system management system. Much of the data in these tables is maintained by the `scfsmgr scan` command. If nodes were down when the `scfsmgr scan` command was run, the data in the tables will be incomplete.

This section describes the following tables:

- The sc_scfs Table (see Section 7.9.1 on page 7–36)

- The sc_scfs_mount Table (see Section 7.9.2 on page 7–37)
- The sc_advfs_vols Table (see Section 7.9.3 on page 7–38)
- The sc_advfs_filesets Table (see Section 7.9.4 on page 7–38)
- The sc_disk Table (see Section 7.9.5 on page 7–38)
- The sc_disk_server Table (see Section 7.9.6 on page 7–39)
- The sc_lsm_vols Table (see Section 7.9.7 on page 7–40)

7.9.1 The sc_scfs Table

The `sc_scfs` table describes the SCFS file systems. This table contains one record for each SCFS file system. Table 7–3 describes the fields in the `sc_scfs` table.

Table 7–3 The sc_scfs Table

Field	Description
`clu_domain`	The name of the FS domain that serves the file system
`advfs_domain`	The name of the AdvFS domain where the file system is stored
`fset_name`	The name of the fileset within the AdvFS domain where the file system is stored
`preferred_server`	The name of the preferred server node
`rw`	Specifies the mount option arguments for the file system. The syntax is: *`common_options`*`:[`*`advfs_options`*`]:[`*`nfs_options`*`]`
`speed`	Specifies whether the file system is `FAST` or `UBC`
`status`	Specifies whether the file system is `ONLINE` or `OFFLINE`
`mount_point`	The pathname of the mount point for the file system
`usr_quota`	Specifies whether user quotas are enabled. A value of `-1` indicates that user quotas are enabled; a value of `0` (zero) indicates that user quotas are disabled.
`grp_quota`	Specifies whether group quotas are enabled. A value of `-1` indicates that group quotas are enabled; a value of `0` (zero) indicates that group quotas are disabled.

Note:

In HP AlphaServer SC Version 2.6, the mount option information is stored in the `rw` field of the `sc_scfs` table. When you make changes to this field, the changes are not reflected automatically on file systems that are already online. To activate the changes, you must unmount the file systems by marking them as OFFLINE, and then mount them again by marking them as ONLINE.

For example, if you change the `rw` field for the `/fs/scfs_1` file system while the file system is mounted, you will need to unmount and remount the file system as follows after you have made the changes to the `rw` field:

```
# scfsmgr offline /fs/scfs_1
# scfsmgr online /fs/scfs_1
```

7.9.2 The sc_scfs_mount Table

The `sc_scfs_mount` table contains the mount status of each SCFS file system on each domain. Each SCFS file system has a record for each domain. Table 7–4 describes the fields in the `sc_scfs_mount` table.

Table 7–4 The sc_scfs_mount Table

Field	Description
`advfs_domain`	The name of the AdvFS domain where the file system is stored
`fset_name`	The name of the fileset within the AdvFS domain where the file system is stored
`cluster_name`	The name of the FS or CS domain to which the mount status applies
`server`	The name of the node that is currently serving the SCFS file system
`state`	The mount status for this SCFS on the specified FS or CS domain

7.9.3 The sc_advfs_vols Table

The `sc_advfs_vols` table specifies which disk or LSM volume is used by an AdvFS domain. Table 7–5 describes the fields in the `sc_advfs_vols` table.

Table 7–5 The sc_advfs_vols Table

Field	Description
`clu_domain`	The name of the FS domain where the disk or LSM volume resides
`name`	The name of the disk partition or LSM volume
`advfs_domain`	The name of the AdvFS domain
`type`	Specifies whether this record is for a disk (`DISK`) or LSM volume (`LSM`)

7.9.4 The sc_advfs_filesets Table

The `sc_advfs_filesets` table specifies the names of all AdvFS file sets within an AdvFS domain. Table 7–6 describes the fields in the `sc_advfs_filesets` table.

Table 7–6 The sc_advfs_filesets Table

Field	Description
`clu_domain`	The name of the FS domain where the disk or LSM volume resides
`advfs_domain`	The name of the AdvFS domain
`fset_name`	The name of the fileset within the AdvFS domain

7.9.5 The sc_disk Table

The `sc_disk` table specifies whether disk partitions are in use or available for use in creating an SCFS file system. There is one record for each disk on all FS domains. Table 7–7 describes the fields in the `sc_disk` table.

Table 7–7 The sc_disk Table

Field	Description
`name`	The name of the disk
`clu_domain`	The name of the FS domain where the disk resides

Table 7–7 The sc_disk Table

Field	Description
status	The device status of the disk (see the drdmgr(8) reference page); this field is updated by the scfsmgr scan command only
type	The type of disk (see the hwmgr(8) reference page)
a	Specifies whether partition a is in use: -1 indicates that the partition is in use, any other value indicates the size of the partition
b	Specifies whether partition b is in use: -1 indicates that the partition is in use, any other value indicates the size of the partition
c	Specifies whether partition c is in use: -1 indicates that the partition is in use, any other value indicates the size of the partition
d	Specifies whether partition d is in use: -1 indicates that the partition is in use, any other value indicates the size of the partition
e	Specifies whether partition e is in use: -1 indicates that the partition is in use, any other value indicates the size of the partition
f	Specifies whether partition f is in use: -1 indicates that the partition is in use, any other value indicates the size of the partition
g	Specifies whether partition g is in use: -1 indicates that the partition is in use, any other value indicates the size of the partition
h	Specifies whether partition h is in use: -1 indicates that the partition is in use, any other value indicates the size of the partition

7.9.6 The sc_disk_server Table

The sc_disk_server table specifies the nodes that are able to serve a given disk. There is one entry for each node that can serve a given disk (that is, if two nodes can serve a disk, there are two entries for that disk). Table 7–8 describes the fields in the sc_disk_server table.

Table 7–8 The sc_disk_server Table

Field	Description
name	The name of the disk
clu_domain	The name of the FS domain where the disk resides
node	The name of the node that can serve the disk

7.9.7 The sc_lsm_vols Table

The `sc_lsm_vols` table specifies the disks used by all of the LSM volumes on a given FS domain. There is one record for each disk partition that is used by an LSM volume. Table 7–9 describes the fields in the `sc_lsm_vols` table.

Table 7–9 The sc_lsm_vols Table

Field	Description
`clu_domain`	The name of the FS domain where the LSM volume resides
`diskgroup`	The name of the diskgroup of the LSM volume
`volume_name`	The name of the LSM volume
`disk`	The name of the disk partition where the LSM volume is stored

8

Managing the Parallel File System (PFS)

This chapter describes the administrative tasks associated with the Parallel File System (PFS).

The information in this chapter is structured as follows:

- PFS Overview (see Section 8.1 on page 8–2)
- Planning a PFS File System to Maximize Performance (see Section 8.2 on page 8–5)
- Managing a PFS File System (see Section 8.3 on page 8–7)
- The PFS Management Utility: pfsmgr (see Section 8.4 on page 8–13)
- Using a PFS File System (see Section 8.5 on page 8–22)
- SC Database Tables Supporting PFS File Systems (see Section 8.6 on page 8–29)

8.1 PFS Overview

A parallel file system (PFS) allows a number of data file systems to be accessed and viewed as a single file system view. The PFS file system stores the data as stripes across the component file systems, as shown in Figure 8–1.

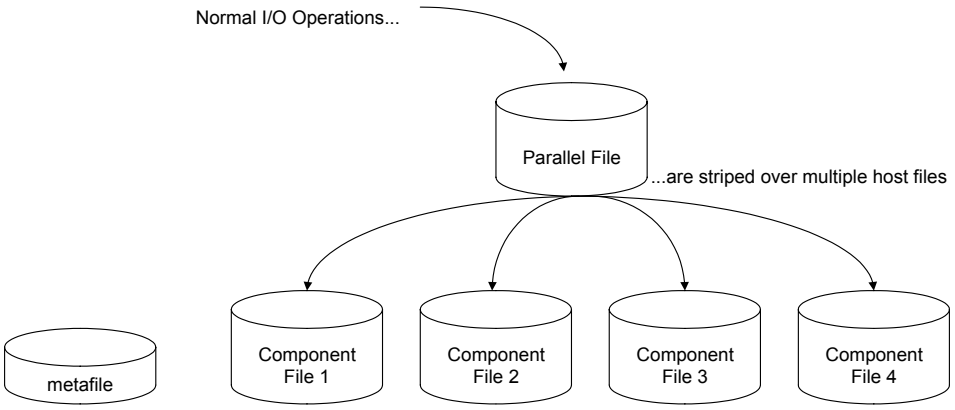

Figure 8–1 Parallel File System

Files written to a PFS file system are written as stripes of data across the set of component file systems. For a very large file, approximately equal portions of a file will be stored on each file system. This can improve data throughput for individual large data read and write operations, because multiple file systems can be active at once, perhaps across multiple hosts.

Similarly, distributed applications can work on large shared datasets with improved performance, if each host works on the portion of the dataset that resides on locally mounted data file systems.

Underlying a component file system is an SCFS file system. The component file systems of a PFS file system can be served by several File-Serving (FS) domains. Where there is only one FS domain, programs running on the FS domain access the component file system *via* the CFS file system mechanisms. Programs running on Compute-Serving (CS) domains access the component file system remotely *via* the SCFS file system mechanisms. If several FS domains are involved in serving components of a PFS file system, each FS domain must import the other domain's SCFS file systems (that is, the SCFS file systems are cross-mounted between domains). See Chapter 7 for a description of FS and CS domains.

See the `pfs(8)` reference page for more information about PFS file systems.

8.1.1 PFS Attributes

A PFS file system has a number of attributes, which determine how the PFS striping mechanism operates for files within the PFS file system. Some of the attributes, such as the set of component file systems, can only be configured when the file system is created, so you should plan these carefully (see Section 8.2 on page 8–5). Other attributes, such as the size of the stride, can be reconfigured after file system creation; these attributes can also be configured on a per-file basis.

The PFS attributes are as follows:

- NumFS (Component File System List)

  A PFS file system is comprised of a number of component file systems. The component file system list is configured when a PFS file system is created.

- Block (Block Size)

  The block size is the maximum amount of data that will be processed as part of a single operation on a component file system. The block size is configured when a PFS file system is created.

- Stride (Stride Size)

  The stride size is the amount (or stride) of data that will be read from, or written to, a single component file system before advancing to the next component file system, selected in a round robin fashion. The stride value must be an integral multiple of the block size (see Block above).

  The default stride value is defined when a PFS file system is created, but this default value can be changed using the appropriate ioctl (see Section 8.5.3.5 on page 8–27). The stride value can also be reconfigured on a per-file basis using the appropriate ioctl (see Section 8.5.3.3 on page 8–26).

- Stripe (Stripe Count)

  The stripe count specifies the number of component file systems to stripe data across, in cyclical order, before cycling back to the first file system. The stripe count must be non-zero, and less than or equal to the number of component file systems (see NumFS above).

  The default stripe count is defined when a PFS file system is created, but this default value can be changed using the appropriate ioctl (see Section 8.5.3.5 on page 8–27). The stripe count can also be reconfigured on a per-file basis using the appropriate ioctl (see Section 8.5.3.3 on page 8–26).

- Base (Base File System)

  The base file system is the index of the file system, in the list of component file systems, that contains the first stripe of file data. The base file system must be between 0 and NumFS – 1 (see NumFS above).

The default base file system is selected when the file is created, based on the modulus of the file `inode` number and the number of component file systems. The base file system can also be reconfigured on a per-file basis using the appropriate ioctl (see Section 8.5.3.3 on page 8–26).

8.1.2 Structure of a PFS Component File System

The root directory of each component file system contains the same information, as described in Table 8–1.

Table 8–1 PFS Component File System Directory Structure

File Name	Description
`README`	Text file that describes the PFS configuration and lists the component file systems. The `mkfs_pfs` command automatically creates this file.
`.pfsid`	Binary file containing the PFS identity and number of component file systems.
`.pfsmap`	Binary file containing PFS mapping information.
`.pfs`*id*	Symbolic link to a component file system. There are *N* of these files in total: `.pfs0`, `.pfs1`, `.pfs2`,..., `.pfs`*N*`-1`, where *N* is the total number of component file systems. *id* is the index number used by the PFS ioctl calls to refer to the component file systems — see Section 8.5.3 on page 8–25 for more information about PFS ioctl calls.
`.pfscontents`	Contents directory that stores PFS file system data, in a way that is meaningful only to PFS.
`.pfsbirth`	Directory used to store PFS files while they are being created. Files are not available to user programs while they are in this directory. You should never modify the contents of this directory manually.
`.pfsdeath`	Directory used to store PFS files while they are being deleted. Files are not available to user programs while they are in this directory. You should never modify the contents of this directory manually. If files remain in the `.pfsdeath` directory for a prolonged period of time (for example, a week or more), please contact your HP Customer Support representative for assistance.

8.1.3 Storage Capacity of a PFS File System

The storage capacity of a PFS file system is primarily dependent on the capacity of the component file systems, but also depends on how the individual files are laid out across the component file systems.

For a particular file, the maximum storage capacity available within the PFS file system can be calculated by multiplying the stripe count (that is, the number of file systems it is striped across) by the actual storage capacity of the smallest of these component file systems.

Note:

The PFS file system stores directory mapping information on the first (root) component file system. The PFS file system uses this mapping information to resolve files to their component data file system block. Because of the minor overhead associated with this mapping information, the actual capacity of the PFS file system will be slightly reduced, unless the root component file system is larger than the other component file systems.

For example, a PFS file system consists of four component file systems (A, B, C, and D), with actual capacities of 3GB, 1GB, 3GB, and 4GB respectively. If a file is striped across all four file systems, then the maximum capacity of the PFS file system for this file is 4GB — that is, 1GB (Minimum Capacity) x 4 (File Systems). However, if a file is only striped across component file systems C and D, then the maximum capacity would be 6GB — that is, 3GB (Minimum Capacity) x 2 (File Systems).

For information on how to extend the storage capacity of PFS file systems, see Section 8.3.2 on page 8–10.

8.2 Planning a PFS File System to Maximize Performance

The primary goal, when using a PFS file system, is to achieve improved file access performance, scaling linearly with the number of component file systems (NumFS). However, it is possible for more than one component file system to be served by the same server, in which case the performance may only scale linearly with the number of servers.

To achieve this goal, you must analyze the intended use of the PFS file system. For a given application or set of applications, determine the following criteria:

- Number of Files

  An important factor when planning a PFS file system is the expected number of files.

  If expecting to use a very large number of files in a large number of directories, then you should allow extra space for PFS file metadata on the first (root) component file system. The extra space required will be similar in size to the overhead required to store the files on an AdvFS file system.

- Access Patterns

  How data files will be accessed, and who will be accessing the files, are two very important criteria when determining how to plan a PFS file system.

  On an FS domain, if a file is to be shared among a number of process elements (PEs) on different nodes on the domain, you can improve performance by ensuring that the file layout matches the access patterns, so that all PEs are accessing the parts of a file that are local to their nodes.

  If files are specific to a subset of nodes, then localizing the file to the component file systems that are local to these nodes should improve performance.

  If a large file is being scanned in a sequential or random fashion, then spreading the file over all of the component file systems should benefit performance.

- File Dynamics and Lifetime

  Data files may exist for only a brief period while an application is active, or they may persist across multiple runs. During this time, their size may alter significantly.

  These factors affect how much storage must be allocated to the component file systems, and whether backups are required.

- Bandwidth Requirements

  Applications that run for very long periods of time frequently save internal state at regular intervals, allowing the application to be restarted without losing too much work.

  Saving this state information can be a very I/O intensive operation, the performance of which can be improved by spreading the write over multiple physical file systems using PFS. Careful planning is required to ensure that sufficient I/O bandwidth is available.

To maximize the performance gain, some or all of the following conditions should be met:

1. PFS file systems should be created so that files are spread over the appropriate component file systems or servers. If only a subset of nodes will be accessing a file, then it may be useful to limit the file layout to the subset of component file systems that are local to these nodes, by selecting the appropriate stripe count.
2. The amount of data associated with an operation is important, as this determines what the stride and block sizes should be for a PFS file system. A small block size will require more I/O operations to obtain a given amount of data, but the duration of the operation will be shorter. A small stride size will cycle through the set of component file systems faster, increasing the likelihood of multiple file systems being active simultaneously.

3. The layout of a file should be tailored to match the access pattern for the file. Serial access may benefit from a small stride size, delivering improved read or write bandwidth. Random access performance should improve as more than one file system may seek data at the same time. Strided data access may require careful tuning of the PFS block size and the file data stride size to match the size of the access stride.
4. The base file system for a file should be carefully selected to match application access patterns. In particular, if many files are accessed in lock step, then careful selection of the base file system for each file can ensure that the load is spread evenly across the component file system servers. Similarly, when a file is accessed in a strided fashion, careful selection of the base file system may be required to spread the data stripes appropriately.

8.3 Managing a PFS File System

The primary tasks involved in managing a PFS file system are as follows:

- Creating and Mounting a PFS File System (see Section 8.3.1 on page 8–7)
- Increasing the Capacity of a PFS File System (see Section 8.3.2 on page 8–10)
- Upgrading a PFS File System (see Section 8.3.3 on page 8–11)
- Checking a PFS File System (see Section 8.3.4 on page 8–11)
- Exporting a PFS File System (see Section 8.3.5 on page 8–12)
- Backing up a PFS File System (see Section 8.3.6 on page 8–12)

8.3.1 Creating and Mounting a PFS File System

A PFS file system consists of a number of component SCFS file systems. The component SCFS file systems must be created, online, and mounted by the first domain before you attempt to create a PFS file system.

The `pfsmgr` command can be used to create, mount, unmount, and delete PFS file systems. It is also responsible for the automatic mounting of PFS file systems when a node boots, once all of the component file systems are available.

Note:

Before you create a PFS file system, you should analyze the intended use and plan the PFS file system accordingly, to maximize performance (see Section 8.2 on page 8–5).

Creating a PFS file system is a two-step process, as follows:

1. Create all component SCFS file systems. When the SCFS file systems are successfully created, place them online and wait until they are mounted by Domain 0 (the first domain in the system). Use the `scfsmgr` command to create, place online, and check the mount status of the SCFS file systems, as described in Chapter 7.
2. Use the `pfsmgr create` command (see Section 8.4.2.1 on page 8–15) to create the PFS file system based on these component file systems.

When the file system has been successfully created, use the `pfsmgr online` command to mount the file system.

8.3.1.1 Example 1: Four-Component PFS File System — /scratch

In this example, we create a PFS file system in domain `atlasD0`, using four 72GB component file systems. These component file systems, described in Table 8–2, have already been created using `scfsmgr`.

Table 8–2 Component File Systems for /scratch

Component Path	Server Node
`/data0_72g`	`atlas0`
`/data1_72g`	`atlas1`
`/data2_72g`	`atlas2`
`/data3_72g`	`atlas3`

We will use these component file systems to create a PFS file system that will be mounted as `/scratch`. The `pfsmgr` command allows a logical tag (PFS Set) name to be associated with a PFS file system — we will call this file system `scratch`. We will assign a stride of 128KB to the `scratch` PFS file system.

To create this PFS file system, run the following command:

```
# pfsmgr create scratch /scratch -numcomps 4 -stride 128k \
/data0_72g /data1_72g /data2_72g /data3_72g
```

This command creates the `scratch` PFS file system by creating the directory structure described in Table 8–1 in each component file system. The PFS file system is marked OFFLINE so is not mounted anywhere. As soon as the PFS file system is placed online, the mount point will be created on each domain as the PFS file system is being mounted.

8.3.1.2 Example 2: 32-Component PFS File System — /data3t

In this example, we create a 3TB PFS file system using 32 component file systems served by four nodes from domain `atlasD0`. Each of these 96GB component file systems, described in Table 8–3, has already been created using `scfsmgr`.

Table 8–3 Component File Systems for /data3t

Component Path	Server Node
/data3t_comps/pfs00	atlas0
/data3t_comps/pfs01	atlas1
/data3t_comps/pfs02	atlas2
/data3t_comps/pfs03	atlas3
/data3t_comps/pfs04	atlas0
/data3t_comps/pfs05	atlas1
/data3t_comps/pfs06	atlas2
/data3t_comps/pfs07	atlas3
/data3t_comps/pfs08	atlas0
/data3t_comps/pfs09	atlas1
/data3t_comps/pfs10	atlas2
/data3t_comps/pfs11	atlas3
/data3t_comps/pfs12	atlas0
/data3t_comps/pfs13	atlas1
/data3t_comps/pfs14	atlas2
/data3t_comps/pfs15	atlas3
/data3t_comps/pfs16	atlas0
/data3t_comps/pfs17	atlas1
/data3t_comps/pfs18	atlas2
/data3t_comps/pfs19	atlas3
/data3t_comps/pfs20	atlas0
/data3t_comps/pfs21	atlas1
/data3t_comps/pfs22	atlas2

Table 8–3 Component File Systems for /data3t

Component Path	Server Node
/data3t_comps/pfs23	atlas3
/data3t_comps/pfs24	atlas0
/data3t_comps/pfs25	atlas1
/data3t_comps/pfs26	atlas2
/data3t_comps/pfs27	atlas3
/data3t_comps/pfs28	atlas0
/data3t_comps/pfs29	atlas1
/data3t_comps/pfs30	atlas2
/data3t_comps/pfs31	atlas3

We will use these component file systems to create a PFS file system that will be mounted as /data3t. The pfsmgr set name associated with this PFS file system will be data3t. We will create the data3t PFS file system with a block size of 128KB, a stride size of 512KB, and a stripe count of 4. The stripe count setting means that, by default, a file will only be distributed across a subset of 4 of the 32 components.

For convenience, we can create a file called data3t_comp_list that lists all of the component file systems, and use this file when creating the data3t PFS file system, as shown below.

To create this PFS file system, run the following command:

```
# pfsmgr create data3t /data3t -block 128k -stride 512k \
-stripe 4 -compfile data3t_comp_list
```

where data3t_comp_list is a file containing a list of component file systems; that is, the contents of the Component Path column in Table 8–3.

8.3.2 Increasing the Capacity of a PFS File System

As stated in Section 8.2 on page 8–5, you should plan your PFS file system carefully. The number of component file systems in a PFS file system cannot be extended — you cannot add more component file systems to a PFS file system when it starts to become full.

However, it is possible to extend the size of the individual component file systems, if the file system type permits this. The AdvFS file system, which is the type of file system created by the `scfsmgr` command, permits a file domain to be extended by adding more volumes to it. Therefore, you can use the `scfsmgr add` command to add disk volumes to an existing component file system.

8.3.3 Upgrading a PFS File System

If a PFS file system fails to mount (after being placed online), it may be because the data format has been detected as being from an earlier PFS version. To find out if this is the case, check the log file in the `/var/sra/adm/log/scmountd` directory and search for a message similar to the following:

```
Obsolete pfs version 5.0 in atlasD0:/cluster/members/member1/pfs/mounts/test/.pfsid
use fsck to upgrade to version 6.0
```

In this case, you must update the PFS data format to the new version before the file system can be mounted. For instructions on how to do this, see Chapter 4 of the *HP AlphaServer SC Installation Guide*.

8.3.4 Checking a PFS File System

A PFS file system that mounted successfully may become corrupt for a number of reasons. One possible cause is that a host crashed before completing an update to a PFS file system and the underlying component file systems.

To check and fix the integrity of a PFS file system, perform the following steps:

1. The component file systems must be online and mounted. Check the status of the component file systems by running the following command:
   ```
   # scfsmgr show pfs_set
   ```
   where *pfs_set* is the name of the PFS file system you are checking. If any of the component file systems are offline, use the `online` command to mount them.
2. Log into the FS domain that is serving the components of the PFS file system. Run the `fsck_pfs` command as shown in the following example:
   ```
   atlasD0# fsck_pfs -o p /data1
   ```
   This command checks and automatically corrects simple errors.
3. To correct other errors, run the `fsck_pfs` command as shown in the following example:
   ```
   atlasD0# fsck_pfs /data1
   ```
   The command prompts you if it finds any errors that need to be corrected.

See the `fsck_pfs(8)` reference page for more information about this command.

8.3.5 Exporting a PFS File System

In HP AlphaServer SC Version 2.6 systems, it is not possible to export a PFS file system directly from a domain to another domain, or to an external system, using NFS or SCFS. If you wish to access a PFS file system on another domain, you must export the PFS component file systems from an FS domain to the target system, using either NFS or SCFS. You can then mount the PFS file system locally on the target system.

To enable sharing of PFS components in an HP AlphaServer SC Version 2.6 system, use the `scfsmgr` command to create PFS component file systems. This command automatically adds entries for SCFS-managed file systems to the `/etc/exports` file of the SCFS FS domain, permitting the Compute-Serving (CS) domains to mount the PFS component file systems. This allows all of the nodes in the CS domains to mount the associated PFS file systems.

If PFS component file systems are already mounted on the original mount paths in a domain, PFS will use these component paths, rather than privately NFS-mounting the file system under the `/pfs/admin` hierarchy. This permits the components to be mounted with specific SCFS settings. The `pfsmgr` command verifies that all of the component file systems for a PFS file system are mounted, and accessible, before attempting to mount a PFS.

Therefore, if the PFS components are created using `scfsmgr`, and the PFS file system is mounted using `pfsmgr`, you do not have to do any work to share a PFS file system between domains in an HP AlphaServer SC Version 2.6 system.

8.3.6 Backing up a PFS File System

There are two important points that you should note when backing up a PFS file system, as follows:

- Standard backup utilities do not preserve stripe counts.

  In general, standard backup utilities such as `vdump`, `tar`, and so on, work on PFS file systems as they do on file systems of other types. However, there is one difference: standard backup utilities do not preserve stripe counts. When these backup utilities are used to back up a PFS file system, no data is lost, but all restored files are created using the default stripe count of the PFS file system and any custom stripe counts are lost.

- PFS file systems must be backed up at PFS file system level.

  You cannot reconstruct a PFS file system from individual backups of its components: you must back up and restore the file system at the PFS file system level. The stripes of a PFS file are held together by the inode number of the stripe on the first (root) component. The other components contain additional data (the other stripes of the file) keyed to

inodes on the root component. If this structure is disrupted in any way, particularly if inodes on the root component change, the links between the stripes of the file will be broken and the PFS file system will be irretrievably corrupted.

To back up a PFS file system, you must start from its mount point. For example, if the PFS file system `/pfs0` consists of the components `/scfs0`, `/scfs1`, and `/scfs2`, backing up and restoring files from the `/pfs0` mount point will work and all data will be saved and restored correctly (with the default stripe count).

Backing up the `/scfs0`, `/scfs1`, and `/scfs2` components individually would not be a useful operation, because you could not reconstruct the PFS file system from these backups.

8.4 The PFS Management Utility: pfsmgr

To manage PFS file systems within an HP AlphaServer SC Version 2.6 system, use the `pfsmgr` command. You can use this command to create, check, and delete PFS file systems on an FS domain, and to manage the mounting and unmounting of the PFS file systems globally across the HP AlphaServer SC system. See the `pfsmgr(8)` and the `mkfs_pfs(8)` reference pages for more information about the `pfsmgr` command.

On a system that is not a domain — for example, a management server — you cannot use the `pfsmgr` command to mount or unmount PFS file systems. Instead, you must use the low-level PFS management commands `mount_pfs` and `umount_pfs` to mount and unmount the PFS file system on the external system,. See the `mount_pfs(8)` and `umount_pfs(8)` reference pages for more information about these commands.

PFS file systems are managed by the file-system management system that also manages SCFS file systems. The `scfsmgr status` and `scfsmgr sync` commands also affect PFS file systems. See Section 7.7 on page 7–28 for an overview of how the file-system management system works.

8.4.1 PFS Configuration Attributes

A PFS file system is described by the following attributes:

- Mount Point

  This is the directory path on which the PFS file system is mounted. The same mount point is used on all nodes in the HP AlphaServer SC system.

- ONLINE or OFFLINE

  You do not directly mount or unmount PFS file systems. Instead, you mark the PFS file system as ONLINE or OFFLINE. When you mark a PFS file system as ONLINE, the system will mount the PFS file system on all nodes. When you mark the PFS file system as OFFLINE, the system will unmount the file system on all nodes.

The state is persistent. For example, if a PFS file system is marked ONLINE and the system is shut down and then rebooted, the PFS file system will be mounted as soon as the system has completed booting.

- Mount Status

  This indicates whether a PFS file system is mounted or not. This attribute is specific to a domain (that is, each domain has a mount status). The mount status values are listed in Table 8–4.

Table 8–4 PFS Mount Status Values

Mount Status	Description
`mounted`	The PFS file system is mounted on all active members of the domain.
`not-mounted`	The PFS file system is not mounted on any member of the domain.
`mounted-busy`	The PFS file system cannot be unmounted on at least one member of the domain, because the PFS file system is in use.
`mounted-partial`	The PFS file system is mounted by some members of a domain. Normally, a file system is mounted or unmounted by all members of a domain. However, errors in the system may mean that a mount or unmount fails on a specific node or that the node cannot be contacted.
`mount-failed`	An attempt was made to mount the PFS file system on every node in the domain, but the `mount_pfs` command failed. If the `mount_pfs` command worked on some nodes but failed on other nodes, the status is set to `mounted-partial` instead of `mount-failed`. To see why the `mount_pfs` command failed, review the `/var/sra/adm/log/scmountd/pfsmgr.`*`nodename`*`.log` file on the domain where the mount failed.

8.4.2 pfsmgr Commands

This section describes the following `pfsmgr` commands:

- pfsmgr create (see Section 8.4.2.1 on page 8–15)
- pfsmgr delete (see Section 8.4.2.2 on page 8–16)
- pfsmgr offline (see Section 8.4.2.3 on page 8–16)
- pfsmgr online (see Section 8.4.2.4 on page 8–17)
- pfsmgr show (see Section 8.4.2.5 on page 8–18)
- pfsmgr try_mount (see Section 8.4.2.6 on page 8–19)
- pfsmgr try_unmount (see Section 8.4.2.7 on page 8–20)

8.4.2.1 pfsmgr create

Use the `pfsmgr create` command to create a new PFS file system, given a list of component file systems. The pathname of a component file system must reside within an SCFS file system.

The syntax of this command is as follows:

```
pfsmgr create pfs_set mountpoint
[-access mode] [-numcomps num_comps] [-block block_size]
[-stride stride_size] [-stripe stripe_count]
[-compfile comp_file | comp ... ]
```

where:

- *pfs_set* specifies a unique PFS Set name — you cannot specify the keyword `all` as a PFS Set name
- *mountpoint* specifies the mount point for the PFS
- *mode* specifies the access mode — the default value is `rw` (read-write); `ro` (read-only) access is not supported
- *num_comps* specifies the number of component file systems — the default is the number of specified components
- *block* specifies the block size of PFS I/O operations
- *stride* specifies the stride size of a PFS component
- *stripe* specifies the number of components a file is striped across by default
- *comp_file* specifies a file containing a list of component file system paths; if '-' is specified, reads from standard input
- *comp* ... is a list of component file system paths specified on the command line

Note:

Values for *block* and *stride* can be specified as byte values, or suffixed with `K` for Kilobytes, `M` for Megabytes, or `G` for Gigabytes.

Example:

```
# pfsmgr create pfs_1t /pfs_1t -numcomps 8 -stride 512K -stripe 4 \
/d128g_a /d128g_b /d128g_c /d128g_d \
/d128g_e /d128g_f /d128g_g /d128g_h
```

8.4.2.2 pfsmgr delete

Use the `pfsmgr delete` command to destroy a PFS file system. This means that the contents of the PFS component file systems will be deleted, along with the associated PFS configuration data. Also, if requested, the mount point will be deleted globally across the HP AlphaServer SC Version 2.6 system.

If you are using the `pfsmgr delete` command to remove a mount point, and the global operation reports an error, please delete the mount point manually, if required, on the other domains within the HP AlphaServer SC Version 2.6 system.

The syntax of this command is as follows:
`pfsmgr delete [-rm] pfs_set|mountpoint`

where:

- *`pfs_set`* specifies a name that matches exactly one configured PFS Set
- *`mountpoint`* specifies a path that matches exactly one configured PFS mount point
- `-rm` specifies that the mount point is removed also

Example:

```
# pfsmgr delete pfs_1t
```

Note:

The `pfsmgr delete` command requires that the PFS file system is offline and not currently mounted.

In addition, the underlying component SCFS file systems must be online and all mounted by at least one FS or CS domain.

If one of the underlying SCFS file systems cannot be mounted (for example, because the underlying AdvFS domain has been deleted), the `pfsmgr delete` command cannot be used to delete the PFS.

A process that allows you to delete the PFS file system manually is described in Section 29.21.2 on page 29–42.

8.4.2.3 pfsmgr offline

The `pfsmgr offline` command changes the desired state of the specified PFS file systems to OFFLINE in the database and triggers a `sync` operation. The `pfsmgr offline` command completes as soon as the `sync` request has been sent to the `scmountd` daemon; the `sync` operation itself is carried out in the background, and there may be a delay of a few minutes before the file systems are actually unmounted. During this time, you can monitor the progress of the `sync` operation using the `scfsmgr status` command.

During the sync operation, the scmountd daemon examines all SCFS and PFS file systems on all domains and nodes of the HP AlphaServer SC system, and mounts or unmounts the file systems as required to achieve their desired states as defined in the database. For more information on the sync operation, see Section 7.4 on page 7–8.

Only one sync operation can be in progress at a time; issuing further online or offline commands while a sync operation is in progress will only queue more sync requests.

If you specify a PFS Set name or mount point, the pfsmgr offline command places the specified PFS file system offline. If you specify the keyword all, all PFS file systems are placed offline.

The syntax of this command is as follows:

```
pfsmgr offline [pfs_set|mountpoint|all]
```

where:

- *pfs_set* specifies a name that matches exactly one configured PFS Set
- *mountpoint* specifies a path that matches exactly one configured PFS mount point
- all specifies that all PFS Sets should be placed offline

Examples:

```
# pfsmgr offline pfs_1t
# pfsmgr offline /pfs_1t
```

8.4.2.4 pfsmgr online

The pfsmgr online command changes the desired state of the specified PFS file systems to ONLINE in the database and triggers a sync operation. The pfsmgr online command completes as soon as the sync request has been sent to the scmountd daemon; the sync operation itself is carried out in the background, and there may be a delay of a few minutes before the file systems are actually mounted. During this time, you can monitor the progress of the sync operation using the scfsmgr status command.

During the sync operation, the scmountd daemon examines all SCFS and PFS file systems on all domains and nodes of the HP AlphaServer SC system, and mounts or unmounts the file systems as required to achieve their desired states as defined in the database. For more information on the sync operation, see Section 7.4 on page 7–8.

Only one sync operation can be in progress at a time; issuing further online or offline commands while a sync operation is in progress will only queue more sync requests.

Note:

PFS file systems must be mounted by each node. This is in contrast to other file systems, which are visible to all members of a domain once the file system is mounted by any member.

If you specify a PFS Set name or mount point, the `pfsmgr online` command places the specified PFS file system online. If you specify the keyword `all`, all PFS file systems are placed online.

The syntax of this command is as follows:

```
pfsmgr online [pfs_set|mountpoint|all]
```

where:

- *`pfs_set`* specifies a name that matches exactly one configured PFS Set
- *`mountpoint`* specifies a path that matches exactly one configured PFS mount point
- `all` specifies that all PFS Sets should be placed online

Examples:

```
# pfsmgr online pfs_1t
# pfsmgr online /pfs_1t
```

8.4.2.5 pfsmgr show

The `pfsmgr show` command shows the state of PFS file systems.

The syntax of this command is as follows:

```
pfsmgr show [pfs_set|mountpoint] ...
```

where:

- *`pfs_set`* specifies a name that matches one or more configured PFS Sets
- *`mountpoint`* specifies a path that matches one or more configured PFS mount points

Examples:

If you do not specify a PFS Set name or mount point, the `pfsmgr show` command shows all PFS file systems, as shown in the following example:

```
# pfsmgr show
offline    /data      not-mounted: atlasD[0-3]
online     /pscr      mounted: atlasD[0-2] not-mounted: atlas3
```

If you specify a PFS Set name or mount point, detailed information about this PFS file system is displayed, as shown in the following example:

```
# pfsmgr show /pscr
PFS Set: data
State: online
Component Filesystems:
State       Mountpoint         Server              Mount status
-----       ----------         ------              ------------
ONLINE      /scr1              atlas0              mounted: atlasD[0-3]
ONLINE      /scr2              atlas1              mounted: atlasD[0-3]
ONLINE      /scr3              atlas2              mounted: atlasD[0-3]
ONLINE      /scr4              atlas3              mounted: atlasD[0-3]

Mount State:
        Domain          State
        atlasD0         mounted
        atlasD1         mounted
        atlasD2         mounted
        atlasD3         mounted
```

8.4.2.6 pfsmgr try_mount

Caution:

The `pfsmgr try_mount` command is only intended for troubleshooting. It is not a substitute for the `pfsmgr online` command.

The `pfsmgr try_mount` command triggers a PFS file system mount on a selected node or domain. You can use this command on FS domains and CS domains to troubleshoot specific PFS client nodes or client domains that are giving problems. While the file system is being mounted, information about the operation is displayed. This allows you to monitor the commands and actions that are being executed during the mount operation.

The `pfsmgr try_mount` command and the `pfsmgr online` command have different purposes, and they behave in a different way internally (even though they may seem to accomplish a similar task). The effects of the `online` command are more persistent — when the command is executed, certain database tables are modified to ensure that if the system fails or is rebooted, or a command such as `scfsmgr sync` is executed, the state of the mounts can be preserved or restored. When the `try_mount` command is executed, only values that are used for informational purposes are updated.

The syntax of the `pfsmgr try_mount` command is as follows:
`pfsmgr try_mount` *`domain`*|*`node pfs_set`*|*`mountpoint`*

where:

- *`domain`* is the name of the domain where the mount is to be performed
  (Example: `atlasD0`)
- *`node`* is the name of the node where the mount is to be performed
  (Example: `atlas0`)
- *`pfs_set`* specifies a name that matches exactly one configured PFS Set
  (Example: `pfs_1t`)
- *`mountpoint`* specifies a path that matches exactly one configured PFS mount point
  (Example: `/data`)

Examples:

```
# pfsmgr try_mount atlas0 /pfs_mnt
# pfsmgr try_mount atlas0 pfs_set_1

# pfsmgr try_mount atlasD0 /pfs_mnt
# pfsmgr try_mount atlasD0 pfs_set_1
```

8.4.2.7 pfsmgr try_unmount

Caution:

The `pfsmgr try_unmount` command is only intended for troubleshooting. It is not a substitute for the `pfsmgr offline` command.

The `pfsmgr try_unmount` command triggers a PFS file system unmount on a selected node or domain. You can use this command on FS domains and CS domains to troubleshoot specific PFS client nodes or client domains that are giving problems. While the file system is being unmounted, information about the operation is displayed. This allows you to monitor the commands and actions that are being executed during the unmount operation.

The `pfsmgr try_unmount` command and the `pfsmgr offline` command have different purposes, and they behave in a different way internally (even though they may seem to accomplish a similar task). The effects of the `offline` command are more persistent — when the command is executed, certain database tables are modified to ensure that if the system fails or is rebooted, the state of the mounts can be preserved or restored. When the `try_unmount` command is executed, only values that are used for informational purposes are updated.

The syntax of the `pfsmgr try_unmount` command is as follows:
`pfsmgr try_unmount [-f] domain|node pfs_set|mountpoint`

where:

- `-f` forces the unmount operation; if this option is used, user processes are killed before the file system is unmounted
- *`domain`* is the name of the domain where the unmount is to be performed

  (Example: `atlasD0`)
- *`node`* is the name of the node where the unmount is to be performed

  (Example: `atlas0`)
- *`pfs_set`* specifies a name that matches exactly one configured PFS Set

  (Example: `pfs_lt`)
- *`mountpoint`* specifies a path that matches exactly one configured PFS mount point

  (Example: `/data`)

Examples:

```
# pfsmgr try_unmount atlas0 /pfs_mnt
# pfsmgr try_unmount atlasD0 pfs_set_1
# pfsmgr try_unmount -f atlas0 /pfs_mnt
# pfsmgr try_unmount -f atlasD0 pfs_set_1
```

8.4.3 Managing PFS File Systems Using sysman

In HP AlphaServer SC Version 2.6, you can also create, mount, unmount, and delete PFS file systems, and show PFS file system configuration information, by using the `sysman` tool.

To manage PFS file systems with `sysman`, either choose AlphaServer SC *Configuration* > *Manage PFS File Systems* from the SysMan Menu, or start `sysman` with `pfsmgr` as the command line argument.

The following PFS management options are available using `sysman`:

- Create...

  This option allows you to create a new PFS file system by specifying the mount point, components, stride size, block size, and stripe count as detailed for the `pfsmgr create` command (see Section 8.4.2.1 on page 8–15).
- Show...

  This option allows you to view the configuration information for the selected PFS file system.

- Online

  This option allows you to mark the specified PFS file system as ONLINE.

- Offline

  This option allows you to mark the specified PFS file system as OFFLINE.

- Delete...

  This option allows you to delete the specified PFS file system, selecting whether to delete the mount point or not.

8.5 Using a PFS File System

A PFS file system supports POSIX semantics and can be used in the same way as any other Tru64 UNIX file system (for example, UFS or AdvFS), except as follows:

- PFS file systems are mounted with the `nogrpid` option implicitly enabled. Therefore, SVID III semantics apply. For more details, see the AdvFS/UFS options for the `mount(8)` command.
- The layout of the PFS file system, and of files residing on it, can be interrogated and changed using special PFS ioctl calls (see Section 8.5.3 on page 8–25).
- The PFS file system does not support file locking using the `lockf(2)`, `fcntl(2)`, or `lockf(3)` interfaces.
- PFS provides support for the `mmap()` system call for multicomponent file systems, sufficient to allow the execution of binaries located on a PFS file system. This support is, however, not always robust enough to support how some compilers, linkers, and profiling tools make use of the `mmap()` system call when creating and modifying binary executables. Most of these issues can be avoided if the PFS file system is configured to use a stripe count of 1 by default; that is, use only a single data component per file.
- PFS does not support quotas or access control lists (ACLs).
- A PFS file system can only be created for read-write (`rw`) access; read-only (`ro`) access is not supported.

The information in this section is organized as follows:

- Creating PFS Files (see Section 8.5.1 on page 8–23)
- Optimizing a PFS File System (see Section 8.5.2 on page 8–23)
- PFS Ioctl Calls (see Section 8.5.3 on page 8–25)

8.5.1 Creating PFS Files

When a user creates a file, it inherits the default layout characteristics for that PFS file system, as follows:

- Stride size — the default value is inherited from the `mkfs_pfs` command.
- Number of component file systems — the default is to use all of the component file systems.
- File system for the initial stripe — the default value for this is chosen at random.

You can override the default layout on a per-file basis using the `PFSIO_SETMAP` ioctl on file creation.

Note:

This will truncate the file, destroying the content. See Section 8.5.3.3 on page 8–26 for more information about the `PFSIO_SETMAP` ioctl.

PFS file systems also have the following characteristics:

- Copying a sequential file to a PFS file system will cause the file to be striped. The stride size, number of component file systems, and start file are all set to the default for that file system.
- Copying a file from a PFS file system to the same PFS file system will reset the layout characteristics of the file to the default values.

8.5.2 Optimizing a PFS File System

The performance of a PFS file system is improved if accesses to the component data on the underlying CFS file systems follow the performance guidelines for CFS. The following guidelines will help to achieve this goal:

1. In general, consider the stripe count of the PFS file system.

   If a PFS is formed from more than 8 component file systems, HP recommends setting the default stripe count to a number that is less than the total number of components. This will reduce the overhead incurred when creating and deleting files, and improve the performance of applications that access numerous small-to-medium-sized files.

   For example, if a PFS file system is constructed using 32 components, HP recommends selecting a default stripe count of 8 or 4. The desired stripe count for a PFS file system can be specified when the file system is created, or using the PFSIO_SETDFLTMAP ioctl. See Section 8.5.3.5 on page 8–27 for more information about the PFSIO_SETDFLTMAP ioctl.

2. For PFS file systems consisting of FAST-mounted SCFS components, consider the stride size.

   As SCFS FAST mode is optimized for large I/O transfers, it is important to select a stride size that takes advantage of SCFS while still taking advantage of the parallel I/O capabilities of PFS. HP recommends setting the stride size to at least 512K.

To make efficient use of both PFS and SCFS capabilities, an application should read or write data in sizes that are multiples of the stride size.

For example, a large file is being written to a 32-component PFS, the stripe count for the file is 8, and the stride size is 512K. If the file is written in blocks of 4MB or more, this will make maximum use of both PFS and SCFS capabilities, as it will generate work for all of the component file systems on every write. However, setting the stride size to 64K and writing in blocks of 512K is not a good idea, as it will not make good use of SCFS capabilities.

3. For PFS file systems consisting of UBC-mounted SCFS components, follow these guidelines:

   - Avoid False Sharing

     Try to lay the file out across the component file systems such that only one node is likely to access a particular stripe of data. This is especially important when writing data.

     False sharing occurs when two nodes try to get exclusive access to different parts of the same file. This causes the nodes to repeatedly seek access to the file, as their privileges are revoked.

   - Maximize Caching Benefits

     A second order effect that can be useful is to ensure that regions of a file are distributed to individual nodes. If one node handles all the operations on a particular region, then the CFS Client cache is more likely to be useful, reducing the network traffic associated with accessing data on remote component file systems.

File system tools, such as backup and restore utilities, can act on the underlying CFS file system without integrating with the PFS file system.

External file managers and movers, such as the High Performance Storage System (HPSS) and the parallel file transfer protocol (pftp), can achieve good parallel performance by accessing PFS files in a sequential (stride = 1) fashion. However, the performance may be further improved by integrating the mover with PFS, so that it understands the layout of a PFS file. This enables the mover to alter its access patterns to match the file layout.

8.5.3 PFS Ioctl Calls

Valid PFS ioctl calls are defined in the `map.h` header file (`sys/fs/pfs/map.h`) on an installed system. A PFS ioctl call requires an open file descriptor for a file (either the specific file being queried or updated, or any file) on the PFS file system.

In PFS ioctl calls, the N different component file systems are referred to by index number (`0` to `N-1`). The index number is that of the corresponding symbolic link in the component file system root directory (see Table 8–1).

The sample program `ioctl_example.c`, provided in the `/Examples/pfs-example` directory on the *HP AlphaServer SC System Software* CD-ROM, demonstrates the use of PFS ioctl calls.

HP AlphaServer SC Version 2.6 supports the following PFS ioctl calls:

- PFSIO_GETFSID (see Section 8.5.3.1 on page 8–26)
- PFSIO_GETMAP (see Section 8.5.3.2 on page 8–26)
- PFSIO_SETMAP (see Section 8.5.3.3 on page 8–26)
- PFSIO_GETDFLTMAP (see Section 8.5.3.4 on page 8–26)
- PFSIO_SETDFLTMAP (see Section 8.5.3.5 on page 8–27)
- PFSIO_GETFSMAP (see Section 8.5.3.6 on page 8–27)
- PFSIO_GETLOCAL (see Section 8.5.3.7 on page 8–28)
- PFSIO_GETFSLOCAL (see Section 8.5.3.8 on page 8–29)

8.5.3.1 PFSIO_GETFSID

Description: For a given PFS file, retrieves the ID for the PFS file system.
This is a unique 128-bit value.

Data Type: `pfsid_t`

Example: `376a643c-000ce681-00000000-4553872c`

8.5.3.2 PFSIO_GETMAP

Description: For a given PFS file, retrieves the mapping information that specifies how it is laid out across the component file systems.
This information includes the number of component file systems, the ID of the component file system containing the first data block of a file, and the stride size.

Data Type: `pfsmap_t`

Example: The PFS file system consists of two components, 64KB stride:
Slice: Base = 0 Count = 2
Stride: 65536
This configures the file to be laid out with the first block on the first component file system, and a stride size of 64KB.

8.5.3.3 PFSIO_SETMAP

Description: For a given PFS file, sets the mapping information that specifies how it is laid out across the component file systems. Note that this will truncate the file, destroying the content.
This information includes the number of component file systems, the ID of the component file system containing the first data block of a file, and the stride size.

Data Type: `pfsmap_t`

Example: The PFS file system consists of three components, 64KB stride:
Slice: Base = 2 Count = 3
Stride: 131072
This configures the file to be laid out with the first block on the third component file system, and a stride size of 128KB. (The stride size of the file can be an integral multiple of the PFS block size.)

8.5.3.4 PFSIO_GETDFLTMAP

Description: For a given PFS file system, retrieves the default mapping information that specifies how newly created files will be laid out across the component file systems.
This information includes the number of component file systems, the ID of the component file system containing the first data block of a file, and the stride size.

Data Type: `pfsmap_t`

Example: See `PFSIO_GETMAP` (Section 8.5.3.2 on page 8–26).

8.5.3.5 PFSIO_SETDFLTMAP

Description: For a given PFS file system, sets the default mapping information that specifies how newly created files will be laid out across the component file systems.
This information includes the number of component file systems, the ID of the component file system containing the first data block of a file, and the stride size.

Data Type: `pfsmap_t`

Example: See `PFSIO_SETMAP` (Section 8.5.3.3 on page 8–26).

Note:

The scope of the PFSIO_SETDFLTMAP ioctl is a node. Adjustments made by the call are not propagated to other nodes. You must apply the same changes to each of the relevant nodes.

8.5.3.6 PFSIO_GETFSMAP

Description: For a given PFS file system, retrieves the number of component file systems, and the default stride size.

Data Type: `pfsmap_t`

Example: The PFS file system consists of eight components, 128KB stride:
Slice: Base = 0 Count = 8
Stride: 131072
This configures the file to be laid out with the first block on the first component file system, and a stride size of 128KB. For `PFSIO_GETFSMAP`, the base is always 0 — the component file system layout is always described with respect to a base of 0.

8.5.3.7 PFSIO_GETLOCAL

Description: For a given PFS file, retrieves information that specifies which parts of the file are local to the host.
This information consists of a list of slices, taken from the layout of the file across the component file systems, that are local. Blocks laid out across components that are contiguous are combined into single slices, specifying the block offset of the first of the components, and the number of contiguous components.

Data Type: `pfsslices_ioctl_t`

Example: a) The PFS file system consists of three components, all local, file starts on first component:
Size: 3
Count: 1
Slice: Base = 0 Count = 3
b) The PFS file system consists of three components, second is local, file starts on first component:
Size: 3
Count: 1
Slice: Base = 1 Count = 1
c) The PFS file system consists of three components, second is remote, file starts on first component:
Size: 3
Count: 2
Slices: Base = 0 Count = 1
Base = 2 Count = 1
d) The PFS file system consists of three components, second is remote, file starts on second component:
Size: 3
Count: 1
Slice: Base = 1 Count = 2

8.5.3.8 PFSIO_GETFSLOCAL

Description: For a given PFS file system, retrieves information that specifies which of the components are local to the host.
This information consists of a list of slices, taken from the set of components, that are local. Components that are contiguous are combined into single slices, specifying the ID of the first component, and the number of contiguous components.

Data Type: `pfsslices_ioctl_t`

Example:
a) The PFS file system consists of three components, all local:
Size: 3
Count: 1
Slice: Base = 0 Count = 3
b) The PFS file system consists of three components, second is local:
Size: 3
Count: 1
Slice: Base = 1 Count = 1
c) The PFS file system consists of three components, second is remote:
Size: 3
Count: 2
Slices: Base = 0 Count = 1
Base = 2 Count = 1

8.6 SC Database Tables Supporting PFS File Systems

Note:

This section is provided for informational purposes only, and is subject to change in future releases.

This section describes the SC database tables that are used by the PFS file-system management system.

This section describes the following tables:

- The sc_pfs Table (see Section 8.6.1 on page 8–30)
- The sc_pfs_mount Table (see Section 8.6.2 on page 8–30)
- The sc_pfs_components Table (see Section 8.6.3 on page 8–31)
- The sc_pfs_filesystems Table (see Section 8.6.4 on page 8–31)

8.6.1 The sc_pfs Table

The sc_pfs table specifies the attributes of a PFS file system. This table contains one record for each PFS Set. Table 8–5 describes the fields in the sc_pfs table.

Table 8–5 The sc_pfs Table

Field	Description
pfs_set	The name of the PFS Set
mount_point	The pathname of the mount point of the PFS file system
rw	Specifies the access mode for the PFS file system. The default value is rw; the ro access mode is not supported.
status	Specifies whether the PFS file system is ONLINE or OFFLINE
root_component_fs	The pathname of the root (that is, first) component file system

8.6.2 The sc_pfs_mount Table

The sc_pfs_mount table specifies the mount status of each PFS file system on each domain. Each PFS file system has a record for each domain. Table 8–6 describes the fields in the sc_pfs_mount table.

Table 8–6 The sc_pfs_mount Table

Field	Description
pfs_set	The name of the PFS Set
cluster_name	The name of the FS or CS domain to which the mount status applies
state	The mount status for this PFS file system on the specified FS or CS domain

8.6.3 The sc_pfs_components Table

The `sc_pfs_components` table specifies the pathnames of each component file system for a given PFS. Table 8–7 describes the fields in the `sc_pfs_components` table.

Table 8–7 The sc_pfs_components Table

Field	Description
`pfs_set`	The name of the PFS Set
`component_fs_path`	The pathname of a component file system for the specified PFS file system, and an index that orders the component file systems

8.6.4 The sc_pfs_filesystems Table

The `sc_pfs_filesystems` table specifies the SCFS file systems that underlie a given PFS file system. If complete SCFS file systems are used as components of PFS file systems, the `sc_pfs_components` and `sc_pfs_filesystems` tables contain the same data. However, if the pathnames of several components are within an SCFS file system, the `sc_pfs_filesystems` table has fewer entries. If a given SCFS has components of several PFS file systems within it, the `sc_pfs_filesystems` table has more entries.

Table 8–8 describes the fields in the `sc_pfs_filesystems` table.

Table 8–8 The sc_pfs_filesystems Table

Field	Description
`pfs_set`	The name of the PFS Set
`scfs_fs_name`	The name (mount point) of the SCFS file system

9
Managing Events

An HP AlphaServer SC system contains many different components: nodes, terminal servers, Ethernet switches, HP AlphaServer SC Interconnect switches, storage subsystems, file systems, partitions, system software, and so on.

A critical part of an HP AlphaServer SC system administrator's job is to monitor the state of the system, and to be ready to take action when certain unusual conditions occur, such as when a disk fills or a processor reports hardware errors. It is also important to verify that certain routine tasks run successfully each day, and to review certain system configuration values. Such conditions or task completions are described as **events**.

An event is an indication that something interesting has occurred — an action has been taken, some condition has been met, or it is time to confirm that an application is still operational. This chapter describes how to manage events.

This information in this chapter is arranged as follows:

- Event Overview (see Section 9.1 on page 9–2)
- HP AlphaServer SC Event Filter Syntax (see Section 9.2 on page 9–6)
- Viewing Events (see Section 9.3 on page 9–9)
- Event Examples (see Section 9.4 on page 9–10)
- Notification of Events (see Section 9.5 on page 9–13)
- Event Handler Scripts (see Section 9.6 on page 9–18)

9.1 Event Overview

When a software component determines that something has happened to either the hardware or software of the system, and that this incident may be of interest to a user or system administrator, it posts an event.

A event comprises the following information:

- *Timestamp*: This indicates when the event occurred.
- *Name*: This is the name of the object affected by the event.
- *Class*: This is the type or class of object affected by the event.
- *Type*: This is the type of event.
- *Description*: This provides additional information about the event.

Events are stored in the SC database, and can be processed or viewed in the following ways:

- Use the SC Viewer or the `scevent` command to view events (see Section 9.3 on page 9–9).
- Use the `scalertmgr` command to configure the system to send e-mail alerts when particular events occur (see Section 9.5 on page 9–13).
- Create site-specific event handler scripts (see Section 9.6 on page 9–18).

There are three important concepts that will help you to analyze events:

- Event Category (see Section 9.1.1 on page 9–3)
- Event Class (see Section 9.1.2 on page 9–3)
- Event Severity (see Section 9.1.3 on page 9–6)

Both the `scevent` command and the `scalertmgr` command display events grouped by severity and category, to make it easier to identify events of interest. However, it is also possible to identify events by their name, class, or type.

9.1.1 Event Category

The event **category** indicates the subsystem that posted the event.

Table 9–1 lists the HP AlphaServer SC event categories in alphabetical order.

Table 9–1 HP AlphaServer SC Event Categories

Category	Description
`domain`	Events that are specific to, or local to, a domain.
`filesystem`	Events that are related to file systems.
`hardware`	Events that are related to hardware components.
`install`	Events that are related to the software installation process.
`interconnect`	Events that are related to the HP AlphaServer SC Interconnect.
`misc`	All events not covered by the other categories.
`network`	Events that are related to networks.
`resource`	Events that are related to resource or job management.
`software`	Events that are related to software.

9.1.2 Event Class

The event **class** provides additional information about the component that posted the event.

Table 9–2 lists the HP AlphaServer SC event classes in alphabetical order.

Table 9–2 HP AlphaServer SC Event Classes

Class	Description
`action`	An action associated with an `sra` command has changed state.
`advfs`	Something of interest has happened to an AdvFS file system — from either a file system or domain perspective.
`appliance`	Something of interest has happened to a SANworks Management Appliance.
`boot_cmd`	An `sra boot` command has been initiated or has changed state.
`caa`	Indicates change to CAA on a particular domain.
`cfs`	Indicates change to a CFS file system.

Table 9–2 HP AlphaServer SC Event Classes

Class	Description
clu	Something of interest has happened to members on a particular domain.
clua	Something of interest has happened to the cluster alias on a particular domain or network.
cmfd	Something of interest has happened to the console network — from either a hardware or software perspective.
cnx	Something of interest has happened to cluster connections — from either a domain or network perspective.
database	The SC database integrity check utility has run.
domain	A domain has changed state.
extreme	Something of interest has happened to the Extreme switch hardware.
hsg	Something of interest has happened to an HSG80 RAID storage system.
hsv	Something of interest has happened to an HSV110 RAID storage system.
install_cmd	An `sra install` command has been initiated or has changed state.
lsf	Something of interest has happened in the Platform LSF for HP AlphaServer SC system.
nfs	Something of interest has happened to an NFS file system.
node	Something of interest has happened to a node — from either a hardware or resource perspective.
partition	Something of interest has happened to a partition, from a resource perspective.
pfs	Something of interest has happened to a PFS file system.
procurve	Something of interest has happened to the ProCurve switch hardware.
scfs	Something of interest has happened to an SCFS file system.
scmon	Something of interest has happened to the SC Monitor (`scmon`) system.
server	Something of interest has happened to an RMS server (daemon).
shutdown_cmd	An `sra shutdown` command has been initiated or has changed state.
switch_module	Something of interest has happened to an HP AlphaServer SC Interconnect switch.
transaction	Something of interest has happened in the RMS transaction system.
tserver	Something of interest has happened to a console network terminal server.
unix.hw	Something of interest has happened to a hardware device managed by Tru64 UNIX.

To display a list of all of the possible events for a particular class, use the `scevent -l` command. For example, to list all possible events for the `advfs` class, run the following `scevent` command:

```
# scevent -l -f '[class advfs]'
Severity     Category              Class     Type                      Description
-------------------------------------------------------------------------------
event        domain filesystem     advfs     fdmn.addvol               (null)
warning      filesystem            advfs     fdmn.bad.mcell.list       (null)
warning      filesystem            advfs     fdmn.bal.error            (null)
event        domain filesystem     advfs     fdmn.bal.lock             (null)
event        domain filesystem     advfs     fdmn.bal.unlock           (null)
warning      filesystem            advfs     fdmn.frag.error           (null)
event        filesystem            advfs     fdmn.frag.lock            (null)
event        filesystem            advfs     fdmn.frag.unlock          (null)
event        filesystem            advfs     fdmn.full                 (null)
event        domain filesystem     advfs     fdmn.mk                   (null)
failed       filesystem            advfs     fdmn.panic                (null)
event        domain filesystem     advfs     fdmn.rm                   (null)
event        filesystem            advfs     fdmn.rmvol.error          (null)
event        domain filesystem     advfs     fdmn.rmvol.lock           (null)
event        filesystem            advfs     fdmn.rmvol.unlock         (null)
warning      filesystem            advfs     fset.backup.error         (null)
event        domain filesystem     advfs     fset.backup.lock          (null)
event        filesystem            advfs     fset.backup.unlock        (null)
warning      filesystem            advfs     fset.bad.frag             (null)
event        domain filesystem     advfs     fset.clone                (null)
event        domain filesystem     advfs     fset.mk                   (null)
info         domain filesystem     advfs     fset.mount                (null)
info         domain filesystem     advfs     fset.options              (null)
info         domain filesystem     advfs     fset.quota.hblk.limit     (null)
info         domain filesystem     advfs     fset.quota.hfile.limit    (null)
info         domain filesystem     advfs     fset.quota.sblk.limit     (null)
info         domain filesystem     advfs     fset.quota.sfile.limit    (null)
event        domain filesystem     advfs     fset.rename               (null)
warning      domain filesystem     advfs     fset.rm.error             (null)
event        domain filesystem     advfs     fset.rm.lock              (null)
event        filesystem            advfs     fset.rm.unlock            (null)
info         filesystem            advfs     fset.umount               (null)
info         domain filesystem     advfs     quota.off                 (null)
info         domain filesystem     advfs     quota.on                  (null)
info         domain filesystem     advfs     quota.setgrp              (null)
info         domain filesystem     advfs     quota.setusr              (null)
warning      domain filesystem     advfs     special.maxacc            (null)
```

9.1.3 Event Severity

The event **severity** indicates the importance of the event. For example, some events indicate a problem with the system, while other events are merely informational messages.

Table 9–3 lists the event severities in decreasing order.

Table 9–3 HP AlphaServer SC Event Severities

Severity	Description
`failed`	Indicates a failure in a component.
`warning`	Indicates an error condition.
`normal`	Indicates that the object in question has returned from the `failed` or `warning` state.
`event`	An event has occurred. Generally, the event is triggered directly or indirectly by user action.
`info`	An event has occurred. Generally, users do not need to be alerted about these events, but the event is worth recording for later analysis.

9.2 HP AlphaServer SC Event Filter Syntax

An HP AlphaServer SC system may generate many events over the course of a day. Therefore, you may want to limit your view to the particular set in which you are interested. For example, you may want to see the events posted for one particular category, or all events with a high severity value. Events can be selected by using an **event filter** — that is, a character string that describes the selection using a predefined filter syntax.

You can use a filter to select events according to several different criteria, including event name, timestamp, severity, and category. Filters can be used both when viewing events (see Section 9.3 on page 9–9) and when setting up alerts (see Section 9.5 on page 9–13).

Table 9–4 describes the supported HP AlphaServer SC event filter syntax.

Note:

The quotation marks and square brackets must be included in the filter specification.

Table 9–4 HP AlphaServer SC Event Filter Syntax

Filter Specification	Description
'[name *name_expr*]'	Selects events based on their name. *name_expr* can be a comma-separated list, use shell-style wildcards (for example, * and ?), and have ranges enclosed in square brackets. Example: '[name atlas[0-31]]'
'[type *type_expr*]'	Selects events based on their type. *type_expr* can be a comma-separated list or use shell-style wildcards. Example: '[type status,membership]'
'[category *category_expr*]'	Selects events based on their category (see Section 9.1.1 on page 9–3). *category_expr* can be a comma-separated list. Example: '[category domain,hardware]'
'[class *class_expr*]'	Selects events based on their class (see Section 9.1.2 on page 9–3). *class_expr* can be a comma-separated list or use shell-style wildcards. Example: '[class node,domain]'
'[severity *severity_expr*]'	Selects events based on their severity (see Section 9.1.3 on page 9–6). *severity_expr* can be a comma-separated list. Example: '[severity failed,warning]'
'[severity *operator severity_expr*]'	Selects events based on their severity. • *severity_expr* is one of the severities listed in Table 9–3 on page 9–6: • *operator* is one of the operators described in Table 9–5 on page 9–9. Example: '[severity > normal]' shows all events with severity warning or failed.
'[age *operator age_expr*]'[1]	Selects events based on their age. • *age_expr* is a number followed by a letter indicating the unit: w (weeks) d (days) h (hours) m (minutes) s (seconds) • *operator* is one of the operators described in Table 9–5 on page 9–9. Example: '[age < 3d]' shows all events from the last three days.
'[before *absolute_time_spec*]'[1]	Selects events that occurred before the specified time. *absolute_time_spec* has six colon-separated number fields: *year*:*month_of_year*:*day_of_month*:*hour*:*minute*:*second* You cannot use an asterisk in any *absolute_time_spec* field. Example: '[before 2001:9:1:13:37:42]' returns all events that occurred before 1:37:42 p.m. on September 1, 2001
'[after *absolute_time_spec*]'[1]	Selects events that occurred after the specified time. *absolute_time_spec* is the same as for the before keyword. Example: '[after 2001:9:1:13:37:42]' returns all events that have occurred since 1:37:42 p.m. on September 1, 2001

Table 9–4 HP AlphaServer SC Event Filter Syntax

Filter Specification	Description
`'[time time_range_spec]'`[1]	Selects events that occur within a specified range of times. *time_range_spec* has seven colon-separated number fields: *year*:*month_of_year*:*day_of_month*:*day_of_week*:*hour*:*minute*:*second* A *time_range_spec* field may be a comma-separated list, a range, or a wildcard. Multiple ranges are not supported. Wildcards must be followed by wildcards only (except for `day_of_week`, which is not restricted). Only events occurring within the given ranges will be displayed. The valid range of values for each field is as follows: Field — Range `year` — 1970 to 2030 `month_of_year` — 1 to 12 `day_of_month` — 1 to 31 `day_of_week` — 0 (Sunday) to 6 `hours` — 0 to 23 `minutes` — 0 to 59 `seconds` — 0 to 59 Example: `'[time 2002:2:13:*:8-10:*:*]'` returns all events that occurred between 8:00:00 a.m. and 10:59:59 a.m. inclusive on 13 February 2002.
'filter `AND` *filter'*	Selects only events that match both filters. The word `AND` is case-insensitive; the `&` symbol can be used instead of `AND`. Example: `'[class node] AND [type status]'`
'filter `OR` *filter'*	Selects events that match either filter. The word `OR` is case-insensitive; the `|` symbol can be used instead of `OR`. Example: `'[class node] OR [class partition]'`
`'NOT` *filter'*	Selects events that do not match the filter. The word `NOT` is case-insensitive; the `!` symbol can be used instead of `NOT`. Example: `'[type status] NOT [class node]'` **The NOT logical operator is not yet supported.**
`'(`*complex_filter*`)'`	Selects events that match *complex_filter*, where *complex_filter* is composed of two or more simple filters combined using the logical operators `AND` (or `&`), `OR` (or `|`), and `NOT` (or `!`). The order of precedence of these operators (highest to lowest) is `( ) ! & |`. Example: `'([class node] OR [class partition]) AND ([type status])'`
`'@`*name'*	Selects events that match the filter specification associated with the given `name`, using a saved filter. Saved filters are stored in the `/var/sra/scevent/filters` directory in *filter_name*`.filter` files (see Example 9–8). If the name contains any character other than a letter, digit, underscore, or dash, you must enclose the name within quotation marks. Example: `'@name with spaces'`

[1]This filter cannot be used with the `scalertmgr` command (see Section 9.5.1 on page 9–13).

9.2.1 Filter Operators

In HP AlphaServer SC event filter syntax, *operator* is case-insensitive. Table 9–5 describes the supported filter operators.

Table 9–5 Supported HP AlphaServer SC Event Filter Operators

Operator	Alternative Syntax	Description
<	lt	Less than
<=	le	Less than or equal to
>	gt	Greater than
>=	ge	Greater than or equal to

operator must specify less than, or less than or equal (meaning "newer than"); or greater than, or greater than or equal (meaning "older than").

The equal or not equal operators are not allowed.

These operators apply to the age and severity filters only. The descriptions "newer than" and "older than" apply only to the age filters. In the severity filters, the operators establish the order based on that specified in Table 9–3 on page 9–6.

9.3 Viewing Events

You can view events in an HP AlphaServer SC system in either of the following ways:

- Using the SC Viewer to View Events (see Section 9.3.1 on page 9–9)
- Using the scevent Command to View Events (see Section 9.3.2 on page 9–9)

9.3.1 Using the SC Viewer to View Events

Run the `scviewer` command to display the SC Viewer, a Graphical User Interface (GUI) that displays status information for various components of the HP AlphaServer SC system. Select the Events tab to view events for specific objects. You can monitor events in real time or view historical events.

See Chapter 10, and the `scviewer(8)` reference page, for more information about SC Viewer.

9.3.2 Using the scevent Command to View Events

Run the `scevent` command to display the Command Line Interface (CLI) version of the SC Viewer Events tab. The `scevent` command displays the events on standard output.

The `scevent` command is described in Section 9.3.2.1 on page 9–10 and in the `scevent(8)` reference page.

9.3.2.1 scevent Command Syntax

The syntax of the `scevent` command is as follows:

```
scevent [-f filter_spec] [-c] [-h] [-l] [-p] [-v]
```

Table 9–6 describes the command-line options, in alphabetical order.

Table 9–6 scevent Command-Line Options

Option	Description
`-c`	Specifies that `scevent` should display new events continuously as they appear. If `-c` is not specified, matching events are displayed once and then `scevent` exits.
`-f filter_spec`	Specifies a filter for events. If no filter is specified, all events are displayed. Filters are specified as described in Section 9.2 on page 9–6.
`-h`	Specifies that `scevent` should display a header for each column of output.
`-l`	Specifies that `scevent` should not show actual events, but instead should list the possible events that can match the filter. Since actual events are not being shown, the following filters should not be used with the `-l` option: `age`, `before`, `after`, or `time`.
`-p`	Specifies that `scevent` should display page-oriented output, with headers at the top of each page. The size of a page is determined in the same way as that used by the `more(1)` command. This option implies –h.
`-v`	Specifies that `scevent` should display a detailed explanation of the event.

9.4 Event Examples

This section provides the following examples:

- Example 9–1: All Existing Hardware-Related Events of Severity “warning” or “failed”
- Example 9–2: All Existing Events Related to Resource Management
- Example 9–3: Description of All Possible Events Related to Resource Management
- Example 9–4: Additional Information about Event, Including Event Source
- Example 9–5: All Events Related to RMS Partitions in the Previous 24 Hours
- Example 9–6: All Events on atlas0 in the Previous 24 Hours
- Example 9–7: Display Events One Page at a Time
- Example 9–8: Creating and Using a Named Filter

Example 9–1 All Existing Hardware-Related Events of Severity “warning” or “failed”

```
$ scevent -h -f '[severity > warning] and [category hardware]'
Time                Name          Class          Type        Description
-------------------------------------------------------------------------
09/27/01 17:17:47   atlas3        node           status      not responding
09/27/01 17:17:47   atlasms       node           status      not responding
09/28/01 13:27:26   extreme1      extreme        status      not-responding
09/28/01 15:55:53   hsg8          hsg            psu         failed
09/28/01 16:12:35   hsg9          hsg            psu         failed
```

Example 9–2 All Existing Events Related to Resource Management

```
$ scevent -f '[category resource]'
10/12/01 11:52:12   parallel      partition      status      running
10/16/01 12:13:12   parallel      partition      status      blocked
10/16/01 12:14:51   atlas3        node           status      active
10/16/01 12:17:10   atlas3        node           status      running
10/16/01 12:17:12   parallel      partition      status      running
```

Example 9–3 Description of All Possible Events Related to Resource Management

```
$ scevent -l -f '[category resource]'
Severity    Category                Class          Type        Description
-------------------------------------------------------------------------------
info        resource                node           runlevel    (null)
warning     resource hardware       node           status      active
normal      resource hardware       node           status      configured out
failed      resource hardware       node           status      not responding
normal      resource hardware       node           status      running
failed      resource                partition      status      blocked
normal      resource                partition      status      down
normal      resource                partition      status      running
```

Example 9–4 Additional Information about Event, Including Event Source

```
$ scevent -h -l -v -f '[class partition] and [type status]'
Severity            Category      Class          Type        Description
Explanation
-------------------------------------------------------------------------------
failed              resource      partition      status      blocked

Partition is blocked.

More Information:
The partition can block if a node in the partition is no longer running RMS (RMS
stopped, or node has halted or crashed). Use rinfo(1) (with -pl option) to
determine which nodes are in which partition. Then used rinfo -n to determine
whether all nodes in the partition are running.

If the partition does not recover by itself, you can configure out the nodes
causing it to block.

Event Source:
This event is generated by pmanager.
...
```

Example 9–5 All Events Related to RMS Partitions in the Previous 24 Hours

```
$ scevent -f '[age < 1d] and [class partition]'
10/16/01 11:56:06      parallel       partition      status       closing
10/16/01 11:56:11      parallel       partition      status       down
10/16/01 12:17:12      parallel       partition      status       running
```

Example 9–6 All Events on atlas0 in the Previous 24 Hours

```
$ scevent -f '[age < 1d] and [name atlas0]'
10/15/01 14:52:06    atlas0   node    rmc.fan.normal      Fan5 has been turned off
10/15/01 14:52:06    atlas0   node    rmc.psu.no_power    Power Supply PS2 is not
present
10/15/01 15:09:49    atlas0   node    rmc.temp.warning    Zone2 temp 36.0C > warning
threshold of 35.0C
10/15/01 16:17:38    atlas0   node    temperature         ambient=29
10/15/01 16:17:43    atlas0   node    status              running
10/15/01 17:55:55    atlas0   node    rmc.temp.warning    Zone2 temp 36.0C > warning
threshold of 35.0C
10/16/01 10:25:03    atlas0   node    status              not responding
10/16/01 10:27:10    atlas0   node    status              running
```

Example 9–7 Display Events One Page at a Time

```
$ scevent -h -p -f '[after 2001:10:16:08:17:00]'
Time                    Name         Class     Type              Description
------------------------------------------------------------------------------
10/16/01 08:17:07      atlasD2      cfs       advfs.served      CFS: AdvFS domain
root31_local is now served by node atlas94
10/16/01 08:17:07      atlasD2      cfs       advfs.served      CFS: AdvFS domain
root31_local is now served by node atlas94
10/16/01 08:17:07      atlasD2      cfs       advfs.served      CFS: AdvFS domain
root31_local is now served by node atlas94
10/16/01 08:17:07      atlasD2      cfs       advfs.served      CFS: AdvFS domain
root31_local is now served by node atlas94
10/16/01 08:17:14      atlasD2      cfs       advfs.served      CFS: AdvFS domain
root31_local is now served by node atlas94
10/16/01 08:17:07      atlasD2      cfs       advfs.served      CFS: AdvFS domain
root28_local is now served by node atlas91
10/16/01 08:17:07      atlasD2      cfs       advfs.served      CFS: AdvFS domain
root28_local is now served by node atlas91
10/16/01 08:17:08      atlasD2      cfs       advfs.served      CFS: AdvFS domain
root28_local is now served by node atlas91

Press Enter for more...

Time                    Name         Class     Type              Description
------------------------------------------------------------------------------
10/16/01 08:17:08      atlasD2      cfs       advfs.served      CFS: AdvFS domain
root28_local is now served by node atlas91
10/16/01 08:17:15      atlasD2      cfs       advfs.served      CFS: AdvFS domain
root28_local is now served by node atlas91
10/16/01 08:18:07      atlas90      node      temperature       ambient=30
10/16/01 08:53:44      atlas12      node      rmc.temp.warning  Zone1 temp 36.0C >
warning threshold of 35.0C
10/16/01 09:41:23      atlas9       node      temperature       ambient=27
```

Example 9–8 Creating and Using a Named Filter

Note:

You must have `root` permission to run the first two commands below (only the `root` user can write to `/var/sra`), but any user can run the `scevent` command.

```
# mkdir -p /var/sra/scevent/filters
# echo '[category hardware] and [age < 1d]' > \
  /var/sra/scevent/filters/recent_hw.filter
$ scevent -f '@recent_hw'
```

9.5 Notification of Events

The HP AlphaServer SC system uses the following methods to alert operators about problems in the system:

- Using the scalertmgr Command (see Section 9.5.1 on page 9–13)
- Event Handlers (see Section 9.5.2 on page 9–16)

9.5.1 Using the scalertmgr Command

You can configure the system to send e-mail to operators when significant events occur. Use the `scalertmgr` command to specify which events should be sent to which e-mail addresses — this data is stored in the SC database. The `scalertd` daemon periodically checks the event information in the SC database, and takes the appropriate action when it finds an event with a matching alert.

By default, the `scalertd` daemon checks the event information in the SC database every 60 seconds. To change this value, set the SCCONSOLE_IVL environment variable to the new interval (specified in seconds), and restart the `scalertd` daemon.

The `scalertd` log files are stored in the `/var/sra/adm/log/scalertd` directory.

You can use the `scalertmgr` command to perform the following tasks:

- Add an Alert (see Section 9.5.1.1)
- Remove an Alert (see Section 9.5.1.2)
- List the Existing Alerts (see Section 9.5.1.3)
- Change the E-Mail Addresses Associated with Existing Alerts (see Section 9.5.1.4)
- Example E-Mail Alert (see Section 9.5.1.5)

For more information about the scalertmgr command, see the scalertmgr(8) reference page.

9.5.1.1 Add an Alert

To add an alert, use the scalertmgr add command. The syntax of this command is as follows:

```
scalertmgr add [-n alert_name] filter_spec email_address...
```

All events that match the *filter_spec* (see Section 9.2 on page 9–6) are sent to the specified e-mail addresses.The *filter_spec* should not use the age, before, after, or time filters. The alert is optionally given a name, which can be used — by the scalertmgr command only — to refer to the alert later. If the user-specified name is not unique, an error message is generated. If no user-specified name is supplied, scalertmgr supplies a name that is guaranteed to be unique.

For example, to send hardware events to the hwmaint@site.com mail address, run the following scalertmgr command:

```
# scalertmgr -n hw '[category hardware]' hwmaint@site.com
Database updated!
```

9.5.1.2 Remove an Alert

To remove an alert, use the scalertmgr remove command. The syntax of this command is as follows:

```
scalertmgr remove [-i] 'wildcard_alert_name'...
```

If the -i option is specified, matching alerts are displayed one at a time and the user is prompted to confirm that they want to remove the alert. If using a wildcard (*) to remove several alerts, you must enclose the alert name specification within single quotation marks.

For example, to remove the hw event, run the following scalertmgr command:

```
# scalertmgr remove hw
Database updated!
```

9.5.1.3 List the Existing Alerts

To list the existing alerts, use the scalertmgr list command. The syntax of this command is as follows:

```
scalertmgr list [-f filter_spec] [-n 'wildcard_alert_name']
[email_address...]
```

Only alerts that match the *filter_spec* (if supplied), the *wildcard_alert_name* (if supplied), and all of the e-mail addresses (if supplied) will be displayed. If using a wildcard (*) to list several alerts, you must enclose the alert name specification within single quotation marks. Note that wildcards cannot be used in the *filter_spec* or in the e-mail address. See Section 9.2 on page 9–6 for more information about *filter_spec* syntax.

For example, to list the existing alerts, run the following `scalertmgr` command:

```
# scalertmgr list
Name        Filter                    Action      Contact
hw          [category hardware]       e           hwmaint@site.com
```

9.5.1.4 Change the E-Mail Addresses Associated with Existing Alerts

To add e-mail addresses to, or remove e-mail addresses from, existing alerts, use the `scalertmgr email` command. The syntax of this command is as follows:

```
scalertmgr email [-n 'wildcard_alert_name']
{-a email_address...} {-r email_address...}
```

Only alerts matching *`wildcard_alert_name`* will be changed. If using a wildcard (`*`) to change several alerts, you must enclose the alert name specification within single quotation marks. One or more `-a` and `-r` options can be specified, to add (`-a`) or remove (`-r`) multiple e-mail addresses at once.

For example, to add the `other@site.com` e-mail address to the existing `hw` alert, run the following `scalertmgr` command:

```
# scalertmgr email -n hw -a other@site.com
Database updated!
```

9.5.1.5 Example E-Mail Alert

The following example e-mail was triggered by starting a partition:

```
From: system PRIVILEGED account
[mailto:root@atlasms-ext1.site.com]
Sent: 01 November 2001 17:03
To: admin@site.com
Subject: Multiple partition events for parallel

Time of event: 11/01/01 17:02:14
Name:          parallel
Class:         partition
Type:          status
Description:   starting
Severity:      normal
Category:      resource

Explanation:

Partition has been started, but has not yet reached the running stage.

Event Source:
This event is generated by pmanager when the partition is started (by rcontrol
start partition), when the system is booted or when a previously blocked
partition recovers.

------------------------------------------------------------
Time of event: 11/01/01 17:02:14
Name:          parallel
```

```
Class:         partition
Type:          status
Description:   running
Severity:      normal
Category:      resource

Explanation:

Partition has been started.

Event Source:
This event is generated by pmanager when the partition is started (by
rcontrol start partition), when the system is booted or when a previously
blocked partition recovers.

------------------------------------------------------------

Alert Information:
Name:   auto2001Nov01165910
Filter: [class partition]
See scalertmgr(8) for more information.
```

9.5.2 Event Handlers

When the status of a node changes, RMS posts the event to the RMS Event Handler (`eventmgr` daemon). This handler scans the `event_handlers` database looking for handlers for the event. An event handler is a script that is run in response to the event. The RMS event handlers are described in Table 9–7.

Table 9–7 RMS Event Handlers

Event Type	Handler Name	Description
Node status changes	`rmsevent_node`	Triggered whenever the status of a node changes. See Section 9.5.2.1 on page 9–17 for more information.
Node environment events (temperature changes, fan failures and power supply failures)	`rmsevent_env`	Triggered whenever: • The temperature of a node changes by more than 2°C • The temperature of a node exceeds 40°C • A fan fails • A power supply fails See Section 9.5.2.2 on page 9–17 for more information.
Unhandled events	`rmsevent_escalate`	Triggered if one of the previous event handlers fails to run within the specified time. See Section 9.5.2.3 on page 9–18 for more information.

Each event handler has an associated attribute that specifies a list of users to e-mail when the event triggers. There is a different attribute for each type of event. This allows you to decide which events are important and which can be ignored. For example, you might be interested in knowing about fan failures, but not about nodes changing state.

Alternatively, if you have a network management system that can process SNMP traps, you can write an event handler that sends SNMP traps, instead of using e-mail. Section 9.6 on page 9–18 describes how to write site-specific event handlers. You can use the `snmp_trapsnd(8)` command to send traps.

HP recommends that you specify an e-mail address for the power supply, fan failure, and high temperature events (as described in Section 9.5.2.2 on page 9–17). The following sections describe each event handler, including how to set the corresponding attribute.

9.5.2.1 rmsevent_node Event Handler

The `rmsevent_node` event handler is triggered whenever the status of a node changes (for example, from `running` to `active`). This handler performs no actions.

See Section 9.6 on page 9–18 for details of how to substitute your own actions for this event handler.

9.5.2.2 rmsevent_env Event Handler

The `rmsevent_env` event handler is triggered by power supply failures, fan failures, or temperature changes in either a node or in the HP AlphaServer SC Interconnect.

The `rmsevent_env` script sends e-mail to the users specified by the `email-module-psu`, `email-module-fan`, `email-module-tempwarn`, or `email-module-temphigh` attributes (in the `attributes` table). If the attribute has no value, the e-mail is not sent. The attribute may contain a space-separated list of mail addresses.

Table 9–8 shows the events that trigger the `rmsevent_env` handler, and the corresponding attribute names.

Table 9–8 Events that Trigger the rmsevent_env Handler

Event	Attribute
Node (or HP AlphaServer SC Interconnect) temperature changes by more than 2°C	`email-module-tempwarn`
Node (or HP AlphaServer SC Interconnect) temperature exceeds 40°C	`email-module-temphigh`
A fan fails	`email-module-fan`
A power supply fails	`email-module-psu`

When you install RMS or build a new database, these attributes do not exist. If you want the `admin` user to receive e-mail when any of these events occur, create the appropriate attribute as follows:

```
# rcontrol create attribute name=email-module-tempwarn val=admin
# rcontrol create attribute name=email-module-temphigh val=admin
# rcontrol create attribute name=email-module-fan val=admin
# rcontrol create attribute name=email-module-psu val=admin
```

If the attribute already exists, you can modify it as follows:

```
# rcontrol set attribute name=email-module-tempwarn val=admin
# rcontrol set attribute name=email-module-temphigh val=admin
# rcontrol set attribute name=email-module-fan val=admin
# rcontrol set attribute name=email-module-psu val=admin
```

See Section 9.6 on page 9–18 for details of how to substitute your own actions for this event handler.

9.5.2.3 rmsevent_escalate Event Handler

The `rmsevent_escalate` event handler is triggered if an event handler does not complete in time. The time is specified in the `timeout` field in the `event_handlers` table. If you rely on any handlers described elsewhere in this section, you should set the `users-to-mail` attribute so that you are notified in the event that one of those event handlers fails to execute correctly.

When you install RMS or build a new database, the `users-to-mail` attribute does not exist. If you want the `admin` user to receive e-mail when an event handler does not complete in time, create the attribute as follows:

```
# rcontrol create attribute name=users-to-mail val=admin
```

If the attribute already exists, you can modify it as follows:

```
# rcontrol set attribute name=users-to-mail val=admin
```

See Section 9.6 on page 9–18 for details of how to substitute your own actions for this event handler.

9.6 Event Handler Scripts

Note:

In HP AlphaServer SC Version 2.6, event handlers are only supported for events that are of class `node`, `partition`, or `switch_module` and of type `status`.

When events occur, RMS reads the `event_handlers` table to determine if an event handler for the event should be executed.

You can read the event_handlers table as follows:

```
# rmsquery -v "select * from event_handlers"
id    name    class     type           timeout   handler
-----------------------------------------------------------------
1             node      status         600       /opt/rms/etc/rmsevent_node
2                       temphigh       300       /opt/rms/etc/rmsevent_env
3                       tempwarn       300       /opt/rms/etc/rmsevent_env
4                       fan            300       /opt/rms/etc/rmsevent_env
5                       psu            300       /opt/rms/etc/rmsevent_env
6             event     escalation     -1        /opt/rms/etc/rmsevent_escalate
```

When an event occurs, it executes the script specified by the handler field. As with the pstartup script (see Section 5.10 on page 5–74), these scripts execute system and site-specific scripts.

If you want to implement your own event-handling scripts, you can do this in two ways:

- Override the existing event-handling script by creating a file called /usr/local/rms/etc/*scriptname*
- Add an entry to the event_handlers table. For example, if your script is called /mine/part_handler, you can add it to the event_handlers table as follows:

  ```
  # rmsquery "insert into event_handlers
    values(7,'','partition','status',300,'/mine/part_handler')"
  ```

  For this change to take effect, you must stop and start the eventmgr daemon, as follows:

  ```
  # rcontrol stop server=eventmgr
  # rcontrol start server=eventmgr
  ```

  Several handlers for the same event are allowed. RMS executes each of them.

  Errors that occur in event-handling scripts are written to the /var/rms/adm/log/error.log file.

10
Viewing System Status

An HP AlphaServer SC Version 2.6 system contains many different components: nodes, terminal servers, Ethernet switches, HP AlphaServer SC Interconnect switches, storage subsystems, system software, and so on. This chapter describes how to view the status of the system using SC Viewer.

The information in this chapter is arranged as follows:

- SC Viewer (see Section 10.1 on page 10–2)
- Overview Tab (see Section 10.2 on page 10–12)
- Failures Tab (see Section 10.3 on page 10–14)
- Domains Tab (see Section 10.4 on page 10–16)
- Infrastructure Tab (see Section 10.5 on page 10–19)
- Physical Tab (see Section 10.6 on page 10–25)
- Events Tab (see Section 10.7 on page 10–27)
- Interconnect Tab (see Section 10.8 on page 10–30)

Note:

Only the `root` user can perform actions in the Interconnect Tab.

10.1 SC Viewer

SC Viewer is a graphical user interface (GUI) that allows you to view the status of various components in an HP AlphaServer SC system.

The information in this section is organized as follows:

- Invoking SC Viewer (see Section 10.1.1 on page 10–2)
- SC Viewer Menus (see Section 10.1.2 on page 10–3)
- SC Viewer Icons (see Section 10.1.3 on page 10–4)
- SC Viewer Tabs (see Section 10.1.4 on page 10–9)
- Properties Pane (see Section 10.1.5 on page 10–11)

10.1.1 Invoking SC Viewer

To invoke SC Viewer, run the following command:

```
# scviewer
```

Note:

If this is the first time that SC Viewer has been run since the database was rebuilt, the Failed to determine Network Type dialog box is displayed. Click Create Network; the Network Parameters dialog box is displayed. Enter the site-specific parameters, and click OK. For more information on setting the network parameters, see Section 7.3.3 of the *HP AlphaServer SC Interconnect Installation and Diagnostics Manual*.

See the `scviewer(8)` reference page for information about the `scviewer` command.

The SC Viewer GUI is displayed, as shown in Figure 10–1.

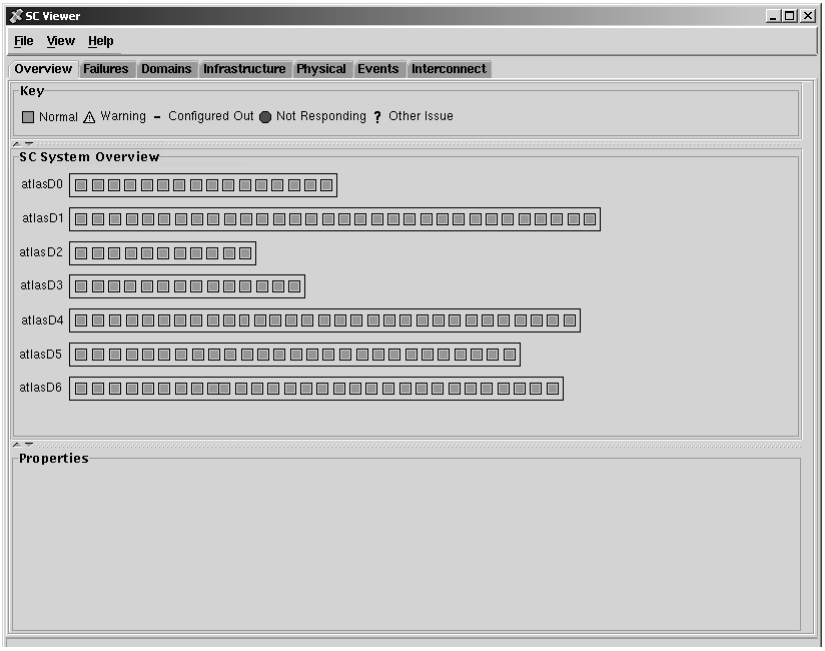

Figure 10–1 SC Viewer GUI

10.1.2 SC Viewer Menus

There are three SC Viewer menus, as shown in Figure 10–2.



Figure 10–2 SC Viewer Menus

10.1.2.1 The File Menu

The File menu contains the following options:

- The *Open* option is enabled when a Domain object is selected. This option opens a Nodes window, displaying the Node objects within the selected domain.
- The *Reload Modules...* option opens the Reload Modules... dialog box, which allows you to select the SC Viewer tabs that you would like to reload. Select the checkbox of the appropriate tab(s) and click on the OK button. By default, the checkbox of the currently displayed tab is selected. To reload all tabs, select the All Domains checkbox.

- The *Select Database...* option allows you to select a database, to view its contents.
- The *Exit* option closes all SC Viewer windows and exits from SC Viewer.

10.1.2.2 The View Menu

The View menu contains the following options:

- The *Events...* option opens the Events Filter dialog box. Enter the appropriate filter to specify what subset of events should be shown in the Events tab. The *Events...* option is enabled — hence the Events Filter dialog box can be displayed — at all times during an SC Viewer session. You can change the filter regardless of which tab is currently displayed — it is not necessary to display the Events tab first.
- The *Show In Domain* option is enabled when a Node object is selected. This option opens a Nodes window, displaying the selected Node object within its domain.
- The *Show In Cabinet* option is enabled only if the following conditions are met:
  - Cabinet data exists in the SC database.
  - A cabinet has been defined for the selected object in the SC database.
  - The selected object is not a Cabinet or a Domain.

  This option opens the Physical tab, scrolling to show the selected object's cabinet in the Room/Area pane. The cabinet's constituent objects are shown in the Cabinet Contents pane, and the selected object's properties are shown in the properties pane.

10.1.2.3 The Help Menu

The Help menu contains the following options:

- The *Help Topics* option opens a Web browser window, displaying the SC Viewer online help.

  Note:

  SC Viewer online help is not available in HP AlphaServer SC Version 2.6.

- The *About SC Viewer* option opens the About dialog box, which displays the SC Viewer version number and copyright information.

10.1.3 SC Viewer Icons

SC Viewer icons can be categorized as follows:

- Overview Status Icons (see Section 10.1.3.1 on page 10–5)
- Object Icons (see Section 10.1.3.2 on page 10–6)

- Status Icons (see Section 10.1.3.3 on page 10–6)
- Event Severity Icons (see Section 10.1.3.4 on page 10–7)

10.1.3.1 Overview Status Icons

The overview status of the node is shown on the first tab of the SC Viewer GUI.

SC Viewer overview status icons are shown in Figure 10–3.

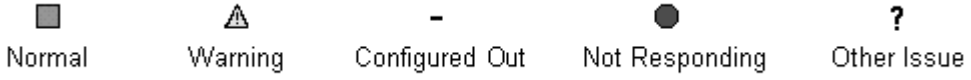

Figure 10–3 SC Viewer Overview Status Icons

The overview status of a node is a combination of the operational status and the environmental status of the node. The operational status indicates whether the node is running, configured out, or not responding. The environmental status of the node indicates whether the cooling fans and power supply units on the node are working correctly, and whether any of the other components are overheating. The operational status and the environmental status of the node are used to determine the overview status of the node as follows:

- A status of Normal indicates that the node is running and the environmental status is OK.
- A status of Warning indicates that the node is running and the environmental status is anything other than OK.
- A status of Configured Out means that the node is configured out; in this case the environmental status is not checked.
- A status of Not Responding indicates that the node is not responding, but the environmental status is OK.
- A status of Other Issue indicates any other combination of operational status and environmental status.

10.1.3.2 Object Icons

SC Viewer object icons are shown in Figure 10–4.

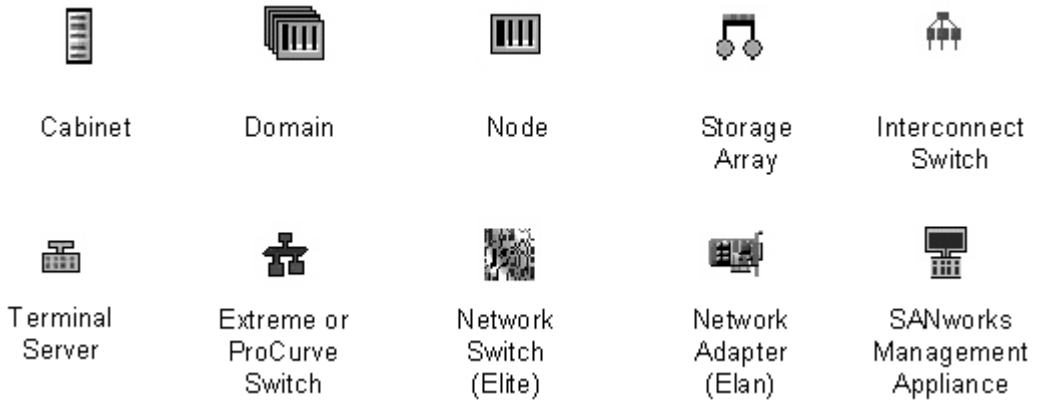

Figure 10–4 SC Viewer Object Icons

The object icons identify the different types of objects about which SC Viewer displays status information. The object icons are used in conjunction with status icons in the object panels (see Section 10.1.3.5) to display the status of various HP AlphaServer SC components.

10.1.3.3 Status Icons

SC Viewer status icons are shown in Figure 10–5.



Figure 10–5 SC Viewer Status Icons

These icons depict the **primary status** or the **contained status** of an object:

- The primary status of an object is the status of the object itself. For example, if a node is `not responding`, its primary status is Failure. If a node is `active`, its primary status is Warning. If a node is `running`, its primary status is Normal/OK.

  For some objects, the primary status concept is meaningless. For example, a cabinet cannot fail, so its primary status is always Normal/OK.

- The contained status of an object indicates the worst status of all of the monitored/reporting items that are contained within the object. For a node, these items include the CPUs, fans, power supplies, and temperatures. For a domain, these items are the nodes of

that domain, and the items within each of those nodes. For example, SC Viewer would indicate a failed fan by displaying a contained status of Failure on its node, and a contained status of Failure on the domain of which the node is a member.

The Missing status icon is used primarily in the properties pane, to indicate that the element is missing from the object; for example, a CPU is missing from the node, a disk is not installed in the HSG80 RAID system.

The status icons are used in conjunction with object icons in the object panels (see Section 10.1.3.5) to display the status of various HP AlphaServer SC components.

10.1.3.4 Event Severity Icons

SC Viewer event severity icons are shown in Figure 10–6.

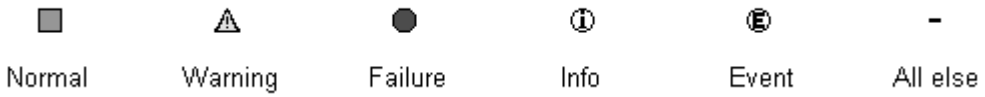

Figure 10–6 SC Viewer Event Severity Icons

These icons appear in the Severity column in the Events tab, and are equivalent to the event severities used by the `scevent` command.

10.1.3.5 Object Panels

An **object panel** shows the type, primary status, label, and contained status of an object:

- The object's type is depicted by the object icon (see Section 10.1.3.2).
- The object's primary status is depicted by a Warning or Failure icon overlaid on the lower right corner of the object's type. The Normal/OK status is not overlaid — if there is no Warning or Failure icon, the status of the object is normal.
- The object's label is depicted by the text underneath the object icon.
- The object's contained status is depicted by a Warning or Failure icon placed below the object label. If the contained status is normal, no icon is shown.

Figure 10–7 shows some example object panels.



Figure 10–7 Example Object Panels

These example object panels provide the following information:

- Domain `atlasD2` has a Failure primary status and a Failure contained status.
- Domain `atlasD10` has a Normal/OK primary status and a Failure contained status.
- Node `atlas65` has a Normal/OK primary status and a Failure contained status.
- Node `atlas69` has a Failure primary status and a Warning contained status.
- Switch `extreme1` has a Failure primary status and a Failure contained status.
- HSV110 RAID system `SCHSV08` has a Normal/OK primary status and a Warning contained status.
- HP AlphaServer SC Interconnect switch `QR0N07` has a Warning primary status and a Normal/OK contained status.

10.1.4 SC Viewer Tabs

Each SC Viewer tab has the same general layout, as shown in Figure 10–8.

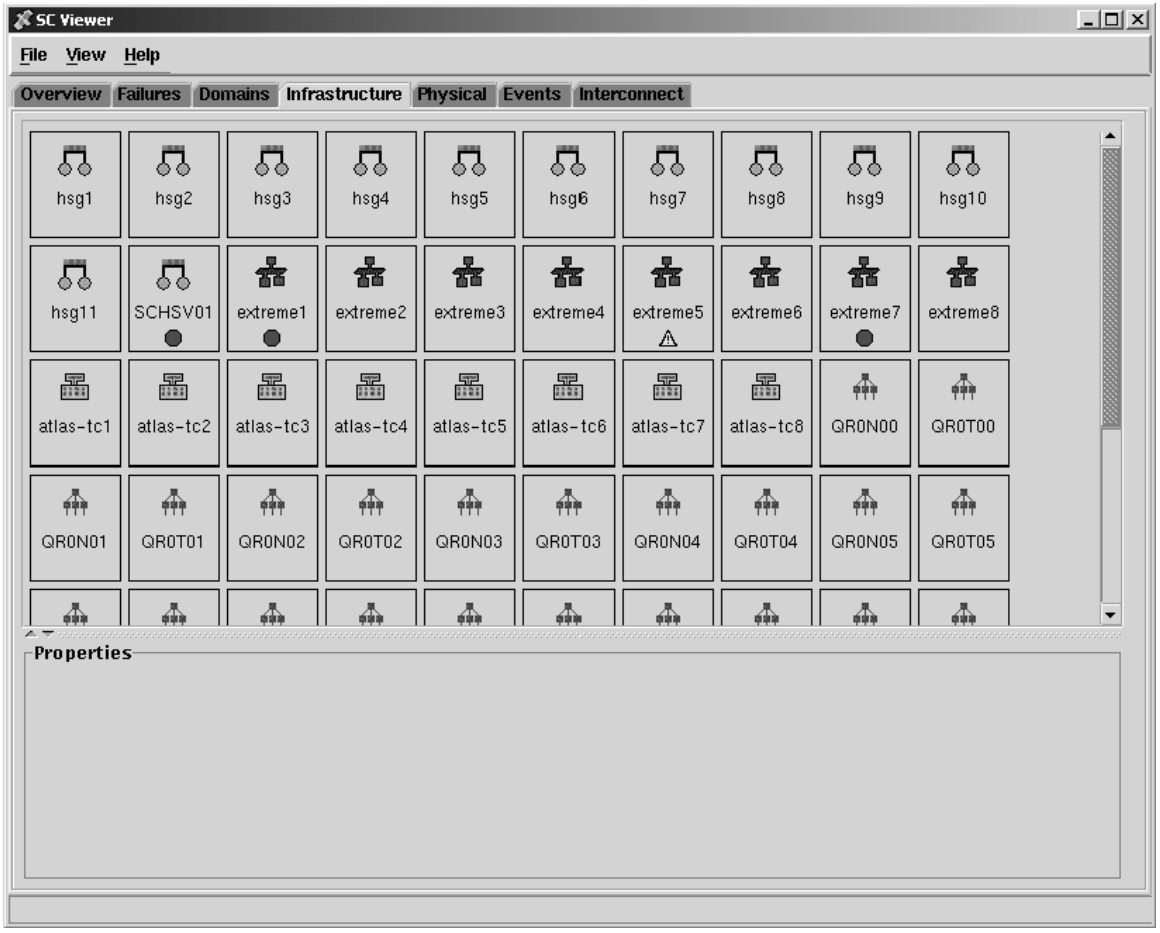

Figure 10–8 SC Viewer Tabs — General Layout

The name of the selected tab is shown with a normal background; the other tabs have a darker background. To change the view, simply click on the desired tab. The Failures tab is displayed by default when SC Viewer starts.

The information area of each tab has two panes:

- Main pane
- Properties pane

The main pane displays the object panel for each system object, as appropriate to the tab. In Figure 10–8, the main pane contains object panels for HSG80 RAID systems, HSV110 RAID systems, SANworks Management Appliances, and so on.

The properties pane displays the attributes of a selected object. To select an object, left-click on its object panel. The selected object panel is highlighted and the corresponding attributes are shown in the properties pane.

The division between the main pane and the properties pane is a splitter that can be moved up or down by the user to display more or less information in each pane. If there is more information to be displayed than will fit in a pane, horizontal and/or vertical scrollbars are displayed as needed.

The display area can also be changed by enlarging or reducing the overall SC Viewer window by dragging its borders or by clicking the Maximize icon.

Right-clicking on an object opens a pop-up menu that allows the user to view the object in a different context. For a Domain object, the choice on the pop-up menu is *Open*. If you select the *Open* option, a Nodes window appears, displaying all of the nodes in the selected domain. You can also display this window by double-clicking on the Domain object.

For a Node object, the pop-up menu choices are *Show In Domain* and *Show In Cabinet*. If you select the *Show In Domain* option, a Nodes window appears for the Domain that contains the node, with the Node selected and its properties displayed in the properties pane. For all other objects except a Cabinet, the pop-up choice is *Show In Cabinet*. If you select the *Show In Cabinet* option, SC Viewer displays the Physical tab, showing the selected object in its cabinet, and the selected object's properties in the properties pane.

Note:

The *Show In Cabinet* choice is disabled if cabinets and unit numbers have not been defined in the SC database.

The Nodes window is described in Section 10.4.1 on page 10–17.

The individual SC Viewer tabs are described in the remaining sections of this chapter.

10.1.5 Properties Pane

Each time you select an object in the main pane, SC Viewer updates the properties pane to show the properties of the object that you have selected.

All properties panes have the following common features:

- Name
  This is the name of the object.
- Primary status
  The primary status of an object is the status of the object itself. The possible primary-status values depend on the type of object, but generally contain a description of the state of the object (running, normal, not responding, and so on). A primary status of Failure usually indicates that no data can be retrieved from the object. For example, if an HSG80 RAID system is "not responding", it is not possible to retrieve the status of individual disks on the HSG80 RAID system. When this happens, the values of properties are not changed — they remain in their last known state. However, until the status of the object returns to normal, the actual state of various properties cannot be known.
- Monitor status
  SC Viewer displays the monitor status of an object when the properties being displayed are gathered by SC Monitor. SC Monitor is distributed throughout the HP AlphaServer SC system, and different nodes are responsible for gathering specific pieces of data. For example, data about an HSG80 RAID system must be gathered by a node that is directly connected to the HSG80 RAID system. If that node is not running, the monitor status is set to `stale`. The values of properties are not changed — they remain in their last known state — but the `stale` monitor status is a hint that the values might not reflect the actual current state.
- Environmental status
  The environmental status of the node indicates whether the cooling fans and power supply units on the node are working correctly, and whether any of the other components are overheating.
- Cabinet
  This identifies the cabinet in which the object is located.

The following types of properties pane are described elsewhere in this chapter:

- Nodes Window (see Section 10.4.1 on page 10–17)
- Switch (see Section 10.5.1 on page 10–20)
- Terminal Server (see Section 10.5.2 on page 10–21)
- SANworks Management Appliance (see Section 10.5.3 on page 10–22)
- HSG80 RAID System (see Section 10.5.4 on page 10–23)

- HSV110 RAID System (see Section 10.5.5 on page 10–24)

10.2 Overview Tab

The Overview tab shows an overview of the status of every node in the system. The main pane is titled SC System Overview and shows a row of nodes for each domain. The overview status of each node is indicated. The key for the status icons is shown in the top pane. The overview status icons are described in Section 10.1.3.1 on page 10–5.

The properties pane displays the attributes of a domain selected in the main pane. Figure 10–9 shows an example Overview tab.

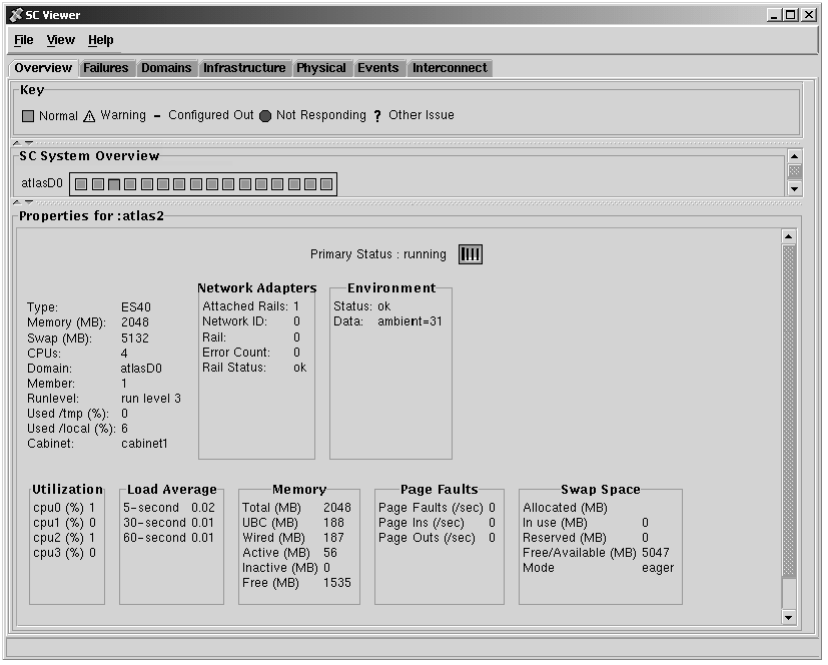

Figure 10–9 Example Overview Tab

If you select a node in the SC System Overview pane, the properties for the node are shown in the properties pane.

Table 10–1 describes the information displayed in the properties pane on the Overview tab. See also Section 10.1.5 on page 10–11.

Table 10–1 Overview Tab Properties Pane

Property	Description
`Primary Status`	Node status as shown by the `rinfo -n` command. See Section 5.8 on page 5–61 for more details about node status.
`Type`	Node type.
`Memory`	Size of memory (MB).
`Swap`	Size of allocated swap space (MB).
`CPUs`	Number of CPUs.
`Domain`	Domain name (if node is a member of a domain).
`Member`	Member number (if node is a member of a domain).
`Runlevel`	Run level of node (usually blank).
`Used /tmp`	Percentage of `/tmp` that is in use.
`Used /local`	Percentage of local disk that is in use. If `/local` and `/local1` are both present, this is the highest of the two values.
`Cabinet`	Cabinet number of the cabinet in which the node is located.
`Network Adapters`	Number of HP AlphaServer SC Elan adapter cards in the node, and the status of each attached rail.
`Environmental Status`	Indicates whether the cooling fans and power supply units on the node are working correctly, and whether any of the other components are overheating.
`Utilization`	Percentage utilization of each CPU in the node.
`Load Average`	Average run-queue lengths for the last 5, 30, and 60 seconds.
`Memory`	Physical memory usage (MB).
`Page Faults`	Page fault rate. This is averaged over a very long period, so it is normal for this value to be very low.
`Swap Space`	Swap usage on the node.

10.3 Failures Tab

The Failures tab shows all system objects whose primary or contained status is Failure or Warning. Figure 10–10 shows an example Failures tab.

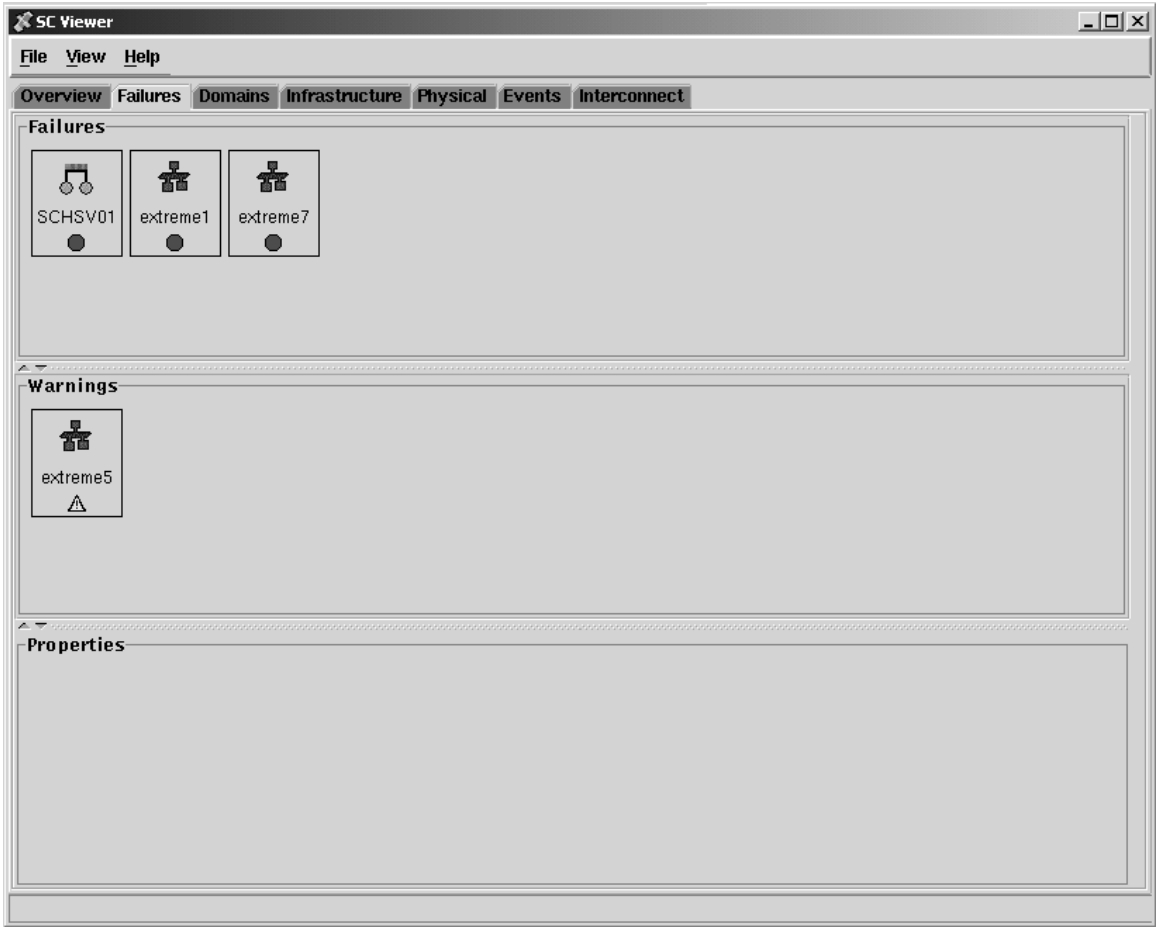

Figure 10–10 Example Failures Tab

In this example, the main pane is divided into two portions: one for Failures and one for Warnings. The subpanes are separated by a splitter that can be moved to increase and decrease the display area of each.

Failure and Warning objects are placed in the subpane of whichever is worst — their primary status or their contained status. For example, Extreme Switch `extreme1` has a Normal primary status and Failure contained status, so it is placed in the Failures sub-pane.

The contents of the Failures tab is expected to be somewhat dynamic — as the primary status and contained status of various objects change, objects will be added to and removed from the display as appropriate.

Figure 10–11 shows a Failure tab with an object selected. This shows the properties pane for Extreme Switch `extreme1`. Note that the overall window size, and thus the properties pane size, can be enlarged by dragging the bottom border.

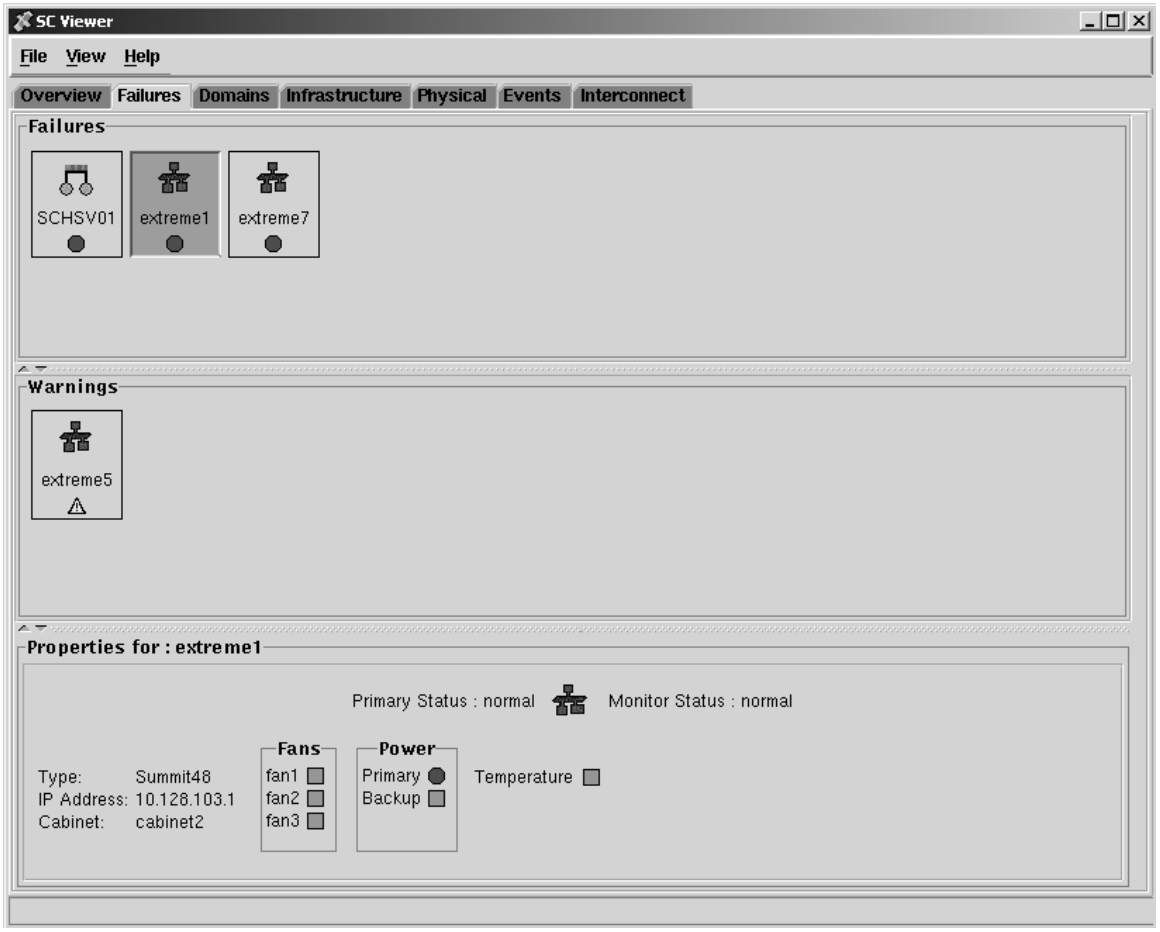

Figure 10–11 Example Failures Tab with Object Selected

10.4 Domains Tab

The Domains tab shows all of the domains in the HP AlphaServer SC system. Figure 10–12 shows an example Domains tab.



Figure 10–12 Example Domains Tab

The Domains tab contains an object panel for each Domain. Each object panel shows the following information for that domain:

- Name
- Primary status
- Contained status

If you select a Domain in the main pane, SC Viewer displays its constituent Nodes in the properties pane, as shown in Figure 10–13.

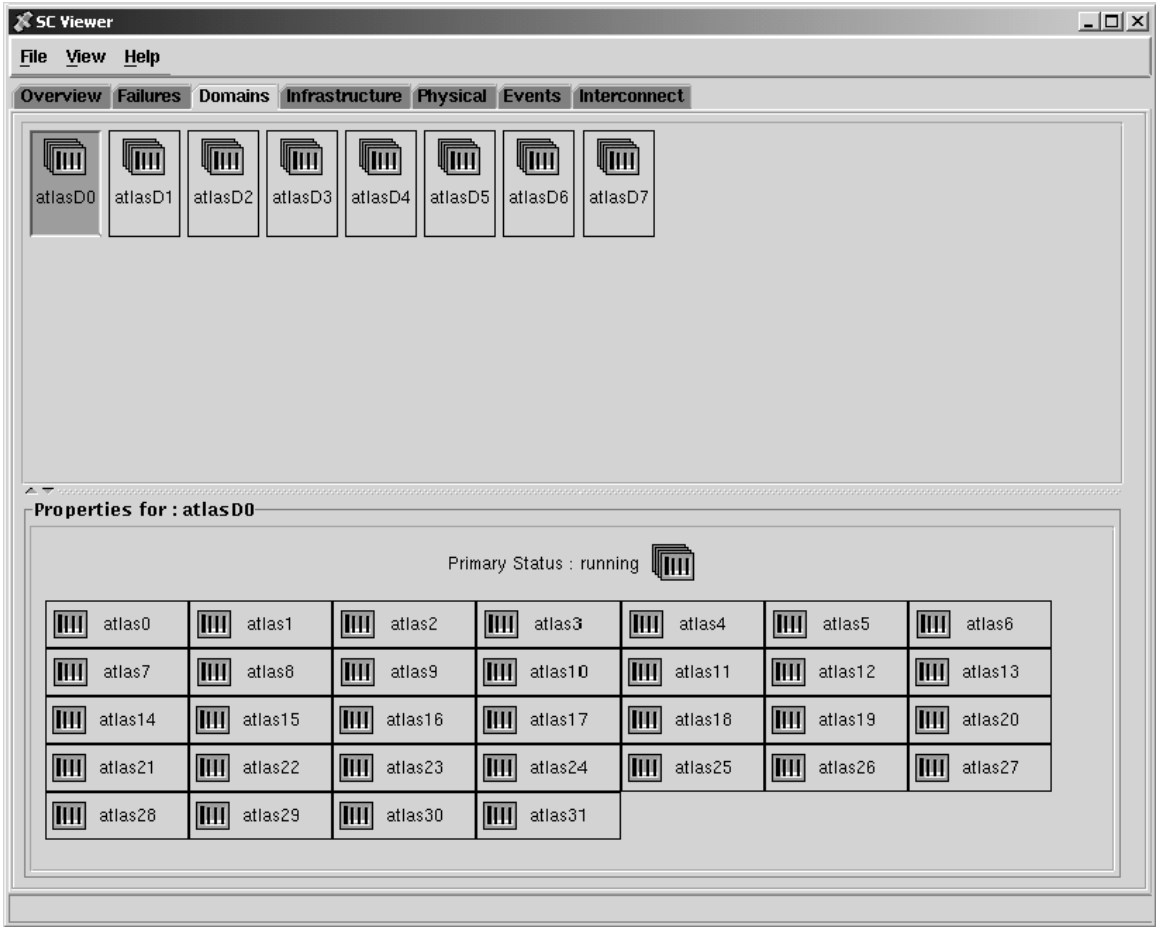

Figure 10–13 Example Domains Tab with Domain Selected

To display the properties for a specific Node, open the Nodes window (see Section 10.4.1).

10.4.1 Nodes Window

The Nodes window shows all of the Nodes in a given Domain. You can open the Nodes window in any of the following ways:

- Double-click on the appropriate domain object panel in the Domains tab.
- Select the appropriate domain object panel in the Domains tab, and choose *Open* from the File menu.

- Select any node's object panel in the Failures tab, and choose *Show In Domain* from the View menu.

The Nodes window is similar to the tabs in both appearance and functionality — it has a main pane that displays an object panel for each of the nodes in the domain, and a properties pane that shows detailed information for a selected Node.

The properties shown for a node are sourced from the RMS system. The data is only valid while the node status is `running`.

Figure 10–14 shows an example Nodes window.

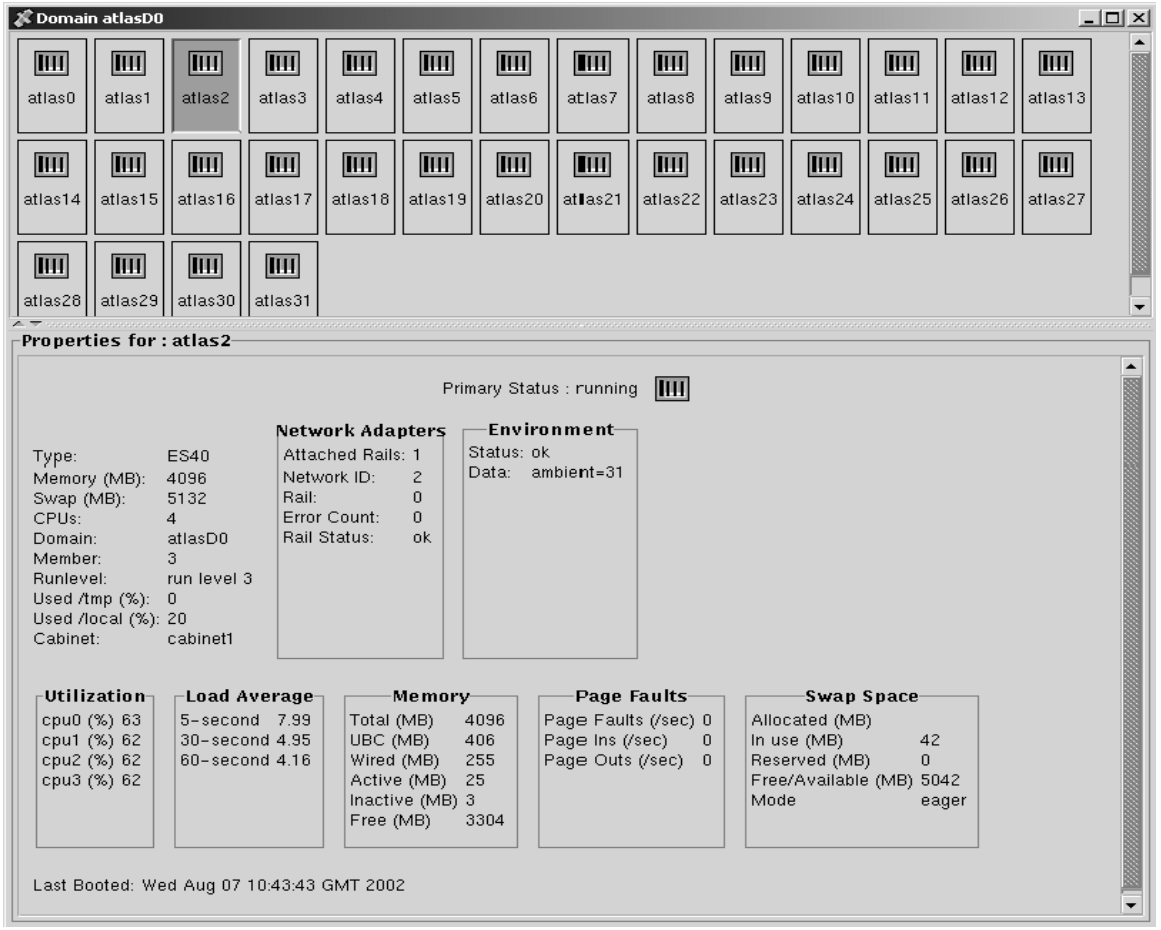

Figure 10–14 Example Nodes Window for an HP AlphaServer ES40

See Table 10–1 on page 10–13 for a description of the information displayed in the properties pane in the Nodes window. See also Section 10.1.5 on page 10–11.

10.5 Infrastructure Tab

The Infrastructure tab shows all of the HSG80 RAID systems, HSV110 RAID systems, SANworks Management Appliances, HP AlphaServer SC Interconnect switches, terminal servers, Extreme and HP ProCurve switches, and network switches (Elites) in the system, and indicates the primary status and contained status of each object.

Figure 10–15 shows an example Infrastructure tab.

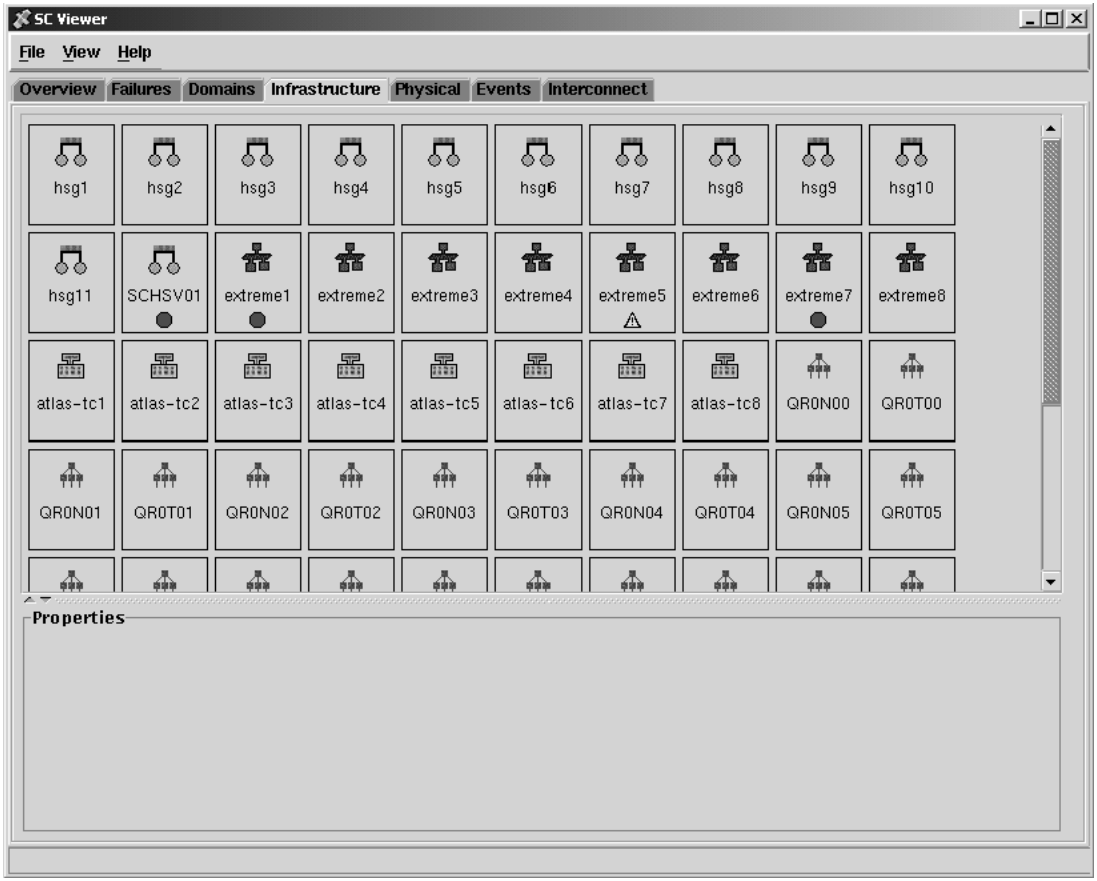

Figure 10–15 Example Infrastructure Tab

Figure 10–16 to Figure 10–20 inclusive show example properties panes for several different objects. For more information about the properties displayed for these objects, see Table 27–1 on page 27–2.

10.5.1 Switch

Figure 10–16 shows an example properties pane for a switch. The switch shown in the example is an Extreme switch; the properties would be similar for an HP ProCurve switch.

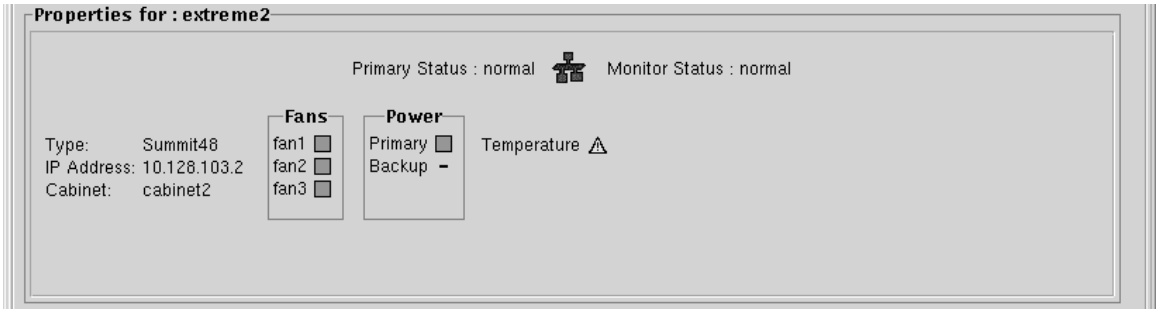

Figure 10–16 Example Properties Pane for a Switch

The properties shown for a switch are sourced from SC Monitor.

Table 10–2 describes the information displayed in the properties pane for a switch. See also Section 10.1.5 on page 10–11.

Table 10–2 Switch Properties Pane

Property	Description

`Primary Status`
`Fans`
`Power`
`Temperature`
	See Table 27–1 on page 27–2 for a detailed description of these properties. This data is valid only if the Primary Status and Monitor Status are `normal`.

`Monitor Status`	SC Monitor monitor status. See Chapter 27 for more information.
`Type`	Type of switch.
`IP Address`	IP address of the switch.
`Cabinet`	Cabinet number of the cabinet in which the switch is located.

10.5.2 Terminal Server

Figure 10–17 shows an example properties pane for a terminal server.

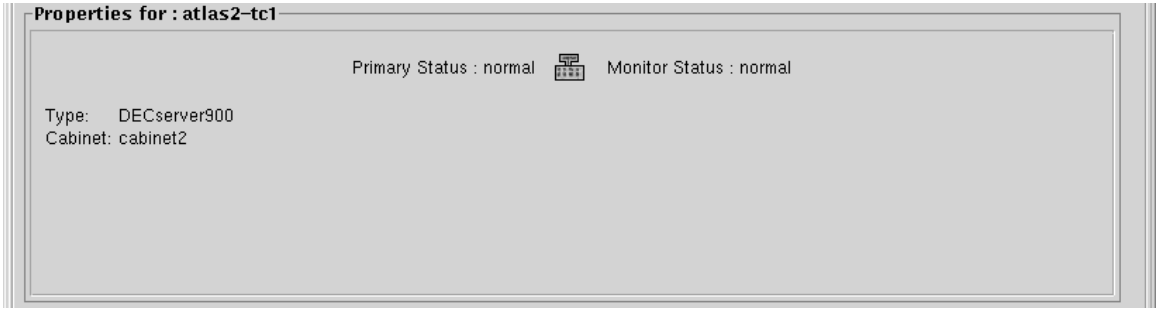

Figure 10–17 Example Properties Pane for a Terminal Server

The properties shown for a terminal server are sourced from SC Monitor.

Table 10–3 describes the information displayed in the properties pane for a terminal server. See also Section 10.1.5 on page 10–11.

Table 10–3 Terminal Server Properties Pane

Property	Description
`Primary Status`	See Table 27–1 on page 27–2 for a detailed description of this property. This data is valid only if the Monitor Status is `normal`.
`Monitor Status`	SC Monitor monitor status. See Chapter 27 for more information.
`Type`	Type of terminal server.
`Cabinet`	Cabinet number of the cabinet in which the terminal server is located.

10.5.3 SANworks Management Appliance

Figure 10–18 shows an example properties pane for a SANworks Management Appliance.

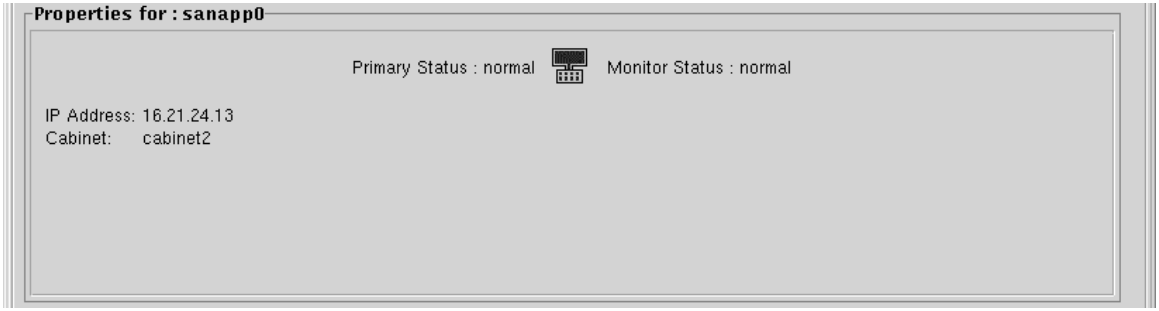

Figure 10–18 Example Properties Pane for a SANworks Management Appliance

The properties shown for a SANworks Management Appliance are sourced from SC Monitor.

Table 10–4 describes the information displayed in the properties pane for a SANworks Management Appliance. See also Section 10.1.5 on page 10–11.

Table 10–4 SANworks Management Appliance Properties Pane

Property	Description
`Primary Status`	See Table 27–1 on page 27–2 for a detailed description of this property. This data is valid only if the Monitor Status is `normal`.
`Monitor Status`	SC Monitor monitor status. See Chapter 27 for more information.
`IP Address`	IP address of the SANworks Management Appliance.
`Cabinet`	Cabinet number of the cabinet in which the SANworks Management Appliance is located.

10.5.4 HSG80 RAID System

The properties pane for an HSG80 RAID system displays different information than that displayed by the properties pane for an HSV110 RAID system, because of the differences in design and capability between these two types of RAID system.

Figure 10–19 shows an example properties pane for an HSG80 RAID system.

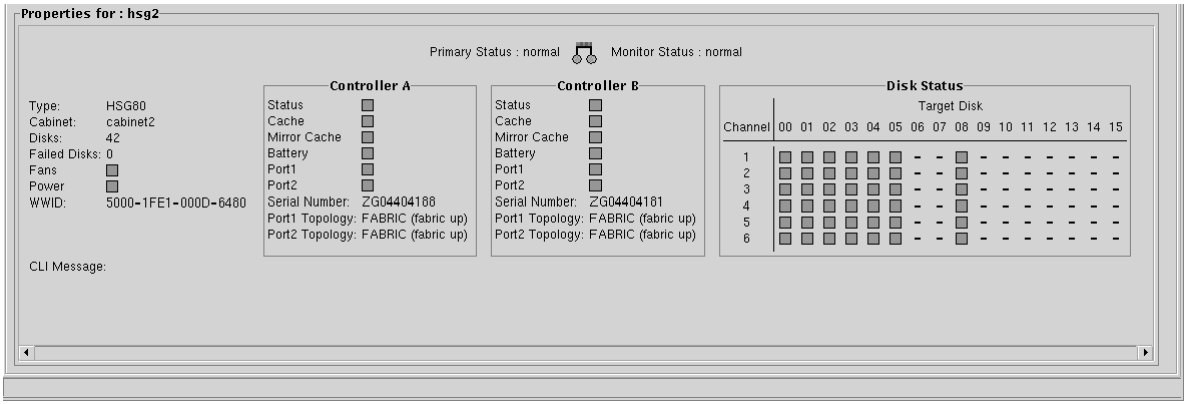

Figure 10–19 Example Properties Pane for an HSG80 RAID System

The properties shown for an HSG80 RAID system are sourced from SC Monitor.

Table 10–5 describes the information displayed in the properties pane for an HSG80 RAID system. See also Section 10.1.5 on page 10–11.

Table 10–5 HSG80 RAID System Properties Pane

Property	Description
`Primary Status`	See Table 27–1 on page 27–2 for a detailed description of this property. This data is valid only if the Monitor Status is `normal`.
`Monitor Status`	SC Monitor monitor status. See Chapter 27 for more information.
`Type`	Type of RAID system.
`Cabinet`	Cabinet number of the cabinet in which the HSG80 RAID system is located.
Other Properties	See Table 27–1 on page 27–2 for a detailed description of these properties. This data is valid only if the Primary Status and Monitor Status are `normal`.

10.5.5 HSV110 RAID System

Figure 10–20 shows an example properties pane for an HSV110 RAID system.

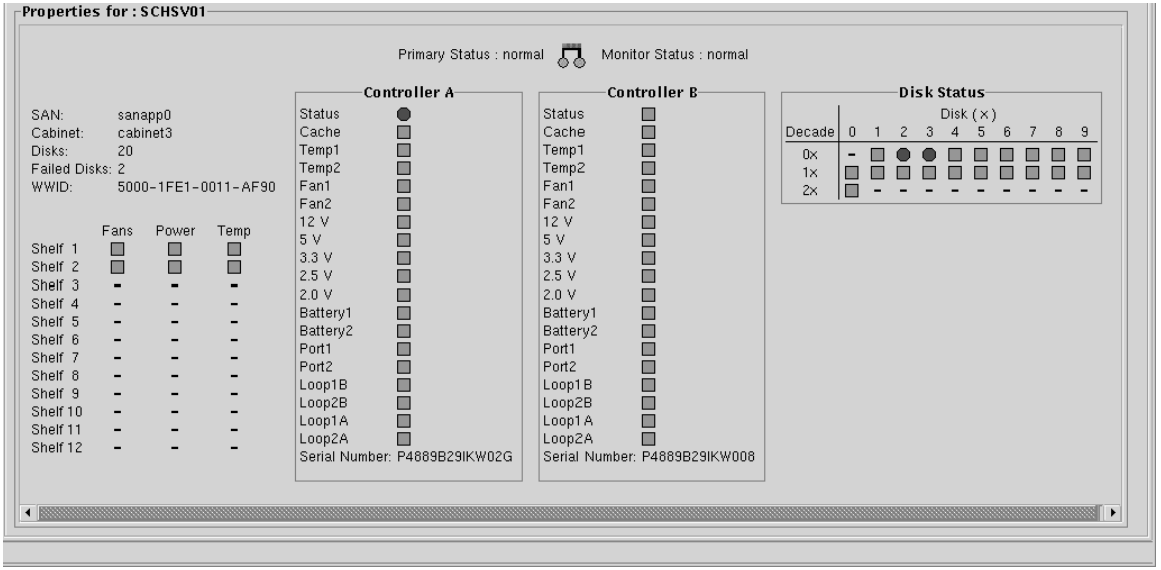

Figure 10–20 Example Properties Pane for an HSV110 RAID System

The properties shown for an HSV110 RAID system are sourced from SC Monitor.

Table 10–6 describes the information displayed in the properties pane for an HSV110 RAID system. See also Section 10.1.5 on page 10–11.

Table 10–6 HSV110 RAID System Properties Pane

Property	Description
`Primary Status`	See Table 27–1 on page 27–2 for a detailed description of this property. This data is valid only if the Monitor Status is `normal`.
`Monitor Status`	SC Monitor monitor status. See Chapter 27 for more information.
`SAN`	SANworks Management Appliance that manages this RAID system.
`Cabinet`	Cabinet number of the cabinet in which the HSV110 RAID system is located.
Other Properties	See Table 27–1 on page 27–2 for a detailed description of these properties. This data is valid only if the Primary Status and Monitor Status are `normal`.

10.6 Physical Tab

If you have updated the SC database with information about the Cabinets in which each object is located, and each object's unit number within that Cabinet, the Physical tab will depict the physical layout of the Cabinets and their contents.

Cabinets may be distributed across multiple rooms or areas — the Physical tab shows each room or area. Within a room or area, the Cabinet objects are positioned according to their specified row and column data.

If you have not updated the SC database with this information, the Physical tab displays a message indicating that the requisite data is not in the database.

Figure 10–21 shows an example Physical tab depicting the Cabinets.

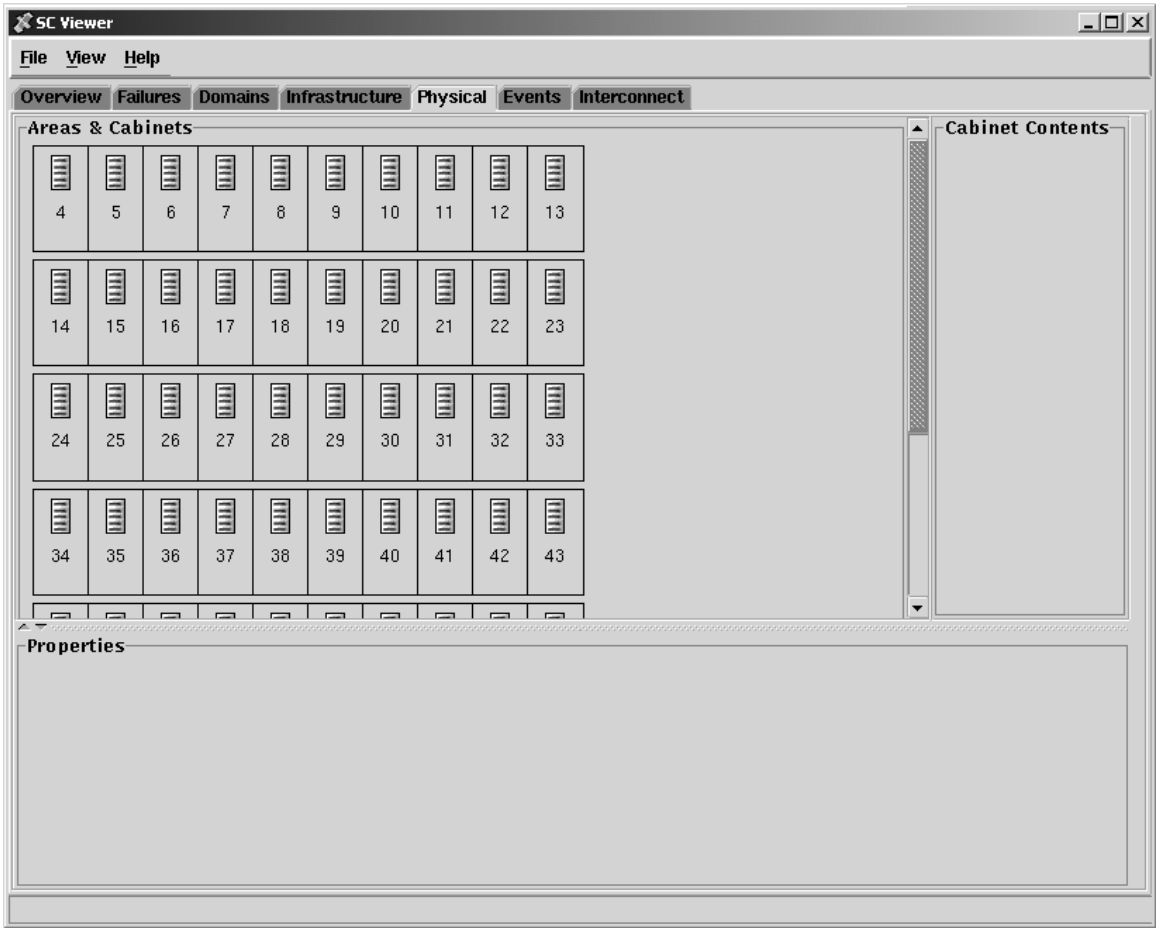

Figure 10–21 Example Physical Tab

If you select a Cabinet, SC Viewer displays detailed information about the Cabinet in the properties pane, and the constituent objects of the Cabinet in the Cabinet Contents subpane. The objects are ordered by unit number, with unit 0 at the bottom of the Cabinet Contents subpane.

Figure 10–22 shows the Physical tab displayed by SC Viewer when Cabinet `4` is selected.

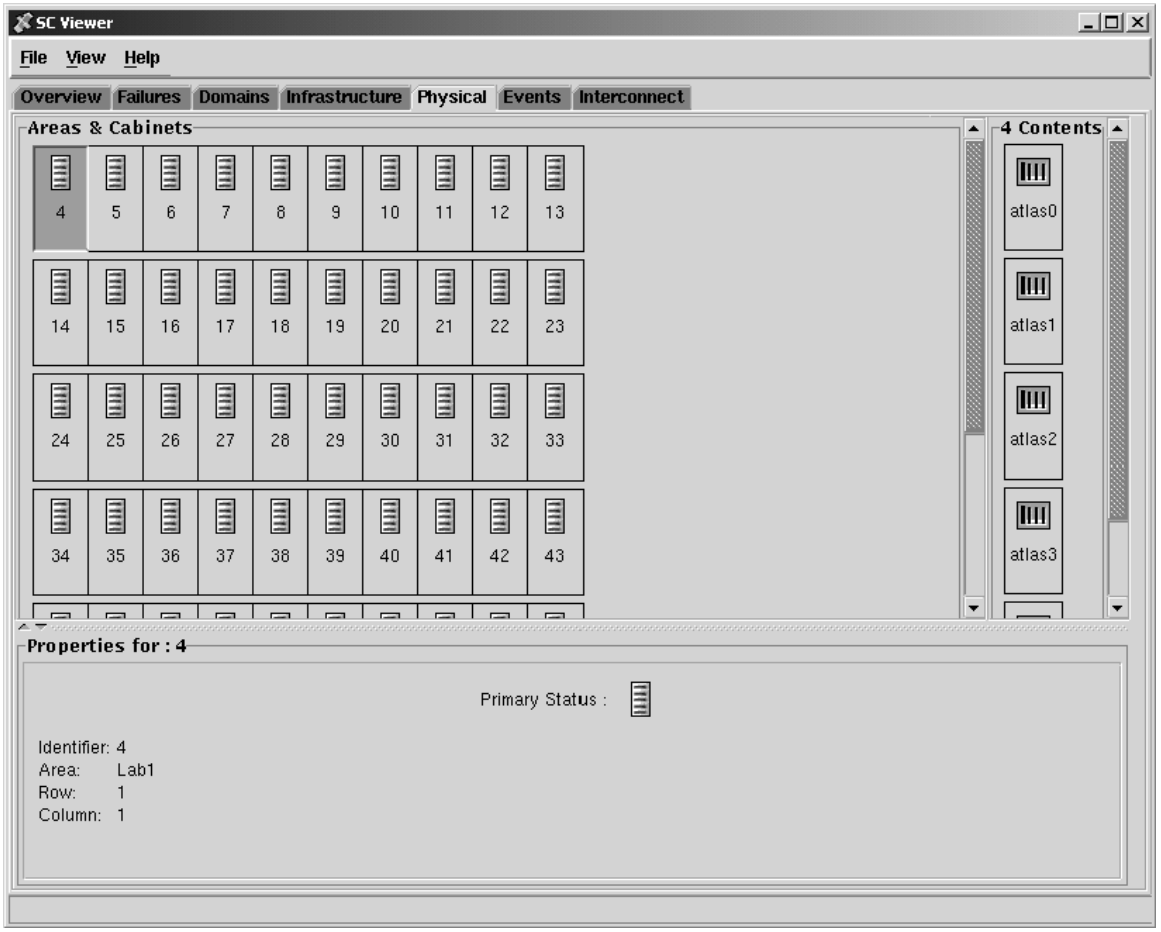

Figure 10–22 Example Physical Tab with Cabinet Selected

If you select an object in the Cabinet 4 Contents subpane, SC Viewer displays detailed information about that object in the properties pane, as shown in Figure 10–23.

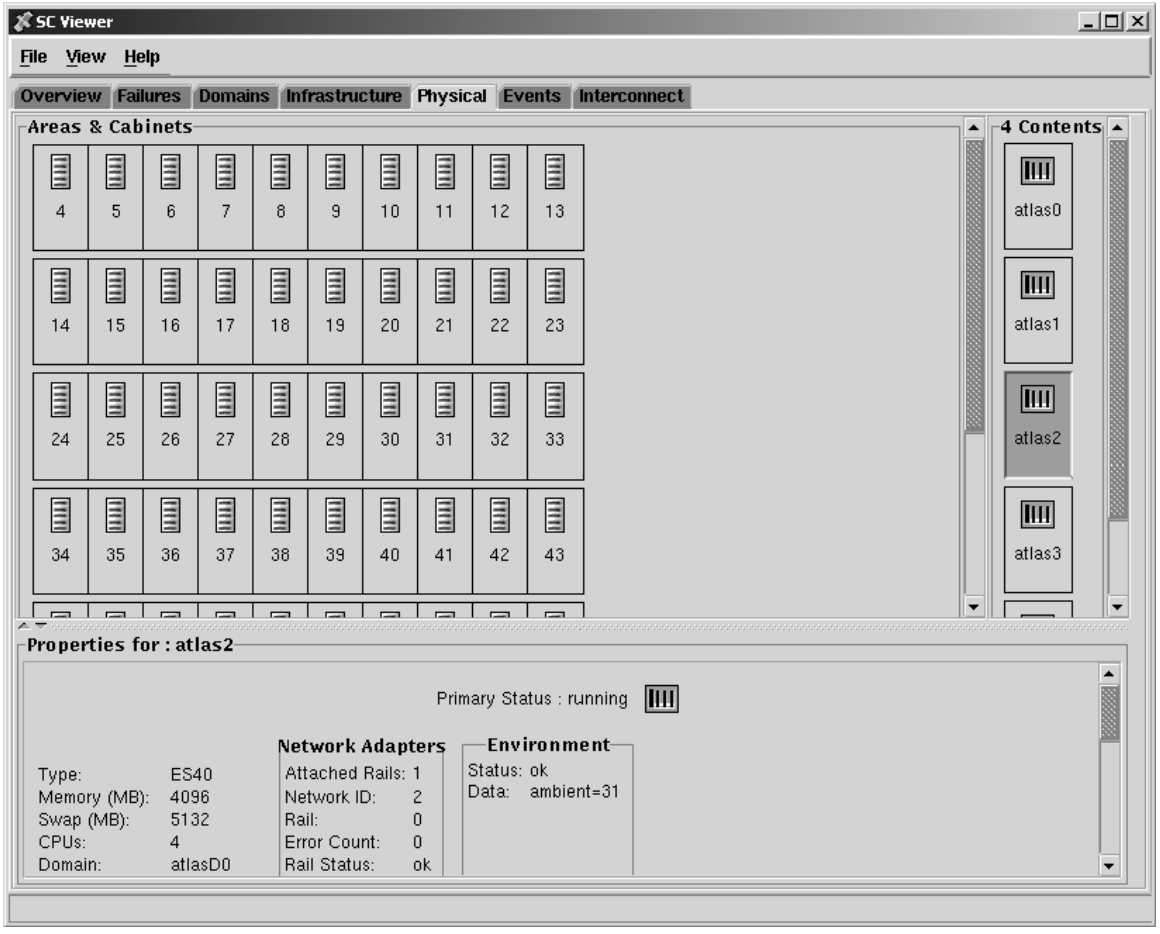

Figure 10–23 Example Physical Tab with Node Selected Within Cabinet

10.7 Events Tab

The Events tab differs from the other tabs in that it primarily presents textual information, rather than pictorial representations. The data shown is comprised of those system events that satisfy the event filter (see Section 9.2 on page 9–6 for more information about event filters).

Figure 10–24 shows an example Events tab.

Events Tab

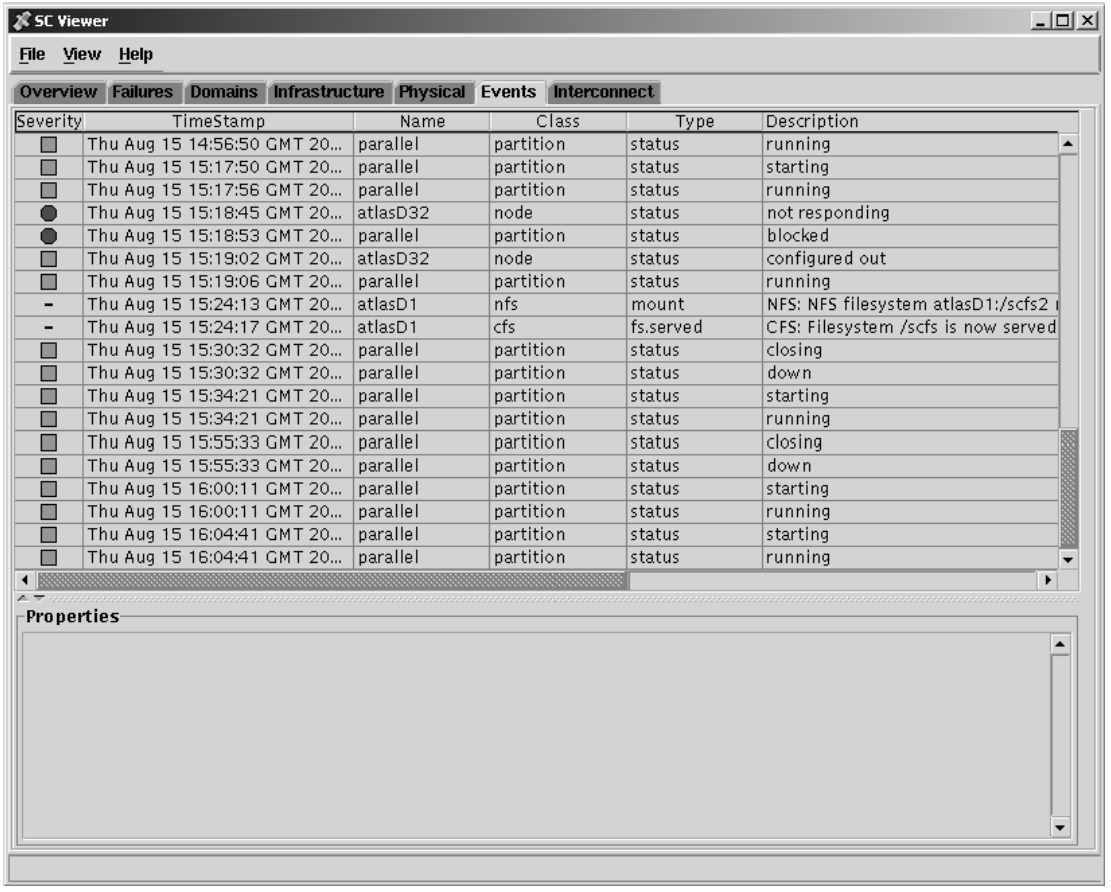

Figure 10–24 Example Events Tab

When SC Viewer is invoked, the default event filter is `[age < 1d]`, which displays all events that have occurred within the past day. To change the event filter, select *Event...* from the View menu. This displays the Event Filter dialog box, as shown in Figure 10–25.

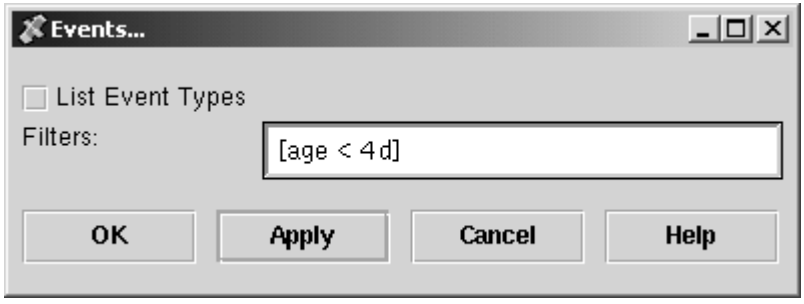

Figure 10–25 Event Filter Dialog Box

The Events Filter dialog box is available at all times during an SC Viewer session. You can change the filter at any time, regardless of which tab is currently displayed — it is not necessary to display the Events tab first. To change the Event Filter, edit the `Filters:` textbox. The filter syntax is the same as that used by the `scevent` command (see Section 9.2 on page 9–6), except that the enclosing single quotes are not needed in SC Viewer. Selecting the List Events checkbox is equivalent to running the `scevent -l` command.

As new events occur, events that satisfy the current filter are added at the bottom of the table.

If you select an event, SC Viewer displays detailed information about the event in the properties pane, as shown in Figure 10–26.

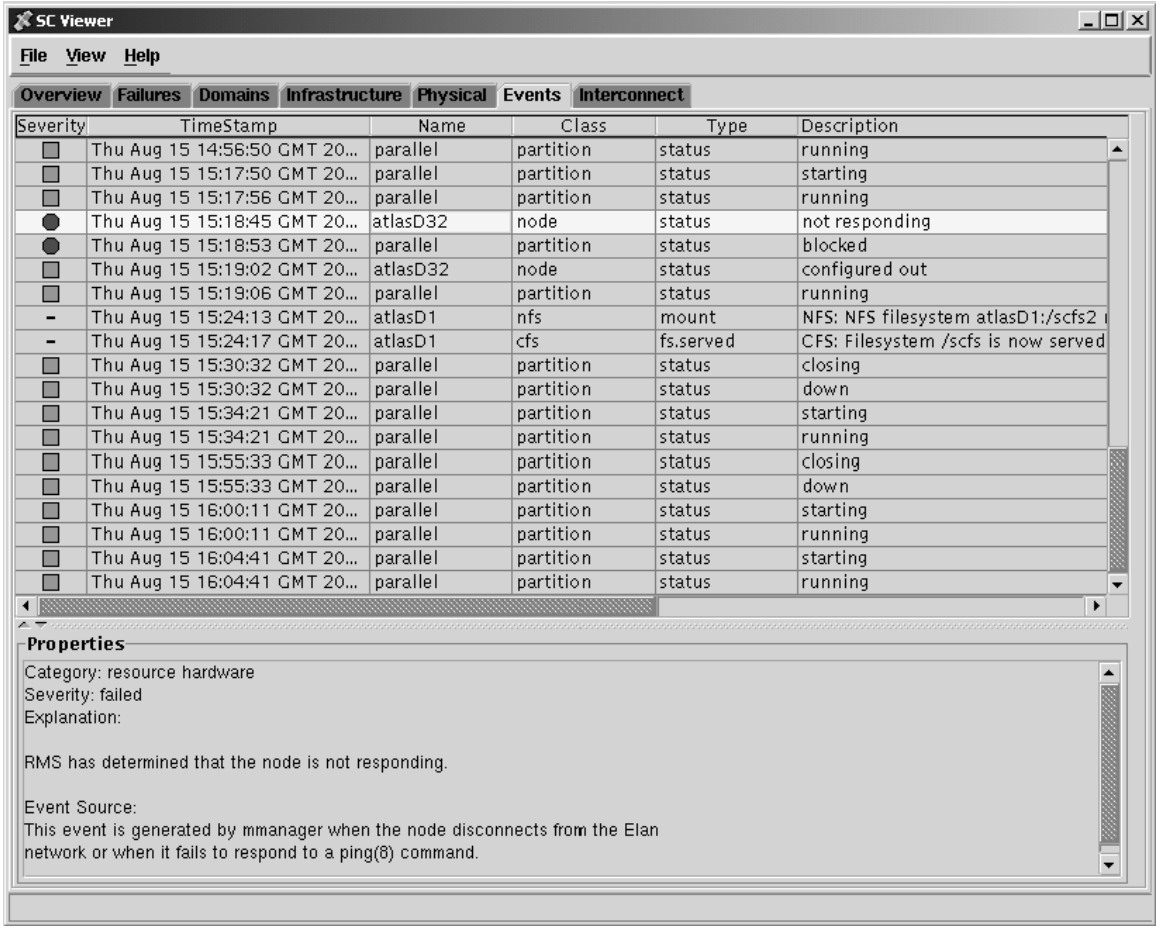

Figure 10–26 Example Event Tab with Event Selected

10.8 Interconnect Tab

An example Interconnect tab is shown in Figure 10–27.

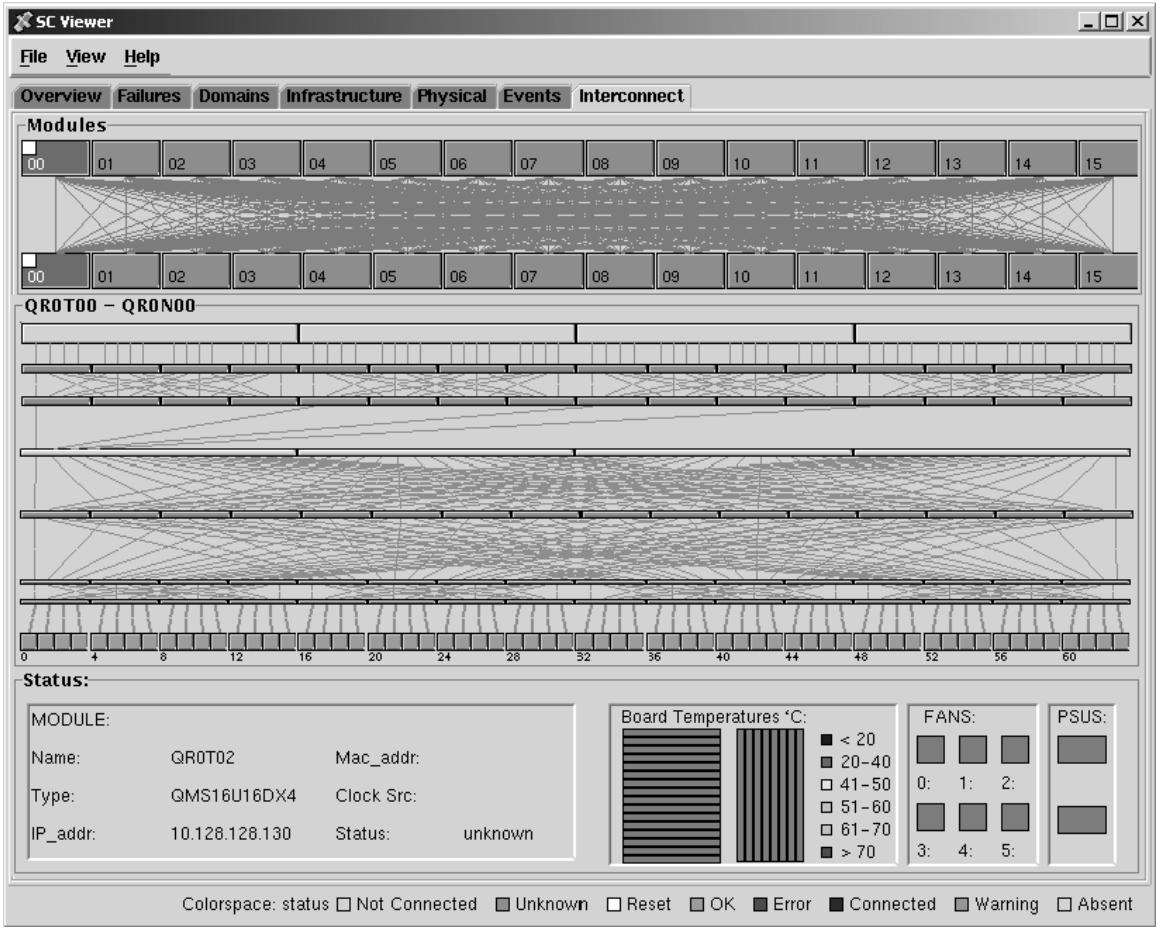

Figure 10–27 Example Interconnect Tab

Note:

The Interconnect tab differs from the other SC Viewer tabs, which show data from the live system and are periodically refreshed by either RMS or SC Monitor. The data shown in the Interconnect tab is not periodically refreshed — instead, the properties shown reflect the results from the last run of the diagnostic programs.

11
SC Performance Visualizer

SC Performance Visualizer (`scpvis`) provides a GUI for monitoring an HP AlphaServer SC system.

Using SC Performance Visualizer, you can view aspects of system performance (such as CPU utilization, memory usage, and page management statistics) at specifiable intervals.

You can specify where the SC Performance Visualizer window should be displayed by setting the DISPLAY environment variable, or by using the `-display` *`displayname`* option with the `scpvis` command. If you set the DISPLAY environment variable and then run `scpvis` with the `-display` *`displayname`* option, the value specified by the `-display` option will overwrite the value specified by the DISPLAY variable.

The `scload` command displays similar information to that displayed by the `scpvis` command, but in a command line interface (CLI) format instead of a GUI format.

The information in this chapter is organized as follows:

- Using SC Performance Visualizer (see Section 11.1 on page 11–2)
- Personal Preferences (see Section 11.2 on page 11–2)
- Online Help (see Section 11.3 on page 11–2)
- The scload Command (see Section 11.4 on page 11–3)

See the `scpvis(1)` and `pvis(1)` reference pages for more information about the `scpvis` command.

11.1 Using SC Performance Visualizer

To use SC Performance Visualizer, perform the following steps:

1. Run the `scpvis` command, as follows:
   ```
   # scpvis
   ```
   The SC Performance Visualizer window appears.
2. Select the system, a domain, or a node.
   SC Performance Visualizer shows an icon for the system and an icon for each domain in the system. When you select one of the domain icons, SC Performance Visualizer shows an icon for each node in the domain.
3. Display performance data for the item selected in step 2.
   From the *View* menu, choose the data that you wish to view for the currently selected object. Alternatively, use MB3 (the third mouse button) to display a pop-up menu containing the same options.

11.2 Personal Preferences

Using the *Options* > *Refresh Intervals* menu, you may set the rate at which the displays will be updated. This does not affect the rate at which the data is gathered. However, using a high update rate will place a load on the `msql3d` daemon. A high update rate will also place a high load on the host running the `scpvis` command.

Using the *Options* > *Preferences* menu, you may set the size of the dialog boxes. You may also set other options, such as whether you wish to use context hints.

SC Performance Visualizer stores information on user preferences, and the nodes that are being monitored, in a configuration file `$HOME/.pvis`. This file may be deleted, but should not be edited.

11.3 Online Help

To run online help, the version of Netscape that is shipped with HP AlphaServer SC Version 2.6 must be running on the user's desktop.

If Netscape is not in the user's PATH, enter the path name of the Netscape executable in the SC Performance Visualizer *Options* > *Preferences* dialog box.

11.4 The scload Command

The `scload` command displays similar information to that displayed by the `scpvis` command, but in a CLI format instead of a GUI format.

Depending on the options specified, the `scload` command displays several different types of information.

The syntax of the `scload` command is as follows:

- `scload [-m metric]`
- `scload [-b|-j|-r] number [-m metric]`
- `scload -d domain|all [-m metric]`
- `scload -p partition [-m metric]`
- `scload -r none [-m metric]`
- `scload -h`

See the `scload(1)` reference page for information about the `scload` command.

11.4.1 scload Options

The `scload` options are described in Table 11–1.

Table 11–1 scload Options

Option	Description
<no option specified>	Displays information about all nodes.
`-b number`	Displays information about the nodes running the specified batch job.
`-d domain|all`	Displays information about the nodes in the specified domain (see Section 11.4.3.3). The domain can be specified by using a number or a name; for example, `2` and `atlasD2` each refer to the third domain in the `atlas` system. If the keyword `all` is specified instead of a domain name or number, the `scload` command displays information about each domain.
`-h`	Displays usage information for the `scload` command.
`-j number`	Displays information about the nodes running the specified job.
`-m metric`	Specifies the metric to be displayed, as described in Table 11–2. If no metric is specified, the `cpu` metric is displayed.
`-p partition`	Displays information about the nodes in the specified partition.
`-r number`	Displays information about the nodes in the specified resource (see Section 11.4.3). This is the default option, if a number is specified without any flag.

Table 11–1 scload Options

Option	Description
`-r none`	Displays information about all nodes that have not been allocated, allowing you to verify that only allocated nodes are busy at a given time (see Example 11–5).

11.4.2 scload Metrics

The `scload` metrics are described in Table 11–2.

Table 11–2 scload Metrics

Metric	Description
`allocation`	Show the processor allocations. This metric only applies to the `-b`, `-j`, and `-r` options.
`cpu`	Show the sum of system CPU usage and user CPU usage. When multiple processors are involved, this is the sum over all of the CPUs: on a 4-CPU system, the value ranges from 0% to 400%. When `scload` produces domain-level statistics, the sum-over-processors value is averaged for the nodes in each domain.
`freemem`	Show the percentage of free memory. When `scload` produces domain-level statistics, the free-memory value is averaged for the nodes in each domain.
`rq5`	Show the five-second run-queue length. When `scload` produces domain-level statistics, the sum of the run-queue lengths for the nodes in each domain is displayed — this value is not averaged for the nodes in each domain.
`system`	Show the system CPU usage. When multiple processors are involved, this is the sum over all of the CPUs: on a 4-CPU system, the value ranges from 0% to 400%. When `scload` produces domain-level statistics, the sum-over-processors value is averaged for the nodes in each domain.
`user`	Show the user CPU usage. When multiple processors are involved, this is the sum over all of the CPUs: on a 4-CPU system, the value ranges from 0% to 400%. When `scload` produces domain-level statistics, the sum-over-processors value is averaged for the nodes in each domain.

11.4.3 Example scload Output

This section provides the following examples:

- Resource Output (see Section 11.4.3.1 on page 11–5)
- Overlapping Resource Output (see Section 11.4.3.2 on page 11–7)
- Domain-Level Output (see Section 11.4.3.3 on page 11–7)

11.4.3.1 Resource Output

```
# rinfo -rl
RESOURCE         CPUS     STATUS           TIME     USERNAME         NODES
parallel.1718      3      allocated        00:12        fred         atlas[0,12-13]
```

The `rinfo` command indicates that resource 1718 exists, consisting of 1 CPU from each of `atlas0`, `atlas12`, and `atlas13`. In these examples, `atlas` is a 16-node system.

Example 11–1 CPU Utilization

To examine the CPU utilization of the nodes assigned to this resource, run the following `scload` command:

```
# scload 1718
CPU Utilisation (system+user) (%) for resource 1718
390-400:
310-390:
290-310:
210-290:
190-210:  [12-13]
110-190:
 90-110:
 10- 90:
  0- 10:     0
```

This output indicates that `atlas[12-13]` are quite busy, whereas `atlas0` is idle.

Example 11–2 Free Memory

To examine the memory of the nodes allocated to the resource, run the following `scload` command:

```
# scload 1718 -m freemem
Free Memory (%) for resource 1718
 90-100:
 80- 90:   0
 70- 80:
 60- 70:
 50- 60:
 40- 50:
 30- 40:   [12-13]
 20- 30:
 10- 20:
  0-  10:
```

This output indicates that much of the memory of atlas[12-13] is in use, whereas atlas0 enjoys a larger proportion of free memory.

Example 11–3 Allocated CPUs

To check how many CPUs per node are allocated to the resource, run the following scload command:

```
# scload 1718 -m allocation
CPU Allocation for resource 1718
1 CPUs:  atlas[0,12-13]
```

This output indicates that one CPU is allocated to resource 1718 from each of the three nodes: atlas0, atlas12, and atlas13.

Example 11–4 Run-Queue Length

To check the run-queue length per node, run the following scload command:

```
# scload 1718 -m rq5
Run Queue Lengths for resource 1718
atlas0 :   0.09
atlas12:   3.51
atlas13:   3.29
```

This output indicates that atlas12 and atlas13 are experiencing a heavy load.

Example 11–5 Unallocated Nodes

Nodes that have not been allocated should have a very low CPU utilization. To verify this, run the following scload command:

```
# scload -r none
CPU Utilisation (system+user) (%) for nodes that have not been allocated
390-400:
310-390:
290-310:
210-290:
190-210:
110-190:
 90-110:
 10- 90:
  0- 10:   [14-15]

No data:   [1-11]
```

This output indicates that atlas14 and atlas15 have not been allocated, and have a very low CPU utilization, as might be expected. Other nodes (atlas[1-11]) have not been allocated, but no valid performance data is available for these nodes; either the data is not present in the node_stats table, or the data is stale. This can happen if rmsd on these nodes has been killed.

11.4.3.2 Overlapping Resource Output

As shown in the following `rinfo` output, resources 1721 and 1722 overlap:

```
# rinfo -rl
RESOURCE         CPUS     STATUS           TIME     USERNAME         NODES
parallel.1721    2        allocated        00:22        fred         atlas[12-13]
parallel.1722    2        allocated        00:18        fred         atlas[12-13]
```

Example 11–6 Overlapping Resources

This example shows how the `scload` command informs the user about overlapping resources:

```
# scload -r 1721
CPU Utilisation (system+user) (%) for resource 1721
390-400:
310-390:
290-310:
210-290:
190-210:
110-190:
 90-110:
 10- 90:
  0- 10:   [12-13]

There are overlapping resources:
resource         nodes
==========       =====
1722             atlas[12-13]
```

11.4.3.3 Domain-Level Output

By default, the `scload` command displays node-level performance statistics. However, on a large HP AlphaServer SC system containing many nodes, this may result in an overwhelming amount of information. Use the `-d` option to summarize the `scload` output — this will display domain-level performance statistics.

Example 11–7 Domain-Level Statistics: Run-Queue Length

The run-queue-length performance data is summed over all of the nodes in each domain, as shown in the following example:

```
# scload -d all -m rq5
Run Queue Lengths per domain (total over nodes in domain)
atlasD0 [0-3]   (4 nodes):  1.51
atlasD1 [4-11]  (8 nodes):  0.00
atlasD2 [12-13] (2 nodes):  0.05
atlasD3 [14-15] (2 nodes):  0.00

No data:   [1-11]
```

Example 11–8 Domain-Level Statistics: Free Memory

For each of the other metrics, the data is averaged over all of the nodes in each domain, as shown in the following example:

```
# scload -d all -m freemem
Free Memory (%) per domain (average over nodes in domain)
atlasD0 [0-3]   (4 nodes):   91
atlasD1 [4-11]  (8 nodes):   91
atlasD2 [12-13] (2 nodes):   91
atlasD3 [14-15] (2 nodes):   88
```

12
Managing Multiple Domains

HP AlphaServer SC Version 2.6 supports multiple domains. Each domain can contain up to 32 HP AlphaServer SC nodes, providing a maximum of 1024 HP AlphaServer SC nodes.

The `scrun` command simplifies the task of maintaining multiple domains.

This information in this chapter is arranged as follows:

- Overview of the scrun Command (see Section 12.1 on page 12–2)
- scrun Command Syntax (see Section 12.2 on page 12–2)
- scrun Examples (see Section 12.3 on page 12–6)
- Interrupting an scrun Command (see Section 12.4 on page 12–7)

12.1 Overview of the scrun Command

The `scrun` command allows you to execute a **global command**; that is, a command that can run on all nodes in the HP AlphaServer SC system. You can specify that the command is to run on the specified nodes either in parallel or in sequence.

The `scrun` command can be issued on any HP AlphaServer SC node, or on the management server. The `scrun` command may only be run by the `root` user; the actual commands executed by `scrun` are also run as `root` on each specified node.

The `scrun` command displays the standard output and standard error of the command being executed, as controlled by the `-o` option (see Table 12–1). The standard input of the command being executed is set to `/dev/null`.

An error message is displayed if any nodes or domains are not available to run the command.

If all commands are executed successfully on all the requested nodes or domains, a successful exit status is returned. Otherwise, an unsuccessful exit status is returned, and an error message indicates which nodes had an unsuccessful exit status and which had a successful exit status.

The `scrun` command is described in Section 12.2 on page 12–2, and in the `scrun(8)` reference page.

12.2 scrun Command Syntax

The syntax of the `scrun` command is as follows:

```
scrun {[-d domain_list|all|self] [-m member_number_list|all|self]
[-n node_list|all|self]} [-w all|n] [-o host|quiet] [-t seconds] command
```

Table 12–1 describes the `scrun` command options in alphabetical order.

Table 12–1 scrun Command Options

Option	Description
`-d`	Specifies the domain(s) on which the *`command`* should run: • Use the case-insensitive keyword `self` to run the *`command`* on the current domain (that is, the domain running the `scrun` command). If `self` is specified when running `scrun` on a management server, an error is displayed. • Use the case-insensitive keyword `all` to run the *`command`* on all domains in the system. • Use a list to specify a particular domain or domains. Domains can be specified using the domain number (for example, `-d 0`) or the domain name (for example, `-d atlasD0`). The effect of the `-d` option may be changed by using the `-m` or `-n` option: • If you use the `-d` option alone, the *`command`* will run on one member of each specified domain — this member is chosen at random on each domain. • If you use the `-d` option with the `-m` option, the *`command`* will run on the specified members of each specified domain. • If you use the `-d` option with the `-n` option, the *`command`* will run on the specified nodes only if the nodes are in the specified domains.
`-m`	Specifies the member(s) on which the *`command`* should run: • Use the case-insensitive keyword `self` to run the *`command`* on the current member (that is, the member running the `scrun` command). If `self` is specified when running `scrun` on a management server, an error is displayed. • Use the case-insensitive keyword `all` to run the *`command`* on all system members. • Use a list to specify a particular member or members. Members are specified using the member number (for example, `-m 1`) — there is a maximum of 32 members in each domain. The effect of the `-m` option may be changed by using the `-d` option: • If you use the `-m` option alone, the *`command`* will run on the specified member of the current domain only. Using the `-m` option alone when running `scrun` from the management server will produce an error. • If you use the `-m` option with the `-d` option, the *`command`* will run on the specified members of each specified domain.

Table 12–1 scrun Command Options

Option	Description
`-n`	Specifies the node(s) on which the *command* should run: • Use the case-insensitive keyword `self` to run the *command* on the current node (that is, the node running the `scrun` command). If `self` is specified when running `scrun` on a management server, an error is displayed. • Use the case-insensitive keyword `all` to run the *command* on all nodes in the system. • Use a list to specify a particular node or nodes. Nodes can be specified using the node number (for example, `-n 0`) or the node name (for example, `-n atlas0`). The effect of the `-n` option may be changed by using the `-d` option: • If the `-n` option is used alone, the *command* will run on the specified nodes. • If the `-n` option is used with the `-d` option, the *command* will run on the specified nodes only if the nodes are in the specified domains.
`-o`	Specifies the format of the command output: • Use `-o host` to specify that each line of output should be prefixed with the nodename. This is the default option. • Use `-o quiet` to specify that each line of output should not be prefixed with the nodename.
`-t`	Specifies that the *command* should timeout after a certain number of seconds. If the *command* does not complete within the specified period, a command termination signal is sent to all nodes on which the *command* has not completed. If the `-w` option is used and set to a number (*n*) to specify that the *command* is to be run sequentially, but on *n* nodes in parallel, the timeout period is reset after each execution of the *command*. If the `-w` option is not used, or is set to `all` to specify that the *command* is to be run on all specified nodes in parallel, the *command* will timeout if it has not completed successfully on all nodes within the specified period.

Table 12–1 scrun Command Options

Option	Description
-w	The width option specifies the maximum number of nodes or domain members the *command* is to be run on in parallel. Use the case-insensitive keyword all to specify that the *command* is to be run on all specified nodes or domain members in parallel. • Use an integer value (*n*) to specify the largest number of nodes the *command* can be executed on in parallel within the system or domain. If the -d option is also specified on the command line, then the value *n* specifies the largest number of members in a particular domain that the *command* can be executed on in parallel. For example, running the following command: scrun -d all -m all -w 1 *command* is the same as specifying: scrun -d all CluCmd *command* The *command* will be executed on each domain in parallel, one member at a time. But running the following command: scrun -n all -w 2 *command* will execute the *command* on the first two nodes, then the next two nodes, and so on, until the *command* has been executed on all of the nodes in the system, regardless of which domains the nodes reside in. If the -w option is not used, the *command* is executed on all specified nodes or domain members in parallel.

The -d, -m, and -n options determine where the *command* will run — at least one of these three options must be specified. Each of these options can specify a single item, a range of items, or a list of items:

- A single item is specified without any additional punctuation.
- A range is surrounded by square brackets and is enclosed within quotation marks. The start or the end of each range may be omitted; this is equivalent to using the minimum or maximum value, respectively.
- List items are separated by commas. Lists may include ranges.

If the *command* includes any items that need to be interpreted on the target host, you must use quotes appropriate to your shell environment to delay the evaluation of those arguments until they are received on the remote machine. For example, in the following command:

```
# scrun -n all 'HOST=`hostname`; echo $HOST'
```

the 'hostname' command is interpreted on the executing nodes, and not on the node where the scrun command is initiated. In this case, the command will not work unless it is enclosed within single quotation marks.

If a list of domains and a list of members are specified, and the `-w` option (width) is set to 1, the command is executed sequentially within each domain, but is run on the specified domains in parallel.

For example, if you specify the following:

```
# scrun -d all -m '3,[5-7]' -w 1 command
```

the *command* starts to run on each domain in parallel, but runs sequentially on member 3, 5, 6 and 7 within each domain.

12.3 scrun Examples

The examples in Table 12–2 show how the `-d`, `-m`, and `-n` options determine where the command runs.

Table 12–2 scrun Examples

Command	Runs on	#Times Run
`scrun -d 0 hostname`	A random member of domain 0	1
`scrun -d '[1-3]' hostname`	A random member of domains 1, 2, and 3	3
`scrun -d '[3-]' hostname`	A random member of all domains except domains 0, 1, and 2	#Domains - 3
`scrun -d '[-2]' hostname`	A random member of domains 0, 1, and 2	3
`scrun -d 1,4,7 hostname`	A random member of domains 1, 4, and 7	3
`scrun -d '1,[3-5],7,[9-11]' hostname`	A random member of domains 1, 3, 4, 5, 7, 9, 10, and 11	8
`scrun -m 1 hostname`	Member 1 of the current domain	1
`scrun -d '1,[3-5],7' -m 3,5 hostname`	Members 3 and 5 of domains 1, 3, 4, 5, and 7	10
`scrun -n 17 hostname`	Node 17	1
`scrun -d 0 -n 17 hostname`	Node 17, if node 17 is in domain 0; otherwise, fails with an error	1 or 0

12.4 Interrupting an scrun Command

Pressing Ctrl/C while an `scrun` command is running has the following effect:

1. Pressing Ctrl/C once will send a SIGINT signal to each of the commands being run by the `scrun` command. The SIGINT signal, which is the usual signal sent by Ctrl/C, will stop most programs. However, if the command being run chooses to ignore this signal, the command will not stop (and, therefore, the `scrun` command will not stop).
2. Pressing Ctrl/C a second time sends a SIGKILL signal to each of the commands being run by the `scrun` command. This is the same signal as that sent by a `kill -9` command. The SIGKILL signal will stop any command, as commands cannot ignore this signal.

See also Section 29.13 on page 29–36 for related troubleshooting information.

13
User Administration

The tasks associated with managing users on an HP AlphaServer SC system are similar to those associated with managing users on a standalone UNIX system. This chapter describes the user administration tasks that are specific to installations in which NIS is not configured and users on the system are local to the HP AlphaServer SC system.

This information in this chapter is arranged as follows:

- Adding Local Users (see Section 13.1 on page 13–2)
- Removing Local Users (see Section 13.2 on page 13–2)
- Managing Local Users Across Domains (see Section 13.3 on page 13–3)
- Managing User Home Directories (see Section 13.4 on page 13–3)

Note:

Management of RMS users is detailed in Chapter 5.

13.1 Adding Local Users

Local users can be added to an HP AlphaServer SC system in the standard UNIX fashion; that is, by using either of the following methods:

- `sysman` command (SysMan Menu)
- `adduser` command

The `sysman` command invokes a graphical interface that allows you to perform integrated system administration tasks, while the `adduser` command is an interactive script. Both of these commands interactively prompt for the information needed to add users:

- Login name
- Userid (UID) — both `sysman` and `adduser` assign the next available UID by default
- Password — enter this twice for confirmation
- Login group — this should be the primary group
- Additional groups — these should be secondary groups
- Shell
- User's home directory

For more information about these commands, see the `sysman(8)` and `adduser(8)` reference pages.

If you are running on a system on which enhanced security is enabled, you must use the `dxaccounts` command to add users. The `dxaccounts` command provides a graphical interface that allows you to manage users in a secure environment. For more information on the `dxaccounts` command, see the `dxaccounts(8)` reference page.

The addition of local users is a domainwide operation. You need only add users once per 32-node domain.

13.2 Removing Local Users

You can remove local users by any of the following methods:

- `sysman` command (SysMan Menu)
- `dxaccounts` command (if you have configured enhanced security)
- `removeuser` command

For more information about these commands, see the `sysman(8)`, `dxaccounts(8)`, and `removeuser(8)` reference pages.

13.3 Managing Local Users Across Domains

If you add users locally on one domain, you must add these users with the same set of attributes on all other domains that are part of the HP AlphaServer SC system. On a fully configured HP AlphaServer SC system that has four domains, this entails adding users four times. This is trivial if only adding or removing a small number of users. However, if adding a large number of users, this can become a time-consuming task.

If enhanced security is not enabled on your HP AlphaServer SC system, this task can be simplified by first adding users on one domain and then replicating the resulting `/etc/passwd` and `/etc/group` files on each of the other domains.

Alternatively, one of the domains can be configured as a NIS master that exports the user information to the other domains. These other domains should be configured as NIS slave servers, as described in Section 22.7 on page 22–21.

13.4 Managing User Home Directories

The user home directories can be on a local file system on one domain, or NFS-mounted from an external location. File system performance will be better within a domain if user home directories are local.

Once the user home directories have been set up on one domain, they must be NFS-exported to the other domains within the HP AlphaServer SC system. The path to the user home directories on each machine must be consistent, to ensure that parallel jobs spanning the complete system will have a consistent view of the user's file system. This consistent view is necessary to ensure that jobs can start correctly and that files can be read and written.

Note:

When NFS-exporting files from a domain, the cluster alias name for that domain is the name from which the files are mounted on the other domains.

File systems that are NFS-mounted by a domain should be placed in the `/etc/member_fstab` file rather than in the `/etc/fstab` file.

14

Managing the Console Network

This chapter describes the console network and the Console Management Facility (CMF).

The information in this chapter is organized as follows:

- Console Network Configuration (see Section 14.1 on page 14–2)
- Console Logger Daemon (cmfd) (see Section 14.2 on page 14–2)
- Configurable CMF Information in the SC Database (see Section 14.3 on page 14–4)
- Console Logger Configuration and Output Files (see Section 14.4 on page 14–5)
- Console Log Files (see Section 14.5 on page 14–8)
- Configuring the Terminal-Server Ports (see Section 14.6 on page 14–9)
- Reconfiguring or Replacing a Terminal Server (see Section 14.7 on page 14–9)
- Manually Configuring a Terminal-Server Port (see Section 14.8 on page 14–10)
- Changing the Terminal-Server Password (see Section 14.9 on page 14–12)
- Configuring the Terminal-Server Ports for New Members (see Section 14.10 on page 14–12)
- Starting and Stopping the Console Logger (see Section 14.11 on page 14–13)
- User Communication with the Terminal Server (see Section 14.12 on page 14–14)
- Backing Up or Deleting Console Log Files (see Section 14.13 on page 14–15)
- Connecting to a Node's Console (see Section 14.14 on page 14–15)
- Connecting to a Terminal Server (see Section 14.15 on page 14–16)
- Monitoring a Node's Console Output (see Section 14.16 on page 14–16)
- Changing the CMF Port Number (see Section 14.17 on page 14–17)
- CMF and CAA Failover Capability (see Section 14.18 on page 14–17)
- Changing the CMF Host (see Section 14.19 on page 14–21)

14.1 Console Network Configuration

The console network comprises console cables and terminal servers. To cable up the terminal servers, you must have the following:

- Terminal server(s)
  Depending on the number of nodes, you may need several terminal servers.
  HP AlphaServer SC Version 2.6 supports DECserver 900TM terminal servers and DECserver 7*XX* terminal servers.
- One AUI-10BaseT adapter for each DECserver 900TM terminal server
- One Cat 5 Ethernet cable for each terminal server
- One cable for each node, to connect each node's console port to the terminal server(s)
- Cables to connect every additional device to the terminal server(s)

Each terminal server manages up to 32 console ports. The serial port of each system node is connected to a port on the terminal server. The order of connections is important: Node 0 is connected to port 1 of the first terminal server, Node 1 to port 2, and so on.

The terminal server is in turn connected to the management network. This configuration enables each node's console port to be accessed using IP over the management network. This facility provides management software with access to a node's console port (for boot, power control, configuration probes, firmware upgrade, and so on).

14.2 Console Logger Daemon (cmfd)

All input and output to a node's console is handled by the console logger daemon (`cmfd`).

On large systems, there may be more than one `cmfd` daemon. By default, an HP AlphaServer SC system is configured to start one `cmfd` daemon per 256 console connections (that is, nodes). If there is a lot of console network activity, it may be necessary to decrease the number of console connections that each daemon handles (option `[12]`), and increase the number of daemons (option `[13]`), as described in Section 14.3 on page 14–4.

The `cmfd` daemon runs on one node in the system, typically on the management server (if used) or Node 0. When started, the `cmfd` daemon connects to each terminal-server port listed in the `sc_cmf` table in the SC database (see Section 14.4 on page 14–5). The `cmfd` daemon then waits for either user connections (using the `sra -c` command) or output from any node's console.

When a user connects to a node's console, the following actions happen:

1. The `sra` command looks up the `sc_cmf` table in the SC database to find the hostname (CMFHOST) and port number (CMFPORT) for the `cmfd` daemon serving that connection.

- CMFHOST

  In most cases, CMF is configured to run on the management server. In such cases, CMFHOST is the hostname of the management server (for example, `atlasms`). However, if CMF is running on the first domain (if the HP AlphaServer SC system does not have a management server) or on a clustered management server, and has been CAA enabled, then CMFHOST is the default cluster alias (for example, `atlasD0` or `atlasms`).

  To identify the node running `cmfd` (CMFHOST), run the following command:
  `# sra dbget cmf.host`

- CMFPORT

  Each `cmfd` daemon requires two ports. The first port is used by SRA clients to access the console connection (USER port). The second port is a control port that is used to perform administration tasks such as log file rotation. In systems with more than one `cmfd` daemon, the port that connects to a node can be determined using the formula `port# = CMFPORT + [(node#/256) * 2]`, as shown in the following examples:
  - For `atlas[0-255]`, the CMF port number is `6500`
  - For `atlas[256-511]`, the CMF port number is `6502`
  - For `atlas[512-767]`, the CMF port number is `6504`

  To identify the `cmfd` port number (CMFPORT), run the following command:
  `# sra dbget cmf.port`

2. The `cmfd` daemon accepts the connection and provides a proxy service to the console of the specified node.
3. As well as passing data between the user and the console, the daemon also logs all data to the node-specific log file, `/var/sra/logs/`*`nodename`*`.log`.

The `cmfd` daemon also controls access to each node — instead of using `telnet`, you can use an `sra` command to connect to a node's console port. This is very useful, because the `sra` command produces log files. Even if somebody else is connected to a node, you can use the `sra -m[l]` command to monitor the node's console output. This command is described in Section 14.16 on page 14–16.

As well as connecting HP AlphaServer SC nodes to the terminal server, you can also use CMF to connect and monitor other devices, as documented in Section 14.8. For example, you may choose to connect the controller for the RAID array — this can be very useful, as all output is captured by CMF. You can also use CMF to connect and monitor the console ports of network equipment.

See the `cmfd(8)` reference page for more information on the `cmfd` daemon.

14.3 Configurable CMF Information in the SC Database

Each HP AlphaServer SC Version 2.6 `cmfd` daemon can support up to 256 nodes. Information about the `cmfd` daemons is recorded in the SC database, and can be changed using the `sra edit` command, as shown in the following example output:

```
# sra edit
sra> sys show system
Id          Description                                                  Value
------------------------------------------------------------------------------

.
.
.
[8  ]       Node running console logging daemon (cmfd)                   atlasD0
[9  ]       cmf home directory                                           /var/sra/
[10 ]       cmf port number                                              6500
[11 ]       cmf port number increment                                    2
[12 ]       cmf max nodes per daemon                                     256
[13 ]       cmf max daemons per host                                     4
[14 ]       Allow cmf connections from this subnet
255.255.0.0,11.222.0.0/255.255.0.0,10.0.0.0/255.255.255.0
[15 ]       cmf reconnect wait time (seconds)                            30
[16 ]       cmf reconnect wait time (seconds) for failed ts              1800

.
.
.
```

Option `[8]` shows the node on which the `cmfd` daemons are running (CMFHOST), and option `[9]` shows the CMF home directory. The number of `cmfd` daemons is shown in option `[13]`. By default, the CMF port number (CMFPORT) starts at 6500 (see option `[10]`) and increments by two (see option `[11]`) after each 256 nodes (see option `[12]`). Therefore, the CMF port number for nodes 0 to 255 inclusive is 6500. For nodes 256 to 511 inclusive, the CMF port number is 6502, and so on.

Option `[14]` controls which connections are accepted by the console logging daemons. The format is `mask,addr1/mask1,addr2/mask2,` and so on. Each address/mask pair is separated by a comma — do not insert a space after the comma. The first connection is specified by a mask only, because the host IP address of the node running the `cmfd` daemon(s) is implicit. Option `[14]` is populated by the `sra setup` command, and should not need to be altered in normal operation. In the above example, `11.222.0.0` is the IP address of a site-specific external network, and connections are allowed from the following subnets: 10.128/16 11.222/16 10/24.

Options `[15]` and `[16]` control how the console logging daemon handles terminal server connection errors:

- If a connection fails, the daemon marks the connection as being down. Option `[15]` controls how long the daemon waits before retrying any connections that are marked as being down. The default value is 30 seconds.

- If there have been more than five connection failures on any given terminal server, the daemon marks the entire terminal server as being down, and will only attempt to reconnect after a length of time determined by option [16] — by default, 30 minutes.

To change these values, use the sra edit command. The sra edit command then asks if you would like to restart or update the daemons, as follows:

```
Modify SC database only (1), update daemons (2), restart daemons (3) [3]:
```

Enter 2 to update the daemons so that they resynchronize with the updated SC database.

14.4 Console Logger Configuration and Output Files

The cmfd configuration details are located in the sc_cmf table in the SC database. This table is populated automatically by the sra command (using either sra setup or sra edit) and should not be changed manually.

The sc_cmf table contains an entry for each node in the system, as shown in the following example:

```
$ rmsquery -v "select * from sc_cmf"
name         cmf_host       cmf_port    ts_host        ts_port
-------------------------------------------------------------
atlas0       atlasms        6500        atlas-tc1      2001
atlas1       atlasms        6500        atlas-tc1      2002
.
.
.
atlas31      atlasms        6500        atlas-tc1      2032
atlas32      atlasms        6500        atlas-tc2      2001
.
.
.
atlas255     atlasms        6500        atlas-tc8      2032
atlas256     atlasms        6502        atlas-tc9      2001
.
.
.
atlas511     atlasms        6502        atlas-tc16     2032
atlas512     atlasms        6504        atlas-tc17     2001
.
.
.
atlas767     atlasms        6504        atlas-tc24     2032
atlas768     atlasms        6506        atlas-tc25     2001
.
.
.
atlas1023    atlasms        6506        atlas-tc32     2032
```

Each entry in the sc_cmf table contains the following five fields:

- name is the name of a node in the HP AlphaServer SC system.
- cmf_host is the hostname or cluster alias of the node on which the cmfd daemons are running.
- cmf_port is the CMF port number, which starts at 6500 and increments by two after each 256 nodes. Therefore, the CMF port number for nodes 0 to 255 inclusive is 6500. For nodes 256 to 511 inclusive, the CMF port number is 6502, and so on.

- ts_host is the name of the terminal server.
- ts_port is the telnet listen number, which starts at 2001; therefore, port 1 is 2001, port 2 is 2002, and so on. There are 32 ports on each terminal server, so the maximum ts_port value is 2032.

The user may create a cmfd configuration file, /var/sra/cmf.conf.local, to include other devices whose serial ports are connected to the terminal server. The format of this file is:

```
name        ts_host       ts_port
```

In the following example cmf.conf.local file, the management server's console port is connected to the first terminal server on port 24, and a RAID controller is connected to the first terminal server on port 25:

```
atlasms     atlas-tc1     2024
raid1       atlas-tc1     2025
```

Note that it is not sufficient to just add an entry to the cmf.conf.local file — you must also manually configure the terminal server to define the port to which the RAID controller is connected, and to set the telnet listen number for that port, as described in Section 14.8 on page 14–10.

Note:

You can only use the sra ds_configure command to configure a terminal-server port for a node — for any device other than a node, such as the RAID controller in the above example, you must configure the terminal server manually.

Once the text file is created, the changes must be written to the SC database, and the daemon(s) must be either restarted or instructed to scan the updated SC database.

You can do this by running the sra edit command on CMFHOST, as follows:

```
# sra edit
sra> sys update cmf
```

The sra edit command repopulates the sc_cmf table, and adds the information from the cmf.conf.local file to the sc_cmf table. The sra edit command then asks if you would like to restart or update the daemons, as follows:

```
Modify SC database only (1), update daemons (2), restart daemons (3) [3]:
```

Enter 2 to update the daemons so that they resynchronize with the updated SC database.

The name assigned to the device — for example, raid1 — is arbitrary and does not need to appear elsewhere. You can use this name to connect to the serial port of the device specified in cmf.conf.local, using the following command:

```
# sra -c raid1
```

The cmfd daemon produces output relating to its current state; this output is stored in the /var/sra/adm/log/cmfd/cmfd_*hostname_port*.log file.

The daemon verbosity level is determined by the -d flag (see Table 14–2). By default, the verbosity level is set to 2. Although the daemon log file is archived each time the daemon starts, the log file can grow to a very large size over time. You can reduce the verbosity by setting the -d flag to 1 or 0.

When cmfd is idle — that is, no users connected to any console — the last entry in the output file will be as follows:

```
CMF [12/Mar/2001 10:39:03 ] : user_mon: , sleeping....
```

If a user connects to a node's console, using the sra -c command, entries similar to the following will appear in the /var/sra/adm/log/cmfd/cmfd_*hostname_port*.log file:

```
CMF [12/Mar/2001 12:28:03 ] : connecting user to atlas1 (port 2002 on server atlas-tc1)
CMF [12/Mar/2001 12:28:03 ] : user_mon(), received wake signal
```

You can connect to a cmfd daemon by specifying the appropriate port number in a telnet command. To connect to the cmfd daemon that serves nodes 0 to 255 inclusive, specify port 6501. To connect to the cmfd daemon that serves nodes 256 to 511 inclusive, specify port 6503, and so on.

Once you have connected, the cmf> prompt appears, as shown in the following example:

```
atlasms# telnet atlasms 6501
Trying 10.128.101.1...
Connected to atlasms.
Escape character is '^]'.
*** CLI starting ***
cmf>
```

Note:

The CMF interpreter session must not be left open for an extended period of time, as this will interfere with normal administration tasks. To get information on connected ports or connected users, use the sra ds_who command, as described in Chapter 16.

Do not use the update db or log rotate commands when running multiple daemons. Use the /sbin/init.d/cmf [update|rotate] commands instead.

Table 14–1 describes the CMF interpreter commands that you can use at the cmf> prompt.

Table 14–1 CMF Interpreter Commands

Command	Description
help	Displays a list of CMF interpreter commands.
update db	Instructs the cmfd daemon to reread the configuration file. This is equivalent to sending a SIGHUP signal to the cmfd daemon.

Table 14–1 CMF Interpreter Commands

Command	Description
`show user` *name*`|all`	Displays information about current user sessions.
`show ts` *name*`|all`	Displays information about terminal server connections.
`disconnect user|ts` *name*	Closes a user or terminal server session.
`disconnect user|ts all`	Closes all user sessions, or all terminal server sessions. Exercise caution before using this command.
`log stop|start`	The `log stop` command stops proxy operations and closes log files. The `log start` command re-opens log files and resumes proxy operations. These commands may be used by an external program to manually rotate the log file directory.
`log rotate`	The `log rotate` command performs the following actions: 1. Stops proxy operations. 2. Closes the current log files. 3. Creates a new directory in `$CMFHOME/cmf.dated`. 4. Moves the symbolic link `$CMFHOME/logs` to point to the new directory created in step 3. 5. Re-opens the log files. 6. Resumes proxy operations.

14.5 Console Log Files

The console logger daemon, `cmfd`, outputs each node's console activity to a node-specific file in the `/var/sra/cmf.dated` directory. This directory contains sub-directories for the various console log file archives. Both `/var/sra/cmf.dated/current` and `/var/sra/logs` are symbolic links to the current logs directory.

To archive console log files, use the `/sbin/init.d/cmf rotate` command. This command uses the CMF interpreter interface to perform the following tasks:

1. Suspend console logging.
2. Create a new directory in the `/var/sra/cmf.dated` directory.
3. Change the `/var/sra/cmf.dated/current` and `/var/sra/logs` symbolic links so that they both point to the new directory created in step 2.
4. Restart console logging.

Data output from the terminal servers is not lost during this process.

The archive duration is controlled by a `crontab` entry on the CMFHOST, as shown in the following example:

```
0 20 13,27 * * /sbin/init.d/cmf rotate
```

This `crontab` entry, which is created by the `sra setup` command, results in the `cmf` startup script being called with the `rotate` option (see Table 14–1). In the above example, it runs on the 13th and 27th day of each month. The archive dates are determined by the date on which `sra setup` is run.

If CMF is running in a domain and is CAA enabled, each member that is a potential CMF host should have a similar crontab entry.

14.6 Configuring the Terminal-Server Ports

The `sra setup` command configures the terminal-server ports so that they can act as a remote console connection, as shown in the following extract from the output of the `sra setup` command:

```
Configure terminal servers ? [no]:y
This may take some time
Continue ? [y|n]: y
connecting to DECserver atlas-tc1 (10.128.100.1)
configuring host atlas0    [port = 1]
configuring host atlas1    [port = 2]
configuring host atlas2    [port = 3]
...
```

This command configures the ports for each node in the system.

14.7 Reconfiguring or Replacing a Terminal Server

If you press the Reset button on the terminal server, it loses its configuration information and is reset to the factory configuration. To reconfigure or replace a terminal server, you must perform the following steps as the `root` user:

1. Configure the terminal-server IP address as described in Chapter 3 of the *HP AlphaServer SC Installation Guide*.
2. Configure the terminal-server ports as described here:
   a. On CMFHOST, stop the console logger daemon(s).

      If CMF is CAA-enabled, stop the daemons as follows:

      ```
      # caa_stop SC10cmf
      ```

      If CMF is not CAA-enabled, stop the daemons as follows:

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

   b. On CMFHOST, start the console logger daemon(s) in maintenance mode, as follows:

      ```
      # /sbin/init.d/cmf start_ts
      ```

c. Configure the ports for the nodes, as follows:
```
# sra ds_configure -nodes nodes
```
where *nodes* is `atlas[0-31]` for the first terminal server, `atlas[32-63]` for the second, and so on.

Alternatively, if using the default configuration of 32 nodes per domain, you can configure the ports for the nodes by running the following command:
```
# sra ds_configure -dom N
```
where *N* is `0` for the first terminal server, `1` for the second, and so on.

To configure all ports on all terminal servers, run the following command:
```
# sra ds_configure -nodes all
```

d. Configure the ports for any other devices, as described in Section 14.8 on page 14–10.

e. On CMFHOST, stop the console logger daemon(s) that are running in maintenance mode, as follows:
```
# /sbin/init.d/cmf stop_ts
```

f. On CMFHOST, restart the console logger daemon(s).

If CMF is CAA-enabled, start the daemons as follows:
```
# caa_start SC10cmf
```

If CMF is not CAA-enabled, start the daemons as follows:
```
# /sbin/init.d/cmf start
```

g. Set the password, if necessary, as described in Section 14.9 on page 14–12.

14.8 Manually Configuring a Terminal-Server Port

When connecting a node to a terminal server, you can automatically configure the terminal-server port using the `sra ds_configure` command. However, if you connect any device other than a node, you must manually configure the terminal-server port.

The following example shows how to configure the terminal-server port for the RAID controller described in Section 14.4 on page 14–5, whose entry in the `cmf.conf.local` file is as follows:
```
raid1    atlas-tc1 2025
```

To configure the terminal-server port for this device, perform the following steps:

1. Connect to the terminal server, as follows:
```
# sra -c atlas-tc1
```

2. Configure the port as follows:
```
# access
Network Access SW V2.4 BL50 for DS732
(c) Copyright 2000, Digital Networks - All Rights Reserved
Please type HELP if you need assistance
```

```
Enter username> system
Local> set priv
Password> password
Local> define port 25 access remote
Local> define port 25 autobaud disabled
Local> define port 25 autoconnect disabled
Local> define port 25 break disabled
Local> define port 25 dedicated none
Local> define port 25 dsrlogout disabled
Local> define port 25 dtrwait enabled
Local> define port 25 inactivity logout disabled
Local> define port 25 interrupts disabled
Local> define port 25 longbreak logout disabled
Local> define port 25 signal check disabled
Local> logout port 25
Local> change telnet listener 2025 ports 25 enabled
Local> change telnet listener 2025 identification raid1
Local> change telnet listener 2025 connections enabled
Local> logout
```

where *password* is a site-specific value (the factory default password is `system`), `25` is the port number, `2025` is the `telnet` listen number (that is, 2000+port_number), and `raid1` is the host name of the device.

3. Rebuild the `sc_cmf` table in the SC database, by running the `sra edit` command on CMFHOST, as follows:

```
# sra edit
sra> sys update cmf
```

The `sra edit` command repopulates the `sc_cmf` table, and adds the information from the `cmf.conf.local` file to the `sc_cmf` table. The `sra edit` command then asks if you would like to restart or update the daemons, as follows:

```
Modify SC database only (1), update daemons (2), restart daemons (3) [3]:
```

Enter 2 to update the daemons so that they resynchronize with the updated SC database.

4. Restart the `cmfd` daemon, as follows:

```
# /sbin/init.d/cmf restart
```

5. Test the configured port by connecting to the serial port of the RAID controller, as follows:

```
# sra -cl raid1
```

For more information, see the terminal-server documentation.

14.9 Changing the Terminal-Server Password

The factory default password for the terminal server is `system`. To change this value, run the `sra ds_passwd` command, as follows (where `atlas-tc1` is an example terminal server name):

```
# sra ds_passwd -server atlas-tc1
This command will set the password on the named
terminal server (atlas-tc1).
Confirm change password for server atlas-tc1 [yes]:
Enter new password for atlas-tc1: site_specific_password[1]
Please re-enter new password for atlas-tc1: site_specific_password[1]
Info: connecting to terminal server atlas-tc1 (10.128.100.1)
Info:    Connected through cmf
```

This command sets the password on the terminal server, and updates the entry in the SC database.

14.10 Configuring the Terminal-Server Ports for New Members

If you wish to add a node to the system after installation, you must first update the SC database using either the `sra edit` command or the `sra setup` command:

- If there are enough spare ports on the terminal server and you are adding a small number of nodes, use the `sra edit` command (see Section 16.2.2.2 on page 16–27).
- If you need to configure a new terminal server, or if adding a large number of nodes, use the `sra setup` command.

Either command will perform the following steps:

- Increase the "number of nodes" entry (`system.num.nodes`) in the SC database.
- Add an entry for the new node to the `sc_cmf` table in the SC database.
- Configure the terminal-server port for the new node.
- Stop and restart `cmfd` console logging to include the new node.
- Probe the new node to determine its ethernet hardware address.
- Add the node to the `/etc/hosts` file.
- Add the node to the RIS client database.

1. The value that you enter is not echoed on screen.

14.11 Starting and Stopping the Console Logger

The console logging service (CMF) is started automatically on system boot:

- If CMF is CAA-enabled, CAA will determine which node should run the console logging daemon(s), and start the daemons as necessary. To stop and start the CMF service *manually* when CMF is CAA enabled, use the `caa_stop` and `caa_start` commands. Use the `caa_stat -t` command to find the status of CAA applications.
- If CMF is not CAA enabled, the `/sbin/init.d/cmf` startup script will start the daemons as necessary, by looking up the `sc_cmf` table in the SC database. To stop and start the CMF service *manually* when CMF is not CAA enabled, use the `/sbin/init.d/cmf stop|start` commands.

CMF can also be started directly from the command line. The syntax is as follows:

```
# /usr/sra/bin/cmfd -D -t -b [options]
```

Table 14–2 lists the `cmfd` options that are valid in HP AlphaServer SC Version 2.6, where *N* is an integer and *B* is boolean (0=no; 1=yes). The `-a` option (archive) is not valid in HP AlphaServer SC Version 2.6.

Table 14–2 cmfd Options

Option	Description
`-b`[1]	Bind local address when connecting to terminal server.
`-D`[1]	Run in distributed mode. This option is for systems with large node counts, in which multiple CMF daemons are running. When this option is specified, the CMF configuration information is read from the SC database, and archiving is disabled.
`-d` *N*	Set debug level to *N*. The debug range is 0 (no debug) to 3 (verbose). Default: *N* = 0
`-f`	Run in foreground mode.
`-h`	Print help information
`-i`	Provide access to telnet port on terminal server(s) only. Do not connect to terminal server ports that are connected to nodes. This mode can be used to log out hung terminal server ports.
`-l`	Specifies the CMF home directory. Default: `/var/sra`
`-p` *N*	Listen on port *N* for user connections. Default: *N* = 6500
`-s` *B*	Strip carriage returns from log files. Default: *B* = 1
`-t`[1]	Provide access to telnet port on terminal server(s).
`-u` *B*	Stamp log files every 15 minutes. Default: *B* = 1

[1]These options must be specified in HP AlphaServer SC Version 2.5 and higher.

The following example shows how to manually start CMF (on the first 256 nodes) with debug enabled and in foreground mode — this can be useful when troubleshooting:

```
# /usr/sra/bin/cmfd -D -t -b -d 3 -f
```

14.12 User Communication with the Terminal Server

Users usually communicate with the terminal server *via* the `cmfd` daemon. When the `cmfd` daemon starts, it connects to all ports (for which it is responsible) on the terminal server. Issuing an `sra -cl atlas0` command connects the user to the `cmfd` daemon, which in turn directs the user interaction to the appropriate terminal server port connection.

When the CMF service is started, the terminal server ports are not logged out prior to starting the `cmfd` daemon. To log out the terminal server ports, use the `sra ds_logout` command.

You can use the `sra ds_logout` command to perform the following tasks:

- Disconnect a User Connection from CMF (see Section 14.12.1)
- Disconnect a Connection Between CMF and the Terminal Server (see Section 14.12.2)
- Bypass CMF and Log Out a Terminal Server Port (see Section 14.12.3)

14.12.1 Disconnect a User Connection from CMF

To disconnect a user connection from CMF, run the `sra ds_logout` command without the `-ts` or `-force` options, as shown in the following example:

```
# sra ds_logout -nodes atlas3
```

This is the preferred method for disconnecting a user from a console session. This command disconnects the user side of the proxy connection, but leaves the terminal-server side open and continues to log data output from the terminal server.

14.12.2 Disconnect a Connection Between CMF and the Terminal Server

To disconnect the connection between CMF and the terminal server, run the `sra ds_logout` command with the `-ts` option, as shown in the following example:

```
# sra ds_logout -nodes atlas3 -ts yes
```

The console logger daemon will reestablish the connection to the terminal server after a configurable delay (see Section 14.3 on page 14–4). This command may be used to reset communication parameters after a console cable has been replaced or moved.

14.12.3 Bypass CMF and Log Out a Terminal Server Port

To log directly onto the terminal server (bypassing CMF) and log out the port, run the `sra ds_logout` command with the `-force` option, as shown in the following example:

```
# sra ds_logout -nodes atlas3 -force yes
```

Alternatively, run the following command to connect to the terminal server:

```
# sra -cl terminal_server
```

and then log the port out manually.

To log out all terminal server ports before starting the `cmfd` daemon, run the following command:

```
# sra ds_logout -nodes all -force yes
```

Once the terminal server ports have been logged out, restart the console logging daemons as described in Section 14.11 on page 14–13.

14.13 Backing Up or Deleting Console Log Files

The console log files in the `/var/sra/cmf.dated/`*`date`* directory are not monitored by any automated process. Although the log file directory is rotated *via* a `crontab` entry, it may become necessary to manually perform this process; for example, if the `/var` file system becomes full.

To back up the console log files, use the `/sbin/init.d/cmf rotate` command. This command is described in Section 14.5 on page 14–8.

To delete the console log files, perform the following steps on the node on which `cmfd` is running (usually Node 0):

1. Stop CMF, as described in Section 14.11 on page 14–13.
2. Delete the log files, as follows:
   ```
   # rm /var/sra/cmf.dated/date/filenames.log
   ```
   Alternatively, move the log files to another location, as follows:
   ```
   # mv /var/sra/cmf.dated/date/filenames.log new_location
   ```
3. Start CMF again, as described in Section 14.11 on page 14–13.

Note:

During this time, the consoles are not being monitored.

14.14 Connecting to a Node's Console

Use the `sra -c[l]` command to connect to a node's console, as shown in the following examples:

- To connect to the console of `atlas3` in a new (Xterm) window:
  Ensure that the DISPLAY environment variable is set, and run the following command:
  ```
  # sra -c atlas3
  ```

- To connect to the console of atlas3 in the current (local) window, run the following command:
  ```
  # sra -cl atlas3
  ```

The sra -c command does not telnet directly to the terminal server; it telnets to the node running cmfd using a particular port (the default port is 6500). Attempting to connect to the node's console by telneting to the terminal server will fail with the following error:

```
# telnet atlas-tc1 2002
Trying x.x.x.x...
telnet: Unable to connect to remote host: Connection refused
```

The connection is refused because the console logger, cmfd, is already connected to the port. This is true whether or not a user is running the sra -c command.

14.15 Connecting to a Terminal Server

The terminal servers are connected to the management network (10.128.100.x). The telnet service has an out_alias attribute in the /etc/clua_services file. This means that when you use telnet, it appears to the receiver that the service is coming from the cluster alias. However, the cluster alias is an IP address on the external network, so responses from the terminal server (on the management network) are not handled correctly.

The sra -c[l] command offers a way around this restriction. To connect to the terminal server, use the sra command as follows:

```
# sra -c atlas-tc1
```

This results in cmfd connecting to the telnet port (port 23) on the terminal server and, therefore, avoiding the telnet out_alias restriction.

14.16 Monitoring a Node's Console Output

Use the sra -m[l] command to monitor a node's console output, as shown in the following examples:

- To monitor the console of atlas3 in a new (Xterm) window:
  Ensure that the DISPLAY environment variable is set, and run the following command:
  ```
  # sra -m atlas3
  ```
- To monitor the console of atlas3 in the current (local) window, run the following command:
  ```
  # sra -ml atlas3
  ```

The sra -ml command connects to the console logging daemon handling the connection, *via* a read-only connection. The cmfd daemon can provide up to 32 of these connections per console connection.

14.17 Changing the CMF Port Number

By default, the console logging daemon(s) listen for connections on ports starting at 6500. You may wish to change this value; for example, if you are using another (inflexible) application that uses a port number in this range.

You can change the `cmfd` port as follows:

1. If not using a management server, modify the `cmf` entries in both the `/etc/services` and `/etc/clua_services` files.

   If using a management server, modify the `cmf` entries in the `/etc/services` file.

2. Run `cluamgr -f` on each node in each domain to reload the `/etc/clua_services` file, as follows:

   ```
   # scrun -n all 'cluamgr -f'
   ```

3. On CMFHOST, use `sra edit` to update the `cmfd` port number in the SC database, as follows:

   ```
   # sra edit
   sra> sys
   sys> edit system
    Id     Description                                          Value
    ------------------------------------------------------------------
   .
   .
   .
    [10 ]   cmf port number                                     6500
   .
   .
   .
   ------------------------------------------------------------------

   Select attributes to edit, q to quit
   eg. 1-5 10 15

   edit? 10
   cmf port number                              [6500]
   new value? 16000

   cmf port number                              [16000]
   correct? [y|n] y
   ```

   The `sra edit` command then asks if you would like to restart or update the daemons, as follows:

   ```
   Modify SC database only (1), update daemons (2), restart daemons (3) [3]:
   ```

   Enter 3 — the `sra edit` command will restart the daemons using the new port number.

14.18 CMF and CAA Failover Capability

If you are running CMF on a clustered management server, or on the first domain (if the HP AlphaServer SC system does not have a management server), then it is possible to enable CMF as a CAA application.

This section describes the following tasks:

- Determining Whether CMF is Set Up for Failover (see Section 14.18.1 on page 14–18)
- Enabling CMF as a CAA Application (see Section 14.18.2 on page 14–18)
- Disabling CMF as a CAA Application (see Section 14.18.3 on page 14–20)

See Chapter 23 for more information on how to manage highly available applications — for example, how to monitor and manually relocate CAA applications.

14.18.1 Determining Whether CMF is Set Up for Failover

To determine whether CMF is set up for failover, run the `caa_stat` command, as follows:

```
# /usr/sbin/caa_stat -t SC10cmf
```

If CMF is not set up for failover, the following message appears:

```
Application is not registered.
```

If failover is enabled, the command prints status information, including the name of the host that is currently the CMF host.

14.18.2 Enabling CMF as a CAA Application

To enable CMF as a CAA application, perform the following steps:

1. Stop the console logging daemons by running the following command on CMFHOST:

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

2. Use the `sra edit` command to set the CMF host in the SC database to be the cluster alias name of the cluster (either a clustered management server, or the first domain) hosting the CMF service, as follows. In this example, the CMF host is the first domain:

   ```
   # sra edit
   sra> sys
   sys> edit system
   Id      Description                                                  Value
   ------------------------------------------------------------------------
   .
   .
   .
   [8  ]   Node running console logging daemon (cmfd)           atlas0
   .
   .
   .
   Select attributes to edit, q to quit
   eg. 1-5 10 15

   edit? 8
   Node running console logging daemon (cmfd)          [atlas0]
   new value? atlasD0

   Node running console logging daemon (cmfd)          [atlasD0]
   correct? [y|n] y
   ```

The `sra edit` command then asks if you would like to restart or update the daemons, as follows:

```
Modify SC database only (1), update daemons (2), restart daemons (3) [3]:
```

Enter 1 to modify the SC database.

3. Check the CMF CAA profile, as follows:

```
atlas0# caa_stat -p SC10cmf
NAME=SC10cmf
TYPE=application
ACTION_SCRIPT=cmf.scr
ACTIVE_PLACEMENT=0
AUTO_START=1
CHECK_INTERVAL=60
DESCRIPTION=AlphaServer SC Console Management Facility
FAILOVER_DELAY=10
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=
OPTIONAL_RESOURCES=
PLACEMENT=balanced
REQUIRED_RESOURCES=
RESTART_ATTEMPTS=1
SCRIPT_TIMEOUT=300
```

When CMFHOST is the first domain (that is, when the HP AlphaServer SC system does not have a management server), the `HOSTING_MEMBERS` field should contain the hostnames of the first two nodes, and the `PLACEMENT` field should contain the text `restricted`, as shown in the following example:

```
HOSTING_MEMBERS=atlas0 atlas1
PLACEMENT=restricted
```

When CAA-enabled, the `cmfd` daemon will run on any node in the cluster (CMFHOST is the default cluster alias). However, it is preferable to use nodes that have a network interface on the subnet on which the cluster alias is defined — that is, the first two nodes in the default configuration — to avoid an extra routing hop.

If the output of the `caa_stat -p SC10cmf` command does not reflect the values specified above for the `HOSTING_MEMBERS` and the `PLACEMENT` fields, use a text editor to make the necessary changes to the `/var/cluster/caa/profile/SC10cmf.cap` file. Alternatively, use the `caa_profile` command to make these changes. For more information, see the HP *TruCluster Server Cluster Highly Available Applications* manual.

If CMFHOST is a clustered management server, the default values should be used for these fields, as follows:

```
HOSTING_MEMBERS=
PLACEMENT=balanced
```

4. On the new CMFHOST (`atlasD0`), register CMF as a CAA application, as follows:

```
# caa_register SC10cmf
```

5. On the new CMFHOST (atlasD0), start the CAA service, as follows:
```
# caa_start SC10cmf
```

14.18.3 Disabling CMF as a CAA Application

To disable CMF as a CAA application, perform the following steps:

1. Stop the cmf CAA service, by running the following command on CMFHOST (for example, atlasD0):
```
# caa_stop SC10cmf
```
2. Unregister the cmf resource, by running the following command on CMFHOST (for example, atlasD0):
```
# caa_unregister SC10cmf
```
3. Use the sra edit command to set the CMF host in the SC database to be the name of the node running the cmfd daemon(s), as follows:
```
# sra edit
sra> sys
sys> edit system
Id      Description                                                Value
------------------------------------------------------------------------
.
.
.
[8  ]   Node running console logging daemon (cmfd)            atlasD0
.
.
.
Select attributes to edit, q to quit
eg. 1-5 10 15

edit? 8
Node running console logging daemon (cmfd)           [atlasD0]
new value? atlasms

Node running console logging daemon (cmfd)           [atlasms]
correct? [y|n] y
```
   The sra edit command then asks if you would like to restart or update the daemons — choose to update the daemons, as follows:
```
Modify SC database only (1), update daemons (2), restart daemons (3) [3]:
```
   Enter 2 to update the daemons so that they resynchronize with the updated SC database.
4. On the new CMFHOST (in this example, atlasms), start the daemon(s), as follows:
```
atlasms# /sbin/init.d/cmf start
```

14.19 Changing the CMF Host

By default, CMFHOST is set to one of the following values:

- On an HP AlphaServer SC system with a management server, CMFHOST is set to the management server hostname; for example, `atlasms`.
- On an HP AlphaServer SC system without a management server, CMFHOST is set to the hostname of Node 0; for example, `atlas0`.

For systems that have a management server, it may become necessary to temporarily move the CMFHOST to Node 0 (for example, if the management server fails). To do this, perform the following steps:

1. If CMF is running on the management server, stop CMF as described in Section 14.11 on page 14–13.
2. Use the `sra edit` command to set the CMF host in the SC database to be the hostname of Node 0, as follows:

```
# sra edit
sra> sys
sys> edit system
Id      Description                                                Value
------------------------------------------------------------------------

.
.
.
[8  ]   Node running console logging daemon (cmfd)           atlasms
.
.
.

Select attributes to edit, q to quit
eg. 1-5 10 15

edit? 8
Node running console logging daemon (cmfd)           [atlasms]
new value? atlas0

Node running console logging daemon (cmfd)           [atlas0]
correct? [y|n] y
```

   The `sra edit` command then asks if you would like to restart or update the daemons — choose to update the daemons, as follows:

```
Modify SC database only (1), update daemons (2), restart daemons (3) [3]:
```

   Enter 2 to update the daemons so that they resynchronize with the updated SC database.

3. On Node 0, start the CMF daemon(s) by running the following command:

```
atlas0# /sbin/init.d/cmf start
```

15
System Log Files

This chapter describes the log files in an HP AlphaServer SC system. These log files provide information about the state of the HP AlphaServer SC system.

This information in this chapter is arranged as follows:

- Log Files Overview (see Section 15.1 on page 15–2)
- LSF Log Files (see Section 15.2 on page 15–3)
- RMS Log Files (see Section 15.3 on page 15–3)
- System Event Log Files (see Section 15.4 on page 15–4)
- Crash Dump Log Files (see Section 15.5 on page 15–4)
- Console Log Files (see Section 15.6 on page 15–4)
- Log Files Created by sra Commands (see Section 15.7 on page 15–6)
- SCFS and PFS File-System Management Log Files (see Section 15.8 on page 15–7)

15.1 Log Files Overview

Table 15–1 describes various log files that are unique to the HP AlphaServer SC system.

Table 15–1 HP AlphaServer SC Log Files

File or Directory Name	Subsystem	Description
/var/sra/cmf.dated/*date*/*nodename*.log	sra/cmf	This file records text written to a node's console.
/var/sra/logs /var/sra/cmf.dated/current	sra/cmf	These are symbolic links to the /var/sra/cmf.dated/*current_date* directory.
/var/sra/adm/log/cmfd/cmfd_*hostname_port*.log	sra/cmf	These are the CMF daemon (cmfd) log files — one log file for each cmfd daemon.
/var/sra/sra.logd	sra	This directory contains a log file for each time that the sra command is used to reference a node. The log files are named sra.log.*N*.
/var/sra/diag	sra	This directory, created by the sra diag command, contains the System Event Analyzer log files.
/var/sra/adm/log/gxdaemons	sra	This is the log directory for the global execution (scrun) daemons.
/var/sra/adm/log/srad/srad.log	sra	The SRA daemon (srad) logs its output in this file.
/var/sra/adm/log/scmountd/srad.log	sra	This file contains a record of the srad daemon running a file system management script.
/var/sra/adm/log/scmountd/fsmgrScripts.log	scfsmgr/pfsmgr	This file contains information from the scripts run by the SCFS and PFS file system management system.
/var/sra/adm/log/scmountd/pfsmgr.*nodename*.log	pfsmgr	Records node-level actions on PFS file systems.
/var/sra/adm/log/scmountd/scmountd.log	scfsmgr/pfsmgr	This file contains information from the scmountd daemon, which manages SCFS and PFS file systems. This file is located on the management server (if used) or on the first domain (if not using a management server).
/var/lsf_logs	LSF	This directory contains the log files for LSF daemons.
/var/rms/adm/log	RMS	This directory contains the RMS log files. The log files for the RMS servers are located on the rmshost system.
/var/log/rmsmhd.log	RMS	This is the log file for the rmsmhd daemon.
/var/log/rmsd.*nodename*.log	RMS	This is the log file for the rmsd daemon.
/local/core/rms	RMS	RMS places core files in /local/core/rms/*resource_name*. See Section 5.5.9 on page 5–25 of this document.

15.2 LSF Log Files

LSF daemons use the `/var/lsf_logs` directory to log information about their activity. You should only need to examine the contents of these log files if the LSF system does not appear to be operating. The `/var/lsf_logs` directory contains the following types of log files, where *`name`* is the name of a host or a domain:

- `lim.log.`*`name`*

  This is the log file of `lim`, the Load Information Manager Daemon. The `lim` daemon reports the status of nodes or domains.

- `sbatchd.log.`*`name`*

  This is the log file of `sbatchd`, the Slave Batch Daemon. The `sbatchd` daemon is involved in the dispatch of jobs.

- `mbatchd.log.`*`name`*

  This is the log file of `mbatchd`, the Master Batch Daemon. The `mbatchd` daemon controls the dispatch of all jobs. Only one `mbatchd` daemon is active at a time.

- `mbschd.log.name`

  This is the log file of `mbschd`, the Master Batch Scheduling Daemon. The `mbschd` daemon controls the scheduling of all jobs. It assigns jobs to specific hosts and sends its scheduling decisions to the Master Batch Daemon (`mbatchd`). Only one `mbschd` daemon is active at a time.

- `pim.log.`*`name`*

  This is the log file of `pim`, the Process Information Manager Daemon. The `pim` daemon monitors jobs and processes, and reports runtime resources to the `sbatchd` daemon.

- `rla.log.`*`name`*

  This is the log file of `rla`, the RMS to LSF Adapter Daemon. The `rla` daemon allocates and deallocates RMS resources on behalf of the LSF system.

- `res.log.`*`name`*

  This is the log file of `res`, the Remote Execution Server Daemon. The `res` daemon executes the jobs once a placement decision has been made.

15.3 RMS Log Files

Check the following log files for error messages. Check these files on the `rmshost` system — that is, the management server (if using a management server) or Node 0 (if not using a management server):

- `/var/rms/adm/log/pmanager-`*`name`*`.log`

- /var/rms/adm/log/mmanager.log
- /var/rms/adm/log/eventmgr.log
- /var/rms/adm/log/swmgr.log

On each node, there are log files for the RMS daemons that run on that node. The log files are /var/log/rmsmhd.log and /var/log/rmsd.*nodename*.log. These files contain node-specific errors.

See Section 5.9.6 on page 5–73 for more information about the log files created by RMS.

15.4 System Event Log Files

The HP *Tru64 UNIX System Administration* guide provides details on mechanisms for logging system events, and details on maintaining those log files.

Note that, as with many other system files, /var/adm/syslog.dated and other files in /var/adm are CDSLs (Context-Dependent Symbolic Links). Therefore, you must perform the following tasks on each node in each domain:

- Review these files for errors.
- Run a cron job to maintain these files.

15.5 Crash Dump Log Files

Crashes generate log files in the /var/adm/crash directory. If crashes occur, you should follow the procedure described in Chapter 29 to report errors. Crash files can be quite large and are generated on a per-node basis. Therefore, maintenance may be required to ensure that the file system does not get full.

See the HP *Tru64 UNIX System Administration* guide for more details on administering crash dump files.

15.6 Console Log Files

Each node's console output is logged by CMF (*via* one or more cmfd daemons) to log files that are stored in the /var/sra/cmf.dated/*date* directory. Each of the following is a symbolic link to the /var/sra/cmf.dated/*latest* directory:

- /var/sra/logs
- /var/sra/cmf.dated/current

The log directory is updated twice a month, *via* a `crontab` entry that is added during the installation process. For more information about CMF log files, see Section 14.5 on page 14–8.

The following types of log files are stored in these directories:

- Console Log Files

  These log files contain output written to the console; there is one such file per node, named `nodename.log`. You should archive and retain log files after the directory is updated. Information in the console log files is often very useful in diagnosing problems.

- CMF Daemon Log File

  The `cmfd` daemon writes status information to the `/var/sra/adm/log/cmfd/cmfd_`*`hostname_port`*`.log` file. See Chapter 14 for more information about this file.

- Device Log Files

  If there are entries in the `/var/sra/cmf.conf.local` file, the CMF utility will provide equivalent console access, logging, and monitoring as for those nodes in the base system. This facility may be used to provide serial consoles for RAID controllers, Summit switches, and so on. If this capability is employed, these files should be reviewed occasionally to monitor for errors with those devices. These files are called `name.log`, where `name` corresponds to the entry in the `/var/sra/cmf.conf.local` file.

  See Chapter 14 for more information about the `cmf.conf.local` file.

All console log files should regularly be monitored for errors. The number and size of these files may grow over time. It is essential that the `/var` filesystem does not become full; therefore, after analysis of the log files, older directories should be deleted at regular intervals.

In particular, the `cmfd` daemon log file, `/var/sra/adm/log/cmfd/cmfd_`*`hostname_port`*`.log`, can grow very large over time. By default, the `cmfd` debug verbosity level is set to 2. The debug may be disabled by setting the `-d` option to 0 (zero) in the CMFOPTIONS specification in the `/sbin/init.d/cmf` startup script, as follows:

Replace: `CMFOPTIONS="-d 2 -t -b"`

with: `CMFOPTIONS="-d 0 -t -b"`

15.7 Log Files Created by sra Commands

Many different log files are produced by the various `sra` commands, as follows:

- `srad` daemon log files

  As documented in Section 16.1.6 on page 16–20 (and in Chapter 7 of the *HP AlphaServer SC Installation Guide*), the installation of the HP AlphaServer SC system is controlled by the installation daemon (`srad`). This is a hierarchical system of commands and controlling processes that automate every aspect of the installation process.

  There are two levels of `srad` daemon:

  – One system-level `srad` daemon runs on the management server (if the system has a management server) or on Node 0 (if the system does not have a management server)

  – A domain-level `srad` daemon runs on each domain

  If the system has a management server:

  – The system-level `srad` daemon writes to the `/var/sra/adm/log/srad/srad.log` file on the management server.

  – The domain-level `srad` daemons write to the `/var/sra/adm/log/srad/srad.log` file on each domain.

  If the system does not have a management server:

  – The system-level `srad` daemon writes to the `/var/sra/adm/log/srad/srad_system.log` file on Node 0.

  – The domain-level `srad` daemon for the first domain writes to the `/var/sra/adm/log/srad/srad_domain.log` file on Node 0.

  – The other domain-level `srad` daemons write to the `/var/sra/adm/log/srad/srad.log` file on each domain.

- `sra install` log files

  When installing an HP AlphaServer SC system, progress is recorded in the `srad` daemon log files on the management server and on each domain, as well as in the CMF log files for each node. However, the domain and domain-member aspects of the installation are logged in the `/cluster/admin/clu_create.log` and `/cluster/admin/clu_add_member.log` files. If there are problems with the domain or member aspects of the installation, these files may help you to diagnose the cause.

- `sra upgrade` log files

  When upgrading an HP AlphaServer SC system to Version 2.6, progress is recorded in the `srad` daemon log files on the management server and on each domain, as well as in the CMF log files for each node. The upgrade process generates other log files that may help you to diagnose the cause of problems during upgrade:

  - A log file in the /var/sra/sra.logd directory on the management server.
  - The file /var/adm/smlogs/sc_upgrade.log on the first member of each domain being upgraded.

    This file records and caches important information that is used during each domain upgrade, and records the stages that have successfully completed.

- sra diag log files

  When you use the sra diag command to examine a node, the results of the diagnosis are placed in the /var/sra/diag directory. The file name is *name*.sra_diag_results, where *name* is the name of the node. For example, the results file for atlas10 is as follows:
  /var/sra/diag/atlas10.sra_diag_results

  If you use System Event Analyzer to analyze the node, the report from the ca analyze command is placed in a file called *name*.ca_report. For example, the report for atlas10 is as follows:
  /var/sra/diag/atlas10.ca_report

- sra dbcheck log files

  The sra dbcheck command checks the internal consistency of the SC database. A log file with the name *currentdate*.log is created, and placed in the /var/sra/dbcheck directory.

- Log files from other (non-daemon-based) sra commands

  Some commands do not involve the srad daemons, and run in the foreground on the controlling terminal. The output from these commands is typically displayed on the controlling terminal, and spooled to a log file. Each time such an sra command is issued, a new sra.log.*N* file is generated in the /var/sra/sra.logd directory. These files log any problems with sra commands.

  Commands that generate log files in this directory include sra info, sra elancheck, and the act of probing for MAC addresses during sra setup. The following commands do not create a log file in the /var/sra/sra.logd directory: sra boot, sra edit, sra install, sra setup (except as described above), and sra shutdown.

See Chapter 16 for more information about sra commands.

15.8 SCFS and PFS File-System Management Log Files

The scmountd daemon manages SCFS and PFS file systems, and has several log files.

- /var/sra/adm/log/scmountd/scmountd.log
  This file contains the date and time at which the scmountd daemon attempted to run scripts on various domains. If a script failed to start, or timed out, the log file will record this fact.

- `/var/sra/adm/log/scmountd/srad.log`
  The `scmountd` daemon invokes scripts on a domain to mount or unmount file systems, as appropriate. The invocation of the scripts are managed by the `srad` daemon, and logged in the `/var/sra/adm/log/scmountd/srad.log` file.
- `/var/sra/adm/log/scmountd/fsmgrScripts.log`
  The scripts run by the SCFS and PFS file system management system write log data to the `/var/sra/adm/log/scmountd/fsmgrScripts.log` file. This log file contains data that is useful if a domain fails to mount or unmount a file system.
- `/var/sra/adm/log/scmountd/pfsmgr.`*`nodename`*`.log`
  Node-level actions on PFS file systems are recorded in the `/var/sra/adm/log/scmountd/pfsmgr.`*`nodename`*`.log` file.

16
The sra Command

This chapter provides information on the `sra` command.

The following commands are documented:

- sra (see Section 16.1 on page 16–2)
- sra edit (see Section 16.2 on page 16–22)
- sra-display (see Section 16.3 on page 16–39)

Note:

In the examples in this chapter, the value *www.xxx.yyy.zzz* represents the (site-specific) cluster alias IP address.

The `sra` commands are also described in the `sra(8)` reference page.

16.1 sra

Most of the sra commands are designed to operate on multiple nodes at the same time. The sra commands may be divided into the following groups:

- Installing the HP AlphaServer SC system

  These commands perform the initial installation of the HP AlphaServer SC system, or expand an existing HP AlphaServer SC system.

- Administering the HP AlphaServer SC system

  These commands perform actions on the system that are required for day-to-day system administration (boot, shutdown, and so on). These commands typically dispatch scripts (from the /usr/opt/sra/scripts directory) to perform an action on the designated nodes.

The sra command resides in the /usr/sra/bin directory. The install process creates a link to this command in the /usr/bin directory (the /usr/bin directory is included in your path during system setup).

All of the administration commands (boot, shutdown, and so on) must be run from the first node of the domain, or from the management server (if used).

To use the sra commands, you must be the root user. Some sra commands prompt for the root password, as follows:

```
# sra shutdown -nodes atlas2
Password:
```

By default, output from sra commands is written to three places:

- Standard output
- Piped to the sra-display program (see Section 16.3 on page 16–39) if the DISPLAY environment variable is set. You can disable this by including the option -display no in the command, as shown in the following example (where atlas is an example system name):

  ```
  # sra boot -nodes atlas2 -display no
  ```

- The /var/sra/sra.logd/sra.log.*n* file. To direct output to a different file, use the -log option, as shown in the following example (where atlas is an example system name):

  ```
  # sra boot -nodes atlas2 -log boot-log.txt
  ```

  To disable the output, use the -log /dev/null option.

  The sra setup and sra edit commands do not generate a log file.

  As you may generate a large number of SRA log files, HP recommends that you set up a cron job to archive or delete these files regularly.

You must specify the nodes on which the sra commands are to operate. This is specified by the -nodes, -domains, or -members option, as shown in the following examples:

```
-nodes atlas0
-nodes 'atlas0,atlas1,atlas10'
-nodes 'atlas[0,1,10]'
-nodes 'atlas[0-4,10-31]'
-domains 'atlasD[0-2]'
-domains 'atlasD[0,3]' -members 1
```

You must enclose the specification in quotes when using square brackets, to prevent the square brackets from being interpreted by the shell.

You can specify domains and nodes in abbreviated form, as follows:

```
-nodes 0-4,10-31
-domains 0-2
```

16.1.1 Nodes, Domains, and Members

Note:

The -domains, -nodes, and -members options interact differently in the sra command than they do in the scrun command. This section describes how these options interact in the sra command. For more information on how these options interact in the scrun command, see Chapter 12.

The -domains and -nodes options are independent of one another (see Example 16–1). The -nodes option is not a qualifier for the -domains option (see Example 16–2). However, the -members option is a qualifier for the -domains option (see Example 16–3). If you specify the -members option without the -domains option, the action is performed on the specified members in each domain.

Example 16–1 Node and Domain are Independent Options

```
-domains atlasD0,atlasD1 -nodes atlas96
```

Specifies all nodes in domains atlasD0 and atlasD1 (that is, nodes atlas0-63), and node atlas96.

Example 16–2 Node is Not a Qualifier for Domain

```
-domains atlasD0 -nodes atlas0
```

Specifies all nodes in atlasD0, not just atlas0. In this example, specifying -nodes atlas0 is redundant.

Example 16–3 Member is a Qualifier for Domain

```
-domains atlasD0,atlasD1 -member 1,2
```

Specifies members 1 and 2 in domains atlasD0 and atlasD1 (that is, nodes atlas0, atlas1, atlas32, and atlas33).

16.1.2 Syntax

The general syntax for `sra` commands is as follows:

```
# sra command_name options
```

for example:

```
# sra boot -nodes atlas2
```

To display help information for the `sra` commands, run the `sra help` command, as follows:

```
# sra help [command_name | -commands]
```

The `sra` commands can be divided into the following categories:

- Installation commands
  - `sra cookie`
  - `sra edit`
  - `sra install`
  - `sra install_info`
  - `sra rischeck`
  - `sra setup`
  - `sra upgrade`
  - `sra upgrade_info`

Note:

With the introduction of the `sra install` command, the following commands are now obsolete:

- `sra add_member`
- `sra clu_create`
- `sra install_unix`

- Diagnostic commands:
  - `sra diag`
  - `sra elancheck`
  - `sra ethercheck`
  - `sra dbcheck`
- Status commands:
  - `sra command_info`
  - `sra ds_who`
  - `sra info`
  - `sra srad_info`
  - `sra sys_info`

- Administration commands:
  - sra abort
  - sra boot
  - sra command
  - sra console
  - sra copy_boot_disk
  - sra dbget
  - sra delete_member
  - sra ds_configure
  - sra ds_logout
  - sra ds_passwd
  - sra halt_in
  - sra halt_out
  - sra kill
  - sra ping
  - sra power_off
  - sra power_on
  - sra reset
  - sra shutdown
  - sra switch_boot_disk
  - sra update_firmware

Table 16–1 provides the syntax for the sra commands, in alphabetical order.

Table 16–1 sra Command Syntax

Command	Syntax
abort	sra abort -command *command_id*
boot	sra boot {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*] [-delay *seconds*|streams|ready|lmf | -delaystring *boot_string*] [-device unix|*sra_device_name*] [-configure in|none] [-file vmunix|genvmunix|*other_kernel_file*] [bootable yes|no] [-sramon yes|no] [-single yes|no] [-init yes|no]
command	sra command {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] -command *command* [-display yes|no] [-log *filename*] [-width *width*] [-silent yes|no] [-limit yes|no] [-telnet yes|no] [-checkstatus yes|no]
command_info	sra command_info [-states *states*]
console	sra [console] {-c|-cl|-m|-ml *node*} | {-c|-cl *terminal_server*}

Table 16–1 sra Command Syntax

Command	Syntax
cookie	sra cookie [-enable yes|no]
copy_boot_disk	sra copy_boot_disk {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*] [-backup yes|no] [-telnet yes|no]
dbcheck	sra dbcheck [-tcl yes|no] [-postevent yes|no] [-backup yes|no] [-fix none|safe|all] [-log *filename*] [-d *database*] [-edir *expect_directory*] [-tdir *test_directory*] [-ddir *description_directory*]
dbget	sra dbget {name | cmf.host | ds.ip | ds.firstport | hwtype | num.nodes | cmf.home | cmf.port}
delete_member	sra delete_member {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-sramon yes|no]
diag	sra diag {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*] [-analyze yes|no] [-rtde *days*]
ds_configure	sra ds_configure {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...]
ds_logout	sra ds_logout {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-ts yes|no | -force yes|no]
ds_passwd	sra ds_passwd -server *terminal_server*
ds_who	sra ds_who {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-ts yes|no]
edit	sra edit
elancheck	sra elancheck {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*] [-test all|link|elan]
ethercheck	sra ethercheck {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*] [-stats yes|no] [-packet_num *number_of_packets(hex)*] [-packet_len *packet_length(hex)*] [-pass *number_of_passes*] [-target_enet *loopback_target_ethernet_address*] [-pattern all|zeros|ones|fives|tens|incr|decr]
halt_in	sra halt_in {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*]
halt_out	sra halt_out {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*]
help	sra help [*command_name* | -commands]

Table 16–1 sra Command Syntax

Command	Syntax
info	sra info {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*]
install	sra install {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*] [-sramon yes|no] [-sckit *sc_kit_path*] [-sysconfig *file*] [-unixpatch *UNIX_patch_path*] [-scpatch *sc_patch_path*] [-endstate *install_state*] [-redo *install_state*] [-nhdkit *NHD_kit_path*]
install_info	sra install_info {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*]
kill	sra kill -command *command_id*
ping	sra ping System|*domain_name*
power_off	sra power_off {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*]
power_on	sra power_on {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*]
reset	sra reset {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*] [-wait yes|no]
rischeck	sra rischeck [-nodes *nodes*|all | -domains *domains*|all | -members *members*] [...] [-display yes|no] [-log *filename*]
setup	sra setup
shutdown	sra shutdown {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*] [-reason *reason_for_shutdown*] [-flags s|h|r] [-configure out|in|none] [-reboot yes|no] [-bootable yes|no] [-single yes|no] [-sramon yes|no] [-force yes|no]
srad_info	sra srad_info [-system yes|no] [-domains *domains*|all] [-log *filename*] [-width *width*]
switch_boot_disk	sra switch_boot_disk {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] [-display yes|no] [-log *filename*] [-width *width*]
sys_info	sra sys_info -domains *domains*|all [-display yes|no] [-log *filename*] [-width *width*]
update_firmware	sra update_firmware {-nodes *nodes*|all | -domains *domains*|all | -members *members*} [...] -file *filename* [-display yes|no] [-log *filename*] [-width *width*] [-force yes|no]
upgrade	sra upgrade -domains *domains*|all -sckit *sc_kit_path* [-backupdev *device*] [-display yes|no] [-log *filename*] [-width *width*] [-endstate *upgrade_state*] [-redo *upgrade_state*][-rishost *RIS_server_name*] [-unixpatch *UNIX_patch_path*][-diskcap *required_disk_capacity*]

Table 16–1 sra Command Syntax

Command	Syntax
upgrade_info	sra upgrade_info -domains *domains*

16.1.3 Description

Table 16–2 describes the sra commands in alphabetical order.

Table 16–2 sra Command Description

Command	Description
abort	Abort an sra command. See Chapter 11 of the *HP AlphaServer SC Installation Guide*.
boot	Boots the specified nodes. If the -file option is specified, it boots that file. The default values are -width 8 -init no -file vmunix -delay lmf -single no -configure none -sramon yes -device *node_boot_disk*. See Chapter 2 of this guide.
command	Runs the specified command on the specified remote system. The default values are -width 32 -silent no -limit yes -telnet no -checkstatus no.
command_info	Displays information on sra commands. Use the -states option to specify which command states to display. The states are: Allocated, Unallocated, Success, Error, and Abort.
console	Connects to or monitors the console of the specified node, or connects to the specified terminal server.
cookie	Determines whether the mSQL daemons are enabled. If used with the -enable option, enables or disables the mSQL daemons. See Section 3.6 on page 3–12 of this guide.
copy_boot_disk	Builds (or rebuilds) either the primary or backup boot disk: if you are booted off the primary boot disk, this command will build the backup boot disk; if you are booted off the backup boot disk, this command will build the primary boot disk. The default values are -backup yes -width 8 -telnet no. See Section 2.8.5 on page 2–14 of this guide.
dbcheck	Checks the internal integrity of the SC database. The default values are -tcl no -postevent no -backup yes -fix none -log /var/sra/dbcheck/currentdate.log -d sc_database -edir /usr/opt/srasysman/dbcheck/exp -tdir /usr/opt/srasysman/dbcheck/check -ddir /usr/opt/srasysman/dbcheck/desc See Section 3.7 on page 3–13 of this guide.

Table 16–2 sra Command Description

Command	Description
dbget	Displays the same information as the `sra edit` command about the following system attributes: • System name — `sra dbget name` • Node running console logging daemon (`cmfd`) — `sra dbget cmf.host` • First DECserver IP address — `sra dbget ds.ip` • First port on the terminal server — `sra dbget ds.firstport` • Hardware type — `sra dbget hwtype` • Number of nodes — `sra dbget num.nodes` • CMF home directory — `sra dbget cmf.home` • CMF port number — `sra dbget cmf.port` See Chapter 14 of this guide.
delete_member	Deletes members from the domain. The default value is `-sramon yes`. See Section 21.6 on page 21–15 of this guide.
diag	Performs an SRM/RMC check if a node is at the SRM prompt. If the system is up, this command analyzes the `binary.errlog` file and generates a report. The default values are `-width 8 -analyze yes -rtde 60`. See Chapter 28 of this guide.
ds_configure	Configures nodes on the terminal server. See Section 14.7 on page 14–9 of this guide.
ds_logout	Disconnects users or nodes from CMF or the terminal server. The default values are `-ts no -force no`. See Section 14.11 on page 14–13 of this guide.
ds_passwd	Sets the password on the terminal server, and updates the entry in the SC database. See Section 14.9 on page 14–12 of this guide.
ds_who	Displays information on user connections to the specified nodes (or all nodes, if none are specified). Specify `-ts yes` to display terminal server connections instead of user connections. The default value is `-ts no`. See Section 14.4 on page 14–5 of this guide.
edit	Displays or modifies the contents of the SC database (interactive mode). See Section 16.2 on page 16–22 of this guide.
elancheck	Checks the HP AlphaServer SC Interconnect network. The default values are `-width 8 -test all`. See Section 21.4 on page 21–6 of this guide.

Table 16–2 sra Command Description

Command	Description
ethercheck	Checks ethernet connectivity by running a live network loopback test. The default values are -width 1 -packet_num 3e8 -packet_len 40 -pass 10 -stats no -pattern all -target_enet *this_host's_management_network_ethernet_address* See Section 21.4 on page 21–6 of this guide.
halt_in	Halts the specified nodes. The default value is -width 32. See Section 2.14 on page 2–19 of this guide.
halt_out	Releases the halt on the specified nodes. The default value is -width 32. See Section 2.14 on page 2–19 of this guide.
help	Displays short help. If *command_name* is specified, displays short help about the specified command. If -commands is specified, lists all of the sra commands documented in (this) Table 16–2. If neither *command_name* nor -commands is specified, displays short help about all of the sra commands listed in (this) Table 16–2.
info	Displays information about the current state of the specified nodes. The default value is -width 32. See Chapter 28 of this guide.
install	RIS-installs Tru64 UNIX on the specified nodes; configures networks, NFS, DNS, NIS, NTP, and mail; installs Tru64 UNIX patch kits; installs HP AlphaServer SC software; installs HP AlphaServer SC patch kits; creates domains, and adds members. The default values are -sramon yes -endstate Member_Added. See Chapter 7 of the *HP AlphaServer SC Installation Guide*.
install_info	Displays information about the installation status of the specified nodes. See Chapter 7 of the *HP AlphaServer SC Installation Guide*.
kill	Kills an sra command. This is similar to the sra abort command, but does not perform the node cleanup.
ping	Sends a wake-up message to the specified SRA daemon.
power_off	Powers off the system on the specified nodes. The default value is -width 32. See Section 2.15 on page 2–19 of this guide.
power_on	Powers on the system on the specified nodes. Note that the power button on the Operator Control Panel (OCP) has precedence. The default value is -width 32. See Section 2.15 on page 2–19 of this guide.

Table 16–2 sra Command Description

Command	Description
`reset`	Resets the specified nodes. The default values are `-wait no -width 32`. Note that if the `-wait yes` option is specified, the default width for the `sra reset` command changes from 32 to 8. See Section 2.13 on page 2–19 of this guide.
`rischeck`	Checks the RIS configuration on the RIS server. The default value is `-nodes all`.
`setup`	Sets up a cluster environment, and builds the SC database. See Chapters 5 and 6 of the *HP AlphaServer SC Installation Guide* for more information.
`shutdown`	Shuts down the specified nodes. If a node is already halted, no action is taken. The default values are `-width 8 -reason 'sra shutdown' -reboot no -single no -sramon yes -flags h -configure out`. There is no default value for the `-bootable` option. See Chapter 2 of this guide.
`srad_info`	Checks the status of the SRA daemons. The default values are `-system yes -domains all -width 32`. See Section 29.17 on page 29–37 of this guide.
`switch_boot_disk`	Toggles between the primary boot disk and the backup boot disk. The specified node(s) must be shut down and must be at the SRM prompt before running this command. The default value is `-width 8`. See Section 2.8.3 on page 2–12 of this guide.
`sys_info`	Checks the status of the nodes, at a domain level. The default value is `-width 32`.
`update_firmware`	Updates firmware on the designated nodes. *`filename`* is a `bootp` file and should be placed in the `/tmp` directory. The default values are `-force no -width 8`. See Section 21.12 on page 21–34 of this guide.
`upgrade`	Upgrades the specified domains to the latest version of the HP AlphaServer SC software. The default values are as follows: • `-width 8` • `-endstate upgraded` • If a management server is used, the default for the `-rishost` option is the management server. If no management server is used, the default for the `-rishost` option is the cluster alias of the first domain. See Chapter 4 of the *HP AlphaServer SC Installation Guide* for more information.
`upgrade_info`	Displays information on the upgrade status of each targeted domain.

16.1.4 Options

Table 16–3 describes the sra options in alphabetical order.

You can abbreviate the sra options. You must specify enough characters to distinguish the option from the other sra options, as shown in the following example:

```
atlasms# sra install -d 0
ambiguous argument -d: must be one of -domains -display

atlasms# sra install -do 0
UNIX patch kit not specified: no UNIX patch will be applied
```

Table 16–3 sra Options

Option	Description
-analyze	Specifies that System Event Analyzer should automatically be run for the user (if appropriate). The default value is yes. This option is only used with the sra diag command.
-backup	This option is used with two sra commands. The default value varies according to the command, as follows: • When used with the sra copy_boot_disk command, specifies that the /local and /tmp file systems should be backed up. The default value is yes. • When used with the sra dbcheck command, specifies whether a backup is taken of the entire database. By default, the -backup option is set to yes, but the database is only backed up if the -fix option is set to safe, or all, and inconsistencies are found during the check.This is because when the -fix option is set to safe, or all, the database integrity check attempts to fix inconsistencies and makes changes to the database. In this case, it is important to have a backup of the database so that you can roll it back after the integrity check, if necessary. You can override this behavior by setting the -backup option to no.
-backupdev	Specifies the name of the backup device, as a UNIX device special file name (the path is not needed). If specified, the upgrade process will write a backup of the cluster root (/), /usr, and /var partitions, and each node's boot disk, to this device. This option can only be used (and is mandatory) when the sra upgrade command is being used to back up the domains. There is no default value for this option.
-bootable	Specifies whether the nodes are bootable or not. Valid values are yes or no. There is no default value for this option. This option is used with the sra boot and sra shutdown commands.
-c	Specifies that you wish to connect to the specified node or terminal server by opening a new window. This option is only used with the sra console command.
-checkstatus	Specifies that if the exit status of the specified command (which runs inside a csh shell) is non-zero, the sra command command should fail. The default value is no. This option is only used with the sra command command.

Table 16–3 sra Options

Option	Description
-cl	Specifies that you wish to connect to the specified node or terminal server from the current (local) window. This option is only used with the `sra console` command.
-command	Specifies the command to be run on the nodes, or the command to be aborted. This option is only used with the `sra command` and `sra abort` commands.
-commands	Specifies that the `sra help` command should list all of the `sra` commands documented in Table 16–2. This option is only used with the `sra help` command.
-configure	Specifies whether the nodes should be configured in, configured out, or left as they are (`none`). The default value varies according to the command. This option is used with the `sra boot` command (default: `-configure none`), and with the `sra shutdown` command (default: `-configure out`).
-d	Specifies the database that is to be checked. If the `-d` *database* option is not specified, the system database is checked by default. This option is used only with the `sra dbcheck` command
-ddir	Specifies the directory where the description files are held. Description files are used to translate the output of the database integrity check into English descriptions and suggested solutions. The default directory where the description files are held is `/usr/opt/srasysman/dbcheck/desc`. If you wish, you can specify a different directory. This option is used only with the `sra dbcheck` command.
-delay	Specifies how long to wait before booting the next node. See also `-delaystring` below. Can be specified as a number of seconds, or as the string “streams”, or as the string “ready”, or as the string “lmf”: • If you specify `-delay 60`, the `sra` command will boot a node and wait for 60 seconds before starting to boot the next node. • If you specify `-delay streams`, the `sra` command will wait until the string “streams” is encountered in the boot output, before starting the next boot. The boot process outputs the string “streams” just after the node has joined the domain. • If you specify `-delay ready`, the `sra` command will wait until the string “ready” is encountered in the boot output, before starting the next boot. The boot process outputs the string “ready” when the node is fully booted. • If you specify `-delay lmf`, the `sra` command will wait until the string “lmf” is encountered in the boot output, before starting the next boot. The boot process outputs the string “lmf” when the LMF licenses are loaded; typically, this happens after all of the disks have been mounted. The default value is `lmf`. This option is only used with the `sra boot` command.

Table 16–3 sra Options

Option	Description
-delaystring	Specifies that the sra command should boot a node and wait until the specified string *boot_string* is encountered in the boot output, before starting to boot the next node. See also -delay above. If neither -delaystring nor -delay is specified, the default used is -delay lmf. This option is only used with the sra boot command**.**
-diskcap	Specifies the disk capacity required for the upgrade. There is no default value. This option is used only with the sra upgrade command.
-device	Specifies the disk (by SRM name) from which the specified nodes should be booted. You can specify unix to boot from the Tru64 UNIX disk — this value is only valid for the lead node in each domain, as these are the only nodes that have a Tru64 UNIX disk. The default value for each node is the boot disk for that specified node, as recorded in the SC database. This option is used only with the sra boot command.
-display[1]	Specifies whether the command output should be piped to the standard output *via* the sra-display command. The default value is yes. This option is used with most sra commands.
-domains[1]	Specifies the domains to operate on. [2-4] specifies domains 2 to 4 inclusive. The default value for the sra srad_info command is all. This option may be used with most sra commands.
-edir	Specifies the directory for sourcing in the "expect" scripts that sra dbcheck needs from another directory. The default is /usr/opt/srasysman/dbcheck/exp. If you wish, you can specify a different directory. This option is used only with the sra dbcheck command.
-enable	Specifies whether to enable the mSQL daemons. There is no default value. This option is only used with the sra cookie command.
-endstate	Specifies the state at which the installation or upgrade process should stop. This option is used with the sra install and the sra upgrade commands. The default value for the sra install command is Member_Added. The default value for the sra upgrade command is Upgraded.
-file	Specifies a file to boot. The default value for the sra boot command is vmunix; there is no default for the sra update_firmware command. This option is used only with the sra boot and sra update_firmware commands.

Table 16–3 sra Options

Option	Description
-fix	Specifies whether the sra dbcheck command attempts to fix some, all, or none of the inconsistencies it finds. The valid values are safe, all, or none. • If safe is specified, the database integrity check attempts to fix inconsistencies it determines are safe to fix, and the output shows whether the inconsistency has been fixed, remains unfixed (the attempt to fix the inconsistency was unsuccessful), or should be fixed manually. • If all is specified, the database integrity check attempts to fix all identified inconsistencies, and the output shows whether the inconsistency has been fixed, remains unfixed (the attempt to fix the inconsistency was unsuccessful), or should be fixed manually. • If none is specified, the database integrity check will not attempt to fix any inconsistencies, but will report each inconsistency, and whether it is safe (can be safely fixed automatically),+ or manual (should be fixed manually). If the -fix option is not specified, the default is none. This option is used only with the sra dbcheck command.
-flags	Specifies how to shut down the system. • If you specify -flags s, the sra command will execute the stop entry point of the run-level transition scripts in /sbin/rc0.d/[K*nn_name*], /sbin/rc2.d/[K*nn_name*], and /sbin/rc3.d/[K*nn_name*] (for example, the stop entry point of /sbin/rc0.d/K45syslog). The run level at which the sra shutdown command is invoked determines which scripts are executed: – If the current run level is level 3 or higher, the K*nn_name* scripts from all three directories are run. – If the run level is 2, only scripts from /sbin/rc0.d and /sbin/rc2.d are run. – If the run level is 1, only scripts from /sbin/rc0.d are run. • If you specify -flags h, the sra command will shut down and halt the system using a broadcast kill signal. • If you specify -flags r, the sra command will shut down the system using a broadcast kill signal, and automatically reboot the system. The default value is h. This option is only used with the sra shutdown command.
-force	When used with the sra update_firmware command, specifies whether to install an earlier revision of the firmware than the currently installed version. When used with the sra ds_logout command, specifies whether to telnet directly to the terminal server, bypassing CMF. When used with the sra shutdown command, specifies whether the node is to be shutdown when it will result in losing quorum. The default value for each of these commands is no. This option is only used with the sra update_firmware, sra ds_logout, and sra shutdown commands.
-init	Specifies that the hardware should be reset before booting. The default value is no. This option is only used with the sra boot command.

Table 16–3 sra Options

Option	Description
-limit	Specifies that the command should stop if the output exceeds 200 lines. The default value is yes. This option is only used with the sra command command.
-log[1]	This option is used with most sra commands. The use and default value varies according to the command, as follows: • When used with the sra dbcheck command, a log file with the name *currentdate*.log is created. By default, the log file is placed in the /var/sra/dbcheck directory. • For other sra commands, specifies the location of the command output. The default value is /var/sra/sra.logd/sra.log.*n*.
-m	Specifies that you wish to monitor the specified node by opening a new window. This option is only used with the sra console command.
-members[1]	Specifies the members to operate on. [2-30] specifies members 2 to 30 inclusive. The -members option qualifies the -domains option; if the -domains option is not specified, the action is performed on the specified members in each domain. The -members option may be used with most sra commands.
-ml	Specifies that you wish to monitor the specified node from the current (local) window. This option is only used with the sra console command.
-nhdkit	Specifies that the sra command should install the New Hardware Delivery software on the specified nodes. This option is only used with the sra install command.
-nodes[1]	Specifies the nodes to operate on. [2-30] specifies Nodes 2 to 30 inclusive. This option may be used with most sra commands.
-packet_len	Specifies, in hex, the length of each packet sent during the ethernet check. The default value is 40. This option is only used with the sra ethercheck command.
-packet_num	Specifies, in hex, the number of packets to send during each pass in the ethernet check. The default value is 3e8. This option is only used with the sra ethercheck command.
-pass	Specifies the number of times to send *packet_num* packets during the ethernet check. The default value is 10. This option is only used with the sra ethercheck command.
-pattern	Specifies the byte pattern of each packet sent during the ethernet check. The default value is all. This option is only used with the sra ethercheck command.
-postevent	Specifies whether an event is generated when the sra dbcheck command runs. Valid values are yes or no. If the -postevent option is not specified, the default is no. This option is only used with the sra dbcheck command.
-reason	Specifies why the nodes were shut down. The default value is 'sra shutdown'. This option is only used with the sra shutdown command.

Table 16–3 sra Options

Option	Description
-reboot	Specifies that the nodes should be rebooted after shutdown. The default value is no. This option is only used with the sra shutdown command.
-redo	Allows you to repeat one or more steps in the installation or upgrade process. Specify the *install_state* or *upgrade_state* that is reached by the step you want to repeat. For example, during an upgrade, if the process has reached the UNIX_Patched state and you want to repeat the last step, specify -redo UNIX_Patched. Note the following restrictions on both installation and upgrade states: • *install_state* or *upgrade_state* must specify a state that is the same as or earlier than the current state of the node. Note the following restrictions on installation states: • The CLU_Added and Bootp_Loaded states do not apply to the lead members of domains. • The states from UNIX_Installed to CLU_Create inclusive do not apply to non-lead members. See Chapter 7 of the *HP AlphaServer SC Installation Guide* for a list of all possible installation states. • Once a node reaches the Bootp_Loaded state, you cannot specify -redo Bootp_Loaded for that node — specify -redo CLU_Added instead. There is no default value for this option. This option is only used with the sra install and sra upgrade commands.
-rishost	Specifies the name of the RIS server. This option is only used with the sra upgrade command. If a management server is used, the default is the management server. If no management server is used, the default is the cluster alias of the first domain.
-rtde	Specifies the period (number of days) for which events should be analyzed, starting with the current date and counting backwards. The default value is 60. This option is only used with the sra diag command.
-sckit	Specifies that the sra command should install all mandatory HP AlphaServer SC subsets on the specified nodes. This option is only used with the sra install command.
-scpatch	Specifies that the sra command should install the HP AlphaServer SC patch kit software on the specified nodes. This option is only used with the sra install command.
-server	Specifies the terminal server whose password you wish to change. This option is only used with the sra ds_passwd command.
-silent	Specifies that the command should run without displaying the command output. The default value is no. This option is only used with the sra command command.

Table 16–3 sra Options

Option	Description
-single	Boots or shuts down the specified nodes in single user mode. The default value is no. This option is only used with the sra boot and sra shutdown commands.
-sramon	Specifies whether details about the progress of the sra command (gathered by the sramon command) should be displayed. The default value is yes. This option is only used with the sra boot, sra delete_member, sra install, and sra shutdown commands. See Section 16.1.6 on page 16–20 for more information about this option.
-stats	Specifies whether to generate network statistics during the ethernet check. The default value is no. This option is only used with the sra ethercheck command.
-sysconfig	Specifies that the configureUNIX phase of the installation process should merge the contents of *file* into the existing /etc/sysconfigtab and /etc/.proto..sysconfigtab files. There is no default value for this option. This option is only used with the sra install command.
-system	Specifies whether to check the System daemon. The default value is yes. This option is only used with the sra srad_info command.
-target_enet	Specifies the target ethernet address to which packets should be sent during the ethernet check. The default value is the management network ethernet address of the host on which the command is being run (loopback). This option is only used with the sra ethercheck command.
-tcl	Specifies whether the output is a formatted output or a summary. Valid values are yes or no. • If yes is specified, the sra dbcheck command produces a formatted output that can be parsed by another script. • If no is specified, the sra dbcheck command produces a summary output of four or five lines for each inconsistency found. If the -tcl option is not specified, the default is no. This option is only used with the sra dbcheck command.
-tdir	Specifies the directory where the test scripts are held. See/usr/opt/srasysman/dbcheck/check/README for test naming conventions. The default directory where the test scripts are held is /usr/opt/srasysman/dbcheck/check. If you wish, you can specify a different directory. This option is only used with the sra dbcheck command.
-telnet	Specifies that the sra command should connect to the specified remote system using telnet, instead of using the default connection method (that is, *via* the cmfd daemon to the node's serial console port). For general commands (for example, stopping or starting RMS), the -telnet option is usually much faster than cmfd. The -telnet option requires that the specified node be up, and running on the network. Output from this command is not logged in /var/sra/logs, but does appear in /var/sra/sra.logd. The default value is no. This option is only used with the sra command and sra copy_boot_disk commands.

Table 16–3 sra Options

Option	Description
-test	Specifies which tests to run: • If you specify -test elan, the sra command will run the elanpcitest script, to test the ability of a node to access its HP AlphaServer SC Elan adapter card through the PCI bus. • If you specify -test link, the sra command will run the elanlinktest script, to test the ability of a node to reach the HP AlphaServer SC 16-port switch card to which it is directly connected *via* the link cable. • If you specify -test all, the sra command will run both the elanpcitest script and the elanlinktest script. The default value is all. This option is only used with the sra elancheck command.
-ts	Specifies that the command should operate on the connection between the terminal server and the CMF daemon, rather than the connection between the user and the CMF daemon. The default value is no. This option is only used with the sra ds_logout and sra ds_who commands.
-unixpatch	Specifies that the sra command should install the Tru64 UNIX patch kit software on the specified nodes. This option is only used with the sra install and sra upgrade commands.
-wait	Specifies that the sra command should wait for an SRM prompt before completing. The default value is no. This option is only used with the sra reset command.
-width[1]	Specifies the number of nodes to target in parallel. This option is used with most sra commands. The default value varies according to the command, as follows: • sra command, sra halt_in, sra halt_out, sra info, sra power_off, sra power_on, sra reset[2], sra srad_info, and sra sys_info (default: -width 32) • sra boot, sra copy_boot_disk, sra diag, sra elancheck, sra shutdown, sra switch_boot_disk, and sra update_firmware, sra upgrade (default: -width 8) • sra ethercheck (default: -width 1)

[1]This option is used with most sra commands, except as indicated.

[2]The default width for the sra reset command is 32. However, if the -wait option is specified, the default width for the sra reset command changes to 8.

16.1.5 Error Messages From sra console Command

When you attempt to connect to a node's console using the `sra console -c` command, you may get one of the following errors:

- `Console-Busy Port already in use`

  This means that someone is already using `sra console -c` to connect to this node's console. This is normal operation for `sra console -c`.

- `cmf-Port This node's port is not connected`

  This means that when the `cmfd` daemon was started, it was unable to connect to the appropriate port of the terminal server. There are two possible causes for this:

  – A `cmfd` daemon (possibly on another node) is already running (and has connected to each port). You can distinguish this condition from other causes by seeing if this error applies to all ports or just to one port. If the error applies to all ports, an existing `cmfd` daemon is the probable cause. Having more than one `cmfd` daemon on the system can only occur by misconfiguring the system.

  – Sometimes the terminal server fails to close connections even though `cmfd` has dropped its connections. You can fix this by logging out the terminal-server ports (for more details, see Section 14.11 on page 14–13).

- `cmf-Fail The cmf port proxy does not appear to be running`

  The `cmfd` daemon is not responding — probably because it is not running. To restart `cmf`, follow the procedure in Section 14.11 on page 14–13.

16.1.6 The sramon Command

The `sramon` command monitors the progress of an `sra` command, by checking the relevant `sc_nodes` entries in the SC database. At a predefined interval, the `sramon` command checks all of the `sc_nodes` entries that are currently being updated by an `sra` command, and displays the contents of the `status` field if that value has changed since the last check.

To change the frequency, run the `sramon` command to display the `sramon` GUI (see Figure 16–1) and specify a new refresh rate.

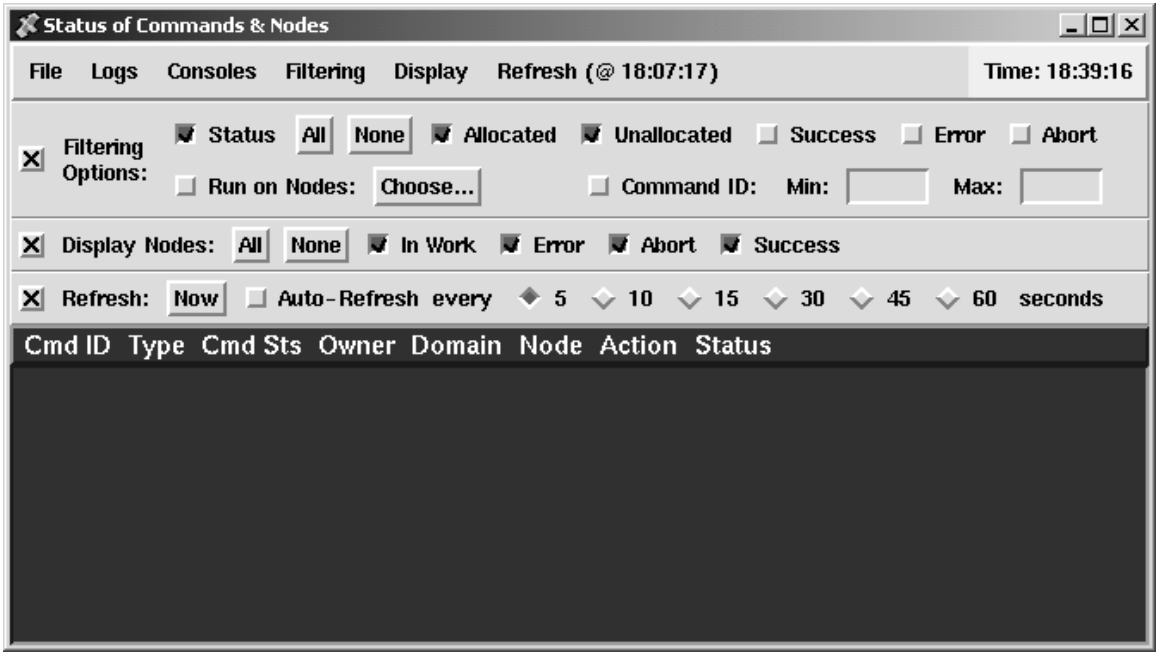

Figure 16–1 sramon GUI

The default refresh rate is 5 seconds. Consequently, `sramon` may occasionally "miss" certain node status changes — if a particular node has more than one 'info' message in a 5-second period, `sramon` will display only the last status message. The only way to see the "missed" lines is to review the `srad.log` file, in the `/var/sra/adm/log/srad` directory.

When reviewing the `srad` log file, remember that two `srad` daemons are involved in most commands. Consider the `install` command: the `srad` daemon on the management server (or Node 0, if not using a management server) manages all of the steps up to and including the installation of the HP AlphaServer SC patches, so all data related to these steps is logged in the `/var/sra/adm/log/srad/srad.log` file on the management server (or Node 0). However, the `srad` daemon on the domain manages the domain creation and adding members, so the data related to these steps is stored in the `srad` log file on the domain.

For certain `sra` commands, the `-sramon` option specifies whether details about the progress of the `sra` command (gathered by the `sramon` command) should be displayed, as follows:

- If you specify `-sramon yes`, the progress details are displayed.
- If you specify `-sramon no`, the progress details are not displayed.

The -sramon yes option intersperses the progress details with the sra command output. To display the sra command output in one window and the progress details in another window, perform the following steps:

1. Start the sra command in the first window, specifying the -sramon no option, as follows:
   ```
   # sra command ... -sramon no ...
   ```
   where *command* is boot, delete_member, install, or shutdown.
2. In the second window, monitor the progress of the command started in step 1, as follows:
   ```
   # sramon command_id
   ```
   where *command_id* is the command ID for *command*.

   If you cannot locate *command_id* in the output in step 1, use the rmsquery command to identify the command ID, as follows:
   ```
   # rmsquery -v "select type,domain,node,status,command_id \
   from sc_command where type='command' and status<>'Success' \
   order by status,domain,node,command_id" | grep -i allocate
   ```

Note:

If no records are returned and *command* has not completed, then either an error has occurred or *command* has been aborted. To identify which, rerun the above rmsquery command, substituting error or abort for allocate.

For more information about the sramon command, see Chapter 7 of the *HP AlphaServer SC Installation Guide*, and the sramon(8) reference page.

16.2 sra edit

During the installation of the HP AlphaServer SC system (see Chapters 5 and 6 of the *HP AlphaServer SC Installation Guide*), the sra setup command builds the SC database, which is a SQL database of information about the components of the HP AlphaServer SC system. The sra edit command allows you to modify the SC database in a controlled way.

You can also use the sra edit command to verify that the sra setup command has run correctly, and to complete the database setup if there are problems probing nodes for hardware information during setup.

During the installation process, several configuration files are built from information stored in the SC database (for example, the /etc/hosts file, and the RIS database). These files depend directly on the SC database. The sra edit command can rebuild these files if necessary. The sra edit command can also force an update of these files, and restart daemons if necessary.

16.2.1 Usage

The `sra edit` command is an interactive command — when you invoke the `sra edit` command, you start an interactive session, as follows:

```
# sra edit
sra>
```

Table 16–4 lists the `sra edit` subcommands.

Table 16–4 sra edit Subcommands

Subcommand	Description
`help`	Show command help.
`node`	Enter the node submenu.
`sys`	Enter the system submenu.
`exit`	Exit from the `sra edit` interactive session.

Table 16–5 provides a quick reference to the `sra edit` command. Each subcommand is discussed in more detail in later sections.

Table 16–5 sra edit Quick Reference

Subcommand	Option	Attributes
`help`	—	—
`node`	`help`	—
	`add`	*nodes*
	`del`	*nodes*
	`show`	`host [`*nodes*`]`
		node `[set|all]`
	`edit`	*node*
	`quit`	—
`sys`	`help`	—

	`show`	`system`
		`clu(ster) [`*name*`]`
		`im(age) [`*name*`]`
		`ip [`*name*`]`
		`ds [`*name*`]`

Table 16–5 sra edit Quick Reference

Subcommand	Option	Attributes
	edit	system clu(ster) [*name*] im(age) [*name*] ip [*name*] ds [*name*]
	update	hosts cmf ris *nodes* ds *nodes*
	add	ds [auto] im(age) *name*
	del	ds *name* im(age) *name*
	quit	–
exit	–	–

16.2.2 Node Submenu

To enter the Node submenu, enter **node** at the sra> prompt, as follows:

```
sra> node
```

Table 16–6 lists the Node submenu options.

Table 16–6 Node Submenu Options

Option	Description
help	Show command help.
add	Add a node to the SC database.
del	Delete a node from the SC database.
show	Show database attributes for a given node.
edit	Edit database attributes for a given node.
quit	Return to the sra prompt; that is, the top-level sra edit menu.

16.2.2.1 Show Node Attributes

Use the `show` command to show the names of nodes in the SC database, or to display a table of key-value pairs for a specific node. The syntax is as follows:

```
node> show host[names] | show hostname [set|all]
```

Note:

Most of the information about a node is derived, by a rule set, from system attributes.

The `set` option displays all node attributes that have been explicitly set, not derived; for example, the node's hardware ethernet address.

The `all` option displays all node attributes. This is the default option.

In Example 16–4 to Example 16–6 inclusive, `atlas` is a four-node system.

Example 16–4

```
node> show host
atlas0
atlas1
atlas2
atlas3
```

Example 16–5

```
node> show atlas1

Id      Description                                          Value
-----------------------------------------------------------------

[0  ]   Hostname                                             atlas1              *
[1  ]   DECserver name                                       atlas-tc1           *
[2  ]   DECserver internal port                              2                   *
[3  ]   cmf host for this node                               atlasms
[4  ]   cmf port number for this node                        6500
[5  ]   TruCluster memberid                                  2                   *
[6  ]   Cluster name                                         atlasD0             *
[7  ]   Hardware address (MAC)                               00-00-F8-1B-2E-BA
[8  ]   Number of votes                                      0                   *
[9  ]   Node specific image_default                          0                   *
[10 ]   Elan Id                                              1
[11 ]   Bootable or not                                      1                   *
[12 ]   Hardware type                                        ES45                *
[13 ]   Current Installation State                           Member_Added
[14 ]   Desired Installation State                           Member_Added
[15 ]   Current Installation Action                          Complete:wait
[16 ]   Command Identifier                                   391
[17 ]   Node Status                                          Finished
[19 ]   im00:Image Role                                      boot
```

```
[20 ]   im00:Image name                              first                *
[21 ]   im00:UNIX device name                        dsk0                 *
[22 ]   im00:SRM device name                         dka0                 *
[23 ]   im00:Disk Location (Identifier)
[24 ]   im00:default or not                          yes
[31 ]   im00:swap partition size (%)                 15
[33 ]   im00:tmp partition size (%)                  42
[35 ]   im00:local partition size (%)                43
[38 ]   im01:Image Role                              boot
[39 ]   im01:Image name                              second
[40 ]   im01:UNIX device name                        dsk1
[41 ]   im01:SRM device name                         dka100
[42 ]   im01:Disk Location (Identifier)
[43 ]   im01:default or not                          no
[50 ]   im01:swap partition size (%)                 15
[52 ]   im01:tmp partition size (%)                  42
[54 ]   im01:local partition size (%)                43
[57 ]   ip00:Interface name                          man
[58 ]   ip00:Hostname suffix                         atlas1               *
[59 ]   ip00:Network address (IP)                    10.128.0.2           *
[60 ]   ip00:UNIX device name                        ee0
[61 ]   ip00:SRM device name                         eia0
[62 ]   ip00:Netmask                                 255.255.0.0
[63 ]   ip00:Cluster Alias Metric
[65 ]   ip01:Interface name                          ext
[66 ]   ip01:Hostname suffix                         atlas1-ext1          *
[67 ]   ip01:Network address (IP)                                         #
[68 ]   ip01:UNIX device name                        alt0
[69 ]   ip01:SRM device name                         eib0
[70 ]   ip01:Netmask                                 255.255.255.0
[71 ]   ip01:Cluster Alias Metric
[73 ]   ip02:Interface name                          ics
[74 ]   ip02:Hostname suffix                         atlas1-ics0          *
[75 ]   ip02:Network address (IP)                    10.0.0.2             *
[76 ]   ip02:UNIX device name                        ics0
[77 ]   ip02:SRM device name
[78 ]   ip02:Netmask                                 255.255.255.0
[79 ]   ip02:Cluster Alias Metric
[81 ]   ip03:Interface name                          eip
[82 ]   ip03:Hostname suffix                         atlas1-eip0          *
[83 ]   ip03:Network address (IP)                    10.64.0.2            *
[84 ]   ip03:UNIX device name                        eip0
[85 ]   ip03:SRM device name
[86 ]   ip03:Netmask                                 255.255.0.0
[87 ]   ip03:Cluster Alias Metric                    16

* = default generated from system
# = no default value exists
------------------------------------------------------------------
```

The character in the right-hand column indicates the source of the node attribute:

- `*` indicates that the attribute has been derived from system attributes by the rule set.

- # indicates that no value exists for this key in the SC database.
- A blank field indicates that the attribute has been specifically set for this node.

Example 16–6

Example 16–6 displays only those attributes that have been explicitly set in the SC database.

```
node> show atlas1 set

Id      Description                                     Value
-----------------------------------------------------------------

[7  ]   Hardware address (MAC)                          00-50-8B-E3-1F-F6
[10 ]   Elan Id                                         1

* = default generated from system
# = no default value exists

-----------------------------------------------------------------
```

16.2.2.2 Add Nodes to, and Delete Nodes from, the SC Database

Use the Node submenu commands `add` and `delete` to add nodes to, and delete nodes from, the SC database. The syntax is as follows:

```
node> add nodes
node> del nodes
```

Note:

Only a limited number of nodes may be added to the SC database using this command — the number of nodes added should not result in a new domain. If the number of nodes to be added would result in a new domain, build the SC database using the `sra setup` command.

Example 16–7

In Example 16–7, an 8-node system named `atlas` is expanded to a 16-node system. As a domain may contain up to 32 nodes, it will not be necessary to create a new domain; therefore, the Node submenu add command may be used.

```
node> add atlas[8-15]
```

The `add` command performs the following actions:

- Updates the terminal server
- Updates the console logging daemon configuration file, and restarts the daemon
- Probes each node for its hardware ethernet address, and updates the RIS database.

At the completion of this command, the SC database will be ready to add members to the domain.

The `delete` command is provided for symmetry, and may be used to undo any changes made when adding nodes.

16.2.2.3 Edit Node Attributes

Use the Node submenu `edit` command to set, or probe for, node-specific SC database attributes.

Example 16–8

In Example 16–8, we use the `sra edit` command to set the node's hardware ethernet address in the SC database (for example, after replacing a faulty ethernet adapter).

```
node> edit atlas1
Id      Description                                          Value
-----------------------------------------------------------------

[0  ]   Hostname                                             atlas1                *
[1  ]   DECserver name                                       atlas-tc1             *
[2  ]   DECserver internal port                              2                     *
[3  ]   cmf host for this node                               atlasms
[4  ]   cmf port number for this node                        6500
[5  ]   TruCluster memberid                                  2                     *
[6  ]   Cluster name                                         atlasD0               *
[7  ]   Hardware address (MAC)                               00-00-F8-1B-2E-BA
[8  ]   Number of votes                                      0                     *
[9  ]   Node specific image_default                          0                     *
[10 ]   Elan Id                                              1
[11 ]   Bootable or not                                      1                     *
[12 ]   Hardware type                                        ES45                  *
[13 ]   Current Installation State                           Member_Added
[14 ]   Desired Installation State                           Member_Added
[15 ]   Current Installation Action                          Complete:wait
[16 ]   Command Identifier                                   391
[17 ]   Node Status                                          Finished
[19 ]   im00:Image Role                                      boot
[20 ]   im00:Image name                                      first                 *
[21 ]   im00:UNIX device name                                dsk0                  *
[22 ]   im00:SRM device name                                 dka0                  *
[23 ]   im00:Disk Location (Identifier)
[24 ]   im00:default or not                                  yes
[31 ]   im00:swap partition size (%)                         15
[33 ]   im00:tmp partition size (%)                          42
[35 ]   im00:local partition size (%)                        43
[38 ]   im01:Image Role                                      boot
[39 ]   im01:Image name                                      second
[40 ]   im01:UNIX device name                                dsk1
[41 ]   im01:SRM device name                                 dka100
[42 ]   im01:Disk Location (Identifier)
[43 ]   im01:default or not                                  no
[50 ]   im01:swap partition size (%)                         15
[52 ]   im01:tmp partition size (%)                          42
[54 ]   im01:local partition size (%)                        43
```

```
[57 ]   ip00:Interface name                          man
[58 ]   ip00:Hostname suffix                         atlas1              *
[59 ]   ip00:Network address (IP)                    10.128.0.2          *
[60 ]   ip00:UNIX device name                        ee0
[61 ]   ip00:SRM device name                         eia0
[62 ]   ip00:Netmask                                 255.255.0.0
[63 ]   ip00:Cluster Alias Metric
[65 ]   ip01:Interface name                          ext
[66 ]   ip01:Hostname suffix                         atlas1-ext1         *
[67 ]   ip01:Network address (IP)                                        #
[68 ]   ip01:UNIX device name                        alt0
[69 ]   ip01:SRM device name                         eib0
[70 ]   ip01:Netmask                                 255.255.255.0
[71 ]   ip01:Cluster Alias Metric
[73 ]   ip02:Interface name                          ics
[74 ]   ip02:Hostname suffix                         atlas1-ics0         *
[75 ]   ip02:Network address (IP)                    10.0.0.2            *
[76 ]   ip02:UNIX device name                        ics0
[77 ]   ip02:SRM device name
[78 ]   ip02:Netmask                                 255.255.255.0
[79 ]   ip02:Cluster Alias Metric
[81 ]   ip03:Interface name                          eip
[82 ]   ip03:Hostname suffix                         atlas1-eip0         *
[83 ]   ip03:Network address (IP)                    10.64.0.2           *
[84 ]   ip03:UNIX device name                        eip0
[85 ]   ip03:SRM device name
[86 ]   ip03:Netmask                                 255.255.0.0
[87 ]   ip03:Cluster Alias Metric                    16
* = default generated from system
# = no default value exists

------------------------------------------------------------------

Select attributes to edit, q to quit
eg. 1-5 10 15

edit? 7

enter a new value, probe or auto
auto  = generate value from system
probe = probe hardware for value

Hardware address (MAC)              [00-00-F8-1B-2E-BA] (set)
new value? probe

info Connected through cmf
info Connected through cmf

Hardware address (MAC)              [00-00-F8-1B-2E-BA] (probed)

correct? [y|n] y

Remote Installation Services (RIS) should be updated
Update RIS ? [yes]: y
Gateway for subnet 10 is 10.128.0.1
Setup RIS for host atlas1
```

Note that for this attribute we chose to probe for the value. The `probe` option is valid for the following node attributes:

- im00: SRM device name
- im01: SRM device name
- ip00: UNIX device name
- ip00: SRM device name
- ip00: Hardware address (MAC)

Use the `auto` option to reset a node attribute to the default value (as derived from the system attributes by the rule set).

Use the `quit` command at the `node>` prompt to exit the Node submenu and return to the `sra>` prompt; that is, the main `sra edit` menu.

16.2.3 System Submenu

To enter the System submenu, enter **`sys`** at the `sra>` prompt:

```
sra> sys
sys>
```

In addition to node-specific attributes, the SC database categorizes information about the HP AlphaServer SC system as follows:

- System
- Cluster
- Image
- Network
- Terminal Server

The System submenu is designed to manage database attributes in these categories.

Table 16–7 lists the System submenu options.

Table 16–7 System Submenu Options

Option	Description
`help`	Show command help.
`show`	Show system attributes.
`edit`	Edit system attributes.
`update`	Update system files; restart daemons.

Table 16–7 System Submenu Options

Option	Description
add	Add a terminal server or image to the SC database.
del	Remove a terminal server or image from the SC database.
quit	Return to the sra prompt; that is, the top-level sra edit menu.

16.2.3.1 Show System Attributes

Use the show command to show the attributes of the system. The syntax is as follows:

```
sys> show system|widths
sys> show clu[ster]|im[age]|ip|ds [name]
```

where *name* is the name of a cluster, image, network interface (ip), or terminal server (ds).

Example 16–9

To show the systemwide attributes, use the show system command, as shown in Example 16–9.

```
sys> show system
Id      Description                                                Value
-----------------------------------------------------------------
[0  ]   System name                                                atlas
[1  ]   SC database revision                                       2.5.4
[2  ]   Connect method                                             cmf
[3  ]   First DECserver IP address                                 10.128.100.01
[4  ]   First port on the terminal server                          6
[5  ]   Hardware type                                              ES45
[6  ]   Default image                                              0
[7  ]   Number of nodes                                            8
[8  ]   Node running console logging daemon (cmfd)                 atlasms
[9  ]   cmf home directory                                         /var/sra/
[10 ]   cmf port number                                            6500
[11 ]   cmf port number increment                                  2
[12 ]   cmf max nodes per daemon                                   256
[13 ]   cmf max daemons per host                                   4
[14 ]   Allow cmf connections from this subnet                     255.255.0.0
[15 ]   cmf reconnect wait time (seconds)                          60
[16 ]   cmf reconnect wait time (seconds) for failed ts            1800
[17 ]   Software selection                                         1
[18 ]   Software subsets
[19 ]   Kernel selection                                           3
[20 ]   Kernel components                                          2 3 4 11
[21 ]   DNS Domain Name                                            site-specific
[22 ]   DNS server IP list                                         site-specific
[23 ]   DNS Domains Searched
[24 ]   NIS server name list                                       site-specific
[25 ]   NTP server name list                                       site-specific
[26 ]   MAIL server name                                           site-specific
[27 ]   Default Internet route IP address                          site-specific
```

```
[28 ]   Management Server name                                  atlasms
[29 ]   Use swap, tmp & local on alternate boot disk            yes
[30 ]   SRA Daemon (srad) port number                           6600
[31 ]   SRA Daemon Monitor host
[32 ]   SRA Daemon Monitor port number
[33 ]   SC Database setup and ready for use                     1
[34 ]   IP address of First Top level switch (rail 0)           10.128.128.128
[35 ]   IP address of First Node level switch (rail 0)          10.128.128.1
[36 ]   IP address of First Top level switch (rail 1)           10.128.129.128
[37 ]   IP address of First Node level switch (rail 1)          10.128.129.1
[38 ]   Port used to connect to the scmountd on MS              5555
[39 ]   /etc/clua_default metric                                2
-----------------------------------------------------------------
```

Example 16–10

To show the `-width` values, use the `show widths` command, as shown in Example 16–10.

```
sys> show widths
Id   Description                                            Value
-----------------------------------------------------------------

[0  ] RIS Install Tru64 UNIX                                 32
[1  ] Configure Tru64 UNIX                                   32
[2  ] Install Tru64 UNIX patches                             32
[3  ] Install AlphaServer SC Software Subsets                32
[4  ] Install AlphaServer SC Software Patches                32
[5  ] Install New Hardware Delivery Subsets                  32
[6  ] Create a One Node Cluster                              32
[7  ] Add Member to Cluster                                  8
[8  ] RIS Download the New Members Boot Partition            8
[9  ] Boot the New Member using the GENERIC Kernel           8
[10 ] Boot                                                   8
[11 ] Shutdown                                               8
[12 ] Cluster Shutdown                                       8
[13 ] Cluster Boot to Single User Mode                       8
[14 ] Cluster Boot Mount Local Filesystems                   4
[15 ] Cluster Boot to Multi User Mode                        32

-----------------------------------------------------------------
```

Example 16–11

To find the object name[s], run the command without specifying a name, as shown in Example 16–11.

```
sys> show clu
valid clusters are [atlasD0 atlasD1 atlasD2 atlasD3 atlasD4 atlasD5]

sys> show image
valid images are [unix-first cluster-first boot-first boot-second cluster-second
gen_boot-first]

sys> show ip
valid ips are [eip ics ext man]
```

```
sys> show ds
valid DECservers are [atlas-tc1 atlas-tc2 atlas-tc3 atlas-tc4]
```

Example 16–12

To show an object's attributes, specify that object's name, as shown in Example 16–12 and Example 16–13.

```
sys> show clu atlasD0

Id    Description                    Value
-----------------------------------------------------------------
[0  ]   Cluster name                                    atlasD0
[1  ]   Cluster alias IP address                        site-specific
[2  ]   Domain Type                                     fs
[3  ]   First node in the cluster                       0
[4  ]   I18n partition device name
[5  ]   SRA Daemon Port Number                          6600
[6  ]   File Serving Partition                          0
[7  ]   Number of Cluster IC Rails                      1
[8  ]   Current Upgrade State                           Unupgrade
[9  ]   Desired Upgrade State                           Unupgrade
[10 ]   Image Role                                      cluster
[11 ]   Image name                                      first
[12 ]   UNIX device name                                dsk3
[13 ]   SRM device name
[14 ]   Disk Location (Identifier)                      IDENTIFIER=1
[15 ]   root partition size (%)                         5
[16 ]   root partition                                  b
[17 ]   usr partition size (%)                          50
[18 ]   usr partition                                   g
[19 ]   var partition size (%)                          45
[20 ]   var partition                                   h
[21 ]   Image Role                                      cluster
[22 ]   Image name                                      second
[23 ]   UNIX device name                                dsk5
[24 ]   SRM device name
[25 ]   Disk Location (Identifier)                      IDENTIFIER=3
[26 ]   root partition size (%)                         5
[27 ]   root partition                                  b
[28 ]   usr partition size (%)                          50
[29 ]   usr partition                                   g
[30 ]   var partition size (%)                          45
[31 ]   var partition                                   h
[32 ]   Image Role                                      gen_boot
[33 ]   Image name                                      first
[34 ]   UNIX device name                                dsk4
[35 ]   SRM device name
[36 ]   Disk Location (Identifier)                      IDENTIFIER=2
[37 ]   default or not
[38 ]   swap partition size (%)                         30
[39 ]   tmp partition size (%)                          35
[40 ]   local partition size (%)                        35
```

```
[41 ]   Image Role                                      unix
[42 ]   Image name                                      first
[43 ]   UNIX device name                                dsk2
[44 ]   SRM device name
[45 ]   Disk Location (Identifier)
[46 ]   root partition size (%)                         10
[47 ]   root partition                                  a
[48 ]   usr partition size (%)                          35
[49 ]   usr partition                                   g
[50 ]   var partition size (%)                          35
[51 ]   var partition                                   h
[52 ]   swap partition size (%)                         20
[53 ]   swap partition                                  b
-----------------------------------------------------------------
```

Example 16–13

```
sys> show ds atlas-tc1
Id      Description                                     Value
-----------------------------------------------------------------

[0  ]   DECserver name                                  atlas-tc1
[1  ]   DECserver model                                 DECserver900
[2  ]   number of ports                                 32
[3  ]   IP address                                      10.128.100.01

-----------------------------------------------------------------
```

16.2.3.2 Edit System Attributes

Use the System submenu `edit` command to set, or probe for, systemwide attributes.

Note:

Changing some systemwide attributes will be reflected in node-specific attributes *via* the rule set.

The syntax is as follows:

```
sys> edit system
sys> edit clu[ster]|im[age]|ip|ds [name]
```

where *name* is the name of a cluster, image, network interface (`ip`), or terminal server (`ds`).

Example 16–14

In Example 16–14, the console logging daemon by default listens on port 6500 for user connections. We change this port using the `edit system` command.

```
sys> edit system

Id      Description                                          Value
-----------------------------------------------------------------
[0  ]   System name                                             atlas
[1  ]   SC database revision                                    2.5.EFT4
```

```
[2  ]   Connect method                                      cmf
[3  ]   First DECserver IP address                          10.128.100.01
[4  ]   First port on the terminal server                   6
[5  ]   Hardware type                                       ES45
[6  ]   Default image                                       0
[7  ]   Number of nodes                                     8
[8  ]   Node running console logging daemon (cmfd)          atlasms
[9  ]   cmf home directory                                  /var/sra/
[10 ]   cmf port number                                     6500
[11 ]   cmf port number increment                           2
[12 ]   cmf max nodes per daemon                            256
[13 ]   cmf max daemons per host                            4
[14 ]   Allow cmf connections from this subnet              255.255.0.0
[15 ]   cmf reconnect wait time (seconds)                   60
[16 ]   cmf reconnect wait time (seconds) for failed ts     1800
[17 ]   Software selection                                  1
[18 ]   Software subsets
[19 ]   Kernel selection                                    3
[20 ]   Kernel components                                   2 3 4 11
[21 ]   DNS Domain Name                                     site-specific
[22 ]   DNS server IP list                                  site-specific
[23 ]   DNS Domains Searched
[24 ]   NIS server name list                                site-specific
[25 ]   NTP server name list                                site-specific
[26 ]   MAIL server name                                    site-specific
[27 ]   Default Internet route IP address                   site-specific
[28 ]   Management Server name                              atlasms
[29 ]   Use swap, tmp & local on alternate boot disk        yes
[30 ]   SRA Daemon (srad) port number                       6600
[31 ]   SRA Daemon Monitor host
[32 ]   SRA Daemon Monitor port number
[33 ]   SC Database setup and ready for use                 1
[34 ]   IP address of First Top level switch (rail 0)       10.128.128.128
[35 ]   IP address of First Node level switch (rail 0)      10.128.128.1
[36 ]   IP address of First Top level switch (rail 1)       10.128.129.128
[37 ]   IP address of First Node level switch (rail 1)      10.128.129.1
[38 ]   Port used to connect to the scmountd on MS          5555
[39 ]   /etc/clua_default metric                            2

------------------------------------------------------------------

Select attributes to edit, q to quit
eg. 1-5 10 15

edit? 10
cmf port number                         [6500]
new value? 6505

cmf port number                         [6505]
correct? [y|n] y

You have modified fields that affect the console logging
system. The SC database will be updated. In addition you
may chose to update (ping) the daemons to reload from
the modified database, or restart the daemons.
```

```
Modify SC database only (1), update daemons (2), restart daemons (3) [3]:3
Finished adding nodes to CMF table
Finished updating nodes in CMF table
CMF reconfigure: succeeded
```

Example 16–15

In Example 16–15, we change the IP addresses of the HP AlphaServer SC management network.

```
sys> show ip man
Id      Description                                     Value
-----------------------------------------------------------------
[0  ]   Interface name                                  man
[1  ]   Hostname suffix
[2  ]   Network address (IP)                            10.128.0.1
[3  ]   UNIX device name                                ee0
[4  ]   SRM device name                                 eia0
[5  ]   Netmask                                         255.255.0.0
[6  ]   Cluster Alias Metric

-----------------------------------------------------------------

sys> edit ip man
Id      Description                                     Value
-----------------------------------------------------------------
[0  ]   Interface name                                  man
[1  ]   Hostname suffix
[2  ]   Network address (IP)                            10.128.0.1
[3  ]   UNIX device name                                ee0
[4  ]   SRM device name                                 eia0
[5  ]   Netmask                                         255.255.0.0
[6  ]   Cluster Alias Metric

-----------------------------------------------------------------

Select attributes to edit, q to quit
eg. 1-5 10 15

edit? 2
network address (IP)                    [10.128.0.1]
new value? 10.128.10.1

network address (IP)                    [10.128.10.1]
correct? [y|n] y
/etc/hosts should be updated
Update /etc/hosts ? [yes]:y
Updating /etc/hosts...
```

Each node's network address will be affected by this change *via* the rule set.

16.2.3.3 Update System Files and Restart Daemons

Use the System submenu `update` command to rebuild system configuration files and restart daemons if necessary. This ensures that those system files and daemons that depend on the SC database are up to date. The syntax of this command is as follows:

```
sys> update hosts | cmf | ris [nodes] | ds [nodes] | diskid filename
```

Example 16–16

In Example 16–16, we rebuild the `/etc/hosts` file from the SC database.

```
sys> update hosts
Updating /etc/hosts...
```

Note:

The `update hosts` command modifies only the section between `#sra start` and `#sra end` in the `/etc/hosts` file. Any local host information is preserved. The `rmshost` alias is also preserved. The `rmshost` alias is not stored in the SC database.

Example 16–17

The console logging daemon (`cmfd`) reads configuration information from the SC database, as described in Chapter 14. In Example 16–17, we update this information in the SC database.

```
sys> update cmf

You have modified fields that affect the console logging
system. The SC database will be updated. In addition you
may chose to update (ping) the daemons to reload from
the modified database, or restart the daemons.

Modify SC database only (1), update daemons (2), restart daemons (3) [3]:3
Finished adding nodes to CMF table
Finished updating nodes in CMF table
CMF reconfigure: succeeded
```

Option (1) will rebuild the `sc_cmf` table in the SC database.
Option (2) will force the `cmfd` daemons to reread the information from the SC database.
Option (3) will stop and restart the `cmfd` daemons.

Example 16–18

In Example 16–18, we update the RIS database.

```
sys> update ris all
Gateway for subnet 10 is 10.128.0.1
Setup RIS for host atlas0
Setup RIS for host atlas1
Setup RIS for host atlas2
⋮
```

Note:

HP recommends that the RIS server be set up on a management server (if used).

If the RIS server is set up on Node 0 and you run the `update ris all` command, you will get a warning similar to the following (where `atlas` is an example system name):

```
The following nodes do not have the hardware ethernet address
set in the database, and were consequently not added to RIS

atlas0
```

Ignore this warning.

Example 16–19

In Example 16–19, we update the connection from a node to the terminal server.

```
sys> update ds atlas0
Info: connecting to terminal server atlas-tc1 (10.128.100.1)
Configuring node atlas0    [port = 1]
```

Example 16–20

In Example 16–20, we update the Disk Location Identifiers.

```
sys> update diskid /var/sra/disk_id
Disk Location Identifiers loaded successfully
```

In this example, `/var/sra/disk_id` is an example file containing the necessary information in the required format. For more information about the `update diskid` command, see Chapter 6 of the *HP AlphaServer SC Installation Guide*.

16.2.3.4 Add or Delete a Terminal Server, Image, or Cluster

Use the System submenu `add` command to add a second boot image, or a terminal server, or a cluster, to the SC database. The syntax of this command is as follows.

`sys> add ds [auto] | im[age] name | cluster start_node [ip_address]`

Use the System submenu `del` command to delete an image, terminal server, or cluster from the SC database. The syntax of this command is as follows.

`sys> del ds name | im[age] name | cluster`

The `del cluster` command deletes the highest-numbered cluster in the HP AlphaServer SC system.

Example 16–21

The `sra setup` command asks the administrator if they require an alternate boot disk. If an alternate boot disk is configured at this point, the SC database will contain two image entries; by default, these images are named `boot-first` and `boot-second`. In Example 16–21, we run the System submenu `add` command to add an alternate boot disk to the SC database without re-running the `sra setup` command.

```
sys> show im
valid images are [unix-first cluster-first cluster-second boot-first
gen_boot-first]
sys> add image boot-second
sys> show im
valid images are [unix-first cluster-first cluster-second boot-first boot-second
gen_boot-first]
```

You can now edit the second image entry and set the SRM boot device and UNIX disk name.

You can remove the alternate boot disk (image) with the following command:

```
sys> del im boot-second
```

16.3 sra-display

When you run an `sra` command, a graphical interface displays the progress of the command (if the DISPLAY environment variable has been set). This interface is called `sra-display`.

`sra-display` scans the data, looking for informational messages. It displays the first word (operation) in the informational message, and prefixes each line with the current date and time. This can be used to monitor the progress of `sra` on a large number of nodes.

The output of the `sra` command is also saved in a log file (the default log file is `sra.log.n`, but you can specify an alternative filename). This allows you to save results for later analysis.

For example, the following command will boot all nodes in the first four domains (where `atlas` is an example system name):

```
# sra boot -nodes 'atlas[0-127]'
Log file is /var/sra/sra.logd/sra.log.5
```

The `sra-display` command can be used to replay previously-saved results, as follows:

```
# cat sra.log.5 | /usr/bin/sra-display
```

Sample output from the `sra-display` command is shown in Figure 16–2.

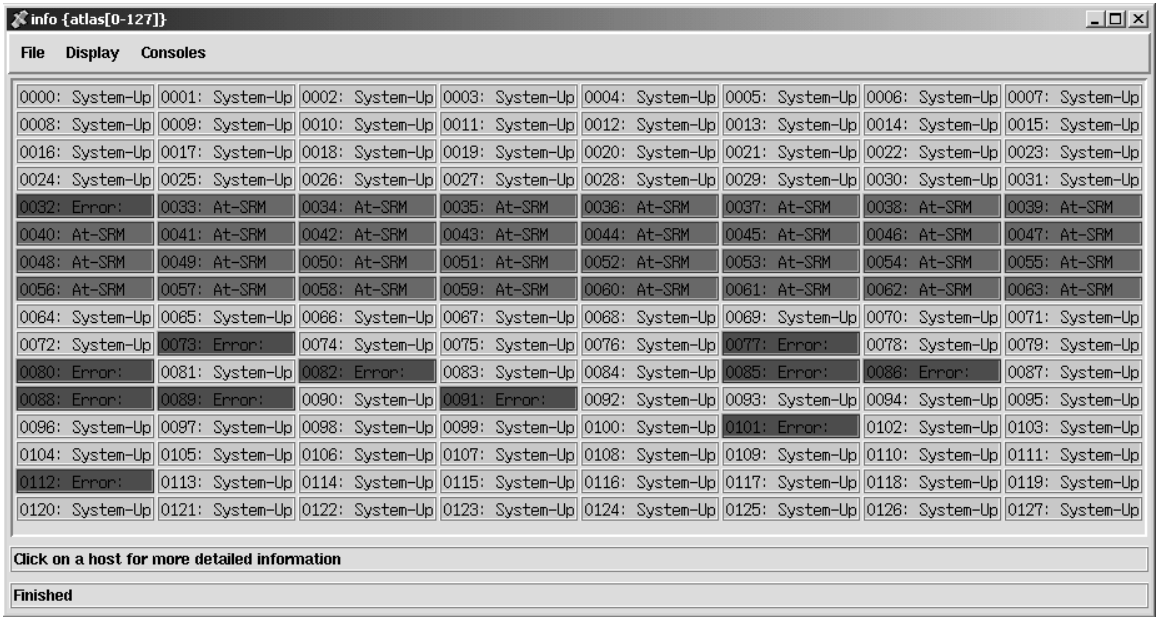

Figure 16–2 sra-display Output

Part 2:

Domain Administration

17

Overview of Managing Domains

This chapter provides an overview of the commands and utilities that you can use to manage domains. It also describes the differences between using these commands and utilities on a domain, and using them on a standalone system.

This chapter is organized as follows:

- Commands and Utilities for Domains (see Section 17.1 on page 17–2)
- Commands Not Supported in HP AlphaServer SC (see Section 17.2 on page 17–3)
- Tru64 UNIX Commands that are Different in a Domain (see Section 17.3 on page 17–4)

17.1 Commands and Utilities for Domains

Table 17–1 lists the commands that are specific to managing domains[1]; some of these commands are TruCluster Server commands, while others are specific to HP AlphaServer SC systems. The commands listed here are used to manipulate or query aspects of a domain. You can find descriptions for these commands in the reference pages.

Table 17–1 Domain Commands

Function	Command	Description
Create and configure domain members	`sra install`, which calls `clu_create(8)` and `clu_add_member(8)`	Creates an initial domain member on an HP AlphaServer SC system, and adds new members to the domain.
	`sra delete_member`, which calls `clu_delete_member(8)`	Deletes a member from a domain.
	`clu_check_config(8)`	Checks that the domain is correctly configured.
	`clu_get_info`	Gets information about a domain and its members.
Define and manage highly available applications	`caad(8)`	Starts the CAA daemon.
	`caa_balance(8)`	Balances application resources across members.
	`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 domain member to another.
	`caa_report(8)`	Reports availability statistics for application resources.
	`caa_start(8)`	Starts a highly available application registered with the CAA daemon.
	`caa_stat(1)`	Provides status of applications registered with CAA.
	`caa_stop(8)`	Stops a highly available application.

1. For information on how the terms “domain” and “cluster” are used in this Guide, see the Terms Used in This Guide section of the Preface on page xxxv.

Table 17–1 Domain Commands

Function	Command	Description
	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 symbolic 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 physical file system in a domain.

17.2 Commands Not Supported in HP AlphaServer SC

Table 17–2 describes commands that are not supported in HP AlphaServer SC systems.

Table 17–2 Features Not Supported in HP AlphaServer SC

Feature	Comments
Archiving bttape(8)	The bttape utility is not supported in domains. For more information about backing up and restoring files, see Section 24.7 on page 24–45.
LSM volrootmir(8) volunroot(8)	The volrootmir and volunroot commands are not supported in domains. See Chapter 25 for details and restrictions on configuring LSM in an HP AlphaServer SC environment.
mount(8)	Network File System (NFS) loopback mounts are not supported. For more information, see Chapter 22. Other commands that run through mountd, such as 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 the /etc/exports.aliases file.
Prestoserve presto(8) dxpresto(8) prestosetup(8) prestoctl_svc(8)	Prestoserve is not supported in HP AlphaServer SC Version 2.6.

Table 17–2 Features Not Supported in HP AlphaServer SC

Feature	Comments

Network Management
`routed(8)`
`netsetup(8)`
	The `routed` daemon is not supported in HP AlphaServer SC Version 2.6 systems. The cluster alias requires `gated`. When you create the initial domain member, `sra install` configures `gated`. When you add a new domain member, `sra install` propagates the configuration to the new member.
	For more information about routers, see Section 22.3 on page 22–7.
	The `netsetup` command has been retired. Do not use it.

Dataless Management Services (DMS)	DMS is not supported in an HP AlphaServer SC environment. A domain can be neither a DMS client nor a server.
`sysman_clone(8)` `sysman -clone(8)`	Configuration cloning and replication is not supported in a domain. Attempts to use the `sysman -clone` command in a domain fail and return the following message: `Error: Cloning in a cluster environment is not supported.`

17.3 Tru64 UNIX Commands that are Different in a Domain

The following tables list Tru64 UNIX commands and subsystems that have options specific to a domain, or that behave differently in a domain 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 17–3 describes the differences in commands and utilities that manage file systems and storage.

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

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

Table 17–3 File Systems and Storage Differences

Command	Differences
`addvol(8)`	In a single system, you cannot use `addvol` to expand `root_domain`. However, in a domain, 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.

Table 17–3 File Systems and Storage Differences

Command	Differences
df(8)	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.
LSM voldisk(8) volencap(8) volmigrate(8) volreconfig(8) volstat(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 domain, the volencap swap command places the swap devices for an individual domain 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 domain to complete the encapsulation. The volreconfig command is called during the domain 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. See Chapter 25 for details and restrictions on configuring LSM in an HP AlphaServer SC environment.
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 domain 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. An MFS file system mount, whether read-only or read/write, is 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 24.9 on page 24–48.

Tru64 UNIX Commands that are Different in a Domain

Table 17–4 describes the differences in commands and utilities that manage networking.

Table 17–4 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 domain. A BIND client configuration is domainwide — all domain members have the same client configuration. Do not configure any member of a domain as a BIND server — HP AlphaServer SC Version 2.6 supports configuring the system as a BIND client only.
	For more information, see Section 22.4 on page 22–8.

Broadcast messages
`wall(1)`
`rwall(1)`
	The `wall -c` command sends messages to all users on all members of the domain. 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 domain members.
	In a domain, a `clu_wall` daemon runs on each domain member to receive `wall -c` messages. If a `clu_wall` daemon is inadvertently stopped on one of the domain members, restart the daemon by using the `clu_wall -d` command.

Dynamic Host Configuration Protocol (DHCP) `joinc(8)`	DHCP is not explicitly configured in HP AlphaServer SC Version 2.6. However, `joind` is enabled if the first node in a domain is configured as a RIS server (see Chapters 5 and 6 of the *HP AlphaServer SC Installation Guide*). A domain can be a DHCP server, but domain members cannot be DHCP clients. Do not run `joinc` in a domain. Domain members must use static addressing.
`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 domain.

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 22.8 on page 22–22 for details.
	The `mailstats` command returns mail statistics for the domain member on which it was run. The mail statistics file, `/usr/adm/sendmail/sendmail.st`, is a member-specific file; each domain member has its own version of the file.

Network File System (NFS)
`nfsconfig(8)`
`rpc.lockd(8)`
`rpc.statd(8)`
	Use the `sysman nfs` command or the equivalent `nfsconfig` command to configure NFS. Do not use the `nfssetup` command; it was retired in Tru64 UNIX Version 5.0.
	Domain members can run client versions of `lockd` and `statd`. Only one domain member runs an additional `lockd` and `statd` pair for the NFS server. These are invoked with the `rpc.lockd -c` and `rpc.statd -c` commands. The server `lockd` and `statd` are highly available and are under the control of CAA.
	For more information, see Chapter 22.

Table 17–4 Networking Differences

Command	Differences

Network Management
`netconfig(8)`
`gated(8)`
	If, as recommended, you configured networks during domain configuration, `gated` was configured as the routing daemon. See the *HP AlphaServer SC Installation Guide* for more information. If you later run `netconfig`, you must select `gated`, not `routed`, as the routing daemon. The `routed` daemon is not supported in HP AlphaServer SC Version 2.6 systems.
	For more information about routers, see Section 22.3 on page 22–7.

Network Interface Failure Finder (NIFF)
`niffconfig(8)`
`niffd(8)`
	For NIFF to monitor the network interfaces in the domain, `niffd`, the NIFF daemon, must run on each domain member.

Network Information Service (NIS) `nissetup(8)`	HP AlphaServer SC Version 2.6 supports configuring the system as a NIS slave only — do not configure the system as a NIS master. For more information about configuring NIS, see Section 22.7 on page 22–21.
Network Time Protocol (NTP) `ntp(1)`	All domain members require time synchronization. NTP meets this requirement. Each domain member is automatically configured as an NTP peer of the other members. You do not need to do any NTP configuration. For more information, see Section 22.5 on page 22–9.

Table 17–5 describes the differences in printing management.

Table 17–5 Printing Differences

Command	Differences
`lprsetup(8)` `printconfig(8)`	A domain-specific printer attribute, `on`, designates the domain members that are serving the printer. The print configuration utilities, `lprsetup` and `printconfig`, provide an easy means of setting the `on` attribute. The file `/etc/printcap` is shared by all members in the domain.
Advanced Printing Software	For information on installing and using Advanced Printing Software in a domain, see the configuration notes chapter in the HP *Tru64 UNIX Advanced Printing Software System Administration and Operation Guide*.

Table 17–6 describes the differences in managing security. For information on enhanced security in a domain, see the HP *Tru64 UNIX Security Administration* manual.

Table 17–6 Security Differences

Command	Differences

`auditd(8)`
`auditconfig(8)`
`audit_tool(8)`
	A domain is a single security domain. To have root privileges on the domain, you can log in as `root` on the cluster alias for the domain or on any one of the domain members. Similarly, Access Control Lists (ACLs) and user authorizations and privileges apply across the domain.
	With the exception of audit log files, security-related files, directories, and databases are shared throughout the domain. 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 domain member fails, auditing continues uninterrupted for the other domain members.
	To generate an audit report for the entire domain, 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 domain creation. You must shut down and boot all domain members to configure enhanced security after domain creation.

`rlogin(1)`
`rsh(1)`
`rcp(1)`
	An `rlogin`, `rsh`, or `rcp` request from the domain uses the default cluster alias as the source address. Therefore, if a noncluster host must allow remote host access from any account in the domain, 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 domain members.

Table 17–7 describes the differences in commands and utilities for configuring and managing systems.

Table 17–7 General System Management Differences

Command	Differences
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 domain. Events with `cluster_event` set to `false` are posted only to the member on which the event was generated.

`halt(8)`
`reboot(8)`
`init(8)`
`shutdown(8)`
	You can use the `sra shutdown` and `sra boot` commands respectively to shut down or boot a number of domain members using one command. You can also use the `sra` command to halt or reset nodes. For more information, see Chapter 16.
	The `halt` and `reboot` commands act only on the member on which the command is executed. The `halt`, `reboot`, and `init` commands have been modified to leave file systems in a domain mounted, because the file systems are automatically relocated to another domain member.
	You can use the `shutdown -c` command to shut down a domain.
	The `shutdown -ch` *time* command fails if a `clu_quorum` command or an `sra delete_member` command is in progress, or if members are being added.
	You can shut down a domain to a halt, but you cannot reboot (`shutdown -r`) the entire domain.
	To shut down a single domain member, execute the `shutdown` command from that member. For more information, see `shutdown(8)`.

`hwmgr(8)`	In a domain, the `-member` option allows you to designate the host name of the domain member that the `hwmgr` command acts upon. Use the `-cluster` option to specify that the command acts across the domain. When neither the `-member` nor `-cluster` option is used, `hwmgr` acts on the system where it is executed. Note that options can be abbreviated to the minimum unique string, such as `-m` instead of `-member`, or `-c` instead of `-cluster`.
Process Control `ps(1)`	A range of possible process identifiers (PIDs) is assigned to each domain member to provide unique PIDs across the domain. 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 which domain member it is running on. If the passed parameter is less than -1, the signal is sent to all processes (domainwide) whose process group ID matches the absolute value of the passed parameter. Even though the PID for `init` on a domain member is not 1, `kill 1` behaves as it would on a standalone system and sends the signal to all processes on the current domain 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 domain. For more information, see Section 21.1 on page 21–2.

Table 17–7 General System Management Differences

Command	Differences

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 domain member on which the command executes. It does not return information for the entire domain.

18
Tools for Managing Domains

This chapter describes the tools that you can use to manage domains in HP AlphaServer SC systems.

The information in this chapter is organized as follows:

- Introduction (see Section 18.1 on page 18–2)
- Domain Configuration Tools and SysMan (see Section 18.2 on page 18–3)
- SysMan Management Options (see Section 18.3 on page 18–4)
- Using SysMan Menu in a Domain (see Section 18.4 on page 18–5)
- Using the SysMan Command-Line Interface in a Domain (see Section 18.5 on page 18–7)

Note:

Neither SysMan Station nor Insight Manager is supported in HP AlphaServer SC Version 2.6.

18.1 Introduction

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

Tru64 UNIX and HP AlphaServer SC provide tools with Web-based, graphical, and command-line interfaces to perform management tasks. In particular, SysMan offers command-line, character-cell terminal, and X Windows interfaces to system and domain management.

SysMan is not a single application or interface. Rather, SysMan is a suite of applications for managing Tru64 UNIX and HP AlphaServer SC systems. HP AlphaServer SC Version 2.6 supports two SysMan components: SysMan Menu and the SysMan command-line interface. Both of these components are described in this chapter.

Because there are numerous domain 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 *HP TruCluster Server Cluster Administration* manual.

For more information about SysMan, see the `sysman_intro(8)` and `sysman(8)` reference pages, and Section 7.6 on page 7–27 of this guide.

Some domain operations do not have graphical interfaces and require that you use the command-line interface (CLI). These operations and commands are described in Section 18.2 on page 18–3.

18.1.1 Domain Tools Quick Start

If you are already familiar with the tools for managing domains and want to start using them, see Table 18–1. This table presents only summary information; additional details are provided later in this chapter.

Table 18–1 Domain 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**
SysMan CLI	Command Line	# **/usr/sbin/sysman -cli**

18.2 Domain Configuration Tools and SysMan

Not all HP AlphaServer SC management tools have SysMan interfaces. Table 18–2 presents the tools for managing domain-specific tasks and indicates which tools are not available through SysMan Menu. In this table, N/A means not available.

Table 18–2 Domain Management Tools

Command	Available in SysMan Menu	Function

`caa_balance(8)`
`caa_profile(8)`
`caa_register(8)`
`caa_relocate(8)`
`caa_report(8)`
`caa_start(8)`
`caa_stat(1)`
`caa_stop(8)`
`caa_unregister(8)`
	`sysman caa`	Manages highly available applications with cluster application availability (CAA).

`cfsmgr(8)`	`sysman cfsmgr`	Manages the cluster file system (CFS).
`cluamgr(8)` `clua_active(8)`	`sysman clu_aliases`	Creates and manages cluster aliases. Determines whether a cluster alias is active and reachable.
`clu_get_info`	`sysman hw_cluhierarchy` (approximate)	Gets information about a domain and its members.
`clu_quorum(8)`	N/A	Manages quorum and votes.
`drdmgr(8)`	`sysman drdmgr`	Manages distributed devices.
`mkcdsl(8)`	N/A	Makes or checks context-dependent symbolic links (CDSLs).
`sra delete_member`	N/A	Deletes a member from a domain.
`sra install`	N/A	Installs and configures an initial domain member on a Tru64 UNIX system, or adds a new member to an existing domain. (This command can only be run on the first node of each domain.)

18.3 SysMan Management Options

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

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.

HP AlphaServer SC Version 2.6 supports two SysMan components: SysMan Menu and the SysMan command-line interface. The following sections describe these components.

18.3.1 Introduction to SysMan Menu

SysMan Menu integrates most available single-system and domain administration utilities in a menu system, as shown in Figure 18–1.

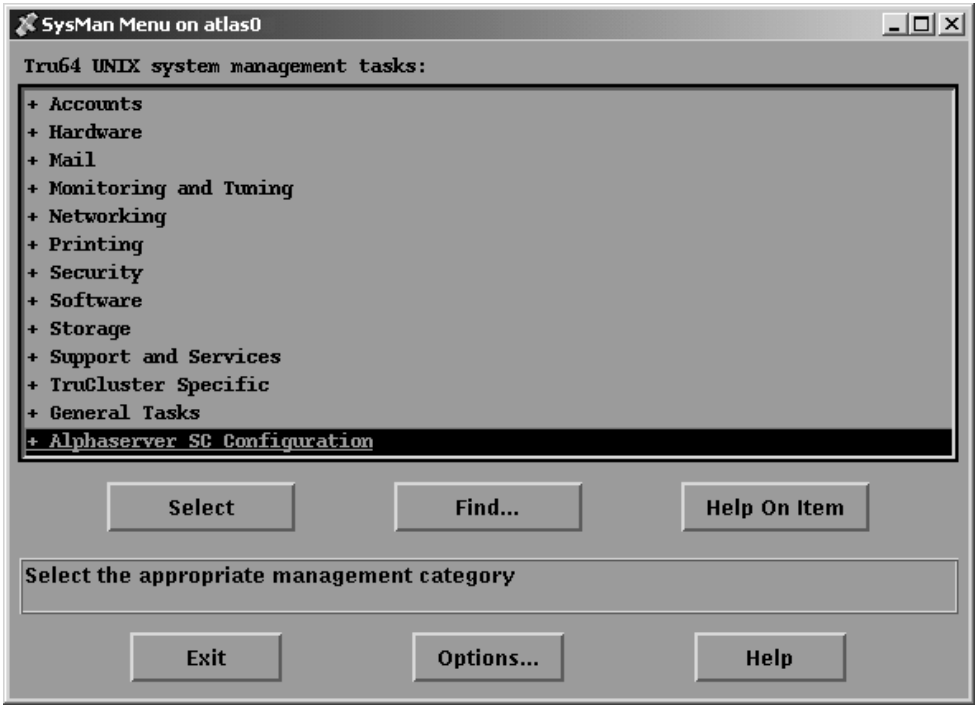

Figure 18–1 The SysMan Menu Hierarchy

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

18.3.2 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 the `sysman_cli(8)` reference page. Use the `sysman -cli -list components` command to list all known components in the SysMan data hierarchy.

18.4 Using SysMan Menu in a Domain

This section describes how to use SysMan Menu in a domain. The section begins with a discussion of **focus** and how it affects SysMan Menu.

18.4.1 Getting in Focus

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

- Domainwide — The operation affects the entire domain.
  This is the default, and does not require a focus.
- Member-specific — The operation affects only the member that you specify.
  The operation requires a focus.
- Both — The operation can be domainwide 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 domainwide and member-specific operations, SysMan Menu lets you select the domain name or a specific member on which to operate. That is, if the domain name and domain 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 domain, which is a domainwide operation, the title bar might appear similar to the following (in this example, atlas0 is a domain member and atlasD0 is the cluster alias):

```
Manage Local Users on atlas0 managing atlasD0
```

18.4.2 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. For example, specifying a focus on the command line affects the `shutdown` command. The `shutdown` command can be domainwide or member-specific.

If you start SysMan Menu from a domain member with the following command, the domain name is the initial focus of the `shutdown` option:

```
# sysman -menu
```

However, if you start SysMan Menu from a domain member with the following command, the `atlas1` domain member is the initial focus of the `shutdown` option:

```
# sysman -menu -focus atlas1
```

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 domain member, you need to select that member only once.

18.4.3 Invoking SysMan Menu

You can invoke SysMan Menu from a variety of interfaces, as explained in Table 18–3.

Table 18–3 Invoking SysMan Menu

User Interface	How to Invoke
Character-cell terminal	Start a terminal session (or open a terminal window) on a domain 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, or 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**
Common Desktop Environment (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. 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.
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 the `sysman_cli(8)` reference page for details on options and flags. See Section 18.5 on page 18–7 for more information.

18.5 Using the SysMan Command-Line Interface in a Domain

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 the `sysman_cli(8)` reference page.

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

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

The following example shows the attributes of the `clua` component for the domain member named `atlas1`:

```
# sysman -cli -focus atlas1 -list attributes -comp clua
Component: clua
  Group: cluster-aliases
    Attribute(s):
      aliasname
      memberlist
  Group: clua-info
    Attribute(s):
      memberid
      aliasname
      membername
      selw
      selp
      rpri
      joined
      virtual
  Group: componentid
    Attribute(s):
      manufacturer
      product
      version
      serialnumber
      installation
      verify
  Group: digitalmanagementmodes
    Attribute(s):
      deferredcommit
      cdfgroups
```

19

Managing the Cluster Alias Subsystem

As system administrator, you control the number of aliases, the membership of each alias, and the attributes specified by each member 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:

- Summary of Alias Features (see Section 19.1 on page 19–2)
- Configuration Files (see Section 19.2 on page 19–5)
- Planning for Cluster Aliases (see Section 19.3 on page 19–6)
- Preparing to Create Cluster Aliases (see Section 19.4 on page 19–8)
- Specifying and Joining a Cluster Alias (see Section 19.5 on page 19–9)
- Modifying Cluster Alias and Service Attributes (see Section 19.6 on page 19–11)
- Monitoring Cluster Aliases (see Section 19.7 on page 19–12)
- Leaving a Cluster Alias (see Section 19.8 on page 19–12)
- Deleting a Cluster Alias (see Section 19.9 on page 19–13)
- Modifying Domainwide Port Space (see Section 19.10 on page 19–14)
- Changing the Default Cluster Alias IP Name (see Section 19.11 on page 19–15)
- Changing the Default Cluster Alias IP Address (see Section 19.12 on page 19–18)
- Cluster Alias and NFS (see Section 19.13 on page 19–19)
- Cluster Alias and Cluster Application Availability (see Section 19.14 on page 19–20)
- Preferred Server Cluster Aliases (see Section 19.15 on page 19–23)
- Third-Party License Managers (see Section 19.16 on page 19–29)

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

- The `cluamgr` command-line interface configures parameters for aliases on the domain 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 domain members. Static parameters are written to the member's `clu_alias.config` file, but do not take effect until the next boot. (If you want the new parameters to take effect immediately, manually run the command lines that SysMan Menu wrote to each member's `clu_alias.config` file.)

19.1 Summary of Alias Features

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

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

- A domain can have multiple cluster aliases with different sets of members, subject to a maximum number of aliases as defined by the `clua:max_aliasd=33` subsystem attribute.
- There is one default cluster alias per domain. The name of the default cluster alias is the name of the domain.
- An alias is defined by an IP address, not by a Domain Name System (DNS) name. The preferred server cluster aliases must live on a common subnet, as described in Section 19.15 on page 19–23. An alias IP address can reside in either a common subnet or a virtual subnet.

  If you are using default cluster alias addresses in the range `10.`*`x`*`.`*`x`*`.`*`x`*, refer to Appendix G of the *HP AlphaServer SC Installation Guide*.
- A domain member must specify an alias in order to advertise a route to that alias. A domain member must join an alias to receive connection requests or packets addressed to that alias.
- Each domain member manages its own set of aliases. Entering a `cluamgr` command on one member affects only that member. For example, if you modify the file `/etc/clua_services`, you must run `cluamgr -f` on all domain members for the change to take effect.

- Only domain members can join cluster aliases. There is no one-to-one correspondence between a cluster alias and a specific network interface card on a domain member. Neither is there a one-to-one correspondence between a cluster alias and a specific application on one or more domain members. Cluster aliasing is a general-purpose mechanism to allow one or more members of a domain to respond to a common IP address. See Section 19.14 on page 19–20 for a description of the differences between the cluster alias and cluster application availability (CAA) subsystems.
- 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 preferred server cluster aliases.

    All members of an FS domain are members of preferred server cluster aliases, as described in Section 19.15 on page 19–23.
  - Specify any other aliases that this member will either advertise a route to or join.
  - Set options for aliases; for example, the selection weight and routing priority.

  Because each domain member reads its copy of `/etc/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 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. For more information, see Section 19.13 on page 19–19.
- Because the mechanisms that cluster alias uses to advertise routes are incompatible with `ogated` and `routed` daemons, `gated` is the required routing daemon in all HP AlphaServer SC domains.

  For further information on the `gated` configuration, see Section 19.15 on page 19–23 and Section 22.1 on page 22–2.
- The network 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 domain 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 or 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:

  Domainwide attributes:

  – *IP address and subnet mask* identifies an alias.

  Per-member attributes:

  – *Router priority* determines which member of a cluster alias responds to proxy Address Resolution Protocol (ARP) requests for this alias 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 of 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.

- In TruCluster Server systems, the cluster alias subsystem monitors network interfaces by configuring Network Interface Failure Finder (NIFF), and updates routing tables on interface failure. HP AlphaServer SC systems implement a pseudo-Ethernet interface, which spans the entire HP AlphaServer SC Interconnect. The IP suffix of this network is `-eip0`. HP AlphaServer SC systems disable NIFF monitoring on this network, to avoid unnecessary traffic on this network.

19.2 Configuration Files

Table 19–1 lists the configuration files that manage cluster aliases and services.

Table 19–1 Cluster Alias Configuration Files

File	Description
`/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.
`/etc/clua_metrics`	This file contains a routing metric for each network interface that will be used by `aliasd` when configuring `gated` to advertise routes for the default cluster alias. For more information, see Chapter 22.
`/etc/clua_services`	Defines port numbers, protocols, and connection attributes for Internet services that are provided by the domain and accessed *via* cluster aliases. The `cluamgr` command reads this file at boot time and registers each service that has one or more service attributes assigned to it. If you modify the `clua_services` file, run `cluamgr -f` on each domain member. For more information, see the `clua_services(4)` and `cluamgr(8)` reference pages.
`/etc/exports.aliases`	By default, the default cluster alias is the only cluster alias that will accept NFS requests. You can specify additional cluster aliases as NFS servers by adding them to the `/etc/exports.aliases` file. You only need to specify the additional cluster aliases in this file if they are to be used for NFS requests. The default alias is not explicitly stored in the file. If there are no cluster aliases other than the default cluster alias defined, then this file is empty on most domain members. However, on members of an FS domain, the file contains a cluster alias entry for each member of the domain; for example, if there are 32 nodes in the FS domain, there will be 32 cluster alias entries in the file.
`/etc/gated.conf.member`*N*	The alias daemon (`aliasd`) writes a member-specific `gated.conf.member`*N* file for each domain member that has an interface to the external network. For more information on the `aliasd` and `gated` daemons, see Section 19.15 on page 19–23 and Section 22.1 on page 22–2.

Table 19–1 Cluster Alias Configuration Files

File	Description
/etc/clua_default	Each domain member that has an interface to the external network is able to serve as a default gateway to those domain members that have no interfaces to the external network. The gated routing daemon advertises these routes using the metric specified in the /etc/clua_default file. For more information, see Section 22.1 on page 22–2.
/etc/gated.conf	On members that do not have an interface to the external network, the gated.conf file has a static configuration. For more information, see section Section 22.1 on page 22–2.

19.3 Planning for Cluster Aliases

Consider the following when planning the alias configuration for a domain:

- What services will the domain provide to clients (for example, login nodes, 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, as follows:

  – If your domain 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 domain 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 domain. However, you can create additional cluster aliases and use them as NFS servers. This feature is described in Section 19.13 of this chapter, in the *HP TruCluster Server Cluster Technical Overview* manual, and in the exports.aliases(4) reference page.
- Which domain members will belong to which aliases?

  If you create aliases that not all domain 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 domain member that is not a member of the alias.

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

  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 wish to associate with the Internet service entries in `/etc/clua_services`? (The file provides default alias port settings for a stock set of Internet services.)

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

  If you are using the standard network configuration in an HP AlphaServer SC system, and you want to define further cluster aliases on a virtual subnet, you will need to use static routing. This is because of the special `gated` configuration that results in incoming RIP packets being ignored by the `gated` daemon on certain members. For more information, see Section 19.15 on page 19–23 and Section 22.1 on page 22–2.

  The preferred server cluster aliases must reside on a common subnet in order to leverage the proxy ARP master functionality. For more information, see Section 19.15 on page 19–23.

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

    Note:

    Because proxy ARP is used for common subnet cluster aliases, if a 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 automatically configures the host routes for aliases on a virtual subnet. If a domain 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 domain 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.

19.4 Preparing to Create Cluster Aliases

To prepare to create cluster aliases, follow these steps:

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

   Because the mechanisms that the cluster alias uses to publish routes are incompatible with `ogated` and `routed` daemons, `gated` is the required routing daemon in all HP AlphaServer SC domains.

   When needed, the alias daemon `aliasd` adds host route entries to a domain 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).

   See the `aliasd(8)` reference page for more information about the alias daemon.

2. 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.

   You will also need to use static routing for aliases on virtual subnets. For more information, see Section 19.15 on page 19–23 and Section 22.1 on page 22–2.

3. If you want an alias other than the default cluster alias to respond to NFS requests from clients, add its name to the `/etc/exports.aliases` file.

4. For Internet services with fixed port assignments, examine the entries in the `/etc/clua_services` file. Add entries for any additional services that will be accessed through the new alias or aliases.
5. The maximum number of cluster aliases is defined by the `clua:max_aliasd=33` subsystem attribute. If the maximum number of aliases is likely to exceed this number, perform the following steps before proceeding to specify the additional aliases:
   a. Increase the maximum number of aliases in the `/etc/sysconfigtab` file.
   b. Reboot the domain so that the new value takes effect.

19.5 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 domain member, that member is aware of the alias and can advertise a route to the alias. When you join a member to an alias, that member can also receive connection requests or packets addressed to that alias.

The simplest command that specifies, but does not join, an alias using the default values for all alias attributes is:

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

After alias routing is restarted on this domain member with the `cluamgr -r start` command, the system advertises a route to the alias and can take packets off the wire and forward them to members of the alias that have joined the alias. Running the `cluamgr -s` *`alias`* command on this member displays a status for this alias of `<ENABLED>`.

The simplest command that both specifies and joins an alias using the default values for all attributes is:

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

After alias routing is restarted on this domain member with the `cluamgr -r start` command, the system advertises a route to the alias, can take packets off the wire and forward them to members of the alias that have joined the alias, and can receive connection requests or packets addressed to the alias.

Running the `cluamgr -s` *`alias_name`* command on this member displays a status for this alias of `<JOINED,ENABLED>`.

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 for one domain 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. The `cluamgr -r` start command updates kernel routing information so that the cluster alias subsystem on this member will route for the new aliases.

As mentioned earlier, to join an alias and accept the default values for the alias attributes, the following command will suffix:

```
# 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 elements that make up this `cluamgr` command do the following:

- `cluamgr -a alias=virtestalias`

  Specifies the alias. This domain member starts advertising a host route to this alias after you run the `cluamgr -r start` command.

- `join`

  Joins the alias. The member is eligible to receive packets addressed to the alias.

- `virtual`

  Advertises a network route to the virtual network where the `virtestalias` address resides.

- `mask=255.255.255.0`

  Explicitly defines a subnet mask. If you do not specify a subnet mask, the alias daemon on this member uses the network mask of the first interface on this system through which the virtual subnet is advertised.

If you do not want a domain 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 domain 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 the `cluamgr(8)` reference page for detailed instructions on configuring an alias on a virtual subnet.

19.6 Modifying Cluster Alias and Service Attributes

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

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

To make this change survive a reboot, modify the entry for this alias in this member's `clu_alias.config` file.

To modify the service port attributes associated with an Internet service in the domainwide `/etc/clua_services` file, follow these steps:

1. Using the information in the `clua_services(4)` reference page, modify the entry in `/etc/clua_services`.
2. On each domain 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.

19.7 Monitoring Cluster Aliases

Use the `cluamgr -s all` command to learn the status of cluster aliases, as shown in the following example:

```
atlas0# cluamgr -s all

Status of Cluster Alias: atlasD0

netmask: 400284c0
aliasid: 1
flags: 7<ENABLED,DEFAULT,IP_V4>
connections rcvd from net: 206
connections forwarded: 133
connections rcvd within cluster: 104
data packets received from network: 926710
data packets forwarded within cluster: 879186
datagrams received from network: 2120
datagrams forwarded within cluster: 1018
datagrams received within cluster: 1623
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=3, selp=1, rpri=1 flags=11<JOINED,ENABLED>
memberid: 3, selw=3, 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 published
atlasD0 (www.xxx.yyy.zzz) at 00-00-f8-24-a9-30 permanent published
```

19.8 Leaving a Cluster Alias

If you want a domain member to leave a cluster alias, enter the following command on that domain member:

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

19.9 Deleting a Cluster Alias

No `cluamgr` command deletes a cluster alias from a member's kernel list of aliases. Even if all domain members that are aware of the alias leave the alias (`cluamgr -a alias=`*`alias_name`*`,leave`), the domain will still accept connection requests addressed to the alias.

When a domain member enables (`cluamgr -a alias=`*`alias_name`*) or joins (`cluamgr -a alias=`*`alias_name`*`,join`), and alias routing is restarted (`cluamgr -r start`), `aliasd` adds an entry for that alias to the member's `gated.conf.member`*`N`* file, and the alias is added to that member's kernel list of aliases.

When a domain member boots, the kernel builds a list of cluster aliases from the information in that member's `/etc/clu_alias.config` file. The `aliasd` daemon gets its information from the alias list in the kernel, not from the `clu_alias.config` file. Therefore removing an alias entry from the file and restarting either the alias daemon or restarting alias routing does not delete the alias.

To delete a cluster alias, do the following:

Note:

Do not attempt to delete the default cluster alias.

---

1. Remove the entry for the cluster alias from the `/etc/hosts` file. (Although not technically required, removing or commenting out the entry is good housekeeping.)
2. On each domain member, run the `cluamgr -s` *`alias_name`* command to determine which domain members are aware of this alias.
3. On each domain member that is aware of the alias, delete any entry associated with that alias from the member's `/etc/clu_alias.config` file. (Removing the entry ensures that the alias will not be reconfigured when the system reboots.)
4. One at a time, reboot each domain member that was a member of the alias you are deleting. When all domain members that were aware of the alias are rebooted, the alias is deleted.

19.10 Modifying Domainwide Port Space

The number of ephemeral (dynamic) ports that are available domainwide for services is determined by the `inet` subsystem attributes `ipport_userreserved_min` (default: 7500) and `ipport_userreserved` (default: 65000).

Note:

The TruCluster Server default values for `ipport_userreserved_min` and `ipport_userreserved` are 1024 and 5000 respectively. These default values have been increased in HP AlphaServer SC to allow for the higher node count.

To avoid conflicting with ephemeral ports, an application should choose ports below `ipport_userreserved_min`. If the application cannot be configured to use ports below `ipport_userreserved_min`, you can prevent the port from being used as an ephemeral port by adding a static entry to the `/etc/clua_services` file.

For example, if an application must bind to port 8000, add the following entry to the `/etc/clua_services` file:

```
MegaDaemon              8000/tcp              static
```

where `MegaDaemon` is application-specific. See the `clua_services(4)` reference page for more detail.

If the application requires a range of ports, you may increase `ipport_userreserved_min`. For example, if `MegaDaemon` requires ports in the range 8000–8500, set `ipport_userreserved` to its maximum value, as follows:

1. Modify the `/etc/sysconfigtab` file on each node, as follows:
   a. Create a `sysconfigtab` fragment in a file system that is accessible on every node in the system. For example, create the fragment `/global/sysconfigtab.frag` with the following contents:
      ```
      inet:
      ipport_userreserved_min=8500
      ```
   b. Merge the changes, by running the following command:
      ```
      # scrun -n all 'sysconfigdb -f /global/sysconfigtab.frag -m inet'
      ```
2. Modify the current value of `ipport_userreserved_min` on each member, by running the following command:
   ```
   # scrun -n all 'sysconfig -r inet ipport_userreserved_min=8500'
   ```

If the number of ports is small, it is preferable to add entries to the `/etc/clua_services` file.

19.11 Changing the Default Cluster Alias IP Name

Note:

HP recommends that you do not change the default cluster alias IP name in an HP AlphaServer SC system.

To change the default cluster alias IP name from `atlasC` to `atlasD2`, perform the following steps:

1. On the management server (or Node 0, if not using a management server), run the following commands:
```
# pfsmgr offline all
# scfsmgr offline all
```
2. On the management server (or Node 0, if not using a management server), run the following commands:
```
# rmsquery -v "update sc_nodes set domain_name='atlasD2' /
    where domain_name='atlasC'"
# rmsquery -v "update sc_scfs set clu_domain='atlasD2' /
    where clu_domain='atlasC'"
# rmsquery -v "update sc_scfs_mount set cluster_name='atlasD2' /
    where cluster_name='atlasC'"
# rmsquery -v "update sc_sc_pfs_mount set cluster_name='atlasD2' /
    where cluster_name='atlasC'"
# rmsquery -v "update sc_disk_server set clu_domain='atlasD2' /
    where clu_domain='atlasC'"
# rmsquery -v "update sc_advfs_vols set clu_domain='atlasD2' /
    where clu_domain='atlasC'"
# rmsquery -v "update sc_advfs_filesets set clu_domain='atlasD2' /
    where clu_domain='atlasC'"
# rmsquery -v "update sc_lsm_vols set clu_domain='atlasD2' /
    where clu_domain='atlasC'"
```
3. On the `atlasC` domain, update the `clubase` stanza on every member, as follows:
   a. Create a clubase fragment containing the new cluster alias name, as follows:
```
[ /etc/sysconfigtab.frag ]
clubase:
      cluster_name=atlasD2
```
   b. Ensure that all nodes are up, and merge the change into the `/etc/sysconfigtab` file on each member, as follows:
```
# CluCmd /sbin/sysconfigdb -f /etc/sysconfigtab.frag -m clubase
```

4. On the management server (or Node 0, if not using a management server), use the `sra edit` command to update the SC database and the `/etc/hosts` file, as follows:

```
# sra edit
sra> sys
sys> edit cluster atlasC

Id       Description                                          Value
------------------------------------------------------------------

[0  ]    Cluster name                                         atlasC
[1  ]    Cluster alias IP address                             site-specific
⋮
------------------------------------------------------------------

Select attributes to edit, q to quit
eg. 1-5 10 15

edit? 0
Cluster name                                                  [atlasC]
new value? atlasD2

Cluster name                                                  [atlasD2]
correct? [y|n] y
sys> update hosts
sys> quit
sra> quit
```

5. On the management server (or Node 0, if not using a management server), perform the following steps:
   a. Change `atlasC` to `atlasD2` in the `/etc/hosts.equiv` file.
   b. Add `atlasD2` to the `/.rhosts` file.
6. On `atlasC`, perform the following steps:
   a. If `atlasC` is an FS domain, update the SCFS_SRV_DOMS variable in the `/etc/rc.config.common` file to contain the new domain name, as follows:
      `# rcmgr -c set SCFS_SRV_DOMS list_of_FS_domains`
   b. If `atlasC` is an FS domain, change the preferred server cluster aliases in the `/etc/exports.alias` file on the domain.
   c. Use the `sra edit` command to update the `/etc/hosts` file, as follows:

```
# sra edit
sra> sys update hosts
sra> quit
```

   d. Change `atlasC` to `atlasD2` in the `/etc/hosts.equiv` file.
   e. Add `atlasD2` to the `/.rhosts` file.

f. If atlasC is an FS domain, update the sc_scfs table, as follows:

```
# rmsquery -v "select clu_domain from sc_scfs"
clu_domain
------------
atlasC
# rmsquery "update sc_scfs set clu_domain='atlasD2' \
 where clu_domain='atlasC'"
```

7. Shut down atlasC, as follows:

   a. Use the sra command to log on to the first node in atlasC, as shown in the following example:

```
# sra -cl atlas64
```

   b. Shut down the domain, by running the following command on atlasC:

```
# shutdown -ch now
```

8. Remove atlasC from the /.rhosts file on the management server (or Node 0, if not using a management server).

9. Update each of the other domains, by performing the following steps on the first node of each domain:

   a. If atlasC is an FS domain, update the SCFS_SRV_DOMS variable in /etc/rc.config.common file to contain the new domain name.

```
# rcmgr -c set SCFS_SRV_DOMS list_of_FS_domains
```

   b. Use the sra edit command to update the /etc/hosts file, as follows:

```
# sra edit
sra> sys update hosts
sra> quit
```

   c. Change atlasC to atlasD2 in the /etc/hosts.equiv file.

   d. Add atlasD2 to the /.rhosts file.

10. Boot atlasD2, by running the following command on the management server (or Node 0, if not using a management server):

```
# sra boot -domain atlasD2
```

11. On the management server (or Node 0, if not using a management server), run the following commands:

```
# rmsctl restart
# /sbin/init.d/scmountd restart
# pfsmgr offline all
# scfsmgr offline all
```

12. On the management server (or Node 0, if not using a management server), use the sra edit command to update the RIS environment to contain the new domain name, as follows:

```
# sra edit
sra> sys update ris domains
sra> quit
```

19.12 Changing the Default Cluster Alias IP Address

It may become necessary to change the IP address of a cluster alias.

To change the cluster alias IP address of atlasD2, perform the following steps:

1. Shut down all nodes on atlasD2, except the minimum required for quorum, as shown in the following example:
```
# sra shutdown -nodes 'atlas[66-95]'
```
2. On the management server (or Node 0, if not using a management server), use the sra edit command to update the SC database and the /etc/hosts file, as follows:
```
# sra edit
sra> sys
sys> edit cluster atlasD2

Id       Description                                          Value
-------------------------------------------------------------------

[0  ]    Cluster name                                         atlasD2
[1  ]    Cluster alias IP address                             site-specific
⋮
-------------------------------------------------------------------

Select attributes to edit, q to quit
eg. 1-5 10 15

edit? 1
Cluster alias IP address                          [site-specific]
new value? new_site-specific

Cluster alias IP address                          [new_site-specific]
correct? [y|n] y
sys> update hosts
sys> quit
sra> quit
```
3. On atlasD2, use the sra edit command to update the /etc/hosts file, as follows:
```
# sra edit
sra> sys update hosts
sra> quit
```
4. Shut down the remaining nodes on atlasD2, by running the following command on the management server (or Node 0, if not using a management server):
```
# scrun -n 'atlas[64-65]' '/sbin/shutdown -h now'
```

Note:

The shutdown -ch command will not work on atlasD2 until the domain is rebooted.

5. Use the `sra edit` command to update the `/etc/hosts` file on each of the other domains, as follows:

```
# sra edit
sra> sys update hosts
sra> quit
```

6. **Note:**

This step is not necessary in this example, because we have changed the IP address of the third domain, not the first domain.

If you have changed the IP address of the first domain, you must update the `sa` entry in the `/etc/bootptab` file on Node 0.

The contents of the `/etc/bootptab` file are similar to the following:

```
.ris.dec:hn:vm=rfc1048
.ris0.alpha:tc=.ris.dec:bf=/ris/r0k1:sa=xxx.xxx.xxx.xxx:rp="atlas0:/ris/r0p1":
atlas1:tc=.ris0.alpha:ht=ethernet:gw=10.128.0.1:ha=08002BC3819D:ip=10.128.0.2:
atlas2:tc=.ris0.alpha:ht=ethernet:gw=10.128.0.1:ha=08002BC39185:ip=10.128.0.3:
atlas3:tc=.ris0.alpha:ht=ethernet:gw=10.128.0.1:ha=08002BC38330:ip=10.128.0.4:
```

where *xxx.xxx.xxx.xxx.* is the cluster alias IP address.

You must change the `sa` entry to reflect the new cluster alias IP address; until you do so, you will not be able to add more nodes to the system.

7. Boot `atlasD2`, by running the following command on the management server (or Node 0, if not using a management server):

```
# sra boot -domain atlasD2
```

8. On the management server (or Node 0, if not using a management server), use the `sra edit` command to update the RIS environment to contain the new domain name, as follows:

```
# sra edit
sra> sys update ris domains
sra> quit
```

19.13 Cluster Alias and NFS

When a domain is configured as an NFS server, NFS client requests must be directed either to the default cluster alias or to an alias listed in the `/etc/exports.aliases` file. NFS mount requests directed at individual domain 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 domain 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. On members of an FS domain, the `/etc/exports.aliases` file contains a cluster alias entry for each member of the domain; for example, if there are 32 nodes in the FS domain, there will be 32 cluster alias entries in the file. The default alias is not explicitly stored in the `/etc/exports.aliases` file.

As described in the *HP TruCluster Server Cluster Technical Overview* manual, 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 domain 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 the `cfsmgr` command 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 *HP TruCluster Server Cluster Technical Overview* manual 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.

19.14 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.

The cluster alias subsystem and CAA are described in the following points:

- CAA is designed to work with applications that run on one domain 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 domain member.
- Because cluster alias can distribute incoming requests and packets among multiple domain members, it is most useful for applications that run on more than one domain member. Cluster alias advertises routes to aliases, and sends requests and packets to members of aliases.

One potential cause of confusion is the term **single-instance application**. CAA uses this term to refer to an application that runs on only one domain 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 domain members or on one domain 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 domain 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 domain member. When the alias subsystem becomes aware that another member is listening, it will send all packets to the new port.

All domain members are members of the default cluster alias. However, you can create a cluster alias whose members are a subset of the entire domain. You can also restrict which domain 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 could create a situation where the application is running on a domain member that is not a member of the alias. The cluster alias subsystem cannot send packets to the domain member that is running the application.

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

In the first scenario, the application is not under CAA control. For each alias, the cluster alias subsystem recognizes which domain 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:

- The values of the attributes set in the `clua_services` file, if the requested service has an entry 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`.

1. The cluster alias subsystem receives a connection request or packet.
2. Of all eligible alias members, only one is listening (because CAA runs the application on only one domain member).
3. 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 domain 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`:

1. The cluster alias subsystem receives a connection request or packet.
2. The port is `in_single`.
3. The cluster alias subsystem picks an eligible member of the alias to receive the connection request or packet.
4. 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:

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

19.15 Preferred Server Cluster Aliases

In an HP AlphaServer SC system, the lead members of a CS domain function as NFS importing nodes for the SCFS file system. These nodes are responsible for importing the file systems' metadata and serving it to the domain *via* CFS.

In previous versions of HP AlphaServer SC, each FS domain defined only a default cluster alias. The default cluster alias was then used by the CS domains when mounting file systems from that FS domain. This functionality introduced one routing complexity for the CS domains, and two for the FS domains as follows:

- On a CS domain, the route table defined a gateway to the default alias based on recently received RIP updates. Typically, the RIP updates favored the management network; however, this was configurable *via* the `/etc/clua_metrics` file. Once a network was selected, the traffic was sent to the node defined as the gateway for the alias. This gateway node was effectively random, and depended on the order in which RIP updates were initially received.
- On an FS domain, the NFS traffic arrived on the node acting as the routing gateway. The NFS traffic was subsequently routed to the node serving the default cluster alias for processing. The corresponding CFS actions were then forwarded to the node in the domain that physically served the file system (CFS server). Within the FS domain, this introduced both a level of indirection that involved the internode communications service (ICS) network and additional latency.

In HP AlphaServer SC Version 2.6, the preferred server cluster alias architecture has been introduced to solve the complexities described above. This architecture consists of a number of changes, on both FS domains and CS domains, that interact with each other.

The new preferred server cluster alias architecture is described in the following sections:

- Overview (see Section 19.15.1 on page 19–24)
- Availability Configuration (see Section 19.15.2 on page 19–24)

- Proxy ARP Master Configuration (see Section 19.15.3 on page 19–25)
- Routing Configuration (see Section 19.15.4 on page 19–25)
- NFS Export Configuration (see Section 19.15.5 on page 19–26)
- Validating the Preferred Server Cluster Aliases Configuration (see Section 19.15.6 on page 19–26)
- Possible Customizations for Preferred Server Cluster Aliases (see Section 19.15.7 on page 19–27)
- Debugging Tools for Preferred Server Cluster Aliases (see Section 19.15.8 on page 19–28)

For background information on the `gated` daemon and routing, see the `gated_intro(7)`, `gated.proto(4)`, `gated.conf(4)`, and `gated.control(4)` reference pages.

19.15.1 Overview

The preferred server cluster aliases are defined for each FS domain. For every member of the FS domain, there is a corresponding preferred server cluster alias, for example, `atlasD0-atlas`*N*. On a CS domain, the NFS mount request uses the preferred server cluster alias instead of the default cluster alias.

For availability reasons, every member of an FS domain joins the other preferred server cluster aliases within the domain as well as joining its own preferred server cluster alias. All members of an FS domain also join the default cluster alias.

The new preferred server cluster alias architecture leverages the proxy ARP master functionality in the FS domain, as opposed to a fully-routed architecture. This has the implication that the preferred server cluster aliases need to be on a common subnet instead of being on a virtual subnet. The common subnet can be on any network that is shared by the preferred server nodes in the FS domains and the importing nodes in the CS domains.

The preferred server cluster aliases are specified in the `/etc/clu_alias.config` file. This is a member-specific file that is automatically created for each member of an FS domain at the time of installation or upgrade.

19.15.2 Availability Configuration

On an FS domain, to guarantee that a member will be favoured as the router for its own preferred server cluster alias, a higher `rpri` value is set for that member in its `/etc/clu_alias.config` file. An `rpri` value of 10 is used to indicate the preferred member, while an `rpri` value of 1 is used for all other members.

To guarantee that an FS member will be selected for new connection requests to its own preferred server cluster alias, a higher `selp` value is set for that member in its `/etc/clu_alias.config` file. A `selp` value of 10 is used to indicate a preferred member, while a `selp` value of 1 is used for all other members.

The following is an example of the contents of the `/etc/clu_alias.config` file on node `atlas0` from a four-node FS domain called `atlasD0`:

```
/usr/sbin/cluamgr -a selw=3,selp=1,join,alias=DEFAULTALIAS
/usr/sbin/cluamgr -a selw=3,selp=10,rpri=10,join,alias=atlasD0-atlas0
/usr/sbin/cluamgr -a selw=3,selp=1,rpri=1,join,alias=atlasD0-atlas1
/usr/sbin/cluamgr -a selw=3,selp=1,rpri=1,join,alias=atlasD0-atlas2
/usr/sbin/cluamgr -a selw=3,selp=1,rpri=1,join,alias=atlasD0-atlas3
```

Assuming a steady state condition, all preferred alias traffic is routed to the preferred member and all new connection requests are handled by the preferred member. This avoids indirection over the ICS network and improves performance.

When a preferred server is unavailable, for example in the case of node failure, traffic to the preferred server cluster alias is routed to the node that has automatically assumed the proxy ARP master role, while connection requests are handled in a round robin manner by the members remaining in the domain. For this failover scenario, the additional latency and indirection of earlier HP AlphaServer SC software versions is reintroduced.

19.15.3 Proxy ARP Master Configuration

On an FS domain, all nodes must run the `aliasd` daemon in order to enable the proxy ARP master functionality for the preferred server cluster aliases. Therefore, on all nodes of an FS domain, the CLUA_PASSIVE runtime configuration variable is set to `no` to indicate that the `aliasd` daemon should start and be active on the nodes.

On a CS domain, only the default cluster alias is defined. The internal nodes are not connected to the external network and do not need to run the `aliasd` daemon to provide the proxy ARP master functionality for the default cluster alias. Therefore, the CLUA_PASSIVE runtime configuration variable on the internal nodes is set to `yes` to indicate that the `aliasd` daemon should not be started, and the nodes run in a passive mode. This reduces the background load on a CS domain.

19.15.4 Routing Configuration

On a CS domain, to guarantee that the `gated` daemon will not learn a gateway for each preferred server cluster alias and so introduce an unwanted level of indirection, the importing node is instructed not to learn RIP updates.

The CFS/NFS design dictates that the importing node will be one of the external nodes in the domain; that node will already be running the `gated` daemon under the control of the `aliasd` daemon. Therefore, on external nodes of all domains, the

CLUAMGR_ROUTE_ARGS runtime configuration variable includes the parameter `noripin=ee0,noripin=ee1,noripin=eip0,noripin=ics0` so that RIP updates are not learned from any available networks.

19.15.5 NFS Export Configuration

On an FS domain, to permit clients to mount file systems using the preferred server cluster aliases, the `/etc/exports.aliases` file contains the preferred server cluster aliases for that domain.

19.15.6 Validating the Preferred Server Cluster Aliases Configuration

To validate the preferred server cluster aliases configuration, run the following commands:

```
# scrun -n all -w 1 "cat /etc/clu_alias.config"
# scrun -n all -w 1 "rcmgr get CLUA_PASSIVE"
# scrun -n all -w 1 "rcmgr get CLUAMGR_ROUTE_ARGS"
# scrun -d all -w 1 "cat /etc/exports.aliases"
```

Table 19–2 shows the configuration that these commands reveal for each of the possible node types on an HP AlphaServer SC Version 2.6 system. Note that some of the rows in this table reveal settings that are needed for the default route configuration. For more information on default route configuration, see Section 22.1 on page 22–2.

Table 19–2 Preferred Server Cluster Aliases Configuration

Configuration Settings / Files	FS Node with External Interface	FS Node with no External Interface	CS Node with External Interface	CS Node with no External Interface
`/etc/clu_alias.config`	As per example above with entries for each preferred server cluster alias.	As per example above with entries for each preferred server cluster alias.	Only contains an entry for the default cluster alias.	Only contains an entry for the default cluster alias.
`CLUA_PASSIVE`	`no`	`no`	`no`	`yes`

`CLUAMGR_ROUTE_ARGS`	`noripin=ee0`
	`noripin=ee1`
	`noripin=eip0`
	`noripin=ics0`
	`noripout=eip0`
		`nogated`	`noripin=ee0`
			`noripin=ee1`
			`noripin=eip0`
			`noripin=ics0`
			`noripout=eip0`
				`nogated`

`/etc/exports.aliases`	Contains a list of preferred server cluster aliases for this domain. One per line.	No additional aliases defined.	Contains a list of preferred server cluster aliases for this domain. One per line.	No additional aliases defined.

19.15.7 Possible Customizations for Preferred Server Cluster Aliases

HP expects that the preferred server cluster aliases architecture will work effectively for all sites, and does not envisage any exceptions. Some site-specific customizations are possible while still maintaining the basic architecture, and some configurations are required, as described here:

- One requirement for the preferred server cluster aliases architecture is that the addresses must be on a common subnet. This common subnet can be on any network that is shared by the preferred server nodes in the FS domains and the importing nodes in the CS domains, for example, the `ee0` or `ee1` subnets.

  This requirement introduces a demand for additional IP address space if a site operates its preferred server cluster aliases over an external network.

- If you wish to operate your preferred server cluster aliases over an external network (for example, Gigabit Ethernet), you can configure the aliases to addresses on that common subnet in one of the following ways:

  – Define the values for the first preferred server cluster aliases when the `sra setup` command is being run at installation time.

  – Alternatively, define values for the first preferred server cluster aliases in the `sys` menu of the `sra edit` command, using the `Preferred Server cluster alias base address` field.

    When you have changed this field, you must update the `/etc/hosts` file on the management server and on all domains using the `sys update hosts` command within the `sra edit` command. You must then reboot the domains so that the routing tables are modified with the new addresses.

    For information on network address conventions in HP AlphaServer SC systems, see Table 1–2 on page 1–10.

- If you wish to operate your preferred server cluster aliases over an external network on which only a subset of FS domain members have interfaces, it is still possible to leverage the preferred server cluster aliases architecture, as follows:

  a. On each FS domain member that does not have an external interface, set the `rpri` value to 0 (zero) for each preferred server cluster alias in the member-specific `/etc/clu_alias.config` file.

  b. On each FS domain member that has an external interface, adjust the `rpri` value to load-balance the routing responsibility between such members. Use an `rpri` value of 10 to indicate the preferred member, and an `rpri` value of 1 (one) for all other members.

You do not need to adjust the `selp` values in the member-specific `/etc/clu_alias.config` files. For each preferred server cluster alias, the `selp` value of 10 continues to be used to indicate a preferred member (CFS server), while a value of 1 (one) is used for all other members.

- The `rpri` setting for the preferred server cluster aliases uses only two setting levels. The preferred server of an alias has an `rpri` value of 10, while the remaining members have their `rpri` for that alias set to 1 (one). However, it is possible to introduce further levels of routing priority and further levels of `selp` weighting so that if a preferred server fails there is a defined hierarchy of failover. This may be desirable where other nodes may have direct access to the AdvFS volumes to be served. Such an `rpri`/`selp` customization should be implemented using a method that ensures that the failover of CFS serving follows the same hierarchy order.
- If your system has further routing requirements that were previously handled *via* RIP on the domains, these routes will need to be statically defined in HP AlphaServer SC Version 2.6. This is because the external nodes ignore incoming RIP updates on each interface (as a result of the `noripin=`*`interface`* parameter setting in the CLUAMGR_ROUTE_ARGS runtime configuration variable on all external nodes) and because of the filtered learning of RIP updates on internal nodes.

19.15.8 Debugging Tools for Preferred Server Cluster Aliases

Commands for displaying the preferred server cluster alias configuration are described in Section 19.15 on page 19–23. Commands for displaying the default route configuration are described in Section 22.1 on page 22–2.

Additional tools are available to check the configuration of a system as follows:

```
# /usr/sbin/netstat -r
# /usr/sbin/ripquery -r <nodename>
# /sbin/arp -a
# /usr/sbin/cluamgr -s all
# /usr/bin/cfsstat ics
# /sbin/cfsmgr
# /sbin/drdmgr
```

For more information on using these tools, see the `netstat(1)` `ripquery(8)` `arp(8)` `cluamgr(8)` `cfsstat(8)` `cfsmgr(8)` and `drdmgr(8)` reference pages.

For background information on the `gated` daemon and routing, see the `gated_intro(7)`, `gated.proto(4)`, `gated.conf(4)`, and `gated.control(4)` reference pages.

19.16 Third-Party License Managers

In general, if you have a network application that communicates with a daemon on a node that is external to the HP AlphaServer SC system, one of the following conditions must apply:

- The application is cluster-aware.
- On nodes that do not have direct external network access, the application is configured to use the cluster alias as a source address.

For example, for the fictitious application `scftp` which uses TCP port 600, add the following entries to the `/etc/services` and `/etc/clua_services` files:

- Add the following entry to the `/etc/services` file:
  ```
  scftp                600/tcp         # scftp port
  ```
- Add the following entry to the `/etc/clua_services` file:
  ```
  scftp                600/tcp     in_multi, static, out_alias
  ```

When you have set up these configuration files, run the following command to force the change to take effect on all domain members:

```
# CluCmd '/usr/sbin/cluamgr -f'
```

Some products must communicate with an external license manager using either UDP or TCP. Applications such as ABAQUS and ANSYS typically use ELM (Elan License Manager), while applications such as FLUENT and STARCD use FLEXlm. Once registered with the license manager, the product must then communicate with a dynamically selected UDP or TCP port.

For example:

- The ABAQUS application usually uses UDP port 1722 or UDP port 7631.
- The ANSYS application usually uses UDP port 1800.
- The FLUENT application usually uses TCP port 1205 or TCP port 1206.
- The STAR-CD application usually uses TCP port 1029 or TCP port 1999.
- The SAMsuite application usually uses TCP port 7274.

All requests must use the cluster alias as a source address. This allows nodes without external network connections (that is, those that have connections only to the management LAN) to communicate with the external license server, and it allows the external license server to communicate back to the nodes.

To ensure that all requests use the cluster alias as a source address, you must specify the required ports with the `out_alias` attribute in the `/etc/clua_services` configuration file.

By additionally configuring the port as `in_multi`, you also allow the port to act as a server on multiple domain members.

The `static` attribute is typically assigned to those ports between 512 and 1023 that you do not want to be assignable using the `bindresvport()` routine, or those ports within the legal range of dynamic ports that you do not want to be dynamically assignable.

The legal range of dynamically assigned ports on an HP AlphaServer SC system is from 7500 to 65000; on a normal Tru64 UNIX system, the default range is from 1024 to 5000. The limits are defined by the `inet` subsystem attributes `ipport_userreserved_min` and `ipport_userreserved`, as described in Section 19.10 on page 19–14.

Your port may or may not lie within the predefined range to which the `static` attribute applies, but it is recommended practice to always add the `static` attribute. It is also good practice to add all applications to the `/etc/services` file.

To enable these products to run on a domain in an HP AlphaServer SC system, perform the following steps:

1. Configure `clua` to manage the license manager ports correctly by adding the appropriate license manager port entries to the `/etc/clua_services` file, as shown in the following examples:

```
abaqusELM1      1722/udp      in_multi,static,out_alias
abaqusELM2      7631/udp      in_multi,static,out_alias
ansysELM1       1800/udp      in_multi,static,out_alias
fluentLmFLX1    1205/tcp      in_multi,static,out_alias
fluentdFLX1     1206/tcp      in_multi,static,out_alias
starFLX1        1999/tcp      in_multi,static,out_alias
starFLX2        1029/tcp      in_multi,static,out_alias
swrap           7274/tcp      in_multi,static,out_alias
```

   To find the ports that these daemons will use, examine the `license.dat` file. A sample `license.dat` is shown below, where italicized text is specific to the license server or application:

```
SERVER Server_Name Server_ID 7127
DAEMON Vendor_Daemon_Name \
      Application_Path/flexlm-6.1/alpha/bin/Vendor_Daemon_Name \
      Application_Path/flexlm-6.1/license.opt 7128
...
License Details
...
```

   The two numbers in bold text are the port numbers used by the master and vendor daemons respectively; enter these numbers into the `/etc/clua_services` file.

   If the `license.dat` file does not display a port number (in the above example, 7128) after the vendor daemon, the port number might be identified in the license manager log file in an entry similar to the following:

```
(lmgrd) Started cdlmd (internet tcp_port 7128 pid 865)
```

If no such port number is specified in either the `license.dat` file or the license manager log file, you can edit the license file to add the port number. You can specify any port, except a port already registered for another purpose.

Once you have established the port numbers being used by the application, and have configured these port numbers in the `/etc/clua_services` file, it is good practice to populate the `license.dat` file with the port numbers being used; for example:
`port=7128`

2. Add the appropriate license manager port entries to the `/etc/services` file, as shown in the following examples:

```
abaqusELM1      1722/udp      # ABAQUS UDP
abaqusELM2      7631/udp      # ABAQUS UDP
ansysELM1       1800/udp      # ANSYS UDP
fluentLmFLX1    1205/tcp      # FLUENT
fluentdFLX1     1206/tcp      # FLUENT
starFLX1        1999/tcp      # STAR-CD
starFLX2        1029/tcp      # STAR-CD
swrap           7274/tcp      # SAMsuite wrapper application
```

3. Reload the `/etc/clua_services` file on every member of the domain, by running the following command:
```
# CluCmd '/usr/sbin/cluamgr -f'
```

4. Repeat steps 1 to 3 on each domain.

Note:

Do not configure the `/etc/clua_services` file as a CDSL.

If you apply this process to a running system, and the ports that you are describing are in the dynamically assigned range (that is, between the reserved ports 512 to 1023, or between `ipport_userreserved_min` and `ipport_userreserved`), the ports may have already been allocated to a process. To check this, run the `netstat -a` command on each domain member, as follows:

```
# CluCmd '/usr/sbin/netstat -a | grep myportname
```

where *myportname* is the name of the port in the `/etc/services` file.

Your new settings may not take full effect until any process that is using the port has released it.

20

Managing Domain 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 domain must have clear criteria for membership and must disallow participation in the domain by systems that do not meet those criteria.

This chapter discusses the following topics:

- Connection Manager (see Section 20.1 on page 20–2)
- Quorum and Votes (see Section 20.2 on page 20–2)
- Calculating Quorum (see Section 20.3 on page 20–5)
- A Connection Manager Example (see Section 20.4 on page 20–7)
- The clu_quorum Command (see Section 20.5 on page 20–10)
- Monitoring the Connection Manager (see Section 20.6 on page 20–12)
- Connection Manager Panics (see Section 20.7 on page 20–13)
- Troubleshooting (see Section 20.8 on page 20–13)

20.1 Connection Manager

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

- Forms a domain, adds members to a domain, and removes members from a domain
- Tracks which members in a domain are active
- Maintains a domain membership list that is consistent on all domain members
- Provides timely notification of membership changes using Event Manager (EVM) events
- Detects and handles possible domain partitions

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

20.2 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 domain only when a majority of **votes** are present. When the majority of votes are present, the domain is said to have **quorum**.

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

Note:

Quorum disks are generally not supported in HP AlphaServer SC Version 2.6, and are not referred to again in this chapter.

Quorum disks are, however, supported on management-server clusters in HP AlphaServer SC Version 2.6. For such cases, refer to the HP *TruCluster Server Cluster Administration* manual.

20.2.1 How a Node Becomes a Domain Member

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

After a node has formed or joined a domain, the connection manager forever considers it to be a domain member (until someone uses the `sra delete_member` command to remove it from the domain). In rare cases, a disruption of communications in a domain (such as that caused by broken or disconnected hardware) might cause an existing domain to divide into two or more domains. In such a case, which is known as a **cluster partition**, nodes may consider themselves to be members of one domain or another. However, as discussed in Section 20.3 on page 20–5, the connection manager at most allows only one of these domains to function.

20.2.2 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 20.2.4 on page 20–4) configured in the domain. Each member brings its own notion of expected votes to the domain; 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 domain members to establish its own internal domainwide 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 domain requires to maintain quorum, as explained in Section 20.3 on page 20–5.

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

The `sra install` command automatically adjusts each member's expected votes as a new voting member is configured in the domain. The `sra delete_member` command automatically lowers expected votes when a member is deleted. Similarly, the `clu_quorum` command adjusts each member's expected votes as 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 domain member and that it is the sum of all node votes.

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.

20.2.3 Current Votes

If expected votes are the number of configured votes in a domain, **current votes** are the number of votes that are contributed by current members. Current votes are the actual number of votes that are visible within the domain.

20.2.4 Node Votes

Node votes are the fixed number of votes that a given member contributes towards quorum. Domain members can have either 1 or 0 (zero) node votes. Each member with a vote is considered to be a **voting member** of the domain. 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 shut down to single-user mode to be a domain member.

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

Although some votes may be assigned by the `sra install` command, you typically assign votes to a member after domain configuration, using the `clu_quorum` command. See Section 20.5 on page 20–10 for additional information.

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` command or the `clu_get_info -full` command to display a member's votes. See Section 20.5.2 on page 20–10 for additional 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.

20.3 Calculating Quorum

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

The quorum algorithm operates as follows:

1. The connection manager selects a set of domain 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 HP AlphaServer SC Interconnect cable or a bad HP AlphaServer SC Interconnect adapter).
2. When a domain is formed, and each time a node boots and joins the domain, 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
   - The previous cluster expected votes value

   Consider a three-member domain. 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 domain. When the new member boots and joins the domain, the connection manager calculates the new value of cluster expected votes as 4, which is the sum of node votes in the domain.
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 domainwide value, based on the value of cluster expected votes, that determines whether a given node can form, join, or continue to participate in a domain. The connection manager computes the domainwide quorum votes value using the following formula:

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

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

Note:

Expected votes (and, hence, quorum votes) are based on domain 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 domain.

4. Whenever a domain member senses that the number of votes it can see has changed (a member has joined the domain, an existing member has been deleted from the domain, 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 domain-accessible storage, and all operations across networks external to the domain until sufficient votes are added (that is, enough members have joined the domain or the communications problem is mended) to bring current votes to a value greater than or equal to quorum.

   The comparison of current votes to quorum votes occurs on a member-by-member basis, although events may make it appear that quorum loss is a domainwide event. When a domain member loses quorum, all of its I/O is suspended and all network interfaces except the HP AlphaServer SC Interconnect interfaces are turned off. No commands that must access a domainwide 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 domain 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, shutting down and booting the entire domain, or resorting to the procedures discussed in Section 29.9 on page 29–30.

20.4 A Connection Manager Example

The connection manager forms a domain when enough nodes with votes have booted for the domain to have quorum.

Consider the three-member `atlasD0` domain shown in Figure 20–1. 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 `atlasD0` domain can survive the failure of any one member.

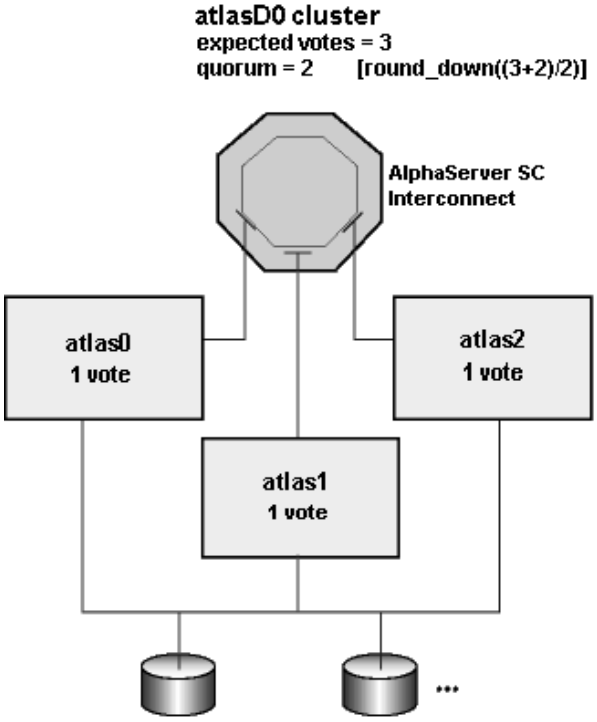

Figure 20–1 The Three-Member atlasD0 Domain

When node `atlas0` was first booted, the console displayed the following messages:

```
CNX MGR: Node atlas0 id 3 incarn 0xbde0f attempting to form or join cluster atlasD0
CNX MGR: insufficient votes to form cluster: have 1 need 2
CNX MGR: insufficient votes to form cluster: have 1 need 2
.
.
.
```

When node atlas1 was booted, its node vote plus atlas0's node vote allowed them to achieve quorum (2) and proceed to form the domain, as evidenced by the following CNX MGR messages:

```
.
.
.
CNX MGR: Cluster atlasD0 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 atlas0 3 incarn 0xbde0f csid 0x10002 has been added to the cluster
CNX MGR: Node atlas1 2 incarn 0x15141 csid 0x10001 has been added to the cluster
```

The boot log of node atlas2 shows similar messages as atlas2 joins the existing domain, 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 atlas2 1 incarn 0x26510f csid 0x10003 has been added to the cluster
```

Of course, if atlas2 is booted at the same time as the other two nodes, it participates in the domain formation and shows cluster formation messages like those nodes.

If atlas2 is then shut down, as shown in Figure 20–2, members atlas0 and atlas1 will each compare their notions of cluster current votes (2) against quorum votes (2). Because current votes equals quorum votes, they can proceed as a domain and survive the shutdown of atlas2. The following log messages describe this activity:

```
.
.
.
CNX MGR: Reconfig operation complete
CNX MGR: membership configuration index: 3 (2 additions, 1 removals)
CNX MGR: Node atlas2 1 incarn 0x80d7f csid 0x10002 has been removed from the
cluster
.
.
.
```

.

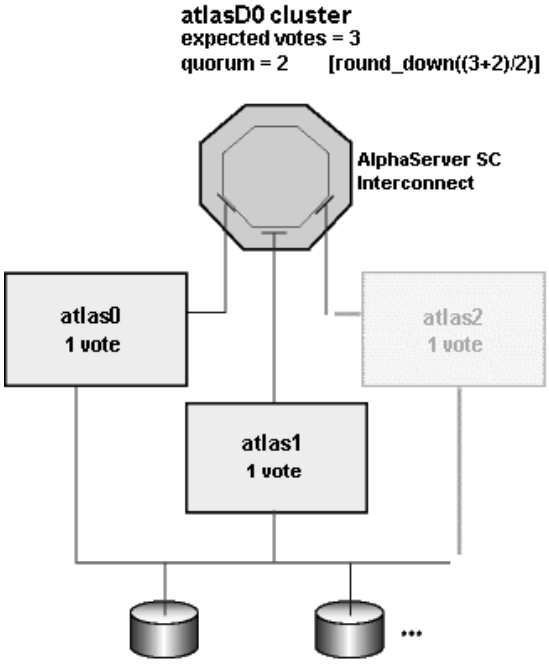

Figure 20–2 Three-Member atlasD0 Domain Loses a Member

However, this domain cannot survive the loss of yet another member. Shutting down either member `atlas0` or `atlas1` will cause the `atlasD0` domain to lose quorum and cease operation with the following messages:

```
.
.
.
CNX MGR: quorum lost, suspending cluster operations.
kch: suspending activity
dlm: suspending lock activity
CNX MGR: Reconfig operation complete
CNX MGR: membership configuration index: 4 (2 additions, 2 removals)
CNX MGR: Node atlas1 2 incarn 0x7dbe8 csid 0x10001 has been removed from the
cluster
.
.
.
```

20.5 The clu_quorum Command

You typically assign votes to a member after domain configuration, using the `clu_quorum` command.

During the installation process, the `sra install` command will assign a vote to the founding member of each domain (member 1). For more information, see Chapter 7 of the *HP AlphaServer SC Installation Guide*.

An HP AlphaServer SC system is typically configured such that at least the first two members of each domain have physical access to the cluster root (`/`), `usr`, and `var` file systems. In such a configuration, the domain can continue to function if either of the first two members should fail. By default, a single vote is given to the first member, and quorum will be lost if this member fails. To improve availability, you can add additional votes after installation. For more information, see Chapter 8 of the *HP AlphaServer SC Installation Guide*.

The section describes the following tasks:

- Using the clu_quorum Command to Manage Cluster Votes (see Section 20.5.1)
- Using the clu_quorum Command to Display Cluster Vote Information (see Section 20.5.2)

20.5.1 Using the clu_quorum Command to Manage Cluster Votes

Use the `clu_quorum -m` command to adjust a particular member's node votes. You can specify a value of 0 (zero) or 1. You must shut down and boot the target member for the change to take effect in its kernel and be broadcast to the kernels of other running members. If you change a member's votes from 1 to 0 (zero), once the member has been shut down and booted, you must issue the `clu_quorum -e` command to reduce expected votes across the domain.

Use the `clu_quorum -e` command to adjust expected votes throughout the domain. The value you specify for expected votes should be the sum total of the node votes assigned to all members in the domain. You can adjust expected votes up or down by one vote at a time. You cannot specify an expected votes value that is less than the number of votes currently available. The `clu_quorum` command warns you if the specified value could cause the domain to partition or lose quorum.

20.5.2 Using the clu_quorum Command to Display Cluster Vote Information

When specified without options (or with the `-f` and/or the `-v` option), the `clu_quorum` command displays information about the current member node votes, and expected votes configuration of the domain.

This information includes:

1. Cluster common quorum data from the domainwide /etc/sysconfigtab.cluster file. The cluster_expected_votes, cluster_qdisk_major, cluster_qdisk_minor, and cluster_qdisk_votes clubase attribute values in this file should be identical to the corresponding values in each member's /etc/sysconfigtab file.

   When a member is booted or rebooted into a domain, a check script compares the values of these attributes in its etc/sysconfigtab file against those in the domainwide /etc/sysconfigtab.cluster file. If the values differ, the check script copies the values in /etc/sysconfigtab.cluster to the member's etc/sysconfigtab and displays a message.

   When the boot completes, you should run the clu_quorum command and examine the running and file values of the cluster_expected_votes, cluster_qdisk_major, cluster_qdisk_minor, and cluster_qdisk_votes clubase attributes for that member. If there are discrepancies between the running and file values, you must resolve them.

   The method that you should use varies. If the member's file values are correct but its running values are not, you typically shut down and boot the member. If the member's running values are correct but its file values are not, you typically use the clu_quorum command.

2. Member-specific quorum data from each member's running kernel and /etc/sysconfigtab file, together with 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.

   The member-specific quorum data includes attribute values from both the clubase and cnx kernel subsystems.

See the clu_quorum(8) reference page for a description of the individual items displayed by the clu_quorum command.

When examining the output from the clu_quorum command, remember the following:

- In a healthy domain, the running and file values of the attributes should be identical. If there are discrepancies between the running and file values, you must resolve them. The method that you use varies. If the member's file values are correct but its running values are not, you typically shut down and boot the member. If the member's running values are correct but its file values are not, you typically use the clu_quorum command.

- With the exception of the member vote value stored in the `clubase cluster_node_votes` attribute, each domain member should have the same value for each attribute. If this is not true, enter the appropriate `clu_quorum` commands from a single domain member to adjust expected votes and quorum disk information.
- The `clubase` subsystem attribute `cluster_expected_votes` should equal the sum of all member votes (`cluster_node_votes`), including those of `DOWN` members. If this is not true, enter the appropriate `clu_quorum` commands from a single domain member to adjust expected votes.
- The `cnx` subsystem attribute `current_votes` should equal the sum of the votes of all `UP` members.
- The `cnx` subsystem attribute `expected_votes` is a dynamically calculated value that is based on a number of factors (discussed in Section 20.3 on page 20–5). Its value determines that of the `cnx` subsystem attribute `quorum_votes`.
- The `cnx` subsystem attribute `qdisk_votes` should be identical to the `clubase` subsystem attribute `cluster_qdisk_votes`.
- The `cnx` subsystem attribute `quorum_votes` is a dynamically calculated value that indicates how many votes must be present in the domain for domain members to be allowed to participate in the domain and perform productive work. See Section 20.3 on page 20–5 for a discussion of quorum and quorum loss.

20.6 Monitoring the Connection Manager

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

- Node joining domain
- Node removed from domain

Each of these events also results in console message output.

The connection manager prints various informational messages to the console during member boots and domain transactions.

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

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

20.7 Connection Manager Panics

The connection manager continuously monitors domain members. In the rare case of a cluster partition, in which an existing domain divides into two or more domains, nodes may consider themselves to be members of one domain or another. As discussed in Section 20.3 on page 20–5, the connection manager at most allows only one of these domains to function.

To preserve data integrity if a domain 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. There is no way to repair a partition without rebooting one or more members to have them rejoin the domain.

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

- The connection manager on a node that is already a domain member discovers a node that is a member of a different domain (which may be a different incarnation of the same domain). 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
```

- A panicking node has discovered a domain 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 domain during a reconfiguration because of communication problems:

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

20.8 Troubleshooting

For information about troubleshooting, see Chapter 29.

21

Managing Domain Members

This chapter discusses the following topics:

- Managing Configuration Variables (see Section 21.1 on page 21–2)
- Managing Kernel Attributes (see Section 21.2 on page 21–3)
- Managing Remote Access Within and From the Domain (see Section 21.3 on page 21–5)
- Adding Domain Members After Installation (see Section 21.4 on page 21–6)
- Converting a CS Domain to an FS Domain (see Section 21.5 on page 21–12)
- Deleting a Domain Member (see Section 21.6 on page 21–15)
- Adding a Deleted Member Back into the Domain (see Section 21.7 on page 21–17)
- Reinstalling a Domain (see Section 21.8 on page 21–17)
- Moving a Domain to Another IP Subnet (see Section 21.9 on page 21–18)
- Updating ifaccess.conf When Changing Network Interfaces (see Section 21.10 on page 21–32)
- Managing Software Licenses (see Section 21.11 on page 21–34)
- Updating System Firmware (see Section 21.12 on page 21–34)
- Updating the Generic Kernel After Installation (see Section 21.13 on page 21–36)
- Changing a Node's Ethernet Card (see Section 21.14 on page 21–36)
- Managing Swap Space (see Section 21.15 on page 21–37)
- Installing and Deleting Layered Applications (see Section 21.16 on page 21–42)
- Managing Accounting Services (see Section 21.17 on page 21–43)

For information about booting members and shutting down members, see Chapter 2.

21.1 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 domain. Table 21–1 presents the uses of the configuration files.

Table 21–1 /etc/rc.config* Files

File	Scope
/etc/rc.config	Member-specific variables. /etc/rc.config is a CDSL. Each domain 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	Domainwide 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	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.site, then /etc/rc.config.common, and finally /etc/rc.config) 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 domainwide, use the -c option. The command then acts on /etc/rc.config.common, the domainwide 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.

21.2 Managing Kernel Attributes

Each member of a domain runs its own kernel and, therefore, has its own `/etc/sysconfigtab` file. This file contains static member-specific attribute settings. Although a domainwide `/etc/sysconfigtab.cluster` file exists, its purpose is different from that of `/etc/rc.config.common`, and it is reserved to utilities that ship in the HP AlphaServer SC product. This section presents a partial list of those kernel attributes 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 list the name and status of all of the subsystems, use the following command:

```
# sysconfig -s
```

In addition to the domain-related kernel attributes presented in this section, two kernel attributes (`vm` subsystem) are set during domain installation. Table 21–2 lists these kernel attributes. Do not change the values assigned to these attributes.

Table 21–2 Kernel Attributes to be Left Unchanged — vm Subsystem

Attribute	Value (Do Not Change)
`vm_page_free_min`	30
`vm_page_free_reserved`	20

Tables 21–3 and 21–4 show the TruCluster Server and SC subsystem names associated with components.

Table 21–3 Configurable TruCluster Server Subsystems

Subsystem Name	Component	Attributes
`cfs`	Cluster file system	`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 (ICS) network service	`sys_attrs_icsnet(5)`
`ics_hl`	Internode communications service (ICS) high level	`sys_attrs_ics_hl(5)`

Table 21–3 Configurable TruCluster Server Subsystems

Subsystem Name	Component	Attributes
token	CFS token subsystem	sys_attrs_token(5)

Table 21–4 Configurable SC Subsystems

Subsystem Name	Component	Attributes
scfs	SCFS file system	scfs_attrs_scfs(5)
scfs_client	SCFS file system	scfs_attrs_scfs_client(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:

- Use the sysconfig command.

  To change the value of an attribute so that its new value takes effect immediately at run time, use the sysconfig command as follows:

  ```
  # sysconfig -r subsystem_name attribute_list
  ```

  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
  ```

  Note that any changes made using the sysconfig command are valid for the current session only, and will be lost during the next system boot.

  Note also that some kernel attributes cannot be reconfigured dynamically. If you receive a message indicating that the attribute does not allow this operation when you attempt to reconfigure it using the sysconfig command, you must change the attribute by setting the value in the /etc/sysconfigtab file, as described in the next section.

- Set the attribute in the /etc/sysconfigtab file.

  To set or change an attribute's value and allow the change to be preserved over the next system boot, set the attribute in the /etc/sysconfigtab file. Do not edit the /etc/sysconfigtab file manually — use the sysconfigdb command to add or edit a subsystem-name stanza entry in the /etc/sysconfigtab file.

  For more information, see the sysconfigdb(8) reference page.

You can also use the configuration manager framework, as described in the HP *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 run the `/sbin/sysconfig -h` command, as in the following example:

```
# sysconfig -h atlas2 -r drd drd-do-local-io=0
drd-do-local-io: reconfigured
```

Note:

In general, it should not be necessary to modify kernel subsystem attributes, as most kernel subsystems try to be self-tuning, where possible. Consult the *HP AlphaServer SC Release Notes* before modifying any kernel subsystem attributes, to see if there are any HP AlphaServer SC-specific restrictions.

21.3 Managing Remote Access Within and From the Domain

An `rlogin`, `rsh`, or `rcp` command from the domain uses the default cluster alias as the source address. Therefore, if a nondomain host must allow remote host access from any account in the domain, the `.rhosts` file on the nondomain 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 NIS or DNS.

The same requirement holds for `rlogin`, `rsh`, or `rcp` to work between domain members. At domain creation, the `sra install` command uses the data in the SC database (which was generated by the `sra setup` command) to put all required host names in the correct locations in the proper format. The `sra install` command does the same when a new member is added to the domain. You do not need to edit the `/.rhosts` file to enable `/bin/rsh` commands from a domain member to the cluster alias or between individual members. Do not change the generated name entries in the `/etc/hosts` and `/.rhosts` files.

If the `/etc/hosts` and `/.rhosts` files are configured incorrectly, many applications will not function properly. For example, the 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` and/or `/.rhosts` files have been configured incorrectly:

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

21.4 Adding Domain Members After Installation

If you had planned (before the initial installation) to install the system in phases, add the new domain members by following the instructions for a phased installation, provided in Section 21.4.1 on page 21–6.

Otherwise, add the new domain members by following the instructions for system expansion, provided in Section 21.4.2 on page 21–7.

21.4.1 Adding Domain Members After Initial Installation — Phased Installation

When planning your HP AlphaServer SC system, you may decide to install your system in phases — this means that you specify the final anticipated number of nodes during the initial installation, and then configure out the nodes that have not yet been added to the system.

In the example in this section, the number of nodes specified in the original `sra setup` was 16: eight nodes set up during the initial installation as described in the *HP AlphaServer SC Installation Guide*, and eight nodes added afterwards as described in this section.

Note:

During the initial installation, you should have specified the final anticipated number of nodes when running the `sra setup` command, to indicate a phased installation.

If not, do not perform a phased installation. Instead, add the new domain members by following the instructions for system expansion (see Section 21.4.2 on page 21–7).

To add the remaining eight nodes, perform the following steps:

1. Connect all hardware (HP AlphaServer SC Interconnect, console, networks).
2. Shut down the non-voting members of the domain that is being expanded; for example, `atlas[1-7]`.
3. Stop the console logger daemon, as follows:
   - If CMF is CAA-enabled: `# caa_stop SC10cmf`
   - If CMF is not CAA-enabled: `# /sbin/init.d/cmf stop`
4. Run the `sra setup` command, as follows:
   ```
   # sra setup
   ```
   You are prompted to indicate whether you want to use the existing database, as follows:
   ```
   An SRA database already exists
   Use the current database? [yes]
   ```
   Answer `yes` to this question.

The `sra setup` command does the following:

a. Adds the new members to the `/etc/hosts` file.

b. Sets up the terminal-server console ports.

c. Restarts the console logger daemon.

d. Probes the systems again and updates RIS with the ethernet address of the new nodes.

5. If you have more than one domain, run the `sra edit` command on each domain in turn, to update the `/etc/hosts` files on the other domains, as follows:
```
# sra edit
sra> sys update hosts
```

6. Run the `sra ethercheck` and `sra elancheck` tests to verify the state of the management network and HP AlphaServer SC Interconnect network on the new nodes. See Chapter 5 or Chapter 6 (depending on the system configuration) of the *HP AlphaServer SC Installation Guide* for details of how to run these tests.

7. Run the following command to check whether the `srad` daemon is running:
```
# sra srad_info
```
The `srad_info` command checks to see if the various `srad` daemons are running correctly on the system (one `srad` daemon running on the management server, and one `srad` daemon for each domain). If any of the `srad` daemons is not running correctly, the `srad_info` command specifies the problem domain. Start the `srad` daemon on that domain by running the following command:
```
# /sbin/init.d/sra start
```

8. Run the `sra install` command to add the new members, as follows:
```
# sra install -nodes 'atlas[8-15]'
```
For more information about adding members to a domain, see Chapter 7 of the *HP AlphaServer SC Installation Guide*.

9. Boot all of the members of the domain, as follows:
```
# sra boot -nodes 'atlas[1-15]'
```

10. Configure the new nodes into the SC database, as follows:
```
# rcontrol configure in nodes='atlas[8-15]'
```

21.4.2 Adding Domain Members After Initial Installation — System Expansion

To expand an HP AlphaServer SC system when the original `sra setup` dialog did not indicate a phased installation, perform the following steps on the management server (or Node 0, if not using a management server):

1. Stop the console logger daemon, as described in Section 14.11 on page 14–13.

2. Run the `sra setup` command as described in Chapter 5 or Chapter 6 (depending on the system configuration) of the *HP AlphaServer SC Installation Guide*, but this time specify the final total number of nodes (16).

   You are prompted to indicate whether you want to use the existing database, as follows:
   ```
   An SRA database already exists
   Use the current database? [yes]
   ```

   Answer `no` to this question, so that you are given the opportunity to re-size the system. The current database is then dropped, and you will be prompted to supply information on the size of the system (node count, domain count, and so on).

   If you answer `yes` to this question, you will not be given the opportunity to re-size the system.

   The `sra setup` command does the following:

   a. Adds the new members to the `/etc/hosts` file.

   b. Sets up the terminal-server console ports.

   c. Restarts the console logger daemon.

   d. Probes the new nodes and updates RIS with their ethernet addresses.

3. The `sra setup` command incorrectly marks existing nodes as being uninstalled. To work around this problem, update the SC database, as follows:
   ```
   # rmsquery "update sc_nodes set bootable='1',
   current_state='Member_Added', desired_state='Member_Added'
   where name='atlas0'"
   ```

   You must issue a similar command for each existing node in the system (you can write a script to do this).

4. If you have more than one domain, run the `sra edit` command on each domain in turn, to update the `/etc/hosts` files on the other domains, as follows:
   ```
   # sra edit
   sra> sys update hosts
   ```

5. With the nodes at the SRM prompt, run the `sra ethercheck` and `sra elancheck` tests to verify the state of the management network and HP AlphaServer SC Interconnect network on the new nodes. See Chapter 5 or Chapter 6 (depending on the system configuration) of the *HP AlphaServer SC Installation Guide* for details of how to run these tests.

6. Change the interconnect nodeset mask for the updated domains, as described in Section 21.4.3 on page 21–9.

7. Run the following command to check whether the `srad` daemon is running:
   ```
   # sra srad_info
   ```

The `srad_info` command checks to see if the various `srad` daemons are running correctly on the system (one `srad` daemon running on the management server, and one `srad` daemon for each domain). If any of the `srad` daemons is not running correctly, the `srad_info` command specifies the problem domain. Start the `srad` daemon on that domain by running the following command:

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

8. Run the `sra install` command to add the new members, as follows:

```
# sra install -nodes 'atlas[16-19]'
```

   For more information about adding members to a domain, see Chapter 7 of the *HP AlphaServer SC Installation Guide*.

9. Boot all of the members of the domain, as follows:

```
# sra boot -nodes 'atlas[16-19]'
```

10. Set up the RMS partitions, as described in Section 8.5 of the *HP AlphaServer SC Installation Guide*.

21.4.3 Changing the Interconnect Nodeset Mask

Note:

This step is not necessary if your HP AlphaServer SC system has less than 32 nodes.

The HP AlphaServer SC Interconnect software uses a nodeset mask to limit the amount of interconnect traffic and software overhead. This nodeset mask, which is called `ics_elan_enable_nodeset`, is specified in the `/etc/sysconfigtab` file of each domain member; it is different for each domain, but the same for all members of one domain.

The nodeset mask in the `/etc/sysconfigtab` file of each domain member is an array of 32 entries; each entry is 32 bits long. Each bit represents an interconnect switch port: typically this maps to the corresponding node number. On each domain member, the only bits that should be set in the array are those that represent the nodes that are members of the domain.

For example, in an HP AlphaServer SC system that has 32 nodes in each of 32 domains, the `ics_elan_enable_nodeset` array in the `/etc/sysconfigtab` file of the members of each domain is set as follows:

On Members of Domain 0:

```
ics_elan_enable_nodeset[0] = 0xffffffff
ics_elan_enable_nodeset[1] = 0x00000000
⋮
ics_elan_enable_nodeset[31] = 0x00000000
```

On Members of Domain 1:

```
ics_elan_enable_nodeset[0] = 0x00000000
ics_elan_enable_nodeset[1] = 0xffffffff
ics_elan_enable_nodeset[2] = 0x00000000
:
ics_elan_enable_nodeset[31] = 0x00000000
.
.
.
```

On Members of Domain 31:

```
ics_elan_enable_nodeset[0] = 0x00000000
ics_elan_enable_nodeset[1] = 0x00000000
ics_elan_enable_nodeset[2] = 0x00000000
:
ics_elan_enable_nodeset[31] = 0xffffffff
```

After you have added nodes to the system, you must manually change the nodeset for any preexisting domains to which you have added nodes. You do not need to create nodeset entries for any domains that are created as a result of adding nodes — the installation process will automatically create the appropriate nodeset entries for such domains.

The following example illustrates this process. In this example, we will add 24 nodes to the `atlas` system by extending an existing domain (`atlasD3`) and creating a new domain (`atlasD4`). Table 21–5 describes the node layout in the `atlas` system.

Table 21–5 Example System — Node Layout

BEFORE: 104 Nodes, 4 Domains			AFTER: 128 Nodes, 5 Domains		
Domain	**#Nodes**	**Nodes**	**Domain**	**#Nodes**	**Nodes**
atlasD0	14	atlas0 - atlas13	atlasD0	14	atlas0 - atlas13
atlasD1	32	atlas14 - atlas45	atlasD1	32	atlas14 - atlas45
atlasD2	32	atlas46 - atlas77	atlasD2	32	atlas46 - atlas77
atlasD3	26	atlas78 - atlas103	atlasD3	32	atlas78 - atlas109
-	-	-	atlasD4	18	atlas110 - atlas117

Table 21–6 describes the `ics_elan_enable_nodeset` values.

Table 21–6 Example System — Nodeset Values

BEFORE: 104 Nodes, 4 Domains		AFTER: 128 Nodes, 5 Domains	
Domain	**Nodeset[1]**	**Domain**	**Nodeset[1]**
atlasD0	ics...nodeset[0] = 0x00003fff ics...nodeset[1] = 0x00000000 . ics...nodeset[31] = 0x00000000	atlasD0	ics...nodeset[0] = 0x00003fff ics...nodeset[1] = 0x00000000 . ics...nodeset[31] = 0x00000000
atlasD1	ics...nodeset[0] = 0xffffc000 ics...nodeset[1] = 0x00003fff ics...nodeset[2] = 0x00000000 . ics...nodeset[31] = 0x00000000	atlasD1	ics...nodeset[0] = 0xffffc000 ics...nodeset[1] = 0x00003fff ics...nodeset[2] = 0x00000000 . ics...nodeset[31] = 0x00000000
atlasD2	ics...nodeset[0] = 0x00000000 ics...nodeset[1] = 0xffffc000 ics...nodeset[2] = 0x00003fff ics...nodeset[3] = 0x00000000 . ics...nodeset[31] = 0x00000000	atlasD2	ics...nodeset[0] = 0x00000000 ics...nodeset[1] = 0xffffc000 ics...nodeset[2] = 0x00003fff ics...nodeset[3] = 0x00000000 . ics...nodeset[31] = 0x00000000
atlasD3	ics...nodeset[0] = 0x00000000 ics...nodeset[1] = 0x00000000 ics...nodeset[2] = 0xffffc000 ics...nodeset[3] = 0x000000ff ics...nodeset[4] = 0x00000000 . ics...nodeset[31] = 0x00000000	atlasD3	ics...nodeset[0] = 0x00000000 ics...nodeset[1] = 0x00000000 ics...nodeset[2] = 0xffffc000 ics...nodeset[3] = 0x00003fff ics...nodeset[4] = 0x00000000 . ics...nodeset[31] = 0x00000000
-	- -	atlasD4	ics...nodeset[0] = 0x00000000 ics...nodeset[1] = 0x00000000 ics...nodeset[2] = 0x00000000 ics...nodeset[3] = 0xffffc000 ics...nodeset[4] = 0x00000000 . ics...nodeset[31] = 0x00000000

[1]For legibility, `ics_elan_enable_nodeset` has been abbreviated to `ics...nodeset` in this table.

As shown in Table 21–5, adding 24 nodes has affected the nodeset mask of two domains:

- atlasD4 (new)

  Because atlasD4 is a new domain, the installation process will add the correct atlasD4 nodeset mask entries to the /etc/sysconfigtab file.

- atlasD3 (changed)

  Because you have added nodes to a preexisting domain (atlasD3), you must manually correct the atlasD3 nodeset mask entries in the /etc/sysconfigtab file. The /etc/sysconfigtab file is member-specific, so you must correct the nodeset mask on each node in atlasD3.

To update the atlasD3 nodeset mask entries, perform the following steps:

1. On any node in the atlasD3 domain, create a temporary file containing the correct atlasD3 nodeset mask entries, as follows:
   ```
   ics_elan:
   ics_elan_enable_nodeset[2] = 0xffffc000
   ics_elan_enable_nodeset[3] = 0x00003fff
   ```
2. Copy the temporary file created in step 1 (for example, newsysconfig) to a file system that is accessible to all nodes in the domain (for example, /global).
3. Run the following command, to apply the changes to every node in the domain:
   ```
   # scrun -n all '/sbin/sysconfigdb -m -f /global/newsysconfig'
   ```

21.5 Converting a CS Domain to an FS Domain

If you wish, you can convert a CS domain to an FS domain, subject to the supported maximum of four FS domains.

To convert a CS domain to an FS domain, perform the following steps (where atlasD1 is the CS domain being converted to an FS domain):

1. Verify that there are currently fewer than four FS domains in the HP AlphaServer SC system, as shown in the following example, which shows a system with eight domains:
   ```
   atlasms# rmsquery -v "select name,type from sc_domain"
   name     type
   atlasD0  fs
   atlasD1  cs
   atlasD2  cs
   atlasD3  cs
   atlasD4  cs
   atlasD5  cs
   atlasD6  cs
   atlasD7  cs
   ```

2. Take all SCFS file systems and PFS file systems offline, as shown in the following example:
```
atlasms# scfsmgr offline all
atlasms# pfsmgr offline all
```
(While it is not mandatory to take all SCFS and PFS file systems offline while changing a CS domain to an FS domain, it is safer to do so.)

3. On the management server (if used) or Node 0, change the domain type, as shown in the following example:
```
atlasms# rmsquery -v "update sc_domain set type='fs' where name='atlasD1'"
```

4. Verify that the domain type has been changed, as shown in the following example:
```
atlasms# rmsquery -v "select name,type from sc_domain"
name     type
atlasD0  fs
atlasD1  fs
atlasD2  cs
atlasD3  cs
atlasD4  cs
atlasD5  cs
atlasD6  cs
atlasD7  cs
```

5. Add `atlasD1` to the list of FS domains in the SCFS_SRV_DOMS variable in the `/etc/rc.config.common` file on the management server and on all domains, as shown in the following example, where `atlasD0` is an existing FS domain, and `atlasD1` is the new FS domain:
```
atlasms# rcmgr -c set SCFS_SRV_DOMS atlasD0, atlasD1
atlasms# scrun -d all "rcmgr -c set SCFS_SRV_DOMS atlasD0, atlasD1"
```

6. Verify that `atlasD1` was added to the list of FS domains in the SCFS_SRV_DOMS variable on the management server and on all domains, as shown in the following examples:
```
atlasms# rcmgr get SCFS_SRV_DOMS
atlasD0,atlasD1

atlasms# scrun -d all "rcmgr -c get SCFS_SRV_DOMS"
atlasD0: atlasD0,atlasD1
atlasD1: atlasD0,atlasD1
atlasD2: atlasD0,atlasD1
atlasD3: atlasD0,atlasD1
atlasD4: atlasD0,atlasD1
atlasD5: atlasD0,atlasD1
atlasD6: atlasD0,atlasD1
atlasD7: atlasD0,atlasD1
```

7. In the `/etc/export.aliases` file, add one new line for each member of the new FS domain, using the notation *domainName-nodeName*. For example, if the new FS domain has three members, the `/etc/export.aliases` file on the new FS domain should contain lines similar to the following:
```
atlasD1-atlas32
```

```
atlasD1-atlas33
atlasD1-atlas34
```

8. In the member-specific /etc/clu_alias.config file on each member of the new FS domain, add one new line for each member of the new FS domain. In the line that applies to the member whose file you are updating, set the selp and rpri values to 10. The rpri value defines the alias member's router priority, and the selp value defines the alias member's selection priority (for more information on these attributes, see the cluamgr(8) reference page).

   For example, if the new FS domain has three members, the /etc/clu_alias.config file on the atlas33 node should contain lines similar to the following:

```
/usr/sbin/cluamgr -a selw=3,selp=1,join,alias=DEFAULTALIAS
/usr/sbin/cluamgr -a selw=3,selp=1,rpri=1,join,alias=atlasD1-atlas32
/usr/sbin/cluamgr -a selw=3,selp=10,rpri=10,join,alias=atlasD1-atlas33
/usr/sbin/cluamgr -a selw=3,selp=1,rpri=1,join,alias=atlasD1-atlas34
```

9. Members of a CS domain that do not have an external interface use the CLUA_PASSIVE attribute with a value of YES in the member-specific /etc/rc.config file to prevent aliasd from running. However, on an FS domain, this attribute should not exist on any member, so that aliasd will run on all members of the FS domain.

   To ensure that the CLUA_PASSIVE attribute is not present in the member-specific /etc/rc.config file on any member of the new FS domain, run the command shown in the following example, where atlasD1 is the CS domain being converted to an FS domain:

```
atlasms# scrun -d atlasD1 -m all "rcmgr delete CLUA_PASSIVE"
```

10. Update the entries in the /etc/hosts file on each domain and on the management server (if present) by running the following commands:

```
atlasms# sra edit
atlasms# sys update hosts
```

11. Reboot the new FS domain by running the following commands on the management server:

```
atlasms# sra shutdown -domains atlasD1
atlasms# sra boot -domains atlasD1
```

12. Stop and restart the scmountd daemon on the management server (if used) or Node 0, as shown in the following example:

```
atlasms# /sbin/init.d/scmountd restart
scmountd: Stopping the daemon
scmountd: Starting the daemon
```

13. Run the command shown in the following example on the management server (if used) or Node 0 to find the disks and file systems on the new domain:

```
atlasms# scfsmgr scan
```

14. Bring all SCFS file systems and PFS file systems online, as shown in the following example:

```
atlasms# scfsmgr online all
atlasms# pfsmgr online all
```

21.6 Deleting a Domain Member

The `sra delete_member` command permanently removes a member from the domain.

Note:

If you are reinstalling HP AlphaServer SC, see the *HP AlphaServer SC Installation Guide*. Do not delete a member from an existing domain and then create a new single-member domain from the member you just deleted. If the new domain has the same name as the old domain, the newly installed system might join the old domain. This can cause data corruption.

The `sra delete_member` command has the following syntax:
`sra delete_member -nodes` *`nodes`*

The `sra delete_member` command performs the following tasks:

- Adjusts the value of `cluster_expected_votes` throughout the domain, if the deleted member has votes.
- Deletes all member-specific directories and files in the domainwide file systems.

Note:

The `sra 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 `sra delete_member`. Otherwise, if you add a new member and re-use 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 HP AlphaServer SC Interconnect interface from the `/.rhosts` and `/etc/hosts.equiv` files.
- Writes a log file of the deletion to `/cluster/admin/clu_delete_member.log`. Appendix D contains a sample `clu_delete_member` log file.

Deleting a Domain Member

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

Note:

If you delete two voting members, the domain will lose quorum and suspend operations.

1. Configure the member out of RMS (see Section 5.8.1 on page 5–61).
2. Shut down the member.

Note:

Before you delete a member from the domain, you must be very careful to shut the node down cleanly. If you halt the system, or run the `shutdown -h` command, local file domains may be left mounted, in particular `rootN_tmp domain`. If this happens, `sra delete_member` will not allow the member to be deleted — before deleting the member, it first checks for any locally mounted file systems; if any are mounted, it aborts the delete. To shut down a node and ensure that the local file domains are unmounted, run the following command:

```
# sra shutdown -nodes node
```

If a member has crashed leaving local disks mounted and the node will not reboot, the only way to unmount the disks is to shut down the entire domain.

3. Ensure that two of the three voting members (Members 1, 2, and 3) are up.
4. Use the `sra delete_member` command (from any node, but typically from the management server) to remove the member from the domain. For example, to delete a halted member whose host name is `atlas2`, enter the following command:

```
# sra delete_member -nodes atlas2
```

5. If the member being deleted is a voting member, the delete operation should have automatically lowered the expected votes for the domain by one vote. Use the `clu_quorum` command to check the expected votes for the domain after the member has been deleted — if necessary, you can manually lower by one vote the expected votes for the domain by running the following command:

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

For an example of the `/cluster/admin/clu_delete_member.log` file created when a member is deleted, see Appendix D.

21.7 Adding a Deleted Member Back into the Domain

To add a member back into the domain after deleting it as described in Section 21.6, run the following command on the management server:

```
atlas0# sra install -nodes atlas2
```

See Chapter 7 of the *HP AlphaServer SC Installation Guide* for information on running the `sra install` command.

21.8 Reinstalling a Domain

If possible, you should reinstall a domain from the Tru64 UNIX operating system disk, as described in Section 21.8.1. If it is not possible to reinstall the domain from the Tru64 UNIX operating system disk, perform a complete reinstall, as described in Section 21.8.2.

In a domain composed of HP AlphaServer ES45 or HP AlphaServer ES40 nodes, the Tru64 UNIX operating system disk is the third local disk on member 1.

In a domain composed of HP AlphaServer DS20L nodes, the Tru64 UNIX operating system disk is the fourth RAID virtual disk within the external storage configuration and is visible only to member 1 through the use of access paths.

21.8.1 Reinstalling a Domain From the Tru64 UNIX Disk

In some cases, a domain can be re-created from the Tru64 UNIX operating system disk — thus avoiding some of the earlier steps of the installation process on that domain.

To reinstall a complete domain (for example, `atlasD2`) from the Tru64 UNIX operating system disk, perform the following steps from the management server:

1. Back up all site-specific changes.
2. Ensure that all of the nodes in the domain are at the SRM prompt, as follows:
   ```
   atlasms# sra reset -domain atlasD2
   ```
3. Boot the first member of the domain from the Tru64 UNIX operating system disk. In the following example, `atlas64` is the first member of `atlasD2`, and `dka2` is the Tru64 UNIX operating system disk:
   ```
   atlasms# sra -cl atlas64
   P00>>>boot dka2
   .
   .
   .
   Compaq Tru64 UNIX V5.1B (Rev. 2650) (atlas64) console

   login:
   ```
4. Press Ctrl/G at the `login:` prompt, to return to the management server prompt.
5. Disable cookies on the management server, as follows:
   ```
   atlasms# sra cookie -enable no
   ```

6. Run the following command to check whether the `srad` daemon is running:
   ```
   atlasms# sra srad_info
   ```
   If necessary, start the `srad` daemon, as follows:
   ```
   atlasms# /sbin/init.d/sra start
   ```
7. Run the `sra install` command, as follows:
   ```
   atlasms# sra install -domain atlasD2 -redo CluCreate
   ```
   See Chapter 7 of the *HP AlphaServer SC Installation Guide* for information on running the `sra install` command.
8. Enable cookies on the management server, as follows:
   ```
   atlasms# sra cookie -enable yes
   ```
9. Restore any site-specific changes recorded in step 1 above.

21.8.2 Reinstalling a Domain Completely

In some cases, it is not possible to re-create a domain from the Tru64 UNIX operating system disk — you must reinstall the domain completely (this process re-creates the Tru64 UNIX operating system disk).

To reinstall a domain (for example, `atlasD2`) completely, perform the following steps from the management server:

1. Back up all site-specific changes.
2. Ensure that all of the nodes in the domain are at the SRM prompt, as follows:
   ```
   atlasms# sra reset -domain atlasD2
   ```
3. Disable cookies on the management server, as follows:
   ```
   atlasms# sra cookie -enable no
   ```
4. Run the `sra install` command, as follows:
   ```
   atlasms# sra install -domain atlasD2 -redo Uninstalled
   ```
   See Chapter 7 of the *HP AlphaServer SC Installation Guide* for information on running the `sra install` command.
5. Enable cookies on the management server, as follows:
   ```
   atlasms# sra cookie -enable yes
   ```
6. Restore any site-specific changes recorded in step 1 above.

21.9 Moving a Domain to Another IP Subnet

This section describes how to move a domain from one IP subnet to another IP subnet. This is usually required only when a site reconfigures its network topology such that the domain's external IP addresses will be on a different IP subnet.

In order of increasing complexity, moving a domain to another IP subnet might involve changing the following items:

- IP addresses associated with the domain's external network interfaces, interface IP aliases, and cluster aliases.
- IP addresses and host names associated with the domain's external network interfaces, interface IP aliases, and cluster aliases.
- IP addresses and host names associated with the domain's external network interfaces, interface IP aliases, and cluster aliases, and the IP addresses and interface names associated with the internal cluster interconnect.

Note:

In order to simplify domain administration, if you change a member's external host name, you might also want to change the host portion of that member's cluster interconnect name. However, because the network used by the cluster interconnect is private to the domain, you can relocate the domain to another network without modifying the IP names and addresses used by the cluster interconnect. The information on how to do this is provided for completeness. Whether or not you modify the host-name portion of cluster interconnect names to match member host names is entirely up to you.

This section provides tables for you to use when gathering information and performing the move. The tables describe the file edits required for each of the three move scenarios. Use the table that corresponds to the type of move you plan to make. Section 21.9.2 on page 21–26 can help you gather information before starting the move, and provides a checklist you can use to keep track of your edits.

Before you apply this procedure, take note of the following requirements:

- The procedure requires a domainwide reboot.
- Several system configuration files require careful editing. Some of these files are context-dependent symbolic links (CDSLs); you must edit the target file for each domain member, and use `cp` rather than `mv` when copying files. (If you use `mv`, you will overwrite the CDSL instead of copying the file to the target of the CDSL.)
- When editing and copying files, you can make mistakes that might stop domain members from booting and forming a domain. Verify edits twice before putting the modified files into place and halting the domain.
- When you modify an IP address, the subnet mask associated with that address might also require modification.

- The tables and steps in this section deal with the most common system configuration files that contain host names and IP addresses. You must review, and possibly edit, any other files such as cluster application availability (CAA) scripts or network resources that might contain host names, cluster aliases, interface IP aliases, or subnet masks.

To move a domain to another IP subnet, perform the following steps:

1. Obtain the IP names and addresses you will need for the move. Use the tables in Section 21.9.2 on page 21–26 to record this information. Note any changes in subnet masks required for the move. If the move will result in the domain using different name servers, note the changes you will need to make to the `/etc/resolv.conf` file.
2. If this move requires changing any physical network connections, make sure the new ones are in place and ready for use.
3. Tell users and other domain and network administrators when the move will occur. If other systems or domains depend on any of the IP names and addresses you plan to change, their administrators will have to coordinate with you; for example, NIS, DNS, or mail servers might be affected. If the domain provides any services that must not be interrupted, make preparations for another system or domain to provide these services while the domain is shut down.
4. Determine where the current domain's IP names and addresses appear in common system configuration files. One way to do this is to use the `clu_get_info` command to get information about the current domain and then use `grep` to search for that information in common system configuration files.
5. Look for any other files, such as CAA scripts, that might contain host names, IP addresses, cluster aliases, interface IP aliases, name aliases, or subnet masks.
6. Make both save and work copies of the configuration files you plan to modify.

   Note:

   If, for example, any CAA scripts or site-specific scripts reference IP addresses, host names, aliases, or subnet masks that will change, make copies of those files and keep track of the changes you make to them.

   ---

7. Use the table that corresponds to the changes you plan to make:
   - Changing external IP addresses only (Section 21.9.1.1 on page 21–22)
   - Changing external IP addresses and host names (Section 21.9.1.2 on page 21–23
   - Changing external and internal IP addresses and host names (Section 21.9.1.3 on page 21–24)

a. Using the information in the table, edit the work copies of the files (not the originals). For CDSLs, remember to edit the work files in the member-specific directories.

Note:

Whether you use SysMan or an editor to make the edits is up to you. Making the correct edits is important. You must be 100 percent certain that the information is correct before you copy the work files to the original file locations and halt all domain members.

When editing the work files, if the subnet mask on the new subnet is not the same as the one for the current subnet, remember to make those edits.

If you edit the `/.rhosts` or `/etc/hosts.equiv` work files, add new host names but do not remove those currently listed there. You will remove the old host names after you have successfully booted the domain on the other subnet.

b. After you have made all edits, compare the contents of the work files to the original files. Verify your edits.

8. After all edits are made to your satisfaction:
   a. Send messages to everyone who will be affected; tell them when you plan to disable access to the domain and start the actual move.
   b. Disable logins and all external network interfaces. For example, you can use the `wall -c` command to tell users how soon you will disable logins, then use the `touch` command to create the `/etc/nologin` file, and then use the `rcinet stop` command on each member to stop the network on that member.
   c. Copy the work files to the original file locations. Keep track of each file you replace with an edited version. (Remember to keep track of any other files you modify.)
   d. After all the edited configuration files are in place, make sure that all the information is correct. When you shut down the domain, the success of the reboot depends on your edits. Depending on the extent of the changes required for the move, verify the correct modifications of some or all of the following items:
      – IP addresses, interface IP aliases, and subnet masks
      – Host names and name aliases
      – Default cluster alias and other cluster aliases
9. Halt each member of the domain. For example, on each member, run the following command:

```
# shutdown -h now
```

Note:

Because the `shutdown -c` command gets some of its information from the files you just edited, you must halt each member. (If you run the `clu_get_info -full` command after the edited files are in place, some of the displayed information will reflect the edits.)

10. When the domain has halted, make any required changes to the network connections.
11. Boot the domain.
12. Use the information in Section 21.9.3 on page 21–31 to determine whether everything is working as expected.
13. Remove outdated entries from `/.rhosts` and `/etc/hosts.equiv`.

21.9.1 File Edit Tables

This section provides tables that list the required edits to common system configuration files for the different types of moves:

- Changing external IP addresses only (Section 21.9.1.1 on page 21–22)
- Changing external IP addresses and host names (Section 21.9.1.2 on page 21–23)
- Changing external and internal IP addresses and host names (Section 21.9.1.3 on page 21–24)

21.9.1.1 Changing External IP Addresses Only

For common configuration files, Table 21–7 lists items that you may need to edit when changing a domain's external IP addresses.

Table 21–7 Changing External IP Addresses - Items to Edit

File	Items to Edit
	Shared Files
`/etc/hosts`	IP addresses associated with each of the domain's external interfaces and cluster aliases.
`/etc/networks`	If it is defined, the address of the network.
`/etc/resolv.conf`	If the domain will use different name servers on the new subnet, their addresses.

Table 21–7 Changing External IP Addresses - Items to Edit

File	Items to Edit
	CDSLs
/etc/clu_alias.config	For each member, any alias that is defined by its IP address (and possibly a subnet mask). (You probably do not have to modify the DEFAULTALIAS entry.)
/etc/inet.local	For each member, if this file is used to configure interface IP aliases.
/etc/ntp.conf	For each member, if the IP address changes for a server or peer system entry.
/etc/rc.config	For each member, the IFCONFIG entries. If necessary, modify subnet masks.
/etc/routes	For each member, if routes are defined. If necessary, modify subnet masks.
	Member-Specific But Not CDSLs
/etc/gated.conf.member*n*	For each member, any entry whose IP address must change. If CLUAMGR_ROUTE_ARGS is set to nogated in a member's rc.config file, modify that member's /etc/gated.conf file (CDSL).
/cluster/admin/.member*n*.cfg	If you use these files, update any changes to IP addresses. (Otherwise you will reset the domain to the old values if you use the files with clu_create or clu_add_member.)

21.9.1.2 Changing External IP Addresses and Host Names

For common configuration files, Table 21–8 lists items that you may need to edit when changing a domain's external IP addresses and host names.

Table 21–8 Changing External IP Addresses and Host Names - Items to Edit

File	Items to Edit
	Shared Files
/.rhosts	Add the new domain name. Do not remove any entries.
/etc/cfgmgr.auth	Member host names.
/etc/hosts	IP addresses, host names, and aliases associated with the domain's external interfaces and cluster aliases.
/etc/hosts.equiv	Add the new domain name. Do not remove any entries.
/etc/networks	If it is defined, the address of the network.

Table 21–8 Changing External IP Addresses and Host Names - Items to Edit

File	Items to Edit
/etc/resolv.conf	If you are changing the domain name or name servers on the new subnet.
CDSLs	
/etc/clu_alias.config	For each member, any alias that is defined by its host name or IP address (and possibly a subnet mask). (You probably do not have to modify the DEFAULTALIAS entry.)
/etc/inet.local	For each member, if this file is used to configure interface IP aliases.
/etc/ntp.conf	For each member, if the IP address or host name changes for a server or peer system entry.
/etc/rc.config	For each member, the HOSTNAME, IFCONFIG, and CLUSTER_NET entries. If necessary, modify subnet masks.
/etc/routes	For each member, if routes are defined. If necessary, modify subnet masks.
/etc/sysconfigtab	For each member, the cluster_name and the cluster_node_name.
Member-Specific But Not CDSLs	
/etc/gated.conf.member*n*	For each member, any entry whose IP address must change. If CLUAMGR_ROUTE_ARGS is set to nogated in a member's rc.config file, modify that member's /etc/gated.conf file (CDSL).
/cluster/admin/.member*n*.cfg	If you use these files, update any changes to domain names, host names, and IP addresses. (Otherwise you will reset the domain to the old values if you use the files with clu_create or clu_add_member.)

21.9.1.3 Changing External and Internal IP Addresses and Host Names

For common configuration files, Table 21–9 lists items that you may need to edit when changing a domain's external and internal IP addresses and host names.

Table 21–9 Changing External and Internal IP Addresses and Host Names - Items to Edit

File	Items to Edit
Shared Files	
/.rhosts	Add the new domain name and the new host names associated with the cluster interconnect. Do not remove any entries.

Table 21–9 Changing External and Internal IP Addresses and Host Names - Items to Edit

File	Items to Edit
/etc/cfgmgr.auth	Member host names.
/etc/hosts	IP addresses, host names, and aliases associated with the domain's external interfaces, cluster aliases, and cluster interconnect interfaces.
/etc/hosts.equiv	Add the new domain name and the new host names associated with the cluster interconnect. Do not remove any entries.
/etc/networks	If it is defined, the address of the network.
/etc/resolv.conf	If you are changing the domain name or name servers on the new subnet.
CDSLs	
/etc/clu_alias.config	For each member, any alias that is defined by its host name or IP address (and possibly a subnet mask). (You probably do not have to modify the DEFAULTALIAS entry.)
/etc/ifaccess.conf	IP addresses associated with the cluster interconnect. If necessary, modify subnet masks.
/etc/inet.local	For each member, if this file is used to configure interface IP aliases.
/etc/ntp.conf	For each member, if the IP address or host name changes for a server or peer system entry. (If you change the names associated with the cluster interconnect, make sure to change those peer names.)
/etc/rc.config	For each member, the HOSTNAME, IFCONFIG, and CLUSTER_NET entries. If necessary, modify subnet masks.
/etc/routes	For each member, if routes are defined. If necessary, modify subnet masks.
/etc/sysconfigtab	For each member, the value of cluster_name, cluster_node_name, and cluster_node_inter_name.
Member-Specific But Not CDSLs	
/etc/gated.conf.member*n*	For each member, any entry whose IP address must change. If CLUAMGR_ROUTE_ARGS is set to nogated in a member's rc.config file, modify that member's /etc/gated.conf file (CDSL).
/cluster/admin/.member*n*.cfg	If you use these files, update any changes to domain names, host names, cluster interconnect names, and IP addresses. (Otherwise you will reset the domain to the old values if you use the files with clu_create or clu_add_member.)

21.9.2 Attribute and Checklist Tables

Use the tables in this section to record the IP name and address information you will need to move the domain to its new subnet. For additional members and aliases, make copies of the pertinent tables and relabel rows as needed.

21.9.2.1 External Host Names and IP Addresses

Use Table 21–10 to record external host names and IP addresses.

Table 21–10 External Host Names and IP Addresses

Member	Attribute	Value	
Member 1	Host Name	Old	
		New	
	IP Address (and subnet mask)	Old	
		New	
Member 2	Host Name	Old	
		New	
	IP Address (and subnet mask)	Old	
		New	
Member 3	Host Name	Old	
		New	
	IP Address (and subnet mask)	Old	
		New	
Member 4	Host Name	Old	
		New	
	IP Address (and subnet mask)	Old	
		New	

21.9.2.2 Domain Name and Cluster Aliases

Use Table 21–11 to record domain name and cluster alias values.

Table 21–11 Domain Name and Cluster Aliases

Cluster Alias	Value	
Fully qualified domain name (the domain name is the default cluster alias)	Old	
	New	
Default cluster alias IP address (and subnet mask)	Old	
	New	
Name of additional cluster alias #1	Old	
	New	
IP Address of additional cluster alias #1 (and subnet mask)	Old	
	New	
Name of additional cluster alias #2	Old	
	New	
IP Address of additional cluster alias #2 (and subnet mask)	Old	
	New	

21.9.2.3 Interface IP Aliases

Use Table 21–12 to record interface IP aliases.

Table 21–12 Interface IP Aliases

Member	Attribute	Value	
Member 1	IP Alias #1 (and subnet mask)	Old	
		New	
	IP Alias #2 (and subnet mask)	Old	
		New	

Table 21–12 Interface IP Aliases

Member	Attribute	Value	
Member 2	IP Alias #1 (and subnet mask)	Old	
		New	
	IP Alias #2 (and subnet mask)	Old	
		New	
Member 3	IP Alias #1 (and subnet mask)	Old	
		New	
	IP Alias #2 (and subnet mask)	Old	
		New	
Member 4	IP Alias #1 (and subnet mask)	Old	
		New	
	IP Alias #2 (and subnet mask)	Old	
		New	

21.9.2.4 External Servers

If the domain will use different servers for network services such as BIND, NIS, or NTP on the new subnet, record the old and new IP addresses used by these services in Table 21–13.

Table 21–13 External Servers

Server	IP Address	
	Old	
	New	
	Old	
	New	
	Old	
	New	
	Old	
	New	

21.9.2.5 Checklist

Table 21–14 provides a checklist for keeping track of edits to the work copies of configuration files.

Table 21–14 Checklist

File	Status	
Shared Files		
/.rhosts		
/etc/cfgmgr.auth		
/etc/hosts		
/etc/hosts.equiv		
/etc/networks		
/etc/resolv.conf		
CDSLs		
/etc/clu_alias.config	Member 1	
	Member 2	
	Member 3	
	Member 4	
/etc/ifaccess.conf	Member 1	
	Member 2	
	Member 3	
	Member 4	
/etc/inet.local	Member 1	
	Member 2	
	Member 3	
	Member 4	

Table 21–14 Checklist

File	Status	
/etc/ntp.conf	Member 1	
	Member 2	
	Member 3	
	Member 4	
/etc/rc.config	Member 1	
	Member 2	
	Member 3	
	Member 4	
/etc/routes	Member 1	
	Member 2	
	Member 3	
	Member 4	
/etc/sysconfigtab	Member 1	
	Member 2	
	Member 3	
	Member 4	
Member-Specific But Not CDSLs		
/etc/gated.conf.member*n*	Member 1	
	Member 2	
	Member 3	
	Member 4	
/cluster/admin/.member*n*.cfg	Member 1	
	Member 2	
	Member 3	
	Member 4	

21.9.3 Verifying Success

After you apply the procedure for moving a domain to another IP subnet, you can verify whether it was successful.

If all edits are made correctly, and the edited files are put in their proper places, the systems will boot, form a domain, and assume their new identities. Use the following commands to verify that the domain and its subsystems are operating correctly:

```
# hostname
# ifconfig -a
# netstat -i
# clu_get_info -full | more
# cluamgr -s all
# caa_stat
```

You can also use standard networking commands such as `ping`, `rlogin`, and `rpcinfo` to verify that the domain members are available for use, will accept logins, and can communicate with other systems.

If the procedure was not successful, see Section 21.9.4 on page 21–32 for information about identifying and solving problems.

21.9.4 Troubleshooting

Table 21–15 shows possible solutions, if you determine that the procedure as described in Section 21.9.3 on page 21–31 was not successful.

Table 21–15 Possible Solutions for Problems Moving Domain to Another IP Subnet

Problem	Possible Solutions
Member cannot boot.	If the failure occurs early in the boot path, you probably made a mistake editing the `sysconfigtab` file. If you can get one member to boot, mount the `boot_partition` of the member that cannot boot, and fix the edits. If no members can boot, first determine whether any members fail because they cannot gain quorum. If so, perform an interactive boot for one member and set the value of expected votes to zero. Boot that member, fix the other members' files, boot those members, and readjust the expected votes for quorum. If you cannot boot any members interactively, boot the Tru64 UNIX operating system, mount the boot_partition for the first member of the domain, fix the edits for that member, halt the Tru64 UNIX system, boot the domain member (interactively, if necessary), and fix the remaining members.
Domain boots but some network problems were encountered in multiuser mode.	Define the problem and decide which files on which members are most likely to have bad edits. Fix the edits, then stop and restart network services on those systems.
Even after applying the preceding solutions, you are unsuccessful.	Restore the saved copies of the original files. Restore the original network connections. Boot the domain on its old subnet so the domain can continue to serve clients while you figure out what went wrong.

21.10 Updating ifaccess.conf When Changing Network Interfaces

Whenever you add a new network interface, or change or replace an existing interface in a manner that causes the known interfaces to change, you must update the `/etc/ifaccess.conf` file on the domain member where the change occurred. You do this to deny access from untrusted subnets to the cluster interconnect.

Make the change as follows:

1. Log in to the member where the network interface changed.

2. Use the `ifconfig` command to learn the names of the network interfaces. For example:
```
# ifconfig -l
ics0 lo0 sl0 ee0 ee1 ee2 tu0 tun0
```

3. Determine the cluster interconnect addresses used by the member. To do this, look in the member's `/etc/ifaccess.conf` file for the `10.1.xxx.xxx` and `10.0.xxx.xxx` entries.

   You can also use the `ifconfig` command to determine the cluster interconnect addresses used by a member and to identify the network interface for the cluster interconnect. The following example shows typical output:
```
# ifconfig -a | grep -p CLUIF
ics0: flags=1100063<UP,BROADCAST,NOTRAILERS,RUNNING,NOCHECKSUM,CLUIF>
    inet 10.0.0.1 netmask ffffff00 broadcast 10.0.0.255 ipmtu 7000
```

4. Edit the `/etc/ifaccess.conf` file appropriately for each changed interface. Use the following format for each line:

   interface-name interconnect_net_address `255.255.255.0 deny`

   This line is read as "deny all packets to *interface-name* from hosts that claim they originated from *interconnect_net_address*." The mask contains 1s (ones) for the bit positions in *interconnect_net_address* that are significant.

   For example, if the address of the cluster interconnect virtual interface is 10.0.0.1 and a new `tu0` interface card has been added to a domain member, add a line similar to the following to the `/etc/ifaccess.conf` file:
```
tu0 10.0.0.1 255.255.255.0 deny
```

   This line is read as "deny all packets to `tu0` from hosts that claim they originated from `10.0.0`."

   Note:

   By default, the `ifaccess.conf` file is set up only for the head node of a domain (that is, member 1) at installation time. If other domain members have external network connections, the `ifaccess.conf` files for these members must be set up manually.

For more information, see the `ifaccess.conf(4)` reference page.

21.11 Managing Software Licenses

If you install a new product on your HP AlphaServer SC system, you must register the software license on each node, using the License Management Facility (LMF).

Copy the LMF registration script (for example, `new.pak`) to a file system that is accessible to all nodes (for example, `/global`), and then run the following command from the management server, to update the license database on every node:

```
# scrun -n all '/global/new.pak'
```

21.12 Updating System Firmware

Table 21–16 lists the minimum firmware versions supported by HP AlphaServer SC Version 2.6.

Table 21–16 Minimum System Firmware Versions

Firmware	HP AlphaServer DS20E	HP AlphaServer DS20L	HP AlphaServer ES40	HP AlphaServer ES45
SRM Console	6.2-1	6.3-1	6.2-1	6.2-8
ARC Console	Not displayed	Not displayed	5.71	Not displayed
OpenVMS PALcode	1.96-77	1.90-71	1.96-103	1.96-39
Tru64 UNIX PALcode	1.90-72	1.86-68	1.90-104	1.90-30
Serial ROM	1.13-44	Not displayed	2.12-F	2.20-F
RMC ROM	Not displayed	Not displayed	1.0	1.0
RMC Flash ROM	Not displayed	Not displayed	2.5	1.9

Check that your system meets these minimum system firmware requirements.

Assuming that nodes `atlas[1-1023]` are at the SRM prompt, you can use the following command to identify the SRM and ARC console firmware revisions:

```
atlasms# sra command -nodes 'atlas[1-1023]' -command 'show config | grep Console'
```

This command will produce output for each node.

Note that this command does not display the version number for all of the firmware components — it does not show PALcode, Serial ROM, RMC ROM, or RMC Flash ROM.

21.12.1 Updating System Firmware When Using a Management Server

To update the system firmware when using a management server, perform the following tasks:

1. Download the `bootp` version of the firmware from the following URL:
   `http://h18007.www1.hp.com/support/files/index.html`
2. Copy the firmware file into the `/tmp` directory on the RIS server — that is, the management server.
3. Shut down all nodes in the system, except the management server.
4. Execute the following command on the management server, where `es40_v6_2.exe` is the firmware file downloaded in step 2 above:
   ```
   atlasms# sra update_firmware -nodes all -file /tmp/es40_v6_2.exe
   ```

21.12.2 Updating System Firmware When Not Using a Management Server

Note:

The instructions in this section are for an HP AlphaServer SC system with the recommended configuration — that is, the first three nodes in each domain have a vote, and so any two of these nodes can form a domain.

To update the system firmware when not using a management server, perform the following tasks:

1. Download the `bootp` version of the firmware from the following URL:
   `http://h18007.www1.hp.com/support/files/index.html`
2. Copy the firmware file into the `/tmp` directory on the RIS server — that is, Node 0 — and into the `/tmp` directory on Node 1.
3. Shut down all domain members except Node 0 and Node 1.
4. Update the firmware on all nodes except Node 0 and Node 1, as follows:
   ```
   atlas0# sra update_firmware -nodes 'atlas[2-31]' -file /tmp/es40_v6_2.exe
   ```
   where `es40_v6_2.exe` is the firmware file downloaded in step 2 above.
5. Boot Node 2, as follows:
   ```
   atlas0# sra boot -nodes atlas2
   ```
6. Shut down Node 1, as follows:
   ```
   atlas0# sra shutdown -nodes atlas1
   ```
7. Update the firmware on Node 1, as follows:
   ```
   atlas0# sra update_firmware -nodes atlas1 -file /tmp/es40_v6_2.exe
   ```

8. Boot Node 1, as follows:
```
atlas0# sra boot -nodes atlas1
```
9. Shut down Node 0 by running the following command on either Node 1 or Node 2:
```
atlas1# sra shutdown -nodes atlas0
```
10. Update the firmware on Node 0, as follows:
```
atlas1# sra update_firmware -nodes atlas0 -file /tmp/es40_v6_2.exe
```
11. Boot the remaining nodes, as follows:
```
atlas1# sra boot -nodes 'atlas[0,3-31]'
```

21.13 Updating the Generic Kernel After Installation

If you rebuild the generic kernel (`genvmunix`) on a domain after installation (for example, after installing a patch), you must ensure that the kernel is accessible to any subsequent `sra install` commands.

Run the following command to copy the generic kernel that has been built in the `/sys/GENERIC/vmunix` file to each member's boot partition, and to the `/usr/opt/TruCluster/clu_genvmunix` file:
```
# Deploykernels -g
```

21.14 Changing a Node's Ethernet Card

The hardware address of each node's Ethernet card is stored in the SC database and in the RIS database. When you change the Ethernet card on a node, you must update each of these databases, as follows:

1. Ensure that the updated node is at the SRM console prompt.
2. Update the SC and RIS databases using the `sra edit` command, as follows:
```
# sra edit
sra> node
node> edit atlas1
```
This displays a list of node-specific settings, including the Ethernet card hardware address:
```
[7]     Hardware address (MAC)                    00-00-F8-1B-2E-BA

edit? 7

enter a new value, probe or auto
auto = generate value from system
probe = probe hardware for value

Hardware address (MAC)                    [00-00-F8-1B-2E-BA] (set)
new value? probe

Hardware address (MAC)                    [08-00-2B-C3-2D-4C] (probed)
correct? [y|n] y

Remote Installation Services (ris) should be updated
Update RIS? [yes]:y
Gateway for subnet 10 is 10.128.0.1
```

```
Setup RIS for host atlas1
node> quit
sra> quit
```

Note:

When prompted for the new value, enter **probe** — this causes the following actions:

a. The `sra` command connects to the node to determine that hardware address.

b. The RIS database is updated.

21.15 Managing Swap Space

Note:

This section is provided for information only. Swap is automatically configured by the `sra install` command. You can change the preconfigured swap space values for new members, by using the `sra edit` command. See Chapter 16 for more information about the `sra edit` command.

Lazy swap is not supported in HP AlphaServer SC Version 2.6 — use eager swap only.

When eager swap is used, swap space is reserved when modifiable virtual address space is created. The Tru64 UNIX default for swap space allocation is eager mode. Lazy swap space is not reserved until the system needs to write a modified virtual page to swap space. Using lazy swap can result in non-deterministic behavior of the system in low swap situations. If swap is not preallocated (eager mode), seemingly random applications may fail if page faults occur, as the applications' previously swapped data may no longer be available. Lazy swap is therefore not supported for use within HP AlphaServer SC systems.

Put each member's swap information in that member's `sysconfigtab` file. Do not put any swap information in the domainwide `/etc/fstab` file. Since Tru64 UNIX Version 5.0, the list of swap devices has been 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 HP *Tru64 UNIX System Administration* manual and the `swapon(8)` reference page.

Swap information is identified by the `swapdevice` attribute in the `vm` section of the `/etc/sysconfigtab` file. The format for swap information is as follows:
`swapdevice=`*`disk_partition`*`,`*`disk_partition`*`,...`

For example:

```
swapdevice=/dev/disk/dsk2b,/dev/disk/dsk2f
```

Specifying swap entries in the `/etc/fstab` file does not work in a domain because the `/etc/fstab` file is not member-specific; it is a domainwide file. If swap were specified in the `/etc/fstab` file, the first member to boot and form a domain would read and mount all the file systems in the `/etc/fstab` file — the other members would never see that swap space.

The file `/etc/sysconfigtab` is a context-dependent symbolic link (CDSL), so 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 an alternate boot disk is in use, that swap space is also added to this device.

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 shutdown-and-boot. For example, to configure `dsk2f` for use as a secondary swap device for a member already using `dsk2b` for swap, enter the following command:

```
# swapon -s /dev/disk/dsk2f
```

Then, edit that member's `/etc/sysconfigtab` and add `/dev/disk/dsk2f`. The final entry in `/etc/sysconfigtab` will look like the following:

```
swapdevice=/dev/disk/dsk2b,/dev/disk/dsk2f
```

21.15.1 Increasing Swap Space

Section 21.15.1.1 on page 21–39 describes how to increase swap space on an HP AlphaServer DS20L node. On an HP AlphaServer ES45 or HP AlphaServer ES40 node, you can increase the swap space by resizing the boot disk. The method varies depending on whether you resize the primary boot disk (see Section 21.15.1.2 on page 21–39) or the alternate boot disk (see Section 21.15.1.3 on page 21–40). You can resize both disks, but this is not mandatory.

Caution:

Increasing the swap space on either the primary or alternate boot disk will involve repartitioning the disk; this may destroy any data on the disk. The boot partition (partition `a`) will be automatically recreated; however, the `/tmp` and `/local` partitions will not. Before resizing the swap partition, you should back up the data on the `/tmp` and `/local` partitions.

21.15.1.1 Increasing Swap Space on an HP AlphaServer DS20L Node

An HP AlphaServer DS20L node does not have an alternate boot disk, so you cannot increase its swap space by resizing the primary boot disk partitions. The only way to revise the partition sizes on the primary boot disk of an HP AlphaServer DS20L node is to perform the following steps:

1. Delete the node from the domain, as described in Section 21.6 on page 21–15.
2. Use the `sra edit` command to change the sizes of the `swap`, `tmp`, and `local` partitions on the primary boot disk, as described in step 4 of Section 21.15.1.2.
3. Reinstall the node, as described in Section 21.7 on page 21–17.

21.15.1.2 Increasing Swap Space by Resizing the Primary Boot Disk

To increase swap space by resizing the primary boot disk, perform the following steps:

1. Shut the member down, as follows (where `atlas5` is an example name of a non-voting member):
   ```
   # sra shutdown -nodes atlas5
   ```
   For more information about shutting down a member, see Chapter 2.
2. Switch to the alternate boot disk by running the following command:
   ```
   # sra switch_boot_disk -nodes atlas5
   ```
3. Boot the system, as follows:
   ```
   # sra boot -nodes atlas5
   ```
   The `sra boot` command will automatically use the alternate boot disk — the primary boot disk is not in the swap device list.
4. Run the `sra edit` command to change the sizes of the `swap`, `tmp`, and `local` partitions on the primary boot disk, as shown in the following example.

   Note:

   If you change the size of any of the boot disk partitions — `swap`, `tmp`, or `local` — you must resize the other partitions so that the total size is always 100(%). Calculate these sizes carefully, as the `sra edit` command does not validate the partition sizes.

```
# sra edit
sra> node
node> edit atlas5
Id        Description                                    Value
------------------------------------------------------------------
[0  ]     Hostname                                       atlas5         *
```

```
[1  ]      DECserver name                              atlas-tc1       *
.
.
.
[24 ]      im00:swap partition size (%)                15              *
[25 ]      im00:tmp partition size (%)                 42              *
[26 ]      im00:local partition size (%)               43              *
.
.
.
* = default generated from system
# = no default value exists
------------------------------------------------------------------
Select attributes to edit, q to quit
eg. 1-5 10 15
edit? 24-26
im00:swap partition size (%)                    [15]
new value? 30
im00:swap partition size (%)                    [30]
correct? [y|n] y
im00:tmp partition size (%)                     [42]
new value? 35
im00:tmp partition size (%)                     [35]
correct? [y|n] y
im00:local partition size (%)                   [43]
new value? 35
im00:local partition size (%)                   [35]
correct? [y|n] y
node> quit
sra> quit
```

5. Re-partition the primary boot disk, and copy the boot partition from the alternate boot disk to the primary boot disk, as follows:
   ```
   # sra copy_boot_disk -nodes atlas5
   ```
6. Switch back to the primary boot disk by running the following commands:
   ```
   # sra shutdown -nodes atlas5
   # sra switch_boot_disk -nodes atlas5
   # sra boot -nodes atlas5
   ```
7. If the updated node is a member of the currently active RMS partition, stop and start the partition.

21.15.1.3 Increasing Swap Space by Resizing the Alternate Boot Disk

To increase swap space by resizing the alternate boot disk, perform the following steps:

1. Edit the `/etc/sysconfigtab` file to find the `swapdevice` list, and delete the entry for the alternate boot disk.

Note:

When you boot off the primary boot disk, the alternate boot disk is included in the list of swap devices. It is not possible to partition a disk when it is in use. Therefore, you must remove the alternate boot disk from the `swapdevice` list in the `/etc/sysconfigtab` file.

2. Set the SC_USE_ALT_BOOT entry to 0 (zero) in the `/etc/rc.config` file, as follows:
```
# rcmgr set SC_USE_ALT_BOOT 0
```
3. Shut down and boot the member, as follows (in this example, `atlas5` is a non-voting member):
```
# sra shutdown -nodes atlas5
# sra boot -nodes atlas5
```
For more information about shutting down and booting a member, see Chapter 2.

4. Run the `sra edit` command to change the sizes of the `swap`, `tmp`, and `local` partitions on the alternate boot disk, as shown in the following example:

Note:

If you change the size of any of the boot disk partitions — `swap`, `tmp`, or `local` — you must resize the other partitions so that the total size is always 100(%). Calculate these sizes carefully, as the `sra edit` command does not validate the partition sizes.

```
# sra edit
sra> node
node> edit atlas5
Id         Description                                    Value
------------------------------------------------------------------
[0  ]      Hostname                                       atlas5          *
[1  ]      DECserver name                                 atlas-tc1       *
.
.
.
[33 ]      im01:swap partition size (%)                   15              *
[34 ]      im01:tmp partition size (%)                    42              *
[35 ]      im01:local partition size (%)                  43              *
.
.
.
* = default generated from system
# = no default value exists
------------------------------------------------------------------
Select attributes to edit, q to quit
eg. 1-5 10 15
edit? 33-35
im01:swap partition size (%)                    [15]
```

```
new value? 30
im01:swap partition size (%)                    [30]
correct? [y|n] y
im01:tmp partition size (%)                     [42]
new value? 35
im01:tmp partition size (%)                     [35]
correct? [y|n] y
im01:local partition size (%)                   [43]
new value? 35
im01:local partition size (%)                   [35]
correct? [y|n] y
node> quit
sra> quit
```

5. Set the SC_USE_ALT_BOOT entry to 1 in the `/etc/rc.config` file, as follows:
   `# rcmgr set SC_USE_ALT_BOOT 1`
6. Re-partition the alternate boot disk, and copy the contents of the primary boot disk to the alternate boot disk, as follows:
   `# sra copy_boot_disk -nodes atlas5`
7. Edit the `/etc/sysconfigtab` file to find the `swapdevice` list, and re-insert the entry for the alternate boot disk.
8. If the updated node is a member of the currently active RMS partition, stop and start the partition.

21.16 Installing and Deleting Layered Applications

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

21.16.1 Installing an Application

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

21.16.2 Deleting an Application

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

For applications managed by CAA, use the following command to check the status of the highly available applications:

```
# caa_stat -t
```

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

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

Once 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 not currently available, the application is automatically removed from the unavailable member when that member rejoins the domain.

21.17 Managing Accounting Services

The system accounting services are not cluster-aware. The services rely on files and databases that are member-specific, but these files and databases reside on the cluster /var file system.

Because of the large number of members in a domain, it is not appropriate to have all members of a domain maintain their accounting records on a single domain file system — this would put too heavy a loading on AdvFS/CFS to the cluster /var file system when all 32 nodes within a domain are recording process accounting information. Therefore, you should reconfigure the standard UNIX accounting configuration into an HP AlphaServer SC-specific UNIX accounting configuration, where the accounting files and databases will actually reside on member-specific file systems, that are in turn locally served by each member.

To create the HP AlphaServer SC-specific UNIX accounting configuration, you must set up and administer the services on each member, as described in Section 21.17.1. If you later add a new member to a domain, set up HP AlphaServer SC-specific UNIX accounting on the new member as described in Section 21.17.2 on page 21–46. To remove the HP AlphaServer SC-specific UNIX accounting configuration, perform the steps described in Section 21.17.3 on page 21–46.

To check whether the HP AlphaServer SC-specific UNIX accounting configuration is in place, run the /usr/sbin/cdslinvchk command. The following output indicates that the modified configuration is in place:

```
Expected CDSL: ./usr/var/adm/acct -> ../cluster/members/{memb}/adm/acct
An administrator or application has modified this CDSL Target to:
/var/cluster/members/{memb}/adm/acct
```

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 HP *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).

21.17.1 Setting Up HP AlphaServer SC-Specific UNIX Accounting

The nature of the HP AlphaServer SC-specific UNIX accounting configuration is to ensure that each member within a domain records its accounting data on member-local file systems, instead of the default TruCluster Server approach of recording data on a clusterwide file system.

Note:

Before setting up the HP AlphaServer SC-specific UNIX accounting configuration, first create the domains, add the domain members, and boot all members to the UNIX prompt.

You must set up the HP AlphaServer SC-specific UNIX accounting configuration once on the first node of each domain; the changes will apply to all members of the domain. Therefore, repeat all steps — indicated by the atlas0 prompt — on the first node of each additional domain (that is, atlas32, atlas64, atlas96, and so on).

On a large system with many domains, you should automate this process by creating a script. Copy this script (for example, accounting_script) to a file system that is accessible to all nodes in the system, across all domains (for example, /global). You can then run the script on each domain in parallel, as follows:

```
# scrun -d all /global/accounting_script
```

To set up the HP AlphaServer SC-specific UNIX accounting configuration, perform the following steps:

1. Stop accounting by running the shutacct command on all members of the domain, as follows:

   ```
   atlas0# /bin/CluCmd /usr/sbin/acct/shutacct
   ```

   After you have stopped accounting on the members, wait for approximately 10 seconds to ensure that all accounting daemons have finished writing to the accounting files.

2. Remove and invalidate the original accounting directory CDSL, and replace with a symbolic link, by running the following commands:

   ```
   atlas0# rm /var/adm/acct
   atlas0# mkcdsl -i /var/adm/acct
   atlas0# ln -s /var/cluster/members/{memb}/adm/acct /var/adm/acct
   ```

3. Move the original accounting directories to the new locations and create symbolic links, as follows:

```
atlas0# mkdir -p /cluster/members/member0/local/var/adm
atlas0# cp -rp /var/cluster/members/member0/adm/acct \
         /cluster/members/member0/local/var/adm

atlas0# /bin/CluCmd /sbin/mkdir -p /cluster/members/{memb}/local/var/adm
atlas0# /bin/CluCmd /sbin/mv /var/cluster/members/{memb}/adm/acct \
         /cluster/members/{memb}/local/var/adm
atlas0# /bin/CluCmd /sbin/ln -s /cluster/members/{memb}/local/var/adm/acct \
         /var/cluster/members/{memb}/adm/acct
```

4. If you have not already done so, change the permissions on certain accounting files, as follows:

```
atlas0# chmod 664 /cluster/members/member0/local/var/adm/acct/fee
atlas0# chmod 664 /cluster/members/member0/local/var/adm/acct/pacct
atlas0# chmod 664 /cluster/members/member0/local/var/adm/acct/qacct

atlas0# /bin/CluCmd /sbin/chmod 664 \
         /cluster/members/{memb}/local/var/adm/acct/fee
atlas0# /bin/CluCmd /sbin/chmod 664 \
         /cluster/members/{memb}/local/var/adm/acct/pacct
atlas0# /bin/CluCmd /sbin/chmod 664 \
         /cluster/members/{memb}/local/var/adm/acct/qacct
```

5. Remove the existing CDSLs for accounting files, and replace with symbolic links to the new locations:

```
atlas0# rm /var/adm/fee
atlas0# mkcdsl -i /var/adm/fee
atlas0# ln -s /var/adm/acct/fee /var/adm/fee

atlas0# rm /var/adm/pacct
atlas0# mkcdsl -i /var/adm/pacct
atlas0# ln -s /var/adm/acct/pacct /var/adm/pacct

atlas0# rm /var/adm/qacct
atlas0# mkcdsl -i /var/adm/qacct
atlas0# ln -s /var/adm/acct/qacct /var/adm/qacct
```

6. To enable accounting, execute the following command on the first node of the domain:

```
atlas0# rcmgr -c set ACCOUNTING YES
```

   If you wish to enable accounting on only certain members, use the `rcmgr -h` 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
```

7. Start accounting on all members of the domain, as follows:

```
atlas0# /bin/CluCmd /usr/sbin/acct/startup
```

   Alternatively, start accounting by rebooting all members.

8. To test that basic accounting is working, check that the size of the `/var/adm/acct/pacct` file is increasing.
9. To create the ASCII accounting report file `/var/adm/acct/sum/rprt`*mmdd* (where *mmdd* is month and day), run the following commands:
```
atlas0# /usr/sbin/acct/lastlogin
atlas0# /usr/sbin/acct/dodisk
atlas0# /usr/sbin/acct/runacct
```
10. The `sa` command, which summarizes UNIX accounting records, has a hard-coded path for the `pacct` file. To summarize the contents of an alternative `pacct` file, specify the alternative `pacct` file location on the `sa` command line, as follows:
```
atlas0# /usr/sbin/sa -a /var/adm/pacct
```
11. Repeat steps 1 to 10 on the first node of each additional domain.

21.17.2 Setting Up HP AlphaServer SC-Specific UNIX Accounting on a Newly Added Member

If you later add Member *N* to a domain, local accounting directories are not automatically created on the new member. This is because the local file system was not available during the early stages of the `sra install` operation. To set up the HP AlphaServer SC-specific UNIX accounting configuration on the new member, perform the following steps:

1. Stop accounting on the new member, as follows:
```
atlasN# /usr/sbin/acct/shutacct
```
   After you have stopped accounting on the member, wait for approximately 10 seconds to ensure that all accounting daemons have finished writing to the accounting files.
2. Create the local accounting directories, as follows:
```
atlasN# mkdir -p /cluster/members/{memb}/local/var/adm
atlasN# cp -r -p /cluster/members/member0/local/var/adm/acct \
         /cluster/members/{memb}/local/var/adm/acct
atlasN# ln -s /cluster/members/{memb}/local/var/adm/acct \
         /var/cluster/members/{memb}/adm/acct
```
3. Start accounting on the new member, as follows:
```
atlasN# /usr/sbin/acct/startup
```

21.17.3 Removing HP AlphaServer SC-Specific UNIX Accounting

Note:

If you want to remove the HP AlphaServer SC-specific UNIX accounting configuration, you must change the configuration once on the first node of each domain; the changes will apply to all members of the domain. Therefore, repeat all steps — indicated by the `atlas0` prompt — on the first node of each additional domain (that is, `atlas32`, `atlas64`, `atlas96`, and so on).

On a large system with many domains, you should automate this process by creating a script. Copy this script (for example, `undo_accounting_script`) to a file system that is accessible to all nodes in the system, across all domains (for example, `/global`). You can then run the script on each domain in parallel, as follows:

```
# scrun -d all /global/undo_accounting_script
```

To remove the HP AlphaServer SC-specific UNIX accounting configuration, perform the following steps:

1. Stop accounting by running the `shutacct` command on all members of the domain, as follows:

   ```
   atlas0# /bin/CluCmd /usr/sbin/acct/shutacct
   ```

   After you have stopped accounting on the members, wait for approximately 10 seconds to ensure that all accounting daemons have finished writing to the accounting files.

2. Remove the accounting-file symbolic links, as follows:

   ```
   atlas0# /usr/sbin/unlink /var/adm/fee
   atlas0# /usr/sbin/unlink /var/adm/pacct
   atlas0# /usr/sbin/unlink /var/adm/qacct
   ```

   Note:

   Do not create replacement links until step 6.

3. Remove the symbolic links to the accounting directories, and move the accounting directories back to their original locations, as follows:

   ```
   atlas0# /bin/CluCmd /usr/sbin/unlink /var/cluster/members/{memb}/adm/acct
   atlas0# /bin/CluCmd /sbin/mv /cluster/members/{memb}/local/var/adm/acct \
       /var/cluster/members/{memb}/adm
   atlas0# cp -rp /cluster/members/member0/local/var/adm/acct \
       /var/cluster/members/member0/adm
   ```

4. Verify that all of the data is back in its original place, and then remove the directory that you created for member 0, as follows:

   ```
   atlas0# rm -rf /cluster/members/member0/local/var/adm/acct
   ```

5. Remove the `/var/adm/acct` symbolic link, and replace it with a CDSL, as follows:

   ```
   atlas0# /usr/sbin/unlink /var/adm/acct
   atlas0# mkcdsl /var/adm/acct
   ```

   This CDSL points to the `/var/cluster/members/{memb}/adm/acct` directory.

6. Create symbolic links for certain accounting files, as follows:

   ```
   atlas0# ln -s ../cluster/members/{memb}/adm/acct/fee /var/adm/fee
   atlas0# ln -s ../cluster/members/{memb}/adm/acct/pacct /var/adm/pacct
   atlas0# ln -s ../cluster/members/{memb}/adm/acct/qacct /var/adm/qacct
   ```

7. Start accounting on all nodes in the domain, as follows:

   ```
   atlas0# /bin/CluCmd /usr/sbin/acct/startup
   ```

8. Check that all of the links are correct, as follows:
   ```
   atlas0# /usr/sbin/cdslinvchk
   ```
   - If all links are correct, the `cdslinvchk` command returns the following message:
     ```
     Successful CDSL inventory check
     ```
   - If any link is not correct, the `cdslinvchk` command returns the following message:
     ```
     Failed CDSL inventory check. See details in /var/adm/cdsl_check_list
     ```
     If the `Failed` message is displayed, take the appropriate corrective action and rerun the `cdslinvchk` command — repeat until all links are correct.
9. Repeat steps 1 to 8 on the first node of each additional domain.

22

Networking and Network Services

This guide describes how to initially configure services. HP strongly recommends that you do this before the HP AlphaServer SC domains are created. If you wait until after the domains have been created to set up services, the process can be more involved. This chapter describes the procedures to set up network services after the domains have been created.

This chapter discusses the following topics:

- Default Route Configuration (see Section 22.1 on page 22–2)
- Running IP Routers (see Section 22.2 on page 22–6)
- Configuring the Network (see Section 22.3 on page 22–7)
- Configuring DNS/BIND (see Section 22.4 on page 22–8)
- Managing Time Synchronization (see Section 22.5 on page 22–9)
- Configuring NFS (see Section 22.6 on page 22–11)
- Configuring NIS (see Section 22.7 on page 22–21)
- Managing Mail (see Section 22.8 on page 22–22)
- Managing inetd Configuration (see Section 22.9 on page 22–25)
- Displaying X Window Applications Remotely (see Section 22.10 on page 22–25)

See the HP *Tru64 UNIX Network Administration* manuals for information about managing networks on single systems.

22.1 Default Route Configuration

There are two kinds of nodes in an AlphaServer SC domain; nodes that have external interfaces to the public/corporate network, and nodes that do not have external interfaces to the public/corporate network. The terms external and internal are used to describe these nodes.

For external nodes, the physical router for the customer site is used as the default route in the `/etc/routes` file. For internal nodes, there have been various implementations in previous versions of HP AlphaServer SC, as follows:

- In HP AlphaServer SC Version 1.0, the default route was defined as the management interface (typically the `ee0` interface) of the first member of the local domain. Failover was not possible without rebooting the domains.
- In HP AlphaServer SC Version 2.0, Version 2.4 and Version 2.4A, the default route was undefined on the internal nodes. This resulted in the internode communications service (ICS) being used as a default route. While this provided failover, it was not optimal, as ICS is a pseudo-ethernet interface implemented over the HP AlphaServer SC Interconnect and is not well-equipped to handle heavy loads.
- In HP AlphaServer SC Version 2.5 and Version 2.5 (UK1), the default route was defined as the management interface (typically the `ee0` interface) of the first member of the local domain. The `/usr/sra/bin/SetDefaultRoute` script was made available to force the routing failover to member 2 if member 1 failed.

In HP AlphaServer SC Version 2.6, external nodes use `gated` to automatically advertise default routes to internal nodes on defined interfaces. The networks on which default routes are advertised are defined by the `/etc/clua_default` file. These default routes are configurable and they will automatically fail over if the node serving them fails.

There are a number of configuration settings and configuration files on both the internal and external nodes that support this feature, as follows

- For internal nodes, the configuration settings and configuration files are as follows:
  - The `/etc/routes` file is empty.
  - The CLUAMGR_ROUTE_ARGS runtime configuration variable includes the `nogated` parameter; as a result, the `gated` daemon no longer runs under the control of the `aliasd` daemon.
  - The `aliasd_niff` daemon is automatically started with the `-g` parameter by the `aliasd` daemon, to indicate that the `gated` daemon is no longer under the control of the `aliasd` daemon.

- The gated daemon is run with a fixed, member-specific /etc/gated.conf configuration file. This fixed configuration file restricts route learning to only those routing information protocol (RIP) updates that are received from the external nodes on the local domain. Therefore, within any one domain, an internal node will not learn RIP updates from external nodes in a different domain.
- The fixed /etc/gated.conf configuration file contains the following lines:

```
interfaces {
    interface all passive ;
} ;
rip yes
{
    nobroadcast;
} ;
# where aaa.bbb.ccc.ddd reflect the local domain alias
static {
    host aaa.bbb.ccc.ddd interface 127.0.0.1;
};
import proto rip interface 10.0.0.11 restrict;
import proto rip interface 10.128.0.11 restrict;
import proto rip interface 10.64.0.11 restrict;
import proto rip gateway  atlas0 atlas1 {
    default ;
};
```

- The GATED_FLAGS runtime configuration variable is set to indicate the location of the fixed /etc/gated.conf file.

• For external nodes, the configuration settings and configuration files are as follows:

- The gated daemon is run under the control of the aliasd daemon. Therefore, if there is an interface change (for example, a cable disconnect) the aliasd daemon will both recreate the /etc/gated.conf.member*N* file and restart the gated daemon. This ensures that failover happens automatically and transparently, and a suitable default route is always available to the internal members.
- For performance reasons, RIP updates are not advertised over the eip0 network. The eip0 network is a pseudo-ethernet interface and is not optimally designed for broadcast traffic. Therefore, on external nodes, the CLUAMGR_ROUTE_ARGS runtime configuration variable includes the parameter noripout=eip0; as a result, RIP updates are not transmitted over the eip0 network.
- The aliasd daemon uses the domain-wide /etc/clua_default file to determine on which networks a default route is to be advertised. The default routes can be provided over any of the networks that are visible to both internal and external nodes; for example, the management network, the ICS network, and so on. The /etc/clua_default file also contains a metric for each network. If the preferred network becomes unavailable, an alternative is chosen based on the metrics.

For example, to specify two default routes, one over the management LAN and the second over the ICS network, but favoring the management LAN, the `/etc/clua_default` file would contain the following:

```
10.128   2
10.0.0   5
```

On a standard HP AlphaServer SC Version 2.6 system, the `/etc/clua_default` file only contains an entry for the management LAN by default; the ICS network is not specified.

- The domain-wide `/etc/clua_metrics` file can be used to define one network as being preferred over another network (as in a standard TruCluster system). This is important for routing decisions where all nodes are connected to multiple networks. Where a network is not specified in the `/etc/clua_metrics` file, that network is assumed to have a metric of 14. By default, the `/etc/clua_metrics` file on a domain in an HP AlphaServer SC system contains the following line:

  ```
  10.64    16
  ```

  As a result, the `10.64` network (`eip0`) will be least preferred. This is in keeping with the fact that the `eip0` network is a pseudo-ethernet interface that is not optimal for heavy-duty tasks.

- On all domains, the ALIASD_NONIFF runtime configuration variable is set to `-N eip0` in the domain-wide configuration file to indicate that the `eip0` interface is not to be monitored by NIFF. As the quantity of broadcast traffic on the `eip0` interface is deliberately kept low, if this interface was monitored by NIFF, the absence of background broadcast traffic could lead to the interface being falsely classified as inactive.

For background information on the `gated` daemon and routing, see the `gated_intro(7)`, `gated.proto(4)`, `gated.conf(4)`, and `gated.control(4)` reference pages.

22.1.1 Validating the Default Route Configuration

To validate the default route configuration in your HP AlphaServer SC Version 2.6 system, run the following commands:

```
# scrun -n all -w 1 "cat /etc/routes"

# scrun -n all -w 1 "rcmgr get GATED"
# scrun -n all -w 1 "rcmgr get CLUAMGR_ROUTE_ARGS"
# scrun -n all -w 1 "rcmgr get GATED_FLAGS"

# scrun -d all -m "1" -w 1 "cat /etc/gated.conf.member1"
# scrun -d all -m "2" -w 1 "cat /etc/gated.conf.member2"
# scrun -d all -m "[3-]" -w 1 "cat /etc/gated.conf"

# scrun -d all -w 1 "cat /etc/clua_default"
```

```
# scrun -d all -w 1 "cat /etc/clua_metrics"

# scrun -n all -w 1 "ps aux" | grep aliasd | grep -v aliasd_niff
# scrun -n all -w 1 "ps aux" | grep aliasd_niff
# scrun -d all -w 1 "rcmgr -c get ALIASD_NONIFF"
```

Table 22–1 shows the default configuration that these commands will reveal for each of the possible node types on an HP AlphaServer SC Version 2.6 system. Note that some of the rows in this table reveal settings that are needed for the preferred server cluster alias configuration. For more information on preferred server cluster alias configuration, see Section 19.15 on page 19–23.

Table 22–1 Default Route Configuration

Configuration Settings / Files	FS Node with External Interface	FS Node with no External Interface	CS Node with External Interface	CS Node with no External Interface
/etc/routes	<site router>	<empty file>	<site router>	<empty file>
GATED	yes	yes	yes	yes
CLUAMGR_ROUTE_ARGS	noripin=ee0 noripin=ee1 noripin=eip0 noripin=ics0 noripout=eip0	nogated	noripin=ee0 noripin=ee1 noripin=eip0 noripin=ics0 noripout=eip0	nogated
GATED_FLAGS	<null>	-f /etc/gated.conf	<null>	-f /etc/gated.conf
/etc/gated.conf	The /etc/gated.conf.member*N* file is created automatically by the aliasd daemon.	The /etc/gated.conf file is a fixed file as defined above	The /etc/gated.conf.member*N* file is created automatically by the aliasd daemon.	The /etc/gated.conf file is a fixed file as defined above.
/etc/clua_default	10.128 2	10.128 2	10.128 2	10.128 2
/etc/clua_metrics	10.64 16	10.64 16	10.64 16	10.64 16
aliasd daemon	aliasd -N eip0	aliasd -N eip0	aliasd -N eip0	<not running>
aliasd_niff daemon	aliasd_niff	aliasd_niff -g	aliasd_niff	<not running>
ALIASD_NONIFF	-N eip0	-N eip0	-N eip0	-N eip0

22.2 Running IP Routers

The default gated configuration in HP AlphaServer SC Version 2.6 systems can be customized, and this section provides instruction on how to do this. However, as the gated configuration is complex, you should ensure that you have read Section 22.1 on page 22–2 and are familiar with the default configuration before attempting any customization.

Domain 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).

Note:

You should not disable gated on a domain unless every member has an external network interface (and an accompanying static default route).

To run a customized gated configuration on a domain 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 the gated.conf file (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 start,nogated 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 the `cluamgr(8)` reference page.

Note:

The `cluamgr` option `nogated` is not a means to allow the use of `routed`. Only `gated` is supported. HP strongly recommends that domain 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.

22.3 Configuring the Network

The recommended time for configuring various network services is at system installation time before building any domains (see Chapters 5 and 6 of the *HP AlphaServer SC Installation Guide*). Configuring services before building any domains ensures that services are automatically configured as nodes are added to the domain.

Typically, you configure the network when you install the Tru64 UNIX 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`)[1]
- Routing services (`gated`, IP router)
- Hosts file (`/etc/hosts`)
- Hosts equivalency file (`/etc/hosts.equiv`)
- Networks file (`/etc/networks`)
- DHCP server (`joind`) — DHCP is used by RIS, but is not supported in any other way

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.

1. For more information on static routes, see Section 19.15 on page 19–23.

The following configuration tasks require a focus member:

- Network interfaces
- Gateway routing daemon (`gated`)
- Static routes (`/etc/routes`)[1]
- Remote `who` daemon (`rwhod`) — not supported in HP AlphaServer SC
- 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 stop/restart of network services domainwide would be disruptive; therefore, these tasks are performed on one member at a time.

If you do not specify a focus member, the configurations performed are considered to be domainwide, and all configurations are placed in the `/etc/rc.config.common` file.

The following configuration tasks must be run domainwide:

- DHCP server daemon — not supported except for use by RIS
- Hosts (`/etc/hosts`)
- Hosts equivalencies (`/etc/hosts.equiv`)
- Networks (`/etc/networks`)

22.4 Configuring DNS/BIND

Note:

HP AlphaServer SC Version 2.6 supports configuring the system as a DNS/BIND client only — do not configure the system as a DNS/BIND server.

Configuring a domain in an HP AlphaServer SC system as a Domain Name Service (DNS) or Berkeley Internet Name Domain (BIND) client is similar to configuring an individual system as a DNS/BIND client. If a domain member is configured as a DNS/BIND client, then the entire domain is configured as a client.

Whether you configure DNS/BIND at the time of domain creation or after the domain is running, the process is the same, as follows:

1. On any member, run the `bindconfig` command or the `sysman dns` command.
2. Configure DNS/BIND by selecting *Configure system as a DNS client*.

3. When prompted to update hostnames to fully qualified Internet hostnames, enter **No**.

   The hostnames of nodes in an HP AlphaServer SC system must not be fully qualified; that is, hostnames must be of the format `atlas0`, not `atlas0.yoursite.com`.

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

For more information about configuring DNS (BIND), see Chapters 5 and 6 of the *HP AlphaServer SC Installation Guide*. See also the HP *Tru64 UNIX Network Administration* manuals.

22.5 Managing Time Synchronization

All domain members in an HP AlphaServer SC system need time synchronization. Network Time Protocol (NTP) meets this requirement. Because of this, the `sra install` command configures NTP on the initial domain member at the time of domain creation, and NTP is automatically configured on each member as it is added to the domain. All members are configured as NTP peers. If your system includes a management server, you can use the management server as the NTP server for the domains that make up your system.

Note:

NTP is the only time service supported in HP AlphaServer SC systems.

22.5.1 Configuring NTP

The peer entries act to keep all domain members synchronized so that the time offset is in microseconds across the domain. 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 domain is running, use the `ntpconfig` command or the `sysman ntp` command on each domain member. This command always acts on a single domain member. You can either log in to each member or you can use the `sysman -focus` option 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 domain, and should be performed on only one member at a time.

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

22.5.2 All Members Should Use the Same External NTP Servers

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

22.5.2.1 Time Drift

If you notice a time drift between nodes, you must resynchronize against an external reference (NTP server). This might be a management server or an external time server. In a large system, HP recommends the following approach, to minimize requests on the external reference.

In the following example, the external reference is the management server (`atlasms`):

1. Synchronize the management server against another external source.
2. Synchronize the first member of each domain against the management server, by running the following command:
   ```
   # scrun -d all -m 1 ntp -s -f atlasms
   ```
3. Synchronize all members within a domain against the first member of that domain, by running the appropriate command on each domain. For example:
   a. To synchronize the first domain, run the following command:
      ```
      # scrun -d 0 -m [2-32] ntp -s -f atlas0
      ```
   b. To synchronize the second domain, run the following command:
      ```
      # scrun -d 1 -m [2-32] ntp -s -f atlas32
      ```

   and so on.

In a large system, you may need to write a shell script to automate this process. You can identify the name of the first member of a domain by running the following command:

```
# scrun -d domain# -m 1 hostname | awk '{print $2}' -
```

For example:

```
# scrun -d 1 -m 1 hostname | awk '{print $2}' -
atlas32
```

For more information about configuring NTP, see Chapters 5 and 6 of the *HP AlphaServer SC Installation Guide*, and the HP *Tru64 UNIX Network Administration: Services* manual.

22.6 Configuring NFS

The HP AlphaServer SC system can provide highly available Network File System (NFS) service, and can be configured to act as a client to external NFS servers. It can also be used to serve its file systems to external clients. You can use AutoFS with CAA (Cluster Application Availability) to automatically fail-over NFS mounts, thus improving the availability of external NFS file systems within a domain.

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

Note:

To serve file systems between domains, do not use NFS — use SCFS (see Chapter 7).

22.6.1 The HP AlphaServer SC System as an NFS Client

When a domain acts as an NFS client, an NFS file system that is mounted by one domain member is accessible to all domain members: the Cluster File System (CFS) funnels file accesses through the mounting member to the external NFS server. That is, the domain 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 domain members, CFS guarantees that all members at all times have the same view of the NFS file system.

External NFS systems can be mounted manually, by executing the `mount` command. The node on which the `mount` command is issued becomes the client of the external NFS server. The node also serves the file system to internal nodes within the domain.

Note:

On NFS servers that are external to the HP AlphaServer SC system, the `/etc/exports` file must specify the hostname associated with the external interface, instead of the cluster alias — for example, `atlas0-ext1`.

However, in the event that the mounting member becomes unavailable, there is no failover. Access to the NFS file system is lost until another domain 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 domain members. Using AutoFS in a domain environment is described in Section 22.6.5 on page 22–16.

When choosing a node to act as the NFS client, you should select one that has the most suitable external interface — that is, high speed and as near as possible (in network terms) to the file server system. Choosing, for example, a node with no external connection as the client would cause all network traffic for the file system to be routed through a node with an external connection. Such a configuration is not optimal.

If you need to mount multiple external file systems, you can use the same node to act as a client for all file systems. Alternatively, you can spread the load over multiple nodes. The choice will depend on the planned level of remote I/O activity, the configuration of external network interfaces, and the desired balance between compute and I/O.

You must configure at least one node in each domain that is to be configured as an NFS client.

If you wish to routinely mount an external file system on a selected node, but you do not wish to use AutoFS, edit that node's `/etc/member_fstab` file. This file has the same format as `/etc/fstab`, but is used to selectively mount file systems on individual nodes. The `/etc/member_fstab` file is a context-dependent symbolic link (CDSL) to the following file: `/cluster/members/member`*M*`/etc/member_fstab`

The startup script `/sbin/init.d/member_mount` is responsible for mounting the file systems listed in the `/etc/member_fstab` file. Note that the `member_mount` script is called by the `nfsmount` command to mount NFS file systems; it is not executed directly.

Note:

In the `/etc/member_fstab` file, use the `nfs` keyword to denote the file system type. Do not use the `nfsv3` keyword, as this is an old unsupported file system type. The default NFS version is Version 3. To explicitly specify the version, you can include the option `vers=`*n*, where *n* is 2 or 3.

22.6.2 The HP AlphaServer SC System as an NFS Server

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

Other commands that run through `mountd`, such as `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 the `/etc/exports.aliases` file.

Before configuring additional aliases for use as NFS servers, read the sections in the HP *TruCluster Server 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.

Domains within an HP AlphaServer SC system can be configured to export file systems *via* NFS. As stated earlier, you should not export file systems *via* NFS between domains — instead, use SCFS for this purpose. For more information about SCFS, see Chapter 7.

22.6.3 How to Configure NFS

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

To configure NFS, use the `nfsconfig` command or the `sysman nfs` command. With these commands, you can:

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

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

When you configure NFS without any focus, the configuration applies to the entire domain and is saved in `/etc/rc.config.common`. If a focus is specified, then the configuration applies only to the specified domain member, and is saved in the (CDSL) `/etc/rc.config` file for that member.

Local NFS configurations override the domainwide configuration. For example, if you configure member atlas4 as not being an NFS server, then atlas4 is not affected when you configure the entire domain as a server; atlas4 continues not to be a server.

For a more interesting example, suppose you have a 32-member domain atlasD0 with members atlas0, atlas1, ... atlas31. Suppose you configure eight TCP server threads domainwide. If you then set focus on member atlas0 and configure ten TCP server threads, the ps command will show ten TCP server threads on atlas0, but only eight on members atlas1...atlas31. If you then set focus domainwide and set the value from eight TCP server threads to 12, you will find that atlas0 still has ten TCP server threads, but members atlas1...atlas31 now each have 12 TCP server threads.

Note that if a member runs nfsd it must also run mountd, and *vice versa*. This is automatically taken care of when you configure NFS with the sysman command.

If locking is enabled on a domain member, then the rpc.lockd and rpc.statd daemons are started on the member. If locking is configured domainwide, then the lockd and statd run domainwide (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 as its address.

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

You can start and stop services either on a specific member or on the entire domain. Typically, you should not need to manage the domainwide 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 23.

For more information about configuring NFS, see the HP *Tru64 UNIX Network Administration: Services* manual.

22.6.4 Considerations for Using NFS in a Domain

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

22.6.4.1 CFS Support of NFS File Systems

CFS supports the Network File System (NFS) client for read/write access. When a file system is NFS-mounted in a domain, CFS makes it available for read/write access from members of the domain. The member that has actually mounted it serves the file system to other domain members.

If the member that has mounted the NFS file system shuts down or fails, the file system is automatically unmounted and CFS begins to clean up the mount points. During the cleanup process, members that access these mount points may see various types of behavior, depending upon how far the cleanup has progressed:

- If members still have files open on that file system, their writes will be sent to a local cache instead of to the actual NFS-mounted file system.
- After all of the files on that file system have been closed, attempts to open a file on that file system will fail with an `EIO` error until the file system is remounted. Applications may encounter "Stale NFS handle" messages. This is normal behavior on a standalone system, as well as in a domain.

Until the CFS cleanup is complete, members may still be able to create new files at the NFS file system's local mount point (or in any directories that were created locally beneath that mount point).

An NFS file system does not automatically fail over to another domain member unless you are using AutoFS. Rather, you must manually remount it, on the same mount point or another, from another domain member to make it available again. Alternatively, booting a domain member will remount those file systems that are listed in the `/etc/member_fstab` file that are not currently mounted and served in the domain.

22.6.4.2 Clients Must Use a Cluster Alias

Clients must use a cluster alias (not necessarily the default cluster alias) to specify the host when mounting file systems served by the domain, as described in Section 22.6.2 on page 22–13.

22.6.4.3 Loopback Mounts Are Not Supported

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

22.6.4.4 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 domain member goes down.

22.6.4.5 Use AutoFS to Mount File Systems

For more information, see Section 22.6.5.

22.6.5 Mounting NFS File Systems using AutoFS

In an HP AlphaServer SC system, the `srad` and `scmountd` daemons will initiate the automatic failover of SCFS and PFS file systems between member 1 and member 2 where appropriate. Automatic failover for standard NFS file systems can be provided by using 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 `autofsd`, the AutoFS daemon. If this member fails, CAA starts `autofsd` on another member.

For detailed instructions on configuring AutoFS, see the HP *Tru64 UNIX Network Administration: Services* manual.

After you have configured AutoFS, you must register it with CAA and start the daemon, as described in the following steps (where `atlas` is an example system name):

1. Run the `caa_stat -t` command to see if AutoFS is registered with CAA. If not, register AutoFS with CAA, as follows:
   ```
   # caa_register autofs
   ```
2. Restrict AutoFS to run on only those nodes with external interfaces, as follows (in this example, `atlas0` and `atlas1` are the only nodes with external interfaces):
   ```
   # caa_profile -update autofs -p restricted -h 'atlas0 atlas1'
   ```
3. Enable AutoFS by setting the AUTOFS variable to 1 in the `/etc/rc.config.common` file, as follows:
   ```
   # rcmgr -c set AUTOFS 1
   ```
4. If you do not use NIS to manage the automount maps (see below), you must set the AUTOFSMOUNT_ARGS variable in the `/etc/rc.config.common` file, as follows:
   ```
   # rcmgr -c set AUTOFSMOUNT_ARGS '-f /etc/auto.master /- /etc/auto.direct'
   ```

5. Start AutoFS, as follows:
   ```
   # caa_start autofs
   ```
   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 similar to the following:
   ```
   # CAAD[564686]: RTD #0: Action Script \
   /var/cluster/caa/script/autofs.scr(start) timed out! (timeout=180)
   ```
   In this situation, you must 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 240 seconds, enter the following command:
   ```
   # caa_profile -update autofs -o st=240
   ```
   For more information about CAA profiles and using the `caa_profile` command, see the `caa_profile(8)` reference page.

AutoFS mounts NFS file systems that are listed in automount maps. Automount maps are files that may be either stored locally in `/etc`, or served by NIS. See the `automount(8)` reference page for more information about automount maps.

The simplest configuration is to use NIS to export two automount maps, `auto.master` and `auto.direct`, from a server. The files are simpler to set up, and NIS is simpler to maintain.

The `auto.master` map should contain a single entry:
```
/- auto.direct   -rw, intr
```

The `auto.direct` map should list the NFS file systems to be mounted:
```
/usr/users  homeserver:/usr/users
/applications homeserver:/applications
```

In this example, whenever a file or directory in `/usr/users` is accessed, the NFS file system is mounted if necessary. If the mount point does not yet exist, `autofs` will create it. If a file system is not accessed within a set period of time (the default is 50 seconds), it is automatically unmounted by `autofs`.

If you change the automount maps, you should update the automount daemon by running the `autofsmount` command — on each domain mounting the file system — as follows:
```
# autofsmount
```

When you mount NFS file systems using AutoFS, the NFS mounts will automatically failover if the node mounting the file systems is unavailable.

When using AutoFS, keep in mind the following:

- On a domain 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 you display the automounted file systems, a file system might be mounted more than once. This can occur if the kernel times out the automount request, but the NFS mount actually succeeds. If another automount of this file system is then requested, a duplicate mount occurs. This condition does not have an operational impact.
- When the `autofsd` daemon starts or when `autofsmount` runs to process maps for automounted file systems, AutoFS makes sure that all domain members are running the same version of the HP AlphaServer SC TruCluster Server software.

22.6.6 Forcibly Unmounting File Systems Mounted by AutoFS

If AutoFS on a domain member is stopped or becomes unavailable (for example, if the CAA `autofs` resource is stopped), intercept points and file systems that are automounted by AutoFS continue to be available. However, in the case where AutoFS is stopped on a domain 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 domain 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 domain member as the server. These intercept points do not allow new automounts.

22.6.6.1 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 automounted file system.
- You move the CAA `autofs` resource.

In the case where you detect an obvious problem accessing an automounted file system, ensure that the automounted file system is being served as expected. To do this, perform the following steps:

1. Use the `caa_stat autofs` command to identify 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 the `autofs` resource is not running, run it and see whether this fixes the problem.
3. Determine the automount 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 automounted 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:

```
# cfsmgr -e
Domain or filesystem name = /etc/auto.direct
Mounted On = /mnt/mytmp
Server Name = atlas4
Server Status : OK
```

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.

22.6.6.2 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 the busy file systems. 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 automounted 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 automounted 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 automounted 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 automounted file systems are still busy.
3. Enter the following command to verify that all AutoFS intercept points and automounted 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 automounted file systems:
   `# cfsmgr -K directory`
5. Specify the directory on which an AutoFS intercept point or automounted file system is mounted. You need enter only one mounted-on directory to remove all of the intercepts and automounted file systems served by the same node.
6. Enter the following command to start the `autofs` resource:
   `# caa_start autofs -c CFS_domain_member_to_be_server`

For more information about forcibly unmounting an AdvFS file system or domain, see Section 29.4.6 on page 29–12.

22.7 Configuring NIS

Note:

HP AlphaServer SC Version 2.6 supports only the Network Information Service (NIS) configurations described in Table 22–2 on page 22–21.

One way to simplify account management in a large HP AlphaServer SC system is to use NIS. NIS can be used to provide consistent password data and other system data to the domain(s) — and to the optional management server — that make up a large HP AlphaServer SC system.

If you have an existing NIS environment, an HP AlphaServer SC system can be added as a series of NIS clients (each domain). If you do not have an existing NIS environment, the management server (if used) can be configured as a NIS master server, or — if you do not have a management server — Node 0 can be configured as a NIS slave server. Table 22–2 summarizes the NIS configurations supported in HP AlphaServer SC Version 2.6.

Table 22–2 Supported NIS Configurations

Existing NIS Environment?	Management Server?	Supported NIS Configuration[1]
Yes	Yes	Configure the management server as a NIS client.
Yes	No	Configure Node 0 as a slave server.
No	Yes	Configure the management server as a master server.
No	No	Configure Node 0 as a NIS slave server only. Do not configure Node 0 as a NIS master.

[1]In each case, the domains are automatically configured as slave servers by the `sra install` command.

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 domain members.

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

It is not mandatory to configure NIS. However, if you do wish to configure NIS after the domain is running, follow these steps:

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

You must configure NIS as a slave server on an externally connected system. You must supply the host names to which NIS binds. When you have configured NIS, you must add the default cluster alias for each domain (for example, `atlasD0`) as an entry in the list of servers on your NIS master. Otherwise, the slave server will not update properly following changes on the NIS master.

Note:

If you do not configure NIS as a slave server, NIS will not work correctly on nodes that do not have an external network connection.

In a large HP AlphaServer SC system comprising several domains, you must configure NIS on each domain.

2. On each domain member, execute the following commands:

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

For more information about configuring NIS, see the HP *Tru64 UNIX Network Administration: Services* manual.

22.7.1 Configuring a NIS Master in a Domain with Enhanced Security

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

22.8 Managing Mail

HP AlphaServer SC Version 2.6 supports the following mail protocols:

- Simple Mail Transfer Protocol (SMTP)
- Message Transport System (MTS)
- UNIX-to-UNIX Copy Program (UUCP)
- X.25

In a domain in an HP AlphaServer SC system, all members must have the same mail configuration. If SMTP or any other protocol is configured on one domain member, it must be configured on all members, and it must have the same configuration on each member. You can configure the domain as a mail server, client, or as a standalone configuration, but the configuration must be domainwide. 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. This means that 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 domain 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 (MTS, UUCP, and X.25) can run in a domain environment, but they act as if each domain member was a standalone system. Incoming e-mail using one of these protocols must be addressed to an individual domain member, not to the cluster alias. Outgoing email using one of these protocols has as its return address the domain member where the message originated.

Configuring MTS, UUCP, or X.25 in a domain in an HP AlphaServer SC system is like configuring it in a standalone system. It must be configured on each domain member, and any hardware required by the protocol must be installed on each domain member.

The following sections describe managing mail in more detail.

22.8.1 Configuring Mail

Configure mail with either the `mailsetup` or `mailconfig` command. Whichever command you choose, you will have to use it for future mail configuration on the domain, because each command understands only its own configuration format.

22.8.2 Mail Files

The following mail files are all common files shared domainwide:

- `/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`, the `/var/adm/sendmail` directory 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 domain 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.

22.8.3 The Cw Macro (System Nicknames List)

Whether you configure mail with mailsetup or mailconfig, the configuration process automatically adds the names of all domain 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 domain. However, if you do a quick setup in mailsetup, you are not prompted to update the nickname list. The domain members and the cluster alias are still automatically added to the Cw macro.

22.8.4 Configuring Mail at Domain Creation

You must configure mail on your system before you run the sra install command. If you run only SMTP, then you will not need to perform further mail configuration when you add new members to the domain. The sra install command takes care of correctly configuring mail on new members as they are added.

If you configure MTS, UUCP, or X.25, then each time you add a new domain member, you must run mailsetup or mailconfig and configure the protocol on the new member.

Each member must also have any hardware required by the protocol. The protocol(s) must be configured for every domain member, and the configuration of each protocol must be the same on every member.

The mailsetup and mailconfig commands cannot be focused on individual domain members. In the case of SMTP, the commands configure mail for the entire domain. For other mail protocols, the commands configure the protocol only for the domain 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.
```

Deleting members from the domain requires no reconfiguration of mail, regardless of the protocols you are running.

For more information about configuring mail, see the HP *Tru64 UNIX Network Administration: Services* manual.

22.9 Managing inetd Configuration

Configuration data for the Internet server daemon (`inetd`) is stored in the following two files:

- `/etc/inetd.conf`
  Shared domainwide 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 domain member. Use it to configure per-member network services.

To disable a domainwide 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 domainwide in `inetd.conf` and you want to disable it on a member, add a line such as 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`.

22.10 Displaying X Window Applications Remotely

You can configure the domain so that a user on a system outside the domain can run X applications on the domain and display them on his or her 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 displayed from domain members.

In `/etc/clua_services`, the `out_alias` attribute is set for the X server port (6000). A user on a system outside the domain wants to run an X application on a domain member and display it back to his or her system.

Because the `out_alias` attribute is set on port 6000 in the domain, the user must specify the name of the default cluster alias when running the `xhost` command to allow X clients access to his or her local system. For example, for a domain named `atlas`, the user would run the following command on the local system:

```
# xhost +atlas
```

This use of `out_alias` allows any X application from any domain member to display on that user's system, and is required in an HP AlphaServer SC system to allow nodes without external interfaces to connect to an X server on an external network.

For more information on cluster aliases, see Chapter 19.

23

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:

- Introduction (see Section 23.1 on page 23–2)
- Learning the Status of a Resource (see Section 23.2 on page 23–3)
- Reporting Resource Availability Measurements (see Section 23.3 on page 23–9)
- Relocating Applications (see Section 23.4 on page 23–11)
- Starting and Stopping Application Resources (see Section 23.5 on page 23–13)
- Balancing Application Resources (see Section 23.6 on page 23–15)
- Registering and Unregistering Resources (see Section 23.7 on page 23–18)
- HP AlphaServer SC Resources (see Section 23.8 on page 23–19)
- Managing Network, Tape, and Media Changer Resources (see Section 23.9 on page 23–20)
- Managing CAA with SysMan Menu (see Section 23.10 on page 23–21)
- Understanding CAA Considerations for Startup and Shutdown (see Section 23.11 on page 23–25)
- Managing the CAA Daemon (caad) (see Section 23.12 on page 23–26)
- Using EVM to View CAA Events (see Section 23.13 on page 23–27)
- Troubleshooting with Events (see Section 23.14 on page 23–29)
- Troubleshooting a Command-Line Message (see Section 23.15 on page 23–30)

For detailed information on setting up applications with CAA, see the HP *TruCluster Server Cluster Highly Available Applications* manual. For a general discussion of CAA, see the HP *TruCluster Server Cluster Technical Overview*.

Note:

Most of the CAA commands are located in the `/usr/sbin` directory, except for the `caa_stat` command, which is located in the `/usr/bin` directory.

23.1 Introduction

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 domain member

  You might want to learn which highly available applications are running on the member to be shut down, by using the `caa_stat` command. Optionally, you might want to manually relocate one or more of those applications, by using the `caa_relocate` command.

- Load balancing

  As the loads on various domain members change, you might want to manually relocate applications to members with lighter loads, by using the `caa_stat` and `caa_relocate` commands. You could run the `caa_balance` command to check the placement of an application and relocate only if there is another more highly preferred domain member available.

- A new application resource profile has been created

  If the resource has not already been registered and started, you must do this with the `caa_register` and `caa_start` commands.

- The resource profile for an application has been updated

  For the updates to take effect, you must update the resource using the `caa_register -u` command.

- An existing application resource is being retired

  You will want to stop and unregister the resource by using the `caa_stop` and `caa_unregister` commands.

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 difficult to look for a resource name in a list of processes running on the domain. When managing an application with CAA, you must use its resource name.

23.2 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 to find out 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 23.9 on page 23–20 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 the following tables: Table 23–1 (application), Table

23–2 (network), and Table 23–3 (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 23–1 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 domain member. Application being relocated due to explicit relocation or failure of domain 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. 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 23–2 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 domain 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 23–3 Target and State Combinations for Tape Device and Media Changer Resources

Target	State	Description
ONLINE	ONLINE	Tape device 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 domain member.

23.2.1 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 domain 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 domain member.

For example:

```
# caa_stat clock
NAME=clock
TYPE=application
TARGET=ONLINE
STATE=ONLINE on atlas3
```

To use a script to learn whether a resource is on line, use the caa_stat -r 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 caa_stat -g 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.

23.2.2 Learning Status of All Resources on One Domain Member

The caa_stat -c *cluster_member* command returns the status of all resources on *cluster_member*. For example:

```
# caa_stat -c atlas1
NAME=dhcp
TYPE=application
TARGET=ONLINE
STATE=ONLINE on atlas1

NAME=named
TYPE=application
TARGET=ONLINE
STATE=ONLINE on atlas1

NAME=xclock
TYPE=application
TARGET=ONLINE
```

```
STATE=ONLINE on atlas1
```

This command is useful when you need to shut down a domain member and want to learn which applications are candidates for failover or manual relocation.

23.2.3 Learning Status of All Resources on All Domain Members

The `caa_stat` command returns the status of all resources on all domain members. For example:

```
# caa_stat

NAME=cluster_lockd
TYPE=application
TARGET=ONLINE
STATE=ONLINE on atlas3

NAME=dhcp
TYPE=application
TARGET=ONLINE
STATE=ONLINE on atlas1

NAME=xclock
TYPE=application
TARGET=ONLINE
STATE=ONLINE on atlas3

NAME=named
TYPE=application
TARGET=OFFLINE
STATE=OFFLINE

NAME=ln0
TYPE=network
TARGET=ONLINE on atlas3
TARGET=ONLINE on atlas1
TARGET=ONLINE on atlas2
STATE=OFFLINE on atlas3
STATE=ONLINE on atlas1
STATE=ONLINE on atlas2
```

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      atlas3
dhcp              application      ONLINE      ONLINE      atlas1
xclock            application      ONLINE      ONLINE      atlas3
named             application      OFFLINE     OFFLINE
ln0               network          ONLINE      OFFLINE     atlas3
ln0               network          ONLINE      ONLINE      atlas1
ln0               network          ONLINE      ONLINE      atlas2
```

23.2.4 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 domain members. For example:

```
# caa_stat -v
NAME=cluster_lockd
TYPE=application
RESTART_ATTEMPTS=30
RESTART_COUNT=0
FAILURE_THRESHOLD=0
FAILURE_COUNT=0
TARGET=ONLINE
STATE=ONLINE on atlas3

NAME=dhcp
TYPE=application
RESTART_ATTEMPTS=1
RESTART_COUNT=0
FAILURE_THRESHOLD=3
FAILURE_COUNT=1
TARGET=ONLINE
STATE=OFFLINE

NAME=named
TYPE=application
RESTART_ATTEMPTS=1
RESTART_COUNT=0
FAILURE_THRESHOLD=0
FAILURE_COUNT=0
TARGET=OFFLINE
STATE=OFFLINE

NAME=ln0
TYPE=network
FAILURE_THRESHOLD=5
FAILURE_COUNT=1 on atlas3
FAILURE_COUNT=0 on atlas1
TARGET=ONLINE on atlas3
TARGET=OFFLINE on atlas1
STATE=ONLINE on atlas3
STATE=OFFLINE on atlas1
```

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      atlas3
dhcp             application      0/1     1/3     ONLINE      OFFLINE
named            application      0/1     0/0     OFFLINE     OFFLINE
ln0              network                  1/5     ONLINE      ONLINE      atlas3
ln0              network                  0/5     OFFLINE     OFFLINE     atlas1
```

This information can be useful for finding resources that frequently fail or have been restarted many times.

23.3 Reporting Resource Availability Measurements

CAA maintains a history of each application resource from the time it is first started. The `caa_report` command can give you a report summarizing the percentage of time that all currently registered application resources have been in the `ONLINE` state. The data is obtained by analyzing the Event Management events related to CAA.

The files tracking the application uptime history are updated automatically at 0300 hours each day and each time that the `caa_report` command is executed by root. The time and frequency of the periodic merge is configurable by changing the `crontab` format specified in `/var/cluster/caa/clustercron/caa_report.clustercronData`.

The command shows the application, the starting time analyzed, and a percentage description of the amount of time that the application was in the `ONLINE` state.

The following shows an example output:

```
# /usr/bin/caa_report
        Application Availability Report for atlas
Applications          starting/ending            uptime
-------------------------------------------------------------------------------------------

 autofs               NEVER STARTED              0.00 %

 cluster_lockd        Fri Jul 27 11:00:48 2001   99.80 %

                      Thu Oct 4 12:31:14 2001

 clustercron          Fri Jul 27 11:01:08 2001   100.00 %

                      Thu Oct 4 12:31:14 2001

 dmiller1             Tue Sep 25 13:57:51 2001   12.51 %

                      Thu Oct 4 12:31:14 2001
```

You can filter out all applications that have not been run during the time specified. For example:

```
# /usr/bin/caa_report -o
Application Availability Report for atlas
Applications              starting/ending           uptime
-------------------------------------------------------------------------------

 cluster_lockd       Fri Jul 27 11:00:48 2001   99.80 %

                     Thu Oct 4 12:31:14 2001

 clustercron         Fri Jul 27 11:01:08 2001   100.00 %

                     Thu Oct 4 12:31:14 2001

 dmiller1            Tue Sep 25 13:57:51 2001   12.51 %

                     Thu Oct 4 12:31:14 2001
```

You can specify a starting time and ending time to measure the percentage uptime within that range. The time for either value may be specified in the formats used by the `date(1)` command or any ISO 8061 format. If your chosen time format contains spaces, you need to surround the time with double quotes. The following rules apply to time bounds:

- The end time must be after the begin time.
- If no begin time is specified, the earliest time that the application was started will be used.
- If no end time is specified, the current time will be used.
- If an end time is entered that is after the current time, then the current time is used.
- If a begin time is entered and an application had never been run at or before that time, then the first time that the application was started will be used.
- If a begin time is entered that is during a period that the application was down, but the application had been running at some point before that period, the time entered is used.
- The time displayed with the output is always the actual time used in the analysis.

The following shows an example with a begin time and end time:

```
# /usr/bin/caa_report -b 10/03/01 -e 10/04/01
Application Availability Report for atlas
Applications              starting/ending          uptime
-------------------------------------------------------------------------------------------

 autofs              NEVER STARTED              0.00 %

 cluster_lockd       Wed Oct 3 00:00:00 2001    100.00 %

                     Thu Oct 4 00:00:00 2001

 clustercron         Wed Oct 3 00:00:00 2001    100.00 %

                     Thu Oct 4 00:00:00 2001

 dmiller1            Wed Oct 3 00:00:00 2001    92.54 %

                     Thu Oct 4 00:00:00 2001
```

23.4 Relocating Applications

There are times when you may want to relocate applications from one domain to another. You may want to:

- Relocate all applications on a domain member (see Section 23.4.1 on page 23–11)
- Relocate a single application to another domain member (see Section 23.4.2 on page 23–12)
- Relocate dependent applications to another domain member (see Section 23.4.3 on page 23–12)

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 'atlas3'
Stop of 'cluster_lockd' on member 'atlas3' succeeded.
Attempting to start 'cluster_lockd' on member 'atlas2'
Start of 'cluster_lockd' on member 'atlas2' succeeded.
```

The following sections discuss relocating applications in more detail.

23.4.1 Manual Relocation of All Applications on a Domain Member

When you shut down a domain 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 domain 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 domain 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 domain member and want to minimize disruption to the work environment.

To relocate all applications from `atlas0` to `atlas1`, enter the following command:

```
# caa_relocate -s atlas0 -c atlas1
```

To relocate all applications on `atlas0` according to each application's placement policy, enter the following command:

```
# caa_relocate -s atlas0
```

Use the `caa_stat` command to verify that all application resources were successfully relocated.

23.4.2 Manual Relocation of a Single Application

You may want to relocate a single application to a specific domain member for one of the following reasons:

- The domain member that is currently running the application is overloaded and another member has a low load.
- You are about to shut down the domain 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 `atlas1`, enter the following command:

```
# caa_relocate resource_name -c atlas1
```

Use the `caa_stat` command to verify that the application resource was successfully relocated.

23.4.3 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 `caa_relocate -f` 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 atlas1, enter the following command:

```
# caa_relocate resource_name -f -c atlas1
```

Use the caa_stat command to verify that the application resources were successfully relocated.

23.5 Starting and Stopping Application Resources

This 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.

23.5.1 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 domain member that is determined by the resource's placement policy, enter the following command:

```
# caa_start clock
```

The output of this command is similar to the following:

```
Attempting to start 'clock' on member 'atlas1'
Start of 'clock' on member 'atlas1' 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 domain member, assuming that the placement policy allows it, enter the following command:

```
# 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 domain member, enter the following command:

```
# caa_start -f clock
```

Note:

When attempting to start an application on a domain member that undergoes a system crash, the `caa_start` command can give indeterminate results. In this scenario, the start section of the action script is executed but the domain member crashes before notification of the start is displayed on the command line. The `caa_start` command returns a failure with the error `Remote start for [resource_name] failed on member [member_name]`. The application resource is actually `ONLINE` and fails over to another member, making the application appear as though it was started on the wrong member.

If a domain member fails while you are starting an application resource on that member, you should check the state of the resource on the domain with the `caa_stat` command to determine the state of that resource.

See the `caa_start(8)` reference page for more information.

23.5.2 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:

```
# 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:

```
# caa_stop -f clock
```

See the `caa_stop(8)` reference page for more information.

23.5.3 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.

23.5.4 Using caa_stop to Reset UNKNOWN State

If an application resource state is set to `UNKNOWN`, first try to run the `caa_stop` command. If this does not reset the resource to `OFFLINE`, use the `caa_stop -f` command. The command will ignore any errors that are 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 (zero). Also ensure that it returns 0 if the application is not currently running.

23.6 Balancing Application Resources

Balancing application resources is the re-evaluation of application resource placement based on the current state of the resources on the domain and the rules of placement for the resources. Balancing applications can be done on a domainwide basis, a member-wide basis, or with specified resources. Balancing decisions are made using the standard placement decision mechanism of CAA and are not based on any load considerations.

Balancing an application will restore it to its most favored member (when either the `favored` or the `restricted` placement policy is used) or more evenly distribute the number of CAA application resources per domain member (when the `balanced` placement policy is used).

You may want to relocate applications from one domain to another, for the following reasons:

- To balance all applications on a domain (see Section 23.6.1 on page 23–16)

- To balance all applications on a domain member (see Section 23.6.2 on page 23–16)
- To balance specified applications (see Section 23.6.3 on page 23–17)
- To balance specified applications at a specific time (see Section 23.6.4 on page 23–17)

Use the `caa_balance` command only with application resources. You cannot balance network, tape, or media changer resources. Balancing on a per domain basis re-evaluates all `ONLINE` application resources on a domain and relocates the resource if it is not running on the domain member chosen by the placement decision mechanism.

See the `caa_balance(8)` reference page for more information.

The following sections discuss balancing applications in more detail.

23.6.1 Balancing All Applications on a Domain

To balance all applications on a domain, enter the following command:

```
# /usr/sbin/caa_balance -all
```

Assuming that applications `test` and `test2` are the only two applications that are `ONLINE` and are running on member `atlas1` with `balanced` placement policies, the following text is displayed:

```
Attempting to stop 'test' on member 'atlas1'
Stop of 'test' on member 'atlas1' succeeded.
Attempting to start 'test' on member 'atlas2'
Start of 'test' on member 'atlas2' succeeded.
Resource test2 is already well placed
test2 is placed optimally. No relocation is needed.
```

If more applications are `ONLINE` in the domain, the output will reflect any actions taken for each application resource.

23.6.2 Balancing All Applications on a Member

To reevaluate placement of the applications running on the domain member `atlas1`, enter the following command:

```
# /usr/sbin/caa_balance -s atlas1
```

Assuming that applications `test` and `test2` are the only two applications that are `ONLINE` and are running on member `atlas1` with `balanced` placement policies, the following text is displayed:

```
Attempting to stop 'test' on member 'atlas1'
Stop of 'test' on member 'atlas1' succeeded.
Attempting to start 'test' on member 'atlas2'
Start of 'test' on member 'atlas2' succeeded.
Resource test2 is already well placed
test2 is placed optimally. No relocation is needed.
```

If more applications are ONLINE in the domain member, the output will reflect any actions taken for each application resource.

23.6.3 Balancing Specific Applications

To balance specified applications only, enter the following command:

```
# /usr/sbin/caa_balance test test2
```

Assuming that applications test and test2 are running on member atlas1 with balanced placement policies, the following text is displayed:

```
Attempting to stop 'test' on member 'atlas1'
Stop of 'test' on member 'atlas1' succeeded.
Attempting to start 'test' on member 'atlas2'
Start of 'test' on member 'atlas2' succeeded.
Resource test2 is already well placed
test2 is placed optimally. No relocation is needed.
```

23.6.4 Balancing an Application at Specific Time

You may also balance a single application at a particular time of day by setting the REBALANCE attribute in the profile to a time. This feature is useful for causing application failback based on time of day. If the REBALANCE attribute is set to a specific time of day for an application resource, that application will relocate itself to its most favored member at that time. Using the Active Placement attribute can also create a failback to a most favored member, but the failback will occur only when the favored domain member rejoins the domain. This could cause a relocation to occur during a period of time that you do not wish there to be any interruption in the application availability. Using the REBALANCE attribute instead of Active Placement ensures that the application only fails back to a preferred member at the time that you specify.

The time value in the profile must be specified in the following format:

t:*day:hour:min*

where:

- day is the day of the week (0-6, where Sunday=0),
- hour is the hour of the day (0-23)
- min is the minute of the hour (0–59)

when the re-evaluation occurs. An asterisk (*) may be used as a wildcard to specify every day, every hour, or every minute.

The REBALANCE attribute makes use of the clustercron resource. This resource is a CAA specific implementation of the domainwide cron implementation discussed in the *Best Practice* document *Using cron in a TruCluster Server Cluster* located at

http://h30097.www3.hp.com/docs/best_practices/BP_CRON/TITLE.HTM. This specific implementation is used internally by CAA to schedule rebalance attempts and is not supported for any other use.

For examples on how to set up the REBALANCE attribute, see the HP *TruCluster Server Cluster Highly Available Applications* manual.

23.7 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 HP *TruCluster Server Cluster Highly Available Applications* manual describes the process for creating resource profiles.

To learn which resources are registered on the domain, enter the following caa_stat command:

```
# caa_stat
```

23.7.1 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:

```
# caa_register dtcalc
```

If an application resource has resource dependencies defined in the REQUIRED_RESOURCES attribute of the profile, all resources that are listed for this attribute must be registered first.

For more information, see the caa_register(8) reference page.

23.7.2 Unregistering Resources

You might want to unregister a resource to prevent 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 23.5.2 on page 23–14 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:

```
# caa_unregister dtcalc
```

For more information, see the caa_unregister(8) reference page. For information on registering or unregistering a resource with the SysMan Menu, see the SysMan online help.

23.7.3 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 HP *TruCluster Server 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 no longer synchronized 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 domain members in that list. If the application is not running on one of the allowed members, run the `caa_relocate` command on the application after running the `caa_register -u` command.

23.8 HP AlphaServer SC Resources

Table 23–4 lists the CAA resources that are specific to an HP AlphaServer SC system.

Table 23–4 HP AlphaServer SC Resources

Resource Name	Description	See...
`SC05msql`	This is the resource file for the RMS `msql3d` daemon.	Chapter 5
`SC10cmf`	This is the resource file for the `cmfd` daemon.	Chapter 14
`SC15srad`	This is the resource file for the `srad` daemon.	Chapter 16
`SC20rms`	This is the resource file for the RMS `rms` daemon.	Chapter 5
`SC25scalertd`	This is the resource file for the `scalertd` daemon.	Chapter 9
`SC30scmountd`	This is the resource file for the `scmountd` daemon.	Chapter 7, Chapter 8

Use the caa_stat command to check the status of all CAA resources, as shown in the following example:

```
# caa_stat -t
Name             Type            Target     State       Host
------------------------------------------------------------
SC05msql         application     ONLINE     ONLINE      atlas1
SC10cmf          application     ONLINE     ONLINE      atlas1
SC15srad         application     ONLINE     ONLINE      atlas1
SC20rms          application     ONLINE     ONLINE      atlas1
SC25scalertd     application     ONLINE     ONLINE      atlas0
SC30scmountd     application     ONLINE     ONLINE      atlas0
autofs           application     OFFLINE    OFFLINE
cluster_lockd    application     ONLINE     ONLINE      atlas0
dhcp             application     ONLINE     ONLINE      atlas0
named            application     OFFLINE    OFFLINE
```

23.9 Managing Network, Tape, and Media Changer Resources

Only application resources can be stopped using the caa_stop command. However, nonapplication resources can be restarted using the caa_start command, 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 domain 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 domain member is set to OFFLINE, as shown by the caa_stat *resource_name* command. For example:

```
# caa_stat network1
NAME=network1
TYPE=network
TARGET=OFFLINE on atlas3
TARGET=ONLINE on atlas1
STATE=ONLINE on atlas3
STATE=ONLINE on atlas1
```

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 becomes OFFLINE though their TARGET remains ONLINE. These dependent applications will relocate to another machine where the resource is ONLINE. If no domain 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 domain after removal of the device while the domain is running or a physical failure occurs, the domain does not automatically detect the reconnection of the device. You must run the drdmgr -a DRD_CHECK_PATH *device_name* command.

23.10 Managing CAA with SysMan Menu

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 domain, see Chapter 18.

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 23–1. 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.

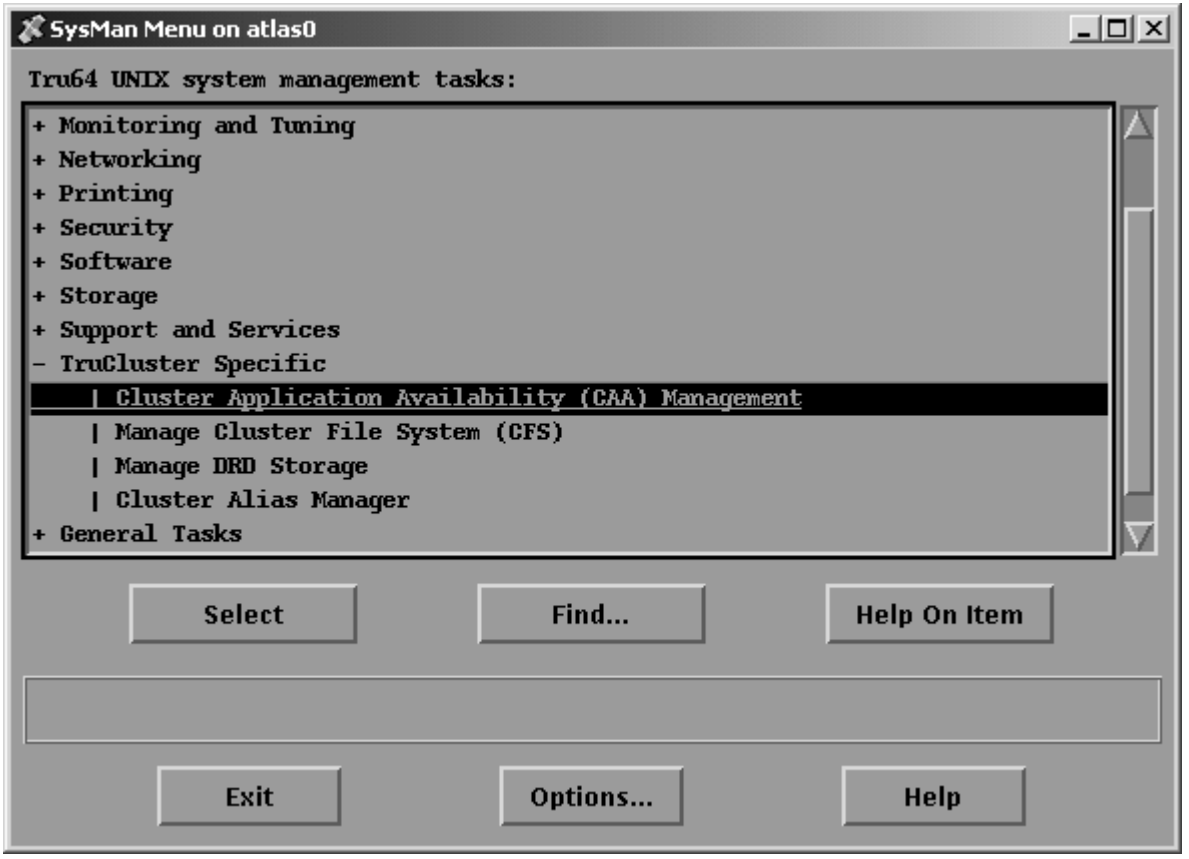

Figure 23–1 CAA Branch of SysMan Menu

23.10.1 CAA Management Dialog Box

The CAA Management dialog box (Figure 23–2) 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.

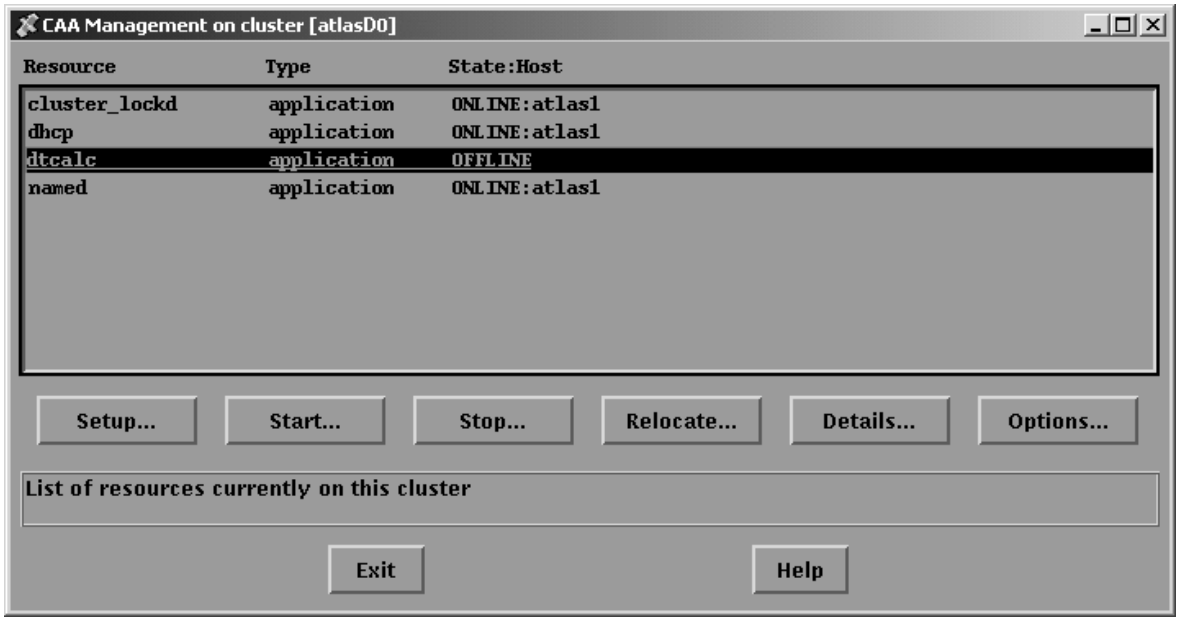

Figure 23–2 CAA Management Dialog Box

23.10.1.1 Start Dialog Box

The Start dialog box (Figure 23–3) 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.

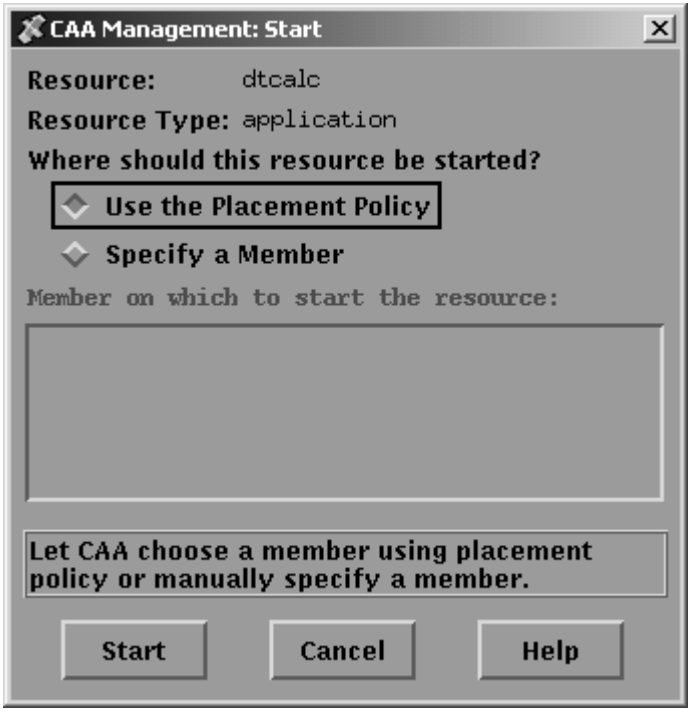

Figure 23–3 Start Dialog Box

23.10.1.2 Setup Dialog Box

To add, modify, register, and unregister profiles of any type, use the Setup dialog box as shown in Figure 23–4. 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.

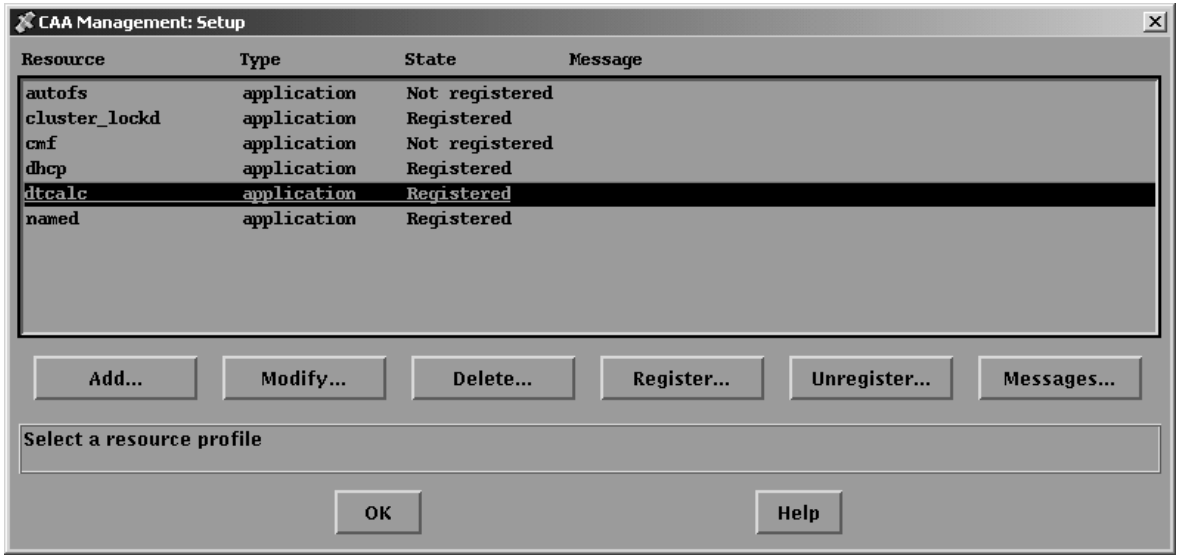

Figure 23–4 Setup Dialog Box

23.11 Understanding 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 domain 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 instructed in Chapters 5 and 6 of the *HP AlphaServer SC Installation Guide*, and start the CAA daemon (`caad`) as follows:

```
# caad
```

When you shut down a domain, CAA notes for each application resource whether it is `ONLINE` or `OFFLINE`. On restart of the domain, 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 domain reboot resolves, use the `caa_start` command to start the application. Any applications that have `AUTO_START` set to 1 will also start when the domain is reformed.

If you want to choose placement of applications before shutting down a domain 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 23.4 on page 23–11.

23.12 Managing the CAA Daemon (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 domain 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. If there was an error with the CAA daemon, the Essential Services Monitor daemon will attempt to restart the CAA daemon. After waiting a few seconds, you can try one of the CAA commands again. If it succeeds, the daemon has been restarted and all features of CAA should function correctly again. To determine whether the daemon is running, see Section 23.12.1.

23.12.1 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   /usr/sbin/caad -0
```

If nothing is displayed, `caad` is not running.

You can determine the status of other `caad` daemons by logging in to the other domain 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 23.12.2, the resources on that machine should be fully manageable by CAA.

23.12.2 Restarting the CAA Daemon

If the caad daemon dies on one domain 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 caad command. The Essential Services Monitor daemon, esmd, will restart the caad daemon automatically and management capabilities will be returned. If the Essential Services Monitor daemon is unable to restart the caad daemon, it will post high priority events to syslog. For more information see the esmd(8) reference page.

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 domain member is booting up.

23.12.3 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 23.13.

23.13 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 file and EVM for information. EVM has the advantage of being a single source of information for the whole domain, while daemon.log information is specific to each member. Some information is available only in the daemon.log files.

You can access EVM events by using the EVM commands at the command line.

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 that are 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 the `EVM(5)`, `evmget(5)`, or `evmshow(5)` reference pages.

23.13.1 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 srad action script /var/cluster/caa/script/srad.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               : 05-Mar-2001 13:23:50
   Cluster IP address      : <site_specific>
   Host Name               : atlas1.xxx.yyy.zzz
   Cluster Name            : atlasD0
   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 @@'
```

To view other events that are logged by the caad daemon, as well as other daemons, enter the following command:

```
# evmget '[name sys.unix.syslog.daemon]' | evmshow -t '@timestamp @@'
```

23.13.2 Monitoring CAA Events

To monitor CAA events with time stamps on the console, enter the following command:

```
# evmwatch -f '[name *.caa.*]' | evmshow -t '@timestamp @@'
```

As events that are related to CAA are posted to EVM, they are displayed on the terminal where this command is executed, as shown in the following example:

```
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 atlas1
```

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
```

23.14 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
```

23.14.1 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. However, this will 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.

23.14.2 Action Script Stop Entry Point Not Returning 0

```
CAAD[524894]: 'foo' on member 'atlas3' 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 (zero) and running `caa_stop` or `caa_stop -f`. In either case, fix the stop action script to return 0 (zero) before you attempt to restart the application resource.

23.14.3 Network Failure

```
CAAD[524764]: 'ee0' has gone offline on member 'atlas9'
```

A message like this for network resource `ee0` indicates that the network has gone down. Make sure that the network card is connected correctly. Replace the card, if necessary.

23.14.4 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-atlas3.yoursite.com
```

A message similar to this is displayed when attempting to start a second `caad`. Determine whether `caad` is running, as described in Section 23.12.1 on page 23–26. If there is no daemon running, remove the lock file that is listed in the message, and restart `caad` as described in Section 23.12.2 on page 23–27.

23.15 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 ensure that the profile exists in the location that is cited in the message.

24

Managing the Cluster File System (CFS), the Advanced File System (AdvFS), and Devices

This chapter describes the Cluster File System (CFS) and the Advanced File System (AdvFS) in an HP AlphaServer SC system, and how to manage devices. The chapter discusses the following subjects:

- CFS Overview (see Section 24.1 on page 24–2)
- Working with CDSLs (see Section 24.2 on page 24–4)
- Managing Devices (see Section 24.3 on page 24–7)
- Managing the Cluster File System (see Section 24.4 on page 24–15)
- Managing AdvFS in a Domain (see Section 24.5 on page 24–37)
- Considerations When Creating New File Systems (see Section 24.6 on page 24–42)
- Backing Up and Restoring Files (see Section 24.7 on page 24–45)
- Managing CDFS File Systems (see Section 24.8 on page 24–47)
- Using the verify Command in a Domain (see Section 24.9 on page 24–48)

For more information on administering devices, see the HP *Tru64 UNIX Hardware Management* manual. For more information on administering file systems and the archiving services, see the HP *Tru64 UNIX System Administration* manual. For more information about managing AdvFS, see the HP *Tru64 UNIX AdvFS Administration* manual.

For information about Logical Storage Manager (LSM) and domains, see Chapter 25 of this guide.

24.1 CFS Overview

CFS is a file system service that integrates all of the underlying file systems within a domain. CFS does not provide disk-structure management; it uses the capabilities of the serving file system for this, without requiring changes to the on-disk structures. The underlying serving file systems used are as follows:

- Advanced File System (AdvFS)
- Network File System (NFS)
- Memory File System (MFS)
- UNIX File System (UFS)
- Named Pipes (FIFO)
- `/proc` File System
- File-on-file Mounting File System (FFM)
- CD-ROM File System (CDFS)
- DVD-ROM File System (DVDFS)

CFS is a POSIX and X/Open compliant file system. CFS provides the following capabilities:

- A single coherent name space

  The same pathname refers to the same file on all nodes. A file system mount on any node is a global operation and results in the file system being mounted at the same point on all nodes.

- Global root

  The point of name space coherency is at the root of the file system and not at a subordinate point; therefore, all files are global and common. This enables all nodes to share the same files; for example, system binaries, and global configuration and administration files.

- Failover

  Because the file system capability is global, CFS will detect the loss of a service node. CFS will automatically move a file service from a failed node to another node that has a path to the same storage. In-flight file system operations are maintained.

- Coherent access

  Multiple accesses of the same file will give coherent results. (Though this mode of access is less common with high-performance applications, and incurs a performance penalty, it is essential for enterprise applications.)

- Client/Server file system architecture

  Each node potentially serves its local file systems to other nodes. Each node is also a client of other nodes. In practice, only a small number of nodes act as file servers of global file systems to the other (client) nodes.

- Support for node-specific files with the same pathname on each node

  This is implemented through a Context Dependent Symbolic Link (CDSL) — a symbolic link with a node identifier in the link name. CDSL is a feature of CFS. The node identifier is evaluated at run time and can be resolved to a node-specific file. This can be used to provide, for example, a node-specific `/tmp` directory. This feature is used to provide node-unique files and to optimize for local performance.

The cluster file system (CFS) provides transparent access to files located anywhere on the domain. Users and applications enjoy a single-system image for file access. Access is the same regardless of the domain member where the access request originates and where in the domain the disk containing the file is connected. CFS follows a server/client model, with each file system served by a domain member. Any domain member can serve file systems on devices anywhere in the domain. If the member serving a file system becomes unavailable, the CFS server automatically fails over to an available domain member.

The primary tool for managing the CFS 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 the `cfsmgr(8)` reference page.

TruCluster Server Version 5.1B includes a load monitoring daemon, `/usr/sbin/cfsd`, that can monitor, report on, and respond to file system-related member and domain activity. The `cfsd` daemon is described in Section 24.4.1.3 on page 24–18.

The `-o` option with the `mount` command can be used to allow file systems to be served by a specific domain member on startup. This option is described in Section 24.4.1.4 on page 24–19.

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 24.4.4.4 on page 24–27. For more information on the command, see the `cfsstat(8)` reference page.

For file systems on devices on a 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 local to a member is faster when the file systems are served by that member.

24.1.1 File System Topology

The HP AlphaServer SC installation utility creates a default CFS file system layout. This consists of file systems resident on storage devices such as the Fibre Channel RAID array and local storage devices on a local system bus. File systems that reside on the RAID array, and are directly accessible by more than one node, are referred to as global storage; file systems that reside on local devices, accessible directly only by the local node, are referred to as local storage. By default, `cluster_root` (`/`), `cluster_usr` (`/usr`) and `cluster_var` (`/var`) are set up on the RAID array and are global storage. Other candidates for global storage are applications and data that will be commonly accessed by (all) other nodes in the domain.

Each node also has some intrinsically local file systems, such as its boot partition and also a number of `/local` and `/tmp` file systems, which are mounted as server-only. For more information about the `server_only` option, see Section 24.4.6 on page 24–34. The information on these file systems is generally of little interest to other nodes in the domain and is only accessed by the host node. The following sections show how the location of a file system affects how it is mounted, its availability, and impact on efficiency of access.

24.2 Working with CDSLs

A context-dependent symbolic link (CDSL) is a link that contains a variable that identifies a domain 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`*`M`*, where *`M`* is the member ID of the current member.

For example, on a domain 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. Domains use this to allow the creation of member-specific files that can be addressed throughout the domain by a single file name. System data and configuration files tend to be CDSLs. They are found in the root (`/`), `/usr`, and `/var` directories.

The information in this section is organized as follows:

- Making CDSLs (see Section 24.2.1 on page 24–5)
- Maintaining CDSLs (see Section 24.2.2 on page 24–6)
- Kernel Builds and CDSLs (see Section 24.2.3 on page 24–6)

- Exporting and Mounting CDSLs (see Section 24.2.4 on page 24–6)

24.2.1 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`, use the following command:

```
# mkcdsl /usr/accounts/usage-history
```

When you check 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.

Note:

The `mkcdsl` command will fail if the parent directory does not exist.

To move a file into a CDSL, use 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 domain 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 domain members and then replace the source file with a CDSL, use the `mkcdsl -a` command as follows:

```
# mkcdsl -a filename
```

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 domain member, use the `mkcdsl -c` *`filename`* command to copy the file and re-create the CDSL.

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 domain'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 the `mkcdsl(8)` reference page.

24.2.2 Maintaining CDSLs

The following tools can help you to 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 domain member, use the `mkcdsl -c` *`filename`* command to copy the file and re-create the CDSL.

24.2.3 Kernel Builds and CDSLs

When you build a kernel in a domain, use the `cp` command to copy the new kernel from `/sys/`*`HOSTNAME`*`/vmunix` to `/vmunix` (which is a CDSL to `/cluster/members/member`*`M`*`/boot_partition/vmunix`), as shown in the following example:

```
# cp /sys/atlas0/vmunix /vmunix
```

Note:

In a domain, you must always copy the new `vmunix` file to `/vmunix`. This is because, in an HP AlphaServer SC system, `/vmunix` is a CDSL:

```
/vmunix -> cluster/members/{memb}/boot_partition/vmunix
```

You should treat a CDSL as you would any other symbolic link: remember that copying a file follows the link, but moving a file replaces the link. If you were to move (instead of copy) a kernel to `/vmunix`, you would replace the symbolic link with the actual file — with the result that the next time that domain member boots, it will use the old `vmunix` in its boot partition, and there will be errors when it or any other domain member next boots.

24.2.4 Exporting and Mounting CDSLs

CDSLs are intended for use when files of the same name must necessarily have different contents on different domain 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 domain 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 domainwide directory, then the 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.

24.3 Managing Devices

Note:

This section generically discusses storage device management. Please see the documentation that is specific to any particular storage device installed on your system.

Storage device management in an HP AlphaServer SC system is a combination of core Tru64 UNIX hardware management (see the HP *Tru64 UNIX Hardware Management* manual), and TruCluster Server management (see the HP *TruCluster Server Cluster Administration* manual).

Because of the typical large size of an HP AlphaServer SC system and the mix of shared and local buses, device naming (device special file name `/dev/disk/dsk3`) and location (logical: bus, target, LUN; physical: host, storage array, and so on) can be complex.

The powerful and flexible ability of the device request dispatcher to allow access to devices from any node in the system further complicates management.

The three main tools used to manage devices are as follows:

- The Hardware Management Utility (hwmgr) (see Section 24.3.1 on page 24–8)
- The Device Special File Management Utility (dsfmgr) (see Section 24.3.2 on page 24–8)
- The Device Request Dispatcher Utility (drdmgr) (see Section 24.3.3 on page 24–8)

This section describes how to use these tools to perform the following tasks:

- Determining Device Locations (see Section 24.3.5 on page 24–11)
- Adding a Disk to the Domain (see Section 24.3.6 on page 24–12)
- Managing Third-Party Storage (see Section 24.3.7 on page 24–13)
- Replacing a Failed Disk (see Section 24.3.8 on page 24–14)

This section also describes how to detect storage devices using SRM commands (see Section 24.3.4 on page 24–11.

Finally, this section describes the following devices:

- Diskettes (see Section 24.3.9 on page 24–15)
- CD-ROM and DVD-ROM (see Section 24.3.10 on page 24–15)

24.3.1 The Hardware Management Utility (hwmgr)

The Tru64 UNIX `hwmgr` utility allows you to view, add, modify, and delete hardware component information. The `hwmgr` command can list all hardware devices in the domain, including those on local buses, and correlate bus-target-LUN names with `/dev/disks/dsk*` names.

For more information about hardware management, see the `hwmgr(8)` reference page and the HP *Tru64 UNIX Hardware Management* manual.

24.3.2 The Device Special File Management Utility (dsfmgr)

The `dsfmgr` utility is used to manage device special files. On an HP AlphaServer SC machine, the device `/dev/disk/dsk10c` is associated with a specific disk (this may be a virtual disk; for example, a unit within a RAID set), no matter where the disk is installed physically. This feature provides the system administrator with great flexibility to move a disk around within a system and not have to worry about name conflicts or improperly mounting a disk.

When using `dsfmgr`, the device special file management utility, in a domain, 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 domain.
- 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 on devices, device naming, and device management, see the `dsfmgr(8)` reference page and the HP *Tru64 UNIX Hardware Management* manual.

24.3.3 The Device Request Dispatcher Utility (drdmgr)

DRD is an operating system software component that provides transparent, highly available access to all devices in the domain. The DRD subsystem makes physical disk and tape storage available to all domain members, regardless of where the storage is physically

located in the domain. It uses a device-naming model to make device names consistent throughout the domain. This provides great flexibility when configuring hardware. A member does not need to be directly attached to the bus on which a disk resides to access storage on that disk.

The device request dispatcher supports domainwide access to both character and block disk devices. You access a raw disk device partition in an HP AlphaServer SC configuration in the same way you do on a Tru64 UNIX standalone system; that is, by using the device's special file name in the `/dev/rdisk` directory.

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 directly above the device drivers.

The primary tool for managing the device request dispatcher is the `drdmgr` command. For more information, see the `drdmgr(8)` reference page.

24.3.3.1 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 domain are either **direct-access I/O devices** or **single-server devices**. A direct-access I/O device supports simultaneous access from multiple domain 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 domain members on that bus. A single-server device, whether on a shared bus or directly connected to a domain 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 24.4.4.4 on page 24–27 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 domain. Initially, such devices are single-server devices. See Section 24.3.3.2 on page 24–10 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. HP recommends that you use direct-access I/O disks in these situations.

24.3.3.2 Devices Supporting Direct-Access I/O

RAID-fronted disks are direct-access I/O capable. The following are RAID-fronted disks:

- HSZ80
- HSG60
- HSG80
- RA3000 (HSZ22)
- HSV110

Any RZ26, RZ28, RZ29, and RZ1CB-CA disks already installed in a system at the time the system becomes a domain member, by using the `sra install` 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 24.3.6 on page 24–12.

24.3.3.3 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 domain 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 the `hwmgr(8)` reference page and the section on replacing a failed SCSI device in the HP *Tru64 UNIX Hardware Management* manual.
4. On each domain member, enter the `clu_disk_install` command:
   ```
   # clu_disk_install
   ```

Note:

If the domain has a large number of storage devices, the `clu_disk_install` command can take several minutes to complete.

24.3.3.4 HSZ Hardware Supported on Shared Buses

For a list of hardware supported on shared buses, see the *HP AlphaServer SC Software Product Description* for Version 2.6.

If you try to use an HSZ that does not have the proper firmware revision on a shared bus, the domain might hang when there are multiple simultaneous attempts to access the HSZ.

24.3.4 Detecting Storage Devices Using SRM Commands

To detect storage devices, enter the following command at the SRM prompt:

```
P00>>>show dev
```

The output from this command shows the devices on the system; for example, the disks. The information about these disks is in the following format:

```
dkxxxx
```

where *xxxx* defines the type of disk.

Note:

SRM cannot show external disks — use the following Tru64 UNIX command to show external disks:

```
# hwmgr -view devices
```

24.3.5 Determining Device Locations

For example, to list all hardware devices in a domain, run the following command:

```
# hwmgr -view devices -cluster
```

```
HWID: Device Name           Mfg   Model               Hostname   Location
------------------------------------------------------------------------------
3:    /dev/kevm                                       atlas0
771:  /dev/disk/floppy13c         3.5in floppy        atlas2     fdi0-unit-0
780:  /dev/disk/cdrom13c    DEC   RRD47 (C) DEC       atlas2     bus-0-targ-5-lun-0
784:  kevm                                            atlas1
40:   /dev/disk/floppy0c          3.5in floppy        atlas0     fdi0-unit-0
49:   /dev/disk/cdrom0c     DEC   RRD47 (C) DEC       atlas0     bus-0-targ-5-lun-0
50:   /dev/disk/dsk0c       DEC   RZ2EA-LA (C) DEC    atlas0     bus-1-targ-0-lun-0
51:   /dev/disk/dsk1c       DEC   RZ2EA-LA (C) DEC    atlas0     bus-1-targ-1-lun-0
52:   /dev/disk/dsk2c       DEC   RZ1EF-CB (C) DEC    atlas0     bus-1-targ-2-lun-0
53:   /dev/disk/dsk3c       DEC   RZ1EF-CB (C) DEC    atlas0     bus-1-targ-3-lun-0
821:  /dev/disk/floppy14c         3.5in floppy        atlas1     fdi0-unit-0
54:   /dev/disk/dsk4c       DEC   RZ1EF-CB (C) DEC    atlas0     bus-1-targ-4-lun-0
55:   /dev/cport/scp0             HSZ70CCL            atlas0     bus-4-targ-0-lun-0
55:   /dev/cport/scp0             HSZ70CCL            atlas1     bus-4-targ-0-lun-0
56:   /dev/disk/dsk5c       DEC   HSZ70               atlas0     bus-4-targ-1-lun-1
56:   /dev/disk/dsk5c       DEC   HSZ70               atlas1     bus-4-targ-1-lun-1
57:   /dev/disk/dsk6c       DEC   HSZ70               atlas0     bus-4-targ-1-lun-2
57:   /dev/disk/dsk6c       DEC   HSZ70               atlas1     bus-4-targ-1-lun-2
58:   /dev/disk/dsk7c       DEC   HSZ70               atlas0     bus-4-targ-1-lun-3
58:   /dev/disk/dsk7c       DEC   HSZ70               atlas1     bus-4-targ-1-lun-3
59:   /dev/disk/dsk8c       DEC   HSZ70               atlas0     bus-4-targ-1-lun-4
59:   /dev/disk/dsk8c       DEC   HSZ70               atlas1     bus-4-targ-1-lun-4
```

```
832:  /dev/disk/cdrom14c    DEC   RRD47 (C) DEC      atlas1    bus-0-targ-5-lun-0
111:  /dev/disk/dsk9c       DEC   RZ2EA-LA (C) DEC   atlas2    bus-1-targ-0-lun-0
164:  /dev/disk/dsk11c      DEC   RZ2EA-LA (C) DEC   atlas1    bus-1-targ-0-lun-0
165:  /dev/disk/dsk12c      DEC   RZ2EA-LA (C) DEC   atlas1    bus-1-targ-1-lun-0
166:  /dev/disk/dsk13c      DEC   RZ2EA-LA (C) DEC   atlas1    bus-1-targ-2-lun-0
736:  kevm                                           atlas2
```

The `drdmgr` *`devicename`* command identifies the members that serve the device. Disks with multiple servers are on a shared bus. With very few exceptions, disks that have only one server are local to that server. For details on the exceptions, see Section 24.3.3.1 on page 24–9.

To locate a physical device such as the RZ2CA known as `/dev/disk/dsk1c`, flash its activity light as follows:

```
# hwmgr -locate component -id 51
```

where 51 is the hardware component ID (HWID) of the device.

To identify a newly installed device, run the following command:

```
# hwmgr -scan scsi
```

To learn the hardware configuration of a domain member, use the following command:

```
# hwmgr -view hierarchy -m member_name
```

If the member is on a shared bus, the command reports devices on the shared bus. The command does not report on devices local to other members.

24.3.6 Adding a Disk to the Domain

For information on physically installing hardware devices, see the HP *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 domain member that needs access to the disk.

   Wait a minute for all members to register the presence of the new disk.

2. To learn the name of the new disk, enter the following command:

   ```
   # hwmgr -view devices -cluster
   ```

For information about creating file systems on the disk, see Section 24.6 on page 24–42.

24.3.7 Managing Third-Party Storage

When a domain 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 domain. This I/O barrier operation inhibits non-domain 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 domain 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 device does not support the SCSI Persistent Reserve command set and uses the Fibre Channel interconnect.
- The shared 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 domain 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 domain, 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 HP *TruCluster Server Cluster Technical Overview* manual.

24.3.8 Replacing a Failed Disk

When a disk fails and is replaced, the new disk is assigned to a new device special file name.

To replace a failed disk and associate its device special file with a new disk, you must remove the previous disk from the system's database and reassign the device special file name. In the following example, the RZ2CA known as `/dev/disk/dsk1c`, with hardware component ID (HWID) `51`, has failed.

To replace the failed disk, perform the following steps:

1. Physically remove the device.
2. Delete it from the hardware component database, as follows:
   ```
   # hwmgr -delete component -id 51
   ```
3. Physically install a new disk.
4. Scan the system for new SCSI devices, as follows:
   ```
   # hwmgr -scan scsi
   ```
5. View all devices again, as follows:
   ```
   # hwmgr -view devices -cluster
   ```

```
HWID: Device Name          Mfg  Model              Hostname  Location
------------------------------------------------------------------------------
3:    /dev/kevm                                   atlas0
771:  /dev/disk/floppy13c       3.5in floppy      atlas2    fdi0-unit-0
780:  /dev/disk/cdrom13c  DEC RRD47 (C) DEC       atlas2    bus-0-targ-5-lun-0
784:  kevm                                        atlas1
40:   /dev/disk/floppy0c       3.5in floppy      atlas0    fdi0-unit-0
49:   /dev/disk/cdrom0c   DEC RRD47 (C) DEC       atlas0    bus-0-targ-5-lun-0
50:   /dev/disk/dsk0c     DEC RZ2EA-LA (C) DEC    atlas0    bus-1-targ-0-lun-0
52:   /dev/disk/dsk2c     DEC RZ1EF-CB (C) DEC    atlas0    bus-1-targ-2-lun-0
53:   /dev/disk/dsk3c     DEC RZ1EF-CB (C) DEC    atlas0    bus-1-targ-3-lun-0
821:  /dev/disk/floppy14c       3.5in floppy      atlas1    fdi0-unit-0
54:   /dev/disk/dsk4c     DEC RZ1EF-CB (C) DEC    atlas0    bus-1-targ-4-lun-0
55:   /dev/cport/scp0           HSZ70CCL          atlas0    bus-4-targ-0-lun-0
55:   /dev/cport/scp0           HSZ70CCL          atlas1    bus-4-targ-0-lun-0
56:   /dev/disk/dsk5c     DEC HSZ70               atlas0    bus-4-targ-1-lun-1
56:   /dev/disk/dsk5c     DEC HSZ70               atlas1    bus-4-targ-1-lun-1
57:   /dev/disk/dsk6c     DEC HSZ70               atlas0    bus-4-targ-1-lun-2
57:   /dev/disk/dsk6c     DEC HSZ70               atlas1    bus-4-targ-1-lun-2
58:   /dev/disk/dsk7c     DEC HSZ70               atlas0    bus-4-targ-1-lun-3
58:   /dev/disk/dsk7c     DEC HSZ70               atlas1    bus-4-targ-1-lun-3
59:   /dev/disk/dsk8c     DEC HSZ70               atlas0    bus-4-targ-1-lun-4
59:   /dev/disk/dsk8c     DEC HSZ70               atlas1    bus-4-targ-1-lun-4
832:  /dev/disk/cdrom14c  DEC RRD47 (C) DEC       atlas1    bus-0-targ-5-lun-0
111:  /dev/disk/dsk9c     DEC RZ2EA-LA (C) DEC    atlas2    bus-1-targ-0-lun-0
164:  /dev/disk/dsk11c    DEC RZ2EA-LA (C) DEC    atlas1    bus-1-targ-0-lun-0
165:  /dev/disk/dsk12c    DEC RZ2EA-LA (C) DEC    atlas1    bus-1-targ-1-lun-0
166:  /dev/disk/dsk13c    DEC RZ2EA-LA (C) DEC    atlas1    bus-1-targ-2-lun-0
167:  /dev/disk/dsk14c    DEC RZ2EA-LA (C) DEC    atlas0    bus-1-targ-1-lun-0
```

```
736:  kevm                                          atlas2
```

6. Locate the new device entry in the listing (HWID 167) by comparing with the previous output.
7. Move the new device special file name to match the old name, as follows:
   `# dsfmgr -m dsk14 dsk1`

24.3.9 Diskettes

HP AlphaServer SC Version 2.6 includes support for read/write UNIX File System (UFS) file systems, as described in Section 24.4.5 on page 24–33, and you can use HP AlphaServer SC Version 2.6 to format a diskette.

24.3.10 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 domain, see Section 24.8 on page 24–47.

24.4 Managing the Cluster File System

This section describes the following topics:

- Mounting CFS File Systems (see Section 24.4.1 on page 24–15)
- Freezing an AdvFS Domain Before Cloning (see Section 24.4.2 on page 24–21)
- File System Availability (see Section 24.4.3 on page 24–22)
- Optimizing CFS — Locating and Migrating File Servers (see Section 24.4.4 on page 24–24)
- MFS and UFS File Systems Supported (see Section 24.4.5 on page 24–33)
- Partitioning File Systems (see Section 24.4.6 on page 24–34)
- Block Devices and Cache Coherency (see Section 24.4.7 on page 24–36)

24.4.1 Mounting CFS File Systems

For a file system on a device to be served, there must be at least one node that can act as a DRD server for the device. Such a node has a physical path to the storage and is booted.

Once a file system has a DRD server, it can be mounted into the cluster file system, and any node in the domain can be its CFS server. Ensure that the same node is both the CFS server and the DRD server, for optimal performance. Mounting a file system is a global operation, and the file system is mounted on each node in the domain synchronously. When a node boots, it attempts to mount each file system referenced in the `/etc/fstab` file.

This is desirable for global storage file systems such as `cluster_usr` (`/usr`); however, it makes little sense for the `/local` file system of, for example, node 5 to be mounted by some other node. Therefore, it is useful to control which nodes may mount which file systems at boot time.

There are four methods of controlling the mounting behavior of a booting domain:

- fstab and member_fstab Files (see Section 24.4.1.1 on page 24–16)
- Start Up Scripts (see Section 24.4.1.2 on page 24–16)
- CFS Load Balancing (see Section 24.4.1.3 on page 24–18)
- Distributing File Systems *via* the mount -o Command (see Section 24.4.1.4 on page 24–19)

24.4.1.1 fstab and member_fstab Files

The `/etc/fstab` file is a global file; each node shares the contents of this file. File systems that reside on global storage have entries in this file, so that the first node in the domain to boot that has access to the global storage will mount the file systems. The `member_fstab` file (`/etc/member_fstab`) is a Context-Dependent Symbolic Link (CDSL, see Section 24.2 on page 24–4) — the contents of this member-specific file differ for each member of the domain. Each member-specific `member_fstab` file describes file systems, residing on local devices, that should only be mounted by the local node. Note, however, that a member-specific `member_fstab` file can be used to mount any file system (global or local), and can be used at the discretion of the system administrator (for example, to distribute fileserving load among a number of file servers). The syntax of the `member_fstab` file is the same as that for the `/etc/fstab` file.

The following example shows the contents of a member-specific `member_fstab` file showing file systems that will be mounted by the selected system:

```
# ls -l /etc/member_fstab
lrwxrwxrwx   1 root     system        42 Jun  6 19:56 /etc/member_fstab ->
../cluster/members/{memb}/etc/member_fstab
# cat /etc/member_fstab
atlasms-ext1:/usr/users         /usr/users            nfs     rw,hard,bg,intr 0 0
atlasms:/usr/kits               /usr/kits             nfs     rw,hard,bg,intr 0 0
```

24.4.1.2 Start Up Scripts

Scripts in the `/sbin/rc3.d` directory are invoked as the node boots. You can install a site-specific script in this directory to mount a specified file system. One such script — `sra_clu_min` — is copied to the `/sbin/rc3.d` directory during the installation process.

The sra_clu_min script is run at boot time on every node. It can be adapted to perform any desired actions. Mounting file systems *via* the sra_clu_min script results in the file systems being mounted earlier in the boot sequence; this is the method used for the default /local and /tmp file systems. Using a startup script has the advantage that it will relocate file systems to the local node if they are currently being served by a different node.

The following example shows the syntax of an sra_clu_min script entry:

```
# Serve the /var file system on the second node for load balance
if [ "$MEMBER_ID" = 2 ] ; then
  echo "Migrate serving of /var to `hostname`"
  cfsmgr -a server=`hostname` /var
fi
```

The script should check 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, the script should check 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, the script should retry the relocation. On any other error, the script should print an appropriate message and exit.

A file system mounted and served by a particular node can be relocated at any stage. Use the drdmgr and cfsmgr commands to relocate file systems (see Section 24.4.4 on page 24–24).

The /etc/member_fstab file is the recommended method to mount member-specific file systems.

After system installation, a typical CFS setup is as follows:

```
atlas0> cfsmgr

 Domain or filesystem name = cluster_root#root
 Mounted On = /
 Server Name = atlas0
 Server Status : OK

Domain or filesystem name = cluster_usr#usr
 Mounted On = /usr
 Server Name = atlas0
 Server Status : OK

Domain or filesystem name = cluster_var#var
 Mounted On = /var
 Server Name = atlas0
 Server Status : OK

Domain or filesystem name = root1_domain#root
 Mounted On = /cluster/members/member1/boot_partition
 Server Name = atlas0
```

```
 Server Status : OK

Domain or filesystem name = root1_local#local
 Mounted On = /cluster/members/member1/local
 Server Name = atlas0
 Server Status : OK

Domain or filesystem name = root1_local1#local1
 Mounted On = /cluster/members/member1/local1
 Server Name = atlas0
 Server Status : OK

Domain or filesystem name = root1_tmp#tmp
 Mounted On = /cluster/members/member1/tmp
 Server Name = atlas0
 Server Status : OK

Domain or filesystem name = root1_tmp1#tmp
 Mounted On = /cluster/members/member1/tmp1
 Server Name = atlas0
 Server Status : OK

Domain or filesystem name = root2_domain#root
 Mounted On = /cluster/members/member2/boot_partition
 Server Name = atlas1
 Server Status : OK

Domain or filesystem name = root2_local#local
 Mounted On = /cluster/members/member2/local
 Server Name = atlas1
 Server Status : OK

Domain or filesystem name = root2_local1#local
 Mounted On = /cluster/members/member2/local1
 Server Name = atlas1
 Server Status : OK

Domain or filesystem name = root2_tmp#tmp
 Mounted On = /cluster/members/member2/tmp
 Server Name = atlas1
 Server Status : OK

Domain or filesystem name = root2_tmp1#tmp
 Mounted On = /cluster/members/member2/tmp1
 Server Name = atlas1
 Server Status : OK
```

24.4.1.3 CFS Load Balancing

When a domain boots, the TruCluster Server software ensures that each file system is directly connected to the member that serves it. 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 bus is served by one of the members that is directly connected to that shared bus.

In the case of AdvFS, the first fileset that is assigned to a CFS server determines that all other filesets in that AdvFS domain will have that same domain member as their CFS server.

When a domain boots, typically the first member up that is connected to a shared 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 and this member can become more heavily loaded than other members, thereby using a larger percentage of its resources (CPU, memory, I/O, and so forth). In this case, CFS can recommend that you relocate file systems to other domain members to balance the load and improve performance.

TruCluster Server Version 5.1B includes a load monitoring daemon, `/usr/sbin/cfsd`, that can monitor, report on, and respond to file-system-related member and domain activity. The `cfsd` daemon is disabled by default and you must explicitly enable it. After being enabled, the `cfsd` daemon can perform the following functions:

- Assist in managing file systems by locating file systems based on your preferences and storage connectivity. You can configure the `cfsd` daemon to automatically relocate file systems when members join or leave the domain, when storage connectivity changes, or as a result of CFS memory usage.
- Collect a variety of statistics on file system usage and system load. You can use this data to understand how the domain's file systems are being used.
- Analyze the statistics that it collects and recommend file system relocations that may improve system performance or balance the file system load across the domain.
- Monitor CFS memory usage on domain members and generate an alert when a member is approaching the CFS memory usage limit.

Note:

The `cfsd` daemon must not be used with SCFS file systems. Instead, SCFS file systems exported from an FS domain can be mounted from preferred server cluster aliases (for more information, see Section 6.2 on page 6–2).

For more information on CFS Load Balancing, see Section 9.3.3 of the HP *TruCluster Server Cluster Administration* manual.

24.4.1.4 Distributing File Systems *via* the mount -o Command

A file system on a device on a shared bus is served by one of the members that is directly connected to that shared bus. When a domain boots, typically the first active member that is connected to a shared 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.

CFS allows you to then relocate file systems to better balance the file system load, as described in Section 24.4.1.3 on page 24–18 of this guide, and Section 9.3.3 of the HP *TruCluster Server Cluster Administration* manual.

As an alternate approach, the `mount -o server=`*`name`* command allows you to specify which domain member serves a given file system at startup. The `-o server=`*`name`* option is particularly useful for those file systems that cannot be relocated, such as NFS, MFS, and read/write UFS file systems:

```
# mount -t nfs -o server=atlas1 smooch:/usr /tmp/mytmp
# cfsmgr -e
.
.
.
Domain or filesystem name = smooch:/usr
Mounted On = /tmp/mytmp
Server Name = atlas1
Server Status : OK
```

If the mount specified by the `mount -o server=`*`name`* command is successful, the specified domain member is the CFS server for the file system. However, if the specified member is not a member of the domain or is unable to serve the file system, the mount attempt fails.

The `mount -o server=`*`name`* command determines where the file system is first mounted; it does not limit or determine the domain members to which the file system might later be relocated or fail over.

If you combine the `-o server=`*`name`* option with the `-o server_only` option, the file system can be mounted only by the specified domain member and the file system is then treated as a partitioned file system. That is, the file system is accessible for both read-only and read/write access only by the member that mounts it. Other domain members cannot read from, or write to, the file system. Remote access is not allowed; failover does not occur. The `-o server_only` option can be applied only to AdvFS, MFS, and UFS file systems.

Note

The `-o server=`*`name`* option bypasses the normal server selection process and may result in a member that has less than optimal connectivity to the file system's devices serving the file system. In addition, if the member you specify is not available, the file system is not mounted by any other domain member.

The combination of the `-o server=`*name* and `-o server_only` options removes many of the high-availability protections of the CFS file system: the file system can be mounted only by the specified domain member, it can be accessed by only that member, and it cannot fail over to another member. Therefore, use this combination carefully.

The `-o server=`*name* option is valid only in a domain, and only for AdvFS, UFS, MFS, NFS, CDFS, and DVDFS file systems. In the case of MFS file systems, the `-o server=`*name* option is supported in a limited fashion: the file system is mounted only if the specified server is the local node. You can use the `-o server=`*name* option with the `/etc/fstab` file to create domain-member-specific `fstab` entries. See the `mount(8)` reference page for additional usage information.

24.4.2 Freezing an AdvFS Domain Before Cloning

To allow coherent hardware snapshots in multivolume AdvFS domain configurations, file system metadata must be consistent across all volumes when the individual volumes are cloned. To guarantee that the metadata is consistent, Tru64 UNIX Version 5.1B includes the `freezefs` command, which is described in the `freezefs(8)` reference page. The `freezefs` command causes an AdvFS domain to enter into a metadata-consistent frozen state and guarantees that it stays that way until the specified freeze time expires or it is explicitly thawed with the `thawfs` command. All metadata, which can be spread across multiple volumes or logical units (LUNs), is flushed to disk and does not change for the duration of the freeze.

Although the `freezefs` command requires that you specify one or more AdvFS file system mount directories, all of the filesets in the AdvFS domain are affected. The `freezefs` command considers an AdvFS domain to be an indivisible entity. Freezing a file system in an AdvFS domain freezes the entire AdvFS domain.

When you freeze a file system in a clustered configuration, all in-process file system operations are allowed to complete. Some file system operations that do not require metadata updates work normally even if the target file system is frozen; for example, the `read` and `stat` operations.

Although there are slight differences in how the `freezefs` command functions on a single system and in a domain, in both cases metadata changes are not allowed on a frozen AdvFS domain. The most notable differences in the behavior of the commands in a domain are the following:

- Shutting down any domain member causes all frozen file systems in the domain to be thawed.

- If any domain member fails, all frozen file systems in the domain are thawed.

24.4.2.1 Determining Whether an AdvFS Domain Is Frozen

By default, the `freezefs` command freezes a file system for 60 seconds. However, you can use the `-t` option to specify a lesser or greater timeout value in seconds, or to specify that the AdvFS domain is to remain frozen until being thawed by the `thawfs` command.

The `freezefs` command `-q` option allows you to query a file system to determine if it is frozen:

```
# freezefs -q /mnt
/mnt is frozen
```

In addition, the `freezefs` command posts an EVM event when a file system is frozen or thawed. You can use the `evmwatch` and `evmshow` commands to determine if any AdvFS domains in the domain are frozen or thawed, as shown in the following example:

```
# /usr/sbin/freezefs -t -1 /freezetest
freezefs: Successful

# evmget -f "[name sys.unix.fs.vfs.freeze]" | evmshow -t "@timestamp @@"
14-Aug-2002 14:16:51 VFS: filesystem test2_domain#freeze mounted on
/freezetest was frozen

# /usr/sbin/thawfs /freezetest
thawfs: Successful

# evmget -f "[name sys.unix.fs.vfs.thaw]" | evmshow -t "@timestamp @@"
14-Aug-2002 14:17:32 VFS: filesystem test2_domain#freeze mounted on
/freezetest was thawed
```

24.4.3 File System Availability

CFS provides transparent failover of served file systems, if the serving node fails and another node has a physical path to the storage. Nodes that have a direct path to the global storage are DRD servers of the file system. Use the `drdmgr` command to view which nodes are potential servers of various file systems.

For example, to identify which nodes are potential servers of the /usr file system, perform the following steps:

1. Identify the AdvFS domain, as follows:

   ```
   atlas0# df -k /usr
   Filesystem         1024-blocks     Used  Available  Capacity  Mounted on
   cluster_usr#usr        8889096   889810    7980736      11%    /usr
   ```

   In this example, `cluster_usr` is the AdvFS domain, and `usr` is the fileset.

2. Identify the devices in this AdvFS domain, as follows:

```
atlas0# showfdmn cluster_usr
                Id              Date Created  LogPgs  Version  Domain Name
3d12209d.000cfffa    Thu Jun 20 19:36:13 2002     512        4  cluster_usr

Vol   512-Blks       Free   % Used   Cmode   Rblks   Wblks    Vol Name
1L    17778192   15961472      10%      on     256     256    /dev/disk/dsk3g
```

   In this example, there is a single device (`dsk3g`) and a single fileset (`usr`) in the `cluster_usr` AdvFS domain.

3. Identify which nodes can serve the `dsk3g` device, as follows:

```
atlas0# drdmgr -a server dsk3g

View of Data from member atlas0 as of 2002-07-12:16:18:50

Device Name: dsk3g
Device Type: Direct Access IO Disk
Device Status: OK
Number of Servers: 2
Server Name: atlas0
Server State: Server
Server Name: atlas1
Server State: Server
```

   The above output shows that both `atlas0` and `atlas1` have the capability to serve the `/usr` file system, which is located on `dsk3g`. The `cfsmgr` command reveals that `atlas0` is the CFS server, as follows:

```
atlas0# cfsmgr -a server /usr
Domain or filesystem name = /usr
Server Name = atlas0
Server Status : OK
```

   If `atlas0` is shut down, `atlas1` will transparently become the CFS server for `/usr`. File system operations that are in progress at the time of the file system relocation will complete normally.

24.4.3.1 When File Systems Cannot Fail Over

In most instances, CFS provides seamless failover for the file systems in the domain. If the domain 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 local bus 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 `Server Status : Not Served` status for the file system (or AdvFS domain).

Attempts to access the file system return the *`filename`* `I/O error` message.

When a domain member 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.

In the following example, `/local` is a CDSL pointing to the member-specific file system `/cluster/members/member3/local`. This file system will not fail over if `atlas2` — the CFS server, and only DRD server — fails.

1. Identify the AdvFS domain, as follows:

```
atlas2# df -k /local
Filesystem          1024-blocks    Used  Available  Capacity  Mounted on
root3_local#local       6131776      16    6127424      0%    /cluster/
members/member3/local
```

   In this example, `root3_local` is the AdvFS domain, and `local` is the fileset.

2. Identify the devices in the AdvFS domain, as follows:

```
atlas2# showfdmn root3_local
               Id              Date Created  LogPgs  Version  Domain Name
3d11df02.000da810    Thu Jun 20 14:56:18 2002     512        4  root3_local

Vol   512-Blks       Free    % Used   Cmode    Rblks    Wblks    Vol Name
1L    12263552   12254800        0%      on     256      256   /dev/disk/dsk10d
```

   In this example, there is a single device (`dsk10d`) and a single fileset (`local`) in the `root3_local` AdvFS domain.

3. Identify which nodes can serve the `dsk10d` device, as follows:

```
atlas2# drdmgr -a server dsk10d

View of Data from member atlas2 as of 2002-07-12:16:24:11

Device Name: dsk10d
Device Type: Direct Access IO Disk
Device Status: OK
Number of Servers: 1
Server Name: atlas2
Server State: Server
```

   The above output shows that only `atlas2` has the capability to serve the `/local` file system, which is located on `dsk10d`. If `atlas2` is shut down, the `/local` file system will not fail over, and all `/local` file system operations will fail.

24.4.4 Optimizing CFS — Locating and Migrating File Servers

This section describes several ways of tuning CFS performance. This section is organized as follows:

- Automatically Distributing CFS Server Load (see Section 24.4.4.1 on page 24–25)
- Tuning the Block Transfer Size (see Section 24.4.4.2 on page 24–25)
- Changing the Number of Read-Ahead and Write-Behind Threads (see Section 24.4.4.3 on page 24–26)

- Taking Advantage of Direct I/O (see Section 24.4.4.4 on page 24–27)
- Adjusting CFS Memory Usage (see Section 24.4.4.5 on page 24–31)
- Using Memory Mapped Files (see Section 24.4.4.6 on page 24–33)
- Avoid Full File Systems (see Section 24.4.4.7 on page 24–33)
- Other Strategies (see Section 24.4.4.8 on page 24–33)

24.4.4.1 Automatically Distributing CFS Server Load

For information on how to automatically have a particular domain member act as the CFS server for a file system or AdvFS domain, see Section 24.4.1 on page 24–15.

24.4.4.2 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 domain member where the attribute is set. If a file system relocates to another domain 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 mounted after the change.

  To change the default size for CFS I/O blocks domainwide, set the `cfsiosize` kernel attribute on each domain member.

  You can set `cfsiosize` at boot time and at run time. The value must be between 8192 bytes (8KB) and 131072 bytes (128KB), inclusive.

  To change the transfer size of a mounted file system, use the `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 the `cfsmgr(8)` reference page.

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 8KB 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.

Note:

HP does not recommend modifying the default cfsiosize and FSBSIZE values on the nodes that are serving the default HP AlphaServer SC file systems (/, /usr, /var, and the member-specific /local and /tmp) — that is, on members 1 and 2.

24.4.4.3 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, you can improve CFS performance by increasing cfs_async_biod_threads, 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 to 128 inclusive. When not in use, the threads consume few system resources.

Note:

We do not recommend modifying the default cfs_async_biod_threads value on the nodes that are serving the default HP AlphaServer SC file systems (/, /usr, /var, and the member-specific /local and /tmp) — that is, on members 1 and 2.

24.4.4.4 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 domain, this arrangement supports concurrent direct I/O on the file from any member in the domain. That is, regardless of which member originates the I/O request, I/O to a file bypasses the CFS layer and goes directly to the DRD layer. If the file resides on an AdvFS domain on a shared storage medium (for example, RAID), I/O does not go through the HP AlphaServer SC Interconnect.

In an HP AlphaServer SC system, direct I/O is only useful in a system with a single domain, and only on AdvFS file systems. For a multidomain HP AlphaServer SC system, you should use SCFS instead. See Chapter 7 for more information about SCFS.

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 HP AlphaServer SC 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 8KB 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 AdvFS 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 domain 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 HP *Tru64 UNIX Programmer's Guide*.

24.4.4.4.1 Differences Between Domain and Standalone AdvFS Direct I/O

The following list presents direct I/O behavior in a domain 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 domain.

24.4.4.4.2 Cloning a Fileset With Files Open in Direct I/O Mode

As described in Section 24.4.4.4, 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 HP AlphaServer SC 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 HP AlphaServer SC 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 HP AlphaServer SC 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 HP AlphaServer SC Interconnect.

24.4.4.4.3 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 are read requests that 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 are write requests that 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 are read requests that 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 are read requests that 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 is not shipped 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 140KB might contain a fragment at the end that is not a multiple of 8KB. Also, small files less than 8KB in size may consist solely of a fragment.

  To ensure that a file of less than 8KB 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 for direct I/O. This request is not shipped 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 copy-in 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 shipped to the CFS server. If the I/O request encompasses an existing location of the file and does not encompass a fragment, this operation is not shipped 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 shipped 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 140KB might contain a fragment at the end that is not a multiple of 8KB.

  Also, small files less than 8KB in size may consist solely of a fragment. To ensure that a file of less than 8KB 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 is shipped to the CFS server.

24.4.4.5 Adjusting CFS Memory Usage

In situations where one domain 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 not actively used, the CFS server must allocate 800 bytes of system memory for the CFS token structure 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 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 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 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 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 domain 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 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 domain 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 domain members. You can use startup scripts to run the `cfsmgr` and automatically relocate file systems around the domain at member startup.

Setting `svrcfstok_max_percent` below the default is recommended only on smaller memory systems that run out of memory because the 25 percent default value is too high.

24.4.4.6 Using Memory Mapped Files

Using memory mapping to share a file across the domain 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.

24.4.4.7 Avoid Full File Systems

If free space in a file system is less than 50MB 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.

24.4.4.8 Other Strategies

The following measures can improve CFS performance:

- Ensure that the domain members have sufficient system memory.
- In general, sharing a file for read/write access across domain 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 AdvFS 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.

24.4.5 MFS and UFS File Systems Supported

HP AlphaServer SC Version 2.6 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 domain for read/write access, or when you mount an MFS file system in a domain 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 24.4.6. That is, the file system is accessible for both read-only and read/write access only by the member that mounts it. Other domain 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 domain members, you must explicitly mount it read-only.

24.4.6 Partitioning File Systems

CFS makes all files accessible to all domain members. Each domain member has the same access to a file, whether the file is stored on a device connected to all domain 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 domain member. This is referred to as **file system partitioning**.

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, domainwide, can see the file system mount.

The `server_only` option can be applied only to AdvFS, UFS, and MFS file systems. In an HP AlphaServer SC system, the local storage associated with temporary and local file systems (`/tmp` and `/local`) is mounted `server_only`.

Starting with Tru64 UNIX Version 5.1B, file systems can be mounted under a partitioned file system if the file systems to be mounted are also partitioned file systems and are served by the same domain member.

Partitioned file systems are subject to the following limitations:

- No failover *via* CFS

  If the domain member serving a partitioned file system fails, the file system is unmounted. You must remount the file system on another domain 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 under the control of CAA will fail over to a running domain 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 19.13 on page 19–19 for additional information on how to best use the `/etc/exports.aliases` file.

  If you use the default cluster alias to NFS-mount file systems that the domain 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 domain. 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 domain. This is not likely to be acceptable in most environments.

- No mixing partitioned and conventional filesets in the same domain

  The `server_only` option applies to all file systems in an AdvFS domain. The type of the first fileset mounted determines the type for all filesets in the AdvFS domain:

  – If a fileset is mounted without the `server_only` option, then attempts to mount another fileset in the AdvFS domain `server_only` will fail.

  – If a fileset in an AdvFS domain is mounted `server_only`, then all subsequent fileset mounts in that AdvFS 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
```

Note:

By default, `/local` and `/tmp` are mounted with the `server_only` option.

If you wish to remove the `server_only` mount option, run the following command:

```
# scrun -d atlasD0 '/usr/sbin/rcmgr -c delete SC_MOUNT_OPTIONS'
```

If you wish to reapply the `server_only` mount option, run the following command:

```
# scrun -d atlasD0 '/usr/sbin/rcmgr -c set SC_MOUNT_OPTIONS -o server_only'
```

24.4.7 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 AdvFS 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 domain, however, cache coherency can be obtained only among all block device file aliases that reside on the same AdvFS domain or file system.

For example, if domain member `atlas5` serves an AdvFS domain with a block device file and member `atlas6` serves an AdvFS 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.

24.4.8 CFS Restrictions

CFS supports the network file system (NFS) client for read/write access.

When a file system is NFS-mounted in a domain, CFS makes it available for read/write access from all members of the domain. The member that has actually mounted it serves the file system to other domain members.

If the member that has mounted the NFS file system shuts down or fails, the file system is automatically unmounted and CFS begins to clean up the mount points. During the cleanup process, members that access these mount points may see various types of behavior, depending upon how far the cleanup has progressed:

- If members still have files open on that file system, their writes will be sent to a local cache instead of to the actual NFS-mounted file system.
- After all of the files on that file system have been closed, attempts to open a file on that file system will fail with an `EIO` error until the file system is remounted. Applications may encounter “Stale NFS handle” messages. This is normal behavior on a standalone system, as well as in a domain.

Until the CFS cleanup is complete, members may still be able to create new files at the NFS file system’s local mount point (or in any directories that were created locally beneath that mount point).

An NFS file system does not automatically fail over to another domain member. Rather, you must manually remount it — on the same mount point or another — from another domain member to make it available again. Alternatively, booting a domain member will remount those file systems that are listed in the `/etc/fstab` file that are not currently mounted and served in the domain. (If you are using AutoFS, the remount will happen automatically.)

24.5 Managing AdvFS in a Domain

For the most part, the Advanced File System (AdvFS) on a domain is like that on a standalone system. However, there are some domain-specific considerations, and these are described in this section:

- Create Only One Fileset in Cluster Root AdvFS Domain (see Section 24.5.1 on page 24–37)
- Adding Filesets to a Member's Boot Partition Not Recommended (see Section 24.5.2 on page 24–37)
- Do Not Add a Volume to a Member's Root AdvFS Domain (see Section 24.5.3 on page 24–38)
- Using the addvol and rmvol Commands in a Domain (see Section 24.5.4 on page 24–38)
- User and Group File Systems Quotas Are Supported (see Section 24.5.5 on page 24–39)
- Storage Connectivity and AdvFS Volumes (see Section 24.5.6 on page 24–42)

24.5.1 Create Only One Fileset in Cluster Root AdvFS Domain

The root AdvFS 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` AdvFS domain needs to fail over.

As an example of when this situation might occur, consider cloned filesets.

As described in the `advfs(4)` reference page, 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 domainwide root (/) and mount it, the cloned fileset is added to the `cluster_root` AdvFS domain. If the `cluster_root` AdvFS domain has to fail over while the cloned fileset is mounted, the domain will panic.

Note:

If you make backups of the domainwide 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.

24.5.2 Adding Filesets to a Member's Boot Partition Not Recommended

Although you are not prohibited from adding filesets to a member's boot partition, HP does not recommend it. If a member leaves the domain, all filesets mounted from that member's boot partition are force-unmounted and cannot be relocated.

24.5.3 Do Not Add a Volume to a Member's Root AdvFS Domain

You cannot use the `addvol` command to add volumes to a member's root AdvFS domain (*rootmemberID_domain*`#root`). Instead, you must delete the member from the domain, use `diskconfig` or `sysman` to configure the disk appropriately, and then add the member back into the domain. For the configuration requirements for a member boot disk, see the *HP AlphaServer SC Installation Guide*.

24.5.4 Using the addvol and rmvol Commands in a Domain

You can manage AdvFS domains from any domain member, regardless of whether the AdvFS 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 AdvFS domain you are managing, the commands use `rsh` to execute remotely on the member that is the CFS server for the AdvFS domain.

This has the following consequences:

- If `addvol` or `rmvol` is entered from a member that is not the server of the AdvFS domain, and the member that is serving the AdvFS 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. Identify the process IDs 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  6253981    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:

     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. Re-enter the `addvol` or `rmvol` command. In the case of `addvol`, you must use the `-F` option. Use of the `-F` option is necessary because the hung `addvol` command might have already changed the disk label type to AdvFS.

Alternatively, before using either the `addvol` or `rmvol` command on an AdvFS domain, you can do the following:

1. Use the `cfsmgr` command to learn the name of the CFS server of the AdvFS domain:
   `# cfsmgr -d domain_name`

   Or, enter only the command `cfsmgr` and get a list of the CFS servers of all AdvFS 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 re-enter the command. The reason is that 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.

  It is a good idea to enter a `showfdmn` command for the target AdvFS 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 AdvFS 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 `sra install` command automatically places the cluster alias in the `/.rhosts` file, 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.
```

24.5.5 User and Group File Systems Quotas Are Supported

HP AlphaServer SC Version 2.6 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 file system on behalf of a given user or group.

Note:

HP AlphaServer SC Version 2.6 also includes quota support for SCFS file systems. For more information on using quota support in SCFS file systems, see Section 7.4.15 on page 7–22.

Quota support for AdvFS file systems in an HP AlphaServer SC environment is similar to quota support in the base Tru64 UNIX operating 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 domainwide. However, you must edit the `/sys/conf/`*`NAME`* system configuration file on each domain member to configure the system to include the quota subsystem. If you do not perform this step on a domain member, quotas are enabled on that member but you cannot enter quota commands from that member.
- Users and groups are managed domainwide. Therefore, user and group quotas are also managed domainwide.

This section describes information that is unique to managing disk quotas for AdvFS file systems in an HP AlphaServer SC environment. For general information about managing quotas, see the HP *Tru64 UNIX System Administration* guide.

24.5.5.1 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 file system. 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 an HP AlphaServer SC 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 domain members. Therefore, the actual disk space quota for a user or group is determined by the hard limit plus the sum `quota_excess_blocks` on all domain 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 1MB of disk space. The value of `quota_excess_blocks` does not have to be the same on all domain members. You might use a larger `quota_excess_blocks` value on domain members on which you expect most of the data to be generated, and accept the default value for `quota_excess_blocks` on other domain members.

24.5.5.2 Setting the quota_excess_blocks Value

The value for `quota_excess_blocks` is maintained in the `cfs` stanza in the `/etc/sysconfigtab` file.

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.

HP recommends accepting the `quota_excess_blocks` default of 1MB, 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.

24.5.6 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 bus.
- Volumes in the AdvFS domain are on different shared buses, but all of those buses are connected to the same domain members.

The `drdmgr` and `hwmgr` commands can give you information about which systems serve which disks.

24.6 Considerations When Creating New File Systems

Most aspects of creating new file systems are the same in a domain and a standalone environment. The HP *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 domain, see Section 24.3.6 on page 24–12.

The following are important domain-specific considerations for creating new file systems:

- To ensure the highest availability, all disks used for volumes in an AdvFS domain should have the same connectivity.

  It is recommended that all LSM volumes placed into an AdvFS domain share the same connectivity. See Section 25.2 on page 25–2 for more on LSM volumes and connectivity.

  See Section 24.6.1 on page 24–43 for more information about checking for disk connectivity.
- When you determine whether a disk is in use, make sure it is not used as any of the following:
  - The domainwide root (`/`) file system, the domainwide `/var` file system, or the domainwide `/usr` file system
  - A member's boot disk, a member's `/local` disk, or a member's `/tmp` disk

    Do not put any data on a member's boot disk, `/local` disk, or `/tmp` disk.

  See Section 24.6.2 on page 24–43 for more information about checking for available disks.
- There is a single `/etc/fstab` file for all members of a domain, and a member-specific `/etc/member_fstab` file for each member of a domain.

24.6.1 Checking for Disk Connectivity

To ensure the highest availability, all disks used for volumes in an AdvFS domain should have the same connectivity.

Disks have the same connectivity when either one of the following conditions is true:

- All disks used for volumes in the AdvFS domain can access the same Fibre Channel storage, using one Fibre Channel switch.
- All disks used for volumes in the AdvFS domain can access the same Fibre Channel storage, using multiple Fibre Channel switches.

  In the latter case, all disks must use the same multiple Fibre Channel switches — the paths must be identical.

You can use the `hwmgr` command to view all the devices on the domain and then pick out those disks that show up multiple times because they are connected to several members. For example:

```
atlas0# hwmgr -view devices -cluster
HWID: Device Name        Mfg         Model          Hostname    Location
------------------------------------------------------------------------------
56:  /dev/disk/dsk0c     COMPAQ      BD018635C4     atlas0      bus-0-targ-0-lun-0
57:  /dev/disk/dsk1c     COMPAQ      BD018122C9     atlas0      bus-0-targ-1-lun-0
58:  /dev/disk/dsk2c     COMPAQ      BD018122C9     atlas0      bus-0-targ-2-lun-0
59:  /dev/disk/dsk3c     DEC         HSG80          atlas0      bus-1-targ-1-lun-1
59:  /dev/disk/dsk3c     DEC         HSG80          atlas1      bus-1-targ-0-lun-1
60:  /dev/disk/dsk4c     DEC         HSG80          atlas0      bus-1-targ-1-lun-2
60:  /dev/disk/dsk4c     DEC         HSG80          atlas1      bus-1-targ-0-lun-2
61:  /dev/disk/dsk5c     DEC         HSG80          atlas0      bus-1-targ-1-lun-3
61:  /dev/disk/dsk5c     DEC         HSG80          atlas1      bus-1-targ-0-lun-3
.
.
.
```

In this partial output, you can see that `dsk0`, `dsk1`, and `dsk2` are local disks connected to `atlas0`'s local bus. None of these could be used for a file system that needs failover capability, and they would not be good choices for LSM volumes.

Looking at `dsk3` (HWID 59), `dsk4` (HWID 60), and `dsk5` (HWID 61), we see that they are connected to `atlas0` and `atlas1`. These three disks all have the same connectivity.

24.6.2 Checking for Available Disks

When you check whether disks are already in use, check for disks containing the domainwide file systems, and member internal disks and swap areas. The boot disk is an internal disk.

24.6.2.1 Checking for Member Boot Disks and Domainwide AdvFS File Systems

To learn the locations of member boot disks and domainwide AdvFS file systems, check 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:
dsk3b
/etc/fdmns/cluster_usr:
dsk3g
/etc/fdmns/cluster_var:
dsk3h
/etc/fdmns/root1_domain:
dsk0a
/etc/fdmns/root1_local:
dsk0d
/etc/fdmns/root1_tmp:
dsk0e
/etc/fdmns/root2_domain:
dsk6a
/etc/fdmns/root2_local:
dsk6d
/etc/fdmns/root2_tmp:
dsk6e
/etc/fdmns/root_domain:
dsk2a
/etc/fdmns/usr_domain:
dsk2d
/etc/fdmns/var_domain:
dsk2e
/etc/fdmns/projects1_data:
dsk9c
/etc/fdmns/projects2_data:
dsk11c
/etc/fdmns/projects_tools:
dsk12c
```

This output from the `ls` command indicates the following:

- Disk `dsk3` is used by the domainwide file system (`/`, `/usr`, `and /var`). You cannot use this disk.
- Disks `dsk0` and `dsk6` are member boot, local, and tmp 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`.

- Disk dsk2 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.
- Disks dsk9, dsk11, and dsk12 appear to be used for data and tools.

24.6.2.2 Checking for Member Swap Areas

A member's primary swap area is always the b partition of the member boot disk.

However, it is possible that a member has 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. Check 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)
```

24.7 Backing Up and Restoring Files

Back up and restore in a domain 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. You should back up the cluster disk immediately after installation, as described in Chapter 10 of the *HP AlphaServer SC Installation Guide*. You can use this backup to restore the cluster disk as detailed in Section 24.7.2 on page 24–46.

Make sure that all restore software you plan to use is available on the Tru64 UNIX disk of Node 0. Treat this disk as the emergency repair disk for the domain. If the domain loses the root AdvFS domain, cluster_root, you can boot the initial domain member from the Tru64 UNIX disk and restore cluster_root.

The bttape utility is not supported in domains.

The clonefset utility, described in the clonefset(8) reference page, enables you to perform online backups of active files by making a read-only copy (clone) of an active fileset. After you create and mount a clone fileset, you can back up the clone using the vdump

command or other supported backup utility. (The `dump` command is not supported by AdvFS.) You might find it useful to use the `clonefset` command to back up cluster file systems. If you do make backups of the domainwide 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. See Section 24.5.1 on page 24–37 for additional information.

24.7.1 Suggestions for Files to Back Up

In addition to data files, you should regularly back up the following file systems:

- The domainwide root (`/`) file system.
  Use the same backup/restore methods that you use for user data. See also Section 29.4.2 on page 29–6 and Section 29.4.3 on page 29–8.
- The domainwide `/usr` file system.
  Use the same backup/restore methods that you use for user data.
- The domainwide `/var` file system.
  Use the same backup/restore methods that you use for user data.

  You must back up `/var` as a file system separate from `/usr`, because `/usr` and `/var` are separate AdvFS file domains.

24.7.2 Booting the Domain Using the Backup Cluster Disk

Note:

Use the procedure described in this section only if you have created a backup cluster disk as described in Chapter 10 of the *HP AlphaServer SC Installation Guide*.

If you did not create a backup cluster disk, follow the instructions in Section 29.4.2 on page 29–6 to recover the cluster root file system.

If the primary cluster disk fails, you can boot the domain using the backup cluster disk — use the `cluster_root_dev` major and minor numbers to specify the correct `cluster_root` device.

To use these attributes, shut down the domain 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` device(s). As other nodes boot into the domain, their member boot disk information is also updated.

To boot the domain using the backup cluster disk, perform the following steps:

1. Ensure that all domain members are shut down.

2. Boot member 1 interactively, specifying the device major and minor numbers of the backup cluster root partition. You should have noted the relevant device numbers for your backup cluster root partition when you created the backup cluster disk (see Chapter 10 of the *HP AlphaServer SC Installation Guide*).

   In the following example, the major and minor numbers of the backup `cluster_root` partition (`dsk5b`) are 19 and 221 respectively.

```
P00>>>b -fl ia
(boot dkb0.0.0.8.1 -flags ia)
block 0 of dkb0.0.0.8.1 is a valid boot block
reading 18 blocks from dkb0.0.0.8.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 2400
initializing HWRPB at 2000
initializing page table at 3ff58000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

UNIX boot - Thursday August 24, 2000

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
```

3. Boot the other domain members.

24.8 Managing CDFS File Systems

In a domain, 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 domain:

- The `cddevsuppl` command is not supported in a domain.
- The following commands work only when executed from the domain 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. Use 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 HP *TruCluster Server Cluster Highly Available Applications* manual.

24.9 Using the verify Command in a Domain

The `verify` command checks the on-disk metadata structures of AdvFS file systems. A utility, `fixfdmn`, allows you to check and repair corrupted AdvFS domains — for more information, see the `fixfdmn(8)` reference page.

You must unmount all filesets in the AdvFS domain to be checked, before running the `verify` command on an AdvFS domain.

If you are running the `verify` command and the domain member on which it is running fails, it is possible that extraneous mounts are left. This can happen because the `verify` command creates temporary mounts of the filesets that are in the AdvFS domain being verified. On a single system these mounts go away if the system fails while running the utility, but in a domain the mounts fail over to another domain 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 AdvFS 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 AdvFS 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` command.

For more information about `verify`, see the `verify(8)` reference page.

24.9.1 Using the verify Command on Cluster Root

The `verify` command has been modified to allow it to run on active AdvFS domains. Use the `-a` option for this purpose. This allows `verify` to check the cluster root file system, `cluster_root`.

You must execute the `verify -a` command on the member serving the AdvFS domain you are checking. Use the `cfsmgr` command to determine which member serves the AdvFS domain.

When `verify` runs with the `-a` option, it performs only checks of the AdvFS domain. No fixes can be done on the active AdvFS domain. The `-f` and `-d` options cannot be used with the `-a` option.

25

Using Logical Storage Manager (LSM) in an HP AlphaServer SC System

This chapter presents configuration and usage information that is specific to Logical Storage Manager (LSM) in an HP AlphaServer SC environment. The chapter discusses the following subjects:

- Overview (see Section 25.1 on page 25–2)
- Differences Between Managing LSM on an HP AlphaServer SC Domain and on a Standalone System (see Section 25.2 on page 25–2)
- Storage Connectivity and LSM Volumes (see Section 25.3 on page 25–3)
- Configuring LSM on an HP AlphaServer SC Domain (see Section 25.4 on page 25–4)
- Creating an LSM Volume with a Single Striped Plex (see Section 25.5 on page 25–4)

For complete documentation on LSM, see the HP *Tru64 UNIX Logical Storage Manager* manual. Information on installing LSM software can be found in that manual and in the HP *Tru64 UNIX Installation Guide*.

25.1 Overview

Using LSM in a domain is similar to using LSM in a single system. The same LSM software subsets are used for both domains and standalone configurations.

In a domain, LSM provides the following features:

- High availability

  LSM operations continue despite the loss of domain members, as long as the domain itself continues operation and a physical path to the storage is available.

- Performance
  - For I/O within the domain 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 domain 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 domain member for handling.

- Ease of management

  The LSM configuration can be managed from any member.

25.2 Differences Between Managing LSM on an HP AlphaServer SC Domain and on a Standalone System

The following restrictions apply to LSM on an HP AlphaServer SC domain:

- 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 RAID5 volumes are not supported in domains.
- System storage (`cluster_root`, `cluster_usr`, and `cluster_var`) and swap storage partitions should not be encapsulated into LSM volumes.
- There are differences in the process of configuring LSM. These differences are described in Section 25.4 on page 25–4.

The following LSM behavior in a domain varies from the single-system image model:

- Statistics 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. In addition, in a domain, the `voldisk list` command displays only those non-LSM disks that are directly connected to the member on which the command was executed.

25.3 Storage Connectivity and LSM Volumes

When adding disks to an LSM disk group on a domain, note the following points:

- Ensure 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 domain 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 directly connected to all domain members.
- Private disk groups (disk groups whose volumes are all connected to the private bus of a single domain member) are supported, but if that member becomes unavailable, then the domain 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.

  Only striped volumes should be created from private disk groups. No other RAID types are supported for private disk groups in HP AlphaServer SC.

  Any AdvFS file domains created from private disk groups should be mounted with the `server_only` option — for more information, see the `mount(8)` reference page.
- 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.

  It is especially important that the disks in the `rootdg` disk group are shared. However, this might not always be possible.

The drdmgr and hwmgr commands can give you information about which systems serve which disks.

25.4 Configuring LSM on an HP AlphaServer SC Domain

LSM should be configured after all members have been added to the domain.

To configure LSM on an established multimember domain, perform the following steps:

1. Ensure that all members of the domain are booted.
2. If you want the default LSM setup, which is suitable for most environments, enter the following command on one domain member. It does not matter which member.
   ```
   atlas0# volsetup
   ```
   You are queried to list disk names or partitions to be added to the rootdg disk group. Take care when choosing these disks or partitions, because it is not possible to remove all disks from the rootdg disk group without deinstalling and reinstalling LSM. For more information, see the volsetup(8) reference page.
3. When you have configured LSM on one domain member by either method in step 1, you must synchronize LSM on each of the other members, as shown in the following example:
   ```
   # sra command -width 1 -nodes 'atlas[1-31]' -command '/usr/sbin/volsetup -s'
   ```

   Note:

   Use the -width 1 option to ensure that the volsetup -s commands execute sequentially.

   Do not run volsetup -s on the domain member where you first configured LSM (in this case, atlas0).

   If a new member is later added to the domain, do not run the volsetup -s command on the new member. The sra install command automatically synchronizes LSM on the new member.

25.5 Creating an LSM Volume with a Single Striped Plex

An AdvFs domain created using an LSM volume striped over disks on different buses can offer faster performance than a domain created from a single disk's block special device. You can specify the number of disks you want LSM to stripe the data over (the number of stripe columns), or you can let LSM stripe the data over as many disks as necessary based on the volume size and the stripe width you specify, if other than the default.

A volume with a single striped plex does not provide data redundancy.

Note:

In general, do not use LSM to stripe data if you also use a hardware controller to stripe data. In some specific cases such a configuration can improve performance but only if:

- Most of the volume I/O requests are large (greater than or equal to 1 MB).
- The LSM volume is striped over multiple RAID sets on different controllers.
- The LSM stripe size is a multiple of the full hardware RAID stripe size.

The number of LSM columns in each plex in the volume should be equal to the number of hardware RAID controllers. See your hardware RAID documentation for information about how to choose the best number of columns for the hardware RAID set.

The following example of creating an LSM volume with a single striped plex assumes that disk labels have been written to each disk and the `volsetup` command has been run as described in Section 25.4 on page 25–4. The commands to create an LSM volume should be run on a node in the domain on which you want to configure LSM.

1. Initialize disks for LSM use as follows:
   ```
   # voldisksetup -i dsk78 dsk79 dsk80 dsk81
   ```
2. Create an LSM disk group containing the disks as follows:
   ```
   # voldg init atlas0_diskgroup dsk78 dsk79 dsk80 dsk81
   ```
3. Query the maximum size for a striped volume using the disks as follows:
   ```
   # volassist -g atlas0_diskgroup maxsize dsk78 dsk79 dsk80 dsk81
   ```
4. Create a striped volume as follows: (In this example, 853532672 was the value returned by the query in step 3).
   ```
   # volassist -g atlas0_diskgroup -U fsgen make atlas0_fs 853532672
   layout=stripe stripewidth=1024k dsk78 dsk79 dsk80 dsk81
   ```
5. Create and mount an AdvFs filesystem using the LSM volume as follows:
   ```
   # mkfdmn /dev/vol/atlas0_diskgroup/atlas0_fs atlas0_domain
   # mkfset atlas0_domain atlas0_fset
   # mount atlas0_domain#atlas0_fset /scratch
   ```

This will create an LSM volume with a stripe width of 1MB, and four columns (1 per disk) using the disks specified on the command line. The 1MB stripe width means that each column will be 1MB in size for a combined “stripe row” of 4MB. In this case, optimum performance will be for writes in blocks of 4MB. In general, the optimum write block size will be the number of stripes (columns) times the stripe width.

Creating an LSM Volume with a Single Striped Plex

The file system created as shown in this example may be exported as an SCFS file system, or incorporated into a PFS in the same way as a non-LSM volume based AdvFS file system. See the `scfsmgr(8)` and `pfsmgr(8)` reference pages for more information.

26

Managing Security

This information in this chapter is arranged as follows:

- General Guidelines (see Section 26.1 on page 26–2)
- Configuring Enhanced Security (see Section 26.2 on page 26–3)
- Secure Shell Software Support (see Section 26.3 on page 26–4)

26.1 General Guidelines

The information in this section is organized as follows:

- RSH (see Section 26.1.1 on page 26–2)
- sysconfig (see Section 26.1.2 on page 26–2)

For general guidelines on security issues for Tru64 UNIX systems, see the HP *Tru64 UNIX Security Administration* manual.

26.1.1 RSH

To implement a more secure supported version of RSH, enable SSH and configure the `rcmd` emulation (`rsh` replacement) option, as described in Section 26.3 on page 26–4.

For security reasons, system administrators may consider disabling the `rsh` command. Each domain is considered to be a single security domain — see Table 17–6 on page 17–8. If a user has root access to any domain member, they have root access to all members of that domain, regardless of the configuration of RSH. Furthermore, the `rsh` command is required between domain members for the following commands:

- `setld` for software installation
- `shutdown -ch` for domainwide shutdown
- `sysman` for miscellaneous system management operations
- `clu_add_member` to add domain members
- `ris` to add domain members

The default installation configures both the `/etc/hosts.equiv` and `/.rhosts` files. The `/etc/hosts.equiv` file may be empty, but the `/.rhosts` file is needed.

26.1.2 sysconfig

The `sysconfig` command has a `-h` argument. To operate, the `sysconfig` command relies on the `/etc/cfgmgr.auth` file containing the nodenames of all domain members, and on the `cfsmgr` service remaining enabled in the `/etc/inetd.conf` file. Both the `hwmgr` command and the `clu_get_info` command rely on this interface being operational.

26.2 Configuring Enhanced Security

Note:

Configure security only after you have installed RMS on the system.

You can use the SysMan Menu to configure enhanced security, as follows:

1. Choose the Security option from the main SysMan Menu.
2. Select Security Configuration.
3. Set the security mode to Enhanced.
4. Select either the SHADOW or CUSTOM profile.

   Sysman interactively prompts for security configuration information. Enter the appropriate information based on your system requirements.

You can also use the SysMan Menu to configure auditing, as follows:

1. Choose the Security option from the main SysMan Menu.
2. Select Audit Configuration

   Sysman interactively prompts for audit configuration information. Enter the appropriate information based on your system requirements.

Configuring enhanced security is a domainwide operation and only needs to be done once per domain. However, for security to take full effect you must shut down and boot the domain.

By configuring enhanced security on your system, the `/etc/passwd` and `/etc/group` files are not used — password and group information is maintained in the `authcap` database, which is maintained by the security system.

To transfer the `authcap` database between domains, use the `edauth` command, as follows:

1. Use the `edauth -g` command to print the database entries, and redirect the output from the `edauth -g` command to a temporary file.
2. On the secondary domains, use the `edauth -s` command to insert the entries from this generated file into the `authcap` database.

For more information on security and auditing, see the HP *Tru64 UNIX Security Administration* manual.

For more information on the SysMan Menu, and on administering users and groups, see the HP *Tru64 UNIX System Administration* manual.

26.3 Secure Shell Software Support

The Secure Shell (SSH) software is a client/server software application that provides a suite of secure network commands that can be used in addition to, or in place of, traditional non-secure network commands (sometimes referred to as the `R*` commands).

By default, the Secure Shell software is installed when you upgrade to, or install, the Tru64 UNIX Version 5.1B or higher operating system software.

This section describes how to configure the Secure Shell software, and is organized as follows:

- Sample Default Configuration (see Section 26.3.1 on page 26–4)
- Secure Shell Software Commands (see Section 26.3.2 on page 26–10)
- Client Security (see Section 26.3.3 on page 26–11)
- Host-Based Security (see Section 26.3.4 on page 26–11)

26.3.1 Sample Default Configuration

This section provides the following sample configurations:

- Sample Default Client Configuration (Example 26–1 on page 26–4)
- Sample Default Server Configuration (Example 26–2 on page 26–7)

Example 26–1 Sample Default Client Configuration

```
## ssh2_config
## SSH 2.0 Client Configuration File
##

## The "*" is used for all hosts, but you can use other hosts as
## well.
*:

## COMPAQ Tru64 UNIX specific

# EnforceSecureRutils: Secure the r* utilities (no, yes)
#  When using this option hostbased authentication must be enabled on the
#  server.  The use of this option affects Sysman, RIS, and any other
#  script or utility which makes use of rsh, rcp, rlogin, or uses rcmd().
#  Please consult the SSH Admin Guide to learn more about its use.

        EnforceSecureRutils             no

## General

        VerboseMode                     no
```

```
#       QuietMode                       yes
#       DontReadStdin                   no
#       BatchMode                       yes
#       Compression                     yes
#       ForcePTTYAllocation             yes
#       GoBackground                    yes
#       EscapeChar                      ~
#       PasswordPrompt                  "%U@%H's password: "
        PasswordPrompt                  "%U's password: "
        AuthenticationSuccessMsg        yes


## Network

        Port                            22
        NoDelay                         no
        KeepAlive                       yes
#       SocksServer                     socks://
mylogin@socks.ssh.com:1080/203.123.0.0/16,198.74.23.0/24


## Crypto

        Ciphers                         AnyStdCipher
        MACs                            AnyMAC
        StrictHostKeyChecking           ask
#       RekeyIntervalSeconds            3600


## User public key authentication

        IdentityFile                    identification
# AuthorizationFile is invalid, commented out by ssh-validate-conf
#       AuthorizationFile               authorization
        RandomSeedFile                  random_seed


## Tunneling

#       GatewayPorts                    yes
#       ForwardX11                      yes
#       ForwardAgent                    yes


# Tunnels that are set up upon logging in

#       LocalForward                    "110:pop3.ssh.com:110"
#       RemoteForward                   "3000:foobar:22"


## SSH1 Compatibility

        Ssh1Compatibility               yes
```

```
        Ssh1AgentCompatibility          none
#       Ssh1AgentCompatibility          traditional
#       Ssh1AgentCompatibility          ssh2
#       Ssh1Path                        /usr/local/bin/ssh1

## Authentication
## Hostbased is not enabled by default.

#       AllowedAuthentications          hostbased,publickey,password
        AllowedAuthentications          publickey,password


# For ssh-signer2 (only effective if set in the global configuration
# file, usually /etc/ssh2/ssh2_config)

#       DefaultDomain                   foobar.com
#       SshSignerPath                   ssh-signer2


## Examples of per host configurations

#alpha*:
#       Host                            alpha.oof.fi
#       User                            user
#       PasswordPrompt                  "%U:s password at %H: "
#       Ciphers                         idea

#foobar:
#       Host                            foo.bar
#       User                            foo_user
```

Example 26–2 Sample Default Server Configuration

```
## sshd2_config
## SSH 2.4 Server Configuration File
##

## General

        VerboseMode                     no
#       QuietMode                       yes
# AllowCshrcSourcingWithSubsystems is invalid, commented out by ssh-validate-conf
#       AllowCshrcSourcingWithSubsystems        no
# ForcePTTYAllocation is invalid, commented out by ssh-validate-conf
#       ForcePTTYAllocation             no
        SyslogFacility                  AUTH
#       SyslogFacility                  LOCAL7

## Network

        Port                            22
        ListenAddress                   0.0.0.0
        RequireReverseMapping           no
        MaxBroadcastsPerSecond          0
#       MaxBroadcastsPerSecond          1
#       NoDelay                         yes
#       KeepAlive                       yes
#       MaxConnections                  50
#       MaxConnections                  0
# 0 == number of connections not limited


## Crypto

        Ciphers                         AnyCipher
#       Ciphers                         AnyStd
#       Ciphers                         AnyStdCipher
#       Ciphers                         3des
        MACs                            AnyMAC
#       MACs                            AnyStd
#       MACs                            AnyStdMAC
#       RekeyIntervalSeconds            3600
```

```
## User

        PrintMotd                       yes
        CheckMail                       yes
        UserConfigDirectory             "%D/.ssh2"
#       UserConfigDirectory             "/etc/ssh2/auth/%U"
        UserKnownHosts                  yes
#       LoginGraceTime                  600
#       PermitEmptyPasswords            no
#       StrictModes                     yes

## User public key authentication

        HostKeyFile                     hostkey
        PublicHostKeyFile               hostkey.pub
        RandomSeedFile                  random_seed
# IdentityFile is invalid, commented out by ssh-validate-conf
#       IdentityFile                    identification
        AuthorizationFile               authorization
        AllowAgentForwarding            yes


## Tunneling

        AllowX11Forwarding              yes
        AllowTcpForwarding              yes
#       AllowTcpForwardingForUsers      sjl, cowboyneal@slashdot.org
#       DenyTcpForwardingForUsers       "2[:isdigit:]*4, peelo"
#       AllowTcpForwardingForGroups     priviliged_tcp_forwarders
#       DenyTcpForwardingForGroups      coming_from_outside


## Authentication
## Hostbased and PAM are not enabled by default.

#       BannerMessageFile               /etc/ssh2/ssh_banner_message
#       BannerMessageFile               /etc/issue.net
        PasswordGuesses                 1
        AllowedAuthentications          hostbased,publickey,password
#       AllowedAuthentications          publickey,password
#       RequiredAuthentications         publickey,password
#       SshPAMClientPath                ssh-pam-client
```

```
## Host restrictions

#       AllowHosts                      localhost, foobar.com, friendly.org
#       DenyHosts                       evil.org, aol.com
#       AllowSHosts                     trusted.host.org
#       DenySHosts                      not.quite.trusted.org
        IgnoreRhosts                    no
#       IgnoreRootRHosts                no
# (the above, if not set, is defaulted to the value of IgnoreRHosts)


## User restrictions

#       AllowUsers                      "sj*,s[:isdigit:]##,s(jl|amza)"
#       DenyUsers                       skuuppa,warezdude,31373
#       DenyUsers                       don@untrusted.org
#       AllowGroups                     staff,users
#       DenyGroups                      guest
#       PermitRootLogin                 nopwd
        PermitRootLogin                 yes

## SSH1 compatibility

#       Ssh1Compatibility               <set by configure by default>
#       Sshd1Path                       <set by configure by default>


## Chrooted environment

#       ChRootUsers                     ftp,guest
#       ChRootGroups                    guest


## subsystem definitions

        subsystem-sftp                  sftp-server
```

26.3.2 Secure Shell Software Commands

Table 26–1 describes some commonly used SSH commands.

Table 26–1 Commonly Used SSH Commands

To...	Run the Command	Example	Notes
Login	`ssh`	`atlasms$` **`ssh atlas0 -l root`**	This command: • Logs into `atlas0` from `atlasms` as `root` • Connects to a server that has a running SSH daemon • Asks for a password, even if you have logged into `atlasms` as `root`
Logout	`exit`	`atlasms$` **`exit`**	
Copy Files	`scp2`	`atlasms$` **`scp2`** ***`user@system:/directory/file user@system:/directory/file`***	This command: • Securely copies files to and from a server • Runs with normal user privileges • Allows local paths to be specified without the `user@system:` prefix • Allows relative paths to be used (interpreted relative to the user's home directory) Alternatively, you can use the `scp` command. The installation process creates a symbolic link from the `scp` executable to the `scp2` executable.
Copy files to and from a server (on a client)	`sftp2`	`atlasms$` **`sftp2 [`*`options`*`] `*`hostname`***	This command: • Is an FTP command • Works in a similar way to the `scp2` command • Does not use the FTP daemon or the FTP client for its connections • Runs with normal user privileges Alternatively, you can enter the `sftp` command. The installation process creates a symbolic link from the `sftp` executable to the `sftp2` executable.
Validate Configuration Files	`ssh-validate-conf`	`atlasms#` **`ssh-validate-conf -r -s /etc/ssh2/sshd2_config`**	This command verifies whether the keyword in the Secure Shell server (or the client configuration file) is valid. With the `-r` option, the invalid keywords in the configuration will be configured out.

26.3.3 Client Security

Consider the following client-based security measures when using the Secure Shell Software application. To maintain client security, all traditional non-secure network commands — such as `rlogin`, `rsh` (`r*`), and so on — should be routed through the SSH protocol. To route the `r*` commands, edit the SSH2 client configuration file `/etc/ssh2/ssh2_config` and change the `EnforceSecureRUtils` field from `no` to `yes`, as follows:

```
EnforceSecureRUtils yes
```

When you have done this, all `r*` commands will appear exactly the same as before, but are routed through the SSH protocol.

If there is no SSH2 daemon running on the system that you are trying to log into (or the SSH daemon running is incompatible with your version of the agent), SSH will automatically and invisibly log into that machine using the standard form of whichever `r*` command you used.

Note:

When the default Secure Shell (SSH) configuration is modified on an HP AlphaServer SC system so that it replaces the `r*` commands (that is, `rlogin`, `rsh`, and so on), the `sra shutdown -domain` *N* command ceases to work correctly, because a password is requested for every `rsh` connection made by the underlying `shutdown -ch` command. For information on how to correct this problem, see Section 29.15 on page 29–37.

26.3.4 Host-Based Security

Consider the following host-based security measures when using the Secure Shell Software application:

- Disabling Root Login (see Section 26.3.4.1 on page 26–12)
- Host Restrictions (see Section 26.3.4.2 on page 26–12)
- User Restrictions (see Section 26.3.4.3 on page 26–13)

Note:

All changes explained in this section require the SSHD daemon to be reset. If changes are made to a domain member, then you must reset all domain members. To do this, run the following command on each member:

```
# /sbin/init.d/sshd reset
```

26.3.4.1 Disabling Root Login

The system can restrict login to a specific user (*via* SSH or `r*` commands) by doing the following:

- Disabling `root` login to an SSH2 daemon
- Removing the `ptys` line from the file `/etc/securettys`

When connected to the system, users can still do the following:

- `su` to `root` and proceed as normal
- Log in directly as `root` to any nodes with an external Ethernet connection

To maintain host security, you must disable `root` login on those nodes.

Note:

In a clustered environment, change only the configuration file on the first member of each domain because all of the members of the domain use the same file to determine their setup.

To disable `root` login, edit the SSH2 client configuration file `/etc/ssh2/sshd2_config` to change the `PermitRootLogin` field from `yes` to `no`, as follows:

```
PermitRootLogin no
```

Caution:

Exercise care when disabling `root` permissions. The default settings for the configuration files ensure that only `root` can edit the settings. In addition, the default system setup ensures that only `root` can control the SSHD daemon. Always ensure that there are users on the system who can `su` to `root` before closing off `root` access, and make sure that console access to the nodes is available.

26.3.4.2 Host Restrictions

In the `sshd_config` file, there are options to restrict the machines that can connect *via* SSH to the running SSH daemon.

Table 26–2 lists the restrictions. The two options for host control in the `sshd_config` file are mutually exclusive. If any node is listed under both `AllowHosts` and `DenyHosts`, that host will be denied access to the SSH daemon.

Each option supports the use of wildcards and is sensitive to the complete hostname. For example, when specifying a hostname, if that host is using a fully qualified domain name, the hostname entry in the `sshd_config` file must also use the fully qualified domain name.

Table 26–2 Host Restrictions

Option	Description
`DenyHosts`	Specify hosts/domains disallowed access to the daemon; this setting overrides the `AllowHosts` setting.
`AllowHosts`	Specify hosts/domains allowed access to the daemon.

26.3.4.3 User Restrictions

The User Restrictions apply in the same way as the Hosts Restrictions described in Section 26.3.4.2, and provide an extra level of security to the system. Table 26–3 lists the restrictions.

Note:

Any users attempting to log in from hosts or domains that have been disallowed as described in Section 26.3.4.2 will still be denied access to the daemon.

Table 26–3 User Restrictions

Option	Description
`DenyUsers`	Specify users disallowed access to the daemon; this setting overrides the `AllowUsers` setting.
`AllowUsers`	Specify users allowed access to the daemon.

Part 3:

System Validation and Troubleshooting

27

SC Monitor

SC Monitor monitors critical hardware components in an HP AlphaServer SC system. The current state of critical hardware is stored in the SC database, and can be viewed using either the SC Viewer or the `scmonmgr` command. When SC Monitor detects changes in hardware state, events are generated. Events can be viewed using either the SC Viewer or the `scevent` command.

The information in this chapter is organized as follows:

- Hardware Components Managed by SC Monitor (see Section 27.1 on page 27–2)
- SC Monitor Events (see Section 27.2 on page 27–4)
- Managing SC Monitor (see Section 27.3 on page 27–6)
- Viewing Hardware Component Properties (see Section 27.4 on page 27–15)

27.1 Hardware Components Managed by SC Monitor

Table 27–1 describes the hardware components that are monitored by SC Monitor, and the properties of each hardware component type.

Table 27–1 Hardware Components Managed by SC Monitor

Component	Description	Property	Description
HSG80 RAID System	The status of the HSG80 RAID system is monitored using the in-band Command Line Interface (CLI). The HSG80 component must be monitored from a node that is directly connected to the RAID system (that is, the RAID controllers are visible to the node). The HSG80 uses a dual-controller configuration. If only one controller responds, the status is set to `warning`. If both controllers have failed, or if the fibre fabric connecting the monitoring node and controllers has failed, the status of the RAID system is set to `failed`.	Status	This indicates whether SC Monitor can communicate with the RAID system.
		WWID	This is the worldwide ID of the RAID system.
		Fan Status	This indicates whether there are any fan alerts on the RAID system.
		Power Supply Status	This indicates whether there are any power supply alerts on the RAID system.
		Temperature Status	This indicates whether there are any temperature alerts on the RAID system.
		CLI Message	When an event occurs on an HSG80 RAID system, the CLI prints a message each time the CLI is used. This property contains the last such message.
		Disk Status	For each disk, this indicates whether the disk is of `normal`, `failed`, or `not-present` status.
		Controller Status	This indicates whether a controller is working. There are two of these status indicators — one for each controller.
		Cache Status	This indicates the cache status of each controller.
		Mirrored Cache Status	This indicates the mirrored-cache status of each controller.
		Battery Status	This indicates the battery status of each controller.
		Port Status	This indicates the status of each port on each controller.
		Port Topology	This indicates the port topology of each port on each controller.

Table 27–1 Hardware Components Managed by SC Monitor

Component	Description	Property	Description
SANworks Management Appliance	The SANworks Management Appliance is used to access HSV110 RAID system status. Other than indicating whether it is responsive or not, SC Monitor does not retrieve any properties from the SANworks Management Appliance.	Status	Indicates if the SANworks Management Appliance is responsive or not.
		IP Address	The IP address of the SANworks Management Appliance.
HSV110 RAID System	The status of a HSV110 RAID system is monitored *via* a SANworks Management Appliance. SC Monitor connects to the SANworks Management Appliance and uses scripting to retrieve data about the HSV110 RAID system. If the SANworks Management Appliance does not respond, SC Monitor is unable to monitor the HSV110 RAID system.	Status	Indicates whether the HSV110 is responsive to the SANworks Management Appliance.
		WWID	This is the worldwide ID of the RAID system.
		Fan Status	Indicates the status of fans on each of the possible 18 shelves.
		Power Supply Status	Indicates the status of power supplies on each of the possible 18 shelves.
		Temperature Status	Indicates the status of temperature sensors on each of the possible 18 shelves.
		Port Status	Indicates the status of port 1 and port 2 on each controller (normally there are two controllers).
		Loop Status	Indicates the status of each loop on each controller. The loops are 1A, 1B, 2A, and 2B.
		Disks	Lists the disks controlled by the HSV110 RAID system. Disks are identified by an integer number.
		Failed Disks	Indicates the identification number of disks that are in the failed state.

Table 27–1 Hardware Components Managed by SC Monitor

Component	Description	Property	Description
Extreme or HP ProCurve Switch	The management network is based on Extreme or HP ProCurve switches. SC Monitor uses SNMP to communicate with the switches. Other types of switches are not monitored.	Status	This indicates whether the Extreme or HP ProCurve switch is responding to SNMP requests.
		Fan Status	This indicates the status of each fan.
		Power Supply Status	This indicates the status of each power supply. There is a primary and a backup power supply. The backup power supply is optional.
		Temperature Status	This indicates whether the temperature is in the normal range or not. The `warning-temp` and `critical-temp` attributes determine whether the temperature is normal or not.
Terminal Server	The console network is based on DECserver 900TM or DECserver 732 terminal servers.	Status	This indicates whether the terminal server responds to the `ping(8)` command or not.

Note:

Modular SAN Array 1000 (MSA1000) systems are not monitored by SC Monitor.

Not all devices support all forms of sensors; for example, a ProCurve switch does not support temperature events.

27.2 SC Monitor Events

This section describes the following types of events:

- Hardware Component Events (see Section 27.2.1 on page 27–5)
- EVM Events (see Section 27.2.2 on page 27–5)

27.2.1 Hardware Component Events

When SC Monitor detects a change in the state of any property of a hardware component that is being monitored, an event is posted. Table 27–2 describes the event class and event type for each type of hardware component.

Table 27–2 Hardware Component Events

Hardware Component	Event Class	Event Type
HSG80	`hsg`	Various
HSV110	`hsv`	Various
SANworks Management Appliance	`appliance`	Various
Extreme Switch	`extreme`	Various
HP ProCurve Switch	`procurve`	Various
Terminal Server	`tserver`	`status`

You can view more detailed explanations of the events and possible types by using the `scevent -v` option. For example, to view the event type associated with the HSG80 RAID system, run the following command:

```
# scevent -lvp -f '[class hsg]'
```

Chapter 9 describes how to use the class and type to select events for a specific type of hardware component. You can select all events associated with SC Monitor by using the `hardware` category, as follows:

```
# scevent -f '[category hardware]'
```

27.2.2 EVM Events

Various Tru64 UNIX subsystems use the `EVM(5)` event management system to report events within the subsystem. For example, if an AdvFS domain panic occurs, the AdvFS system posts the `sys.unix.fs.advfs.fdmn.panic` event to the EVM system. The EVM system is domainwide. This means that it is possible to receive the event anywhere in the domain. To see such events systemwide, SC Monitor escalates selected EVM events to the SC event system. For example, the `sys.unix.fs.advfs.fdmn.panic` EVM event is posted as `class=advfs` and `type=fdmn.panic`, and the description field contains the name of the AdvFS domain.

SC Monitor escalates events of these classes: `advfs`, `caa`, `cfs`, `clu`, `nfs`, `unix.hw`. See Chapter 9 for more information about these event classes.

27.3 Managing SC Monitor

This section is organized as follows:

- SC Monitor Attributes (see Section 27.3.1 on page 27–6)
- Specifying Which Hardware Components Should Be Monitored (see Section 27.3.2 on page 27–7)
- Distributing the Monitor Process (see Section 27.3.3 on page 27–9)
- Managing the Impact of SC Monitor (see Section 27.3.4 on page 27–13)
- Monitoring the SC Monitor Process (see Section 27.3.5 on page 27–14)

27.3.1 SC Monitor Attributes

Table 27–3 describes the attributes that you can adjust to affect the operation of SC Monitor. These attributes apply to the Extreme Switch and HSV110 hardware components.

Table 27–3 SC Monitor Attributes

Attribute Name	Description
`warning-temp`	If a temperature sensor exceeds this value (in degrees Celsius, °C), the temperature is considered to be in a `warning` state.
`critical-temp`	If a temperature sensor exceeds this value (in degrees Celsius, °C), the temperature is considered to be in a `failed` state.

You can use the `rcontrol` command to modify the value of an attribute. For example, to change the `warning-temp` attribute, use the `rcontrol` command as follows:

```
# rcontrol set attribute name=warning-temp val=32
```

The change will come into effect the next time you either start the `scmond` daemon, or send a SIGHUP signal to the `scmond` daemon. You can trigger a reload of all `scmond` daemons by running the following command once on any node:

```
# scrun -d all '/sbin/init.d/scmon reload'
```

This sends a SIGHUP signal to one node in each domain — this is sufficient to trigger the `scmond` daemon to reload on each node in that domain.

If your system has a management server, send a SIGHUP signal to the `scmond` daemon by running the following command on the management server:

```
atlasms# /sbin/init.d/scmon reload
```

Note:

Reloading all `scmond` daemons at once will put a considerable load on the `msql3d` daemon.

27.3.2 Specifying Which Hardware Components Should Be Monitored

SC Monitor uses records in the SC database to determine which hardware components to monitor. These records are created in one or more of the following ways:

- Automatically created when the SC database is first built.
- Created by the `scmond` daemon when SC Monitor starts.
- Manually created.

Table 27–4 describes each of the component types.

Table 27–4 Hardware Components Monitored by SC Monitor

Component	Description	Database Entry
HSG80 RAID System	You must use the `scmonmgr` command to add an HSG80 entry to the SC database. You can either use the `scmonmgr detect` command or the `scmonmgr add` command, as follows: • The `scmonmgr detect` command will detect all HSG80 devices on the system, and add the appropriate entries to the SC database so that these HSG80 devices are monitored by SC Monitor. The `scmonmgr detect` command is a domain-level command. You can run this command on all domains, as follows: # **`scrun -d all '/usr/bin/scmonmgr detect -c hsg'`** • The `scmonmgr add` command will add the appropriate entry to the SC database so that the specified HSG80 device is monitored by SC Monitor. For more information on how to detect HSG80 devices, see Chapter 11 of the *HP AlphaServer SC Installation Guide*. You can also use the `scmonmgr` command to remove an HSG80 entry from the SC database.	Manually created
SANworks Management Appliance	You must use the `scmonmgr` command to add a SANworks Management Appliance entry to the SC database. By convention, if the SANworks Management Appliance is connected to the management network, the IP address should be of the form: `10.128.104.<`*`number`*`>`. You can also use the `scmonmgr` command to remove a SANworks Management Appliance entry from the SC database.	Manually created

Table 27–4 Hardware Components Monitored by SC Monitor

Component	Description	Database Entry
HSV110 RAID System	When a SANworks Management Appliance is first scanned by SC Monitor, the `scmond` daemon detects the HSV110 RAID systems. If it finds an HSV110 RAID system that does not already have an entry in the SC database, `scmond` adds an entry for that HSV110 to the SC database. Instead of relying on SC Monitor to detect the HSV110, you can use the `scmonmgr` command to add an HSV110 entry to the SC database. You can also use the `scmonmgr` command to remove an HSV110 entry from the SC database. However, when you next scan the SANworks Management Appliance, the detect process will re-create the HSV110 entry in the SC database.	Created by the `scmond` daemon
Extreme or HP ProCurve Switch	When you build the SC database, the installation process creates entries for the appropriate number of switches for the standard configuration. This default number is the minimum number of switches needed for the number of nodes in the HP AlphaServer SC system. For example, a 16-node system has one switch; a 128-node system has three switches. The installation process also detects whether the switches attached to the system are Extreme switches or HP ProCurve switches. By default, the IP address of the first switch is 10.128.103.1, the second is 10.128.103.2, and so on. If you have more switches, you must use the `scmonmgr` command to add the other switch entries to the SC database. You can also use the `scmonmgr` command to remove a switch entry from the SC database, including the switch entries that were automatically created when the SC database was built.	Created when the SC database is built, and also Manually created

Table 27–4 Hardware Components Monitored by SC Monitor

Component	Description	Database Entry
Terminal Server	When you build the SC database, the installation process creates an entry for each of a default number of terminal servers. This default number is the minimum number of terminal servers needed for the number of nodes in the HP AlphaServer SC system. For example, a 16-node system has one terminal server; a 128-node system has four terminal servers. By default, the name of the first terminal server is `atlas-tc1`, the second is `atlas-tc2`, and so on, where `atlas` is the system name. If you have more terminal servers, you must use the `scmonmgr` command to add the other terminal server entries to the SC database. You must also add entries to the `/etc/hosts` file for these terminal servers. You can also use the `scmonmgr` command to remove a terminal server entry from the SC database, including the terminal server entries that were automatically created when the SC database was built.	Created when the SC database is built, and also Manually created

If you use the `scmonmgr` command to add an object to or remove an object from the SC database, you must send a SIGHUP signal to the `scmond` daemon that is monitoring the object. To determine which server is serving an object, use the `scmonmgr` command, as shown in the following example:

```
# scmonmgr move -o sanapp0
Moving sanapp0 (class appliance):
   from server atlasms (local name: (none))
   to   server atlasms (local name: (none))
No change occured.
```

In this example, `atlasms` is serving `sanapp0`. Use this command before removing an object, so that you can send the SIGHUP signal to the appropriate `scmond` daemon.

HSV110 RAID systems are not monitored as standalone objects; instead, they are monitored while monitoring a SANworks Management Appliance. For more information, see Section 27.3.3.3 on page 27–12. For an explanation of the term “server”, and information on how to send SIGHUP signals, see Section 27.3.3 on page 27–9.

27.3.3 Distributing the Monitor Process

This section is organized as follows:

- Overview (see Section 27.3.3.1 on page 27–10)
- Managing the Distribution of HSG80 RAID Systems (see Section 27.3.3.2 on page 27–11)
- Managing the Distribution of HSV110 RAID Systems (see Section 27.3.3.3 on page 27–12)

27.3.3.1 Overview

The `scmond` daemon manages the monitoring of hardware components. This daemon runs on all management servers and nodes in the system. However, only some of the `scmond` daemons actually perform monitoring. The SC database contains information that controls which daemons monitor which hardware component. With the exception of the HSG RAID system, any node or management server can monitor all hardware components. An HSG RAID system can only be monitored by nodes that are directly connected to the RAID system.

When the `sra setup` command builds the SC database, all hardware components (or objects) are monitored by the management server (if present) by default. However, in a large system, the monitoring functions can be distributed throughout the system, to minimize the impact of the monitoring activities.

You can examine the monitoring distribution by running the `scmonmgr` command as follows:

```
# scmonmgr distribution
Class: hsg
    Server atlas2 monitors: hsg[4-11]
    Server atlas0 monitors: hsg[1-3,1000]
Class: extreme
    Server atlasms monitors: extreme[1-8]
Class: tserver
    Server atlasms monitors: atlas-tc[1-8]
Class: appliance
    Server atlasms monitors: sanapp0
SAN Appliance: sanapp0
    SCHSV01 (hsv)
    SCHSV02 (hsv)
    SCHSV05 (hsv)
    SCHSV03 (hsv)
    SCHSV06 (hsv)
    SCHSV04 (hsv)
    SCHSV07 (hsv)
    SCHSV08 (hsv)
```

In this example, the management server monitors 17 devices: all Extreme Switches (`extreme1` to `extreme8` inclusive), all terminal servers (`atlas-tc1` to `atlas-tc8` inclusive), and one SANworks Management Appliance (`sanapp0`).

The monitoring server is either a specific hostname or a domain name. If the server is a specific hostname, that host performs the monitor function. If the server is a domain name, SC Monitor automatically selects one member of the domain to perform the monitor function. Normally, this is member 1 of the domain, but can be member 2 if member 1 fails.

You can move the monitoring of an object from one server to another using the `scmonmgr move` command. There are several reasons why you might want to rebalance the distribution:

- To spread the server load over more servers.
- Because the original server has failed.

For example, to move the monitoring of `atlas-tc1` to the `atlasD1` domain, run the following command:

```
# scmonmgr move -o atlas-tc1 -s atlasD1
```

Instead of designating `atlasD1` as the server, you could have designated `atlas32` as the server. However, this would mean that the monitoring of `atlas-tc1` would cease if `atlas32` was shut down. By designating the domain name, `atlas-tc1` will continue to be monitored as long as `atlasD1` maintains quorum.

27.3.3.2 Managing the Distribution of HSG80 RAID Systems

HSG80 RAID systems must be monitored by nodes that are directly connected to the RAID system. Because of this requirement, HSG80 RAID systems must name a specific host as server, instead of a domain name. You can use the `scmonmgr distribution` command to examine the distribution of HSG80 server nodes, as shown in the following example:

```
# scmonmgr distribution -c hsg
Class: hsg
    atlas0 monitors: hsg[0-3]
    atlas32 monitors: hsg4
```

In this example, `atlas0` is responsible for monitoring `hsg0`, `hsg1`, `hsg2`, and `hsg3`, and `atlas32` monitors `hsg4`. You can move the monitoring of HSG80 RAID systems as described in Section 27.3.3.1. However, before doing so, you must determine whether the planned server node can "see" the HSG80 RAID system, and you must identify the "local name" of the HSG80 RAID system. The local name is the device name by which the HSG80 RAID system is accessed. You can determine the local name of an existing HSG80 RAID system as follows (that is, by using the `move` command without designating a new server):

```
# scmonmgr move -o hsg2
Moving hsg2 (class hsg):
   from server atlas0 (local name: scp2)
   to   server atlas0 (local name: scp2)
```

Since you did not designate a new server (by omitting the `-s` flag), no move actually takes place, but `scmonmgr` prints the local name (`scp2`).

You can determine whether a host is capable of serving the HSG80 RAID system as follows:

- If the node is in the same domain as the original server, log onto each node in turn and use the following command:

  ```
  # hwmgr -v dev
  ```

  If the `hwmgr` command lists the same device as the local name (in the previous example, `scp2`), then this node is also able to monitor the HSG80 RAID system.

- If the node is in a different domain, perform the following steps:

  a. Find the WWID of the HSG80 RAID system. You can do so using the `scmonmgr object` command, as shown in the following example:

```
atlas0# scmonmgr object -o hsg2 | grep WWID
HSG80: hsg2 WWID: 5000-1FE1-0009-5180
```

  b. Log into each candidate node in turn and find the HSG80 devices. Use the `hwmgr` command as shown in the following example:

```
atlas33# hwmgr -v dev
 HWID:   Device Name                 Mfg         Model               Location
------------------------------------------------------------------------------
    55:   /dev/disk/floppy0c                     3.5in floppy        fdi0-unit-0
    60:   /dev/disk/dsk0c               COMPAQ   BD018635C4          bus-0-targ-0-lun-0
    65:   /dev/disk/dsk5c               DEC      HSG80               IDENTIFIER=8
    66:   /dev/disk/dsk6c               DEC      HSG80               IDENTIFIER=7
    67:   /dev/disk/dsk7c               DEC      HSG80               IDENTIFIER=6
    70:   /dev/disk/cdrom0c             COMPAQ   CRD-8402B           bus-2-targ-0-lun-0
    71:   /dev/cport/scp0                        HSG80CCL            bus-1-targ-0-lun-0
```

  Identify the devices whose model is `HSG80` or `HSG80CCL` (in this example, such devices are `dsk5c`, `dsk6c`, `dsk7c`, `scp0`).

  c. Determine the WWID of the HSG80 RAID system associated with the device, as shown in the following example:

```
atlas33# /usr/lbin/hsxterm5 -F scp0 'show this' | grep NODE_ID
NODE_ID          = 5000-1FE1-0009-5180
```

  d. If the NODE_ID found in step b matches the WWID of the original object found in step a, this node is also able to monitor the HSG80 RAID System. In this example, the local name on this node is `scp0`.

  e. Having identified the server name (`atlas33`) and the local name on that server (`scp0`), you can move the HSG80 RAID System to the new server, as follows:

```
# scmonmgr move -o hsg2 -s atlas33 -l scp0
Moving hsg2 (class hsg):
   from server atlas0 (local name: scp2)
   to   server atlas33 (local name: scp0)
```

27.3.3.3 Managing the Distribution of HSV110 RAID Systems

HSV110 RAID systems are not monitored independently; instead, they are monitored while monitoring a SANworks Management Appliance. All HSV110 RAID systems that are attached to a SANworks Management Appliance are monitored by the same server. To see which HSV110 RAID systems are monitored by which SANworks Management Appliance, use the `scmonmgr` command as follows:

```
# scmonmgr distribution -c appliance
Class: appliance
    Server atlasms monitors: sanapp[0,2,5]
SAN Appliance: sanapp0
    SCHSV2 (hsv)
```

```
    SCHSV4 (hsv)
    hsv12 (hsv)
    pbs (hsv)
    jun8 (hsv)
SAN Appliance: sanapp2
    no attached objects
SAN Appliance: sanapp5
    no attached objects
```

You can manage the distribution of the SANworks Management Appliance as described in Section 27.3.3.1 on page 27–10.

27.3.4 Managing the Impact of SC Monitor

SC Monitor has some impact on system performance, both in terms of database load and SC Monitor daemons. You can reduce or modify the impact of SC Monitor in the following ways:

- Relocate monitoring to other nodes.

  This applies especially to the monitoring of nodes. This is because member 1 on each domain monitors the other members of the domain. Although node monitoring is infrequent and not very intensive, this small load can have a disproportionate impact on parallel programs. Instead of distributing the monitoring throughout the system, you could concentrate the monitoring onto a smaller number of nodes. Section 27.3.3 on page 27–9 describes how to redistribute monitoring.

- Reduce the frequency of monitoring.

  You can reduce the frequency at which objects are monitored, by modifying the SC database. For example, to see the current frequency, run the following command:

  ```
  # rmsquery "select monitor_period from sc_classes where name='hsg' "
  300
  ```

  This output indicates that HSG80 devices are monitored once every 300 seconds. To reduce the frequency so that HSG80 devices are monitored once every 1000 seconds, run the following command:

  ```
  # rmsquery "update sc_classes set monitor_period=1000 \
  where name='hsg' "
  ```

  The `name` field specifies the type of object whose monitor period is being changed. Table 27–5 gives the name field value for various types of objects.

Table 27–5 Name Field Values in sc_classes Table

Object Type	Name Field in sc_classes Table
HSG80 RAID System	`hsg`
Extreme Switch	`extreme`

Table 27–5 Name Field Values in sc_classes Table

Object Type	Name Field in sc_classes Table
ProCurve Switch	procurve
Terminal Server	tserver
SANworks Management Appliance	appliance

The HSV110 RAID system is monitored as part of the SANworks Management Appliance, so it does not have a distinct record in the sc_classes table.

After you have modified the sc_classes table, you must reload the scmond daemon on those nodes on which you want the new monitor frequency to take effect. The process for reloading the daemon is described in Section 27.3.1 on page 27–6.

27.3.5 Monitoring the SC Monitor Process

Normally, SC Monitor operates in the background and needs no supervision. However, monitoring may cease if nodes or domains are shut down, if scmond daemons die, or if an error occurs in the monitoring process itself.

All hardware components monitored by SC Monitor have two associated status properties:

- The status property indicates the status of the object itself, as described in Table 27–1.
- The monitor_status property indicates whether the object is being monitored normally, as described in Table 27–6.

Table 27–6 Monitoring the SC Monitor Process

monitor_status	Description
normal	The monitor process is working normally.
stale	The object is not being monitored. This is generally because the node or domain responsible for monitoring is shut down. If the node or domain is not shut down, the scmond daemon on that node or domain may have failed.
other_error_message	The monitor process cannot function because of a system failure. For example, the name of a terminal server may be missing from /etc/hosts and cannot be resolved.

You can use the `scmonmgr errors` command to determine whether SC Monitor is operating normally, as shown in the following example:

```
# scmonmgr errors
Class: hsg  Object: hsg4   monitor_status: stale (none)
```

In this example, `hsg4` is not being updated. The `(none)` text indicates that SC Monitor did not have a specific error when processing `hsg4`. The probable cause of this error is that the monitor server for `hsg4` is not running. Use the `scmonmgr distribution` command to determine which node is monitoring `hsg4`.

27.4 Viewing Hardware Component Properties

SC Viewer provides the primary mechanism for viewing hardware component properties. SC Viewer also allows you to view the properties of hardware components that are managed by other subsystems; for example, the RMS `swmgr` daemon manages the HP AlphaServer SC Interconnect. For more information about SC Viewer, see Chapter 10.

You can also use the `scmonmgr object` command to view the properties of hardware components. Use the `scmonmgr distribution` command to list the objects of interest, and then use the `scmonmgr object` command as shown in the following example:

```
# scmonmgr object -o hsg1

HSG80: hsg1  WWID: 5000-1FE1-000D-6460
  Status: normal (Monitor status: normal)
  Fans: N   PSUs: N   Temperature: N
  CLI Message: (none)
  Controller: ZG04404038  Status: N
    Cache:  N  Mirrored Cache: N  Battery: N
    Port 1: N Topology: FABRIC (fabric up)
    Port 2: N Topology: FABRIC (fabric up)
  Controller: ZG04404123  Status: N
    Cache:  N  Mirrored Cache: N  Battery: N
    Port 1: N Topology: FABRIC (fabric up)
    Port 2: N Topology: FABRIC (fabric up)
  Disks:                              Target ID
  Channel     0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15
  -------     ---------------------------------------------------------------
     1        N   N   N   N   N   N   -   -   N   -   -   -   -   -   -   -
     2        N   N   N   N   N   N   -   -   N   -   -   -   -   -   -   -
     3        N   N   N   N   N   N   -   -   N   -   -   -   -   -   -   -
     4        N   N   N   N   N   N   -   -   N   -   -   -   -   -   -   -
     5        N   N   N   N   N   N   -   -   N   -   -   -   -   -   -   -
     6        N   N   N   N   N   N   -   -   N   -   -   -   -   -   -   -

  Rack: 0   Unit: 0   (Key Normal:N  Warning:W Failed:F Not present:-)
```

27.4.1 The scmonmgr Command

Use the `scmonmgr` command to manage SC Monitor. The syntax of the `scmonmgr` command is as follows:

- `scmonmgr add -c appliance -o name -s server -i ip_addr [-r rack -u unit]`
- `scmonmgr add -c extreme -o name -i ip_addr -s server [-r rack -u unit]`
- `scmonmgr add -c procurve|extreme -o name -i ip_addr -s server [-r rack -u unit]`
- `scmonmgr add -c hsg -o name -i WWID [-s server] [-r rack -u unit]`
- `scmonmgr add -c hsv -o name -i WWID -a appliance [-r rack -u unit]`
- `scmonmgr add -c tserver -o name [-t type] -s server [-r rack -u unit]`
- `scmonmgr classes`
- `scmonmgr detect -c class [-d 0|1]`
- `scmonmgr distribution [-c class]`
- `scmonmgr errors [-c class] [-o name]`
- `scmonmgr help`
- `scmonmgr move -o name [-c class] [-s server] [-l localname]`
- `scmonmgr object -o name`
- `scmonmgr remove [-c class] -o name`

Table 27–7 describes the `scmonmgr` commands in alphabetical order.

Table 27–7 scmonmgr Commands

Command	Description
`add`	Adds a record to the SC database for an object of the specified class. You must specify certain object properties — these vary depending on the class of object, as shown in the above syntax.
`classes`	Shows the types (classes) of objects being monitored.
`detect`	Detects all monitored devices in the domain in which the `scmonmgr detect` command is run, and adds them as monitored devices. In HP AlphaServer SC Version 2.6, only HSG80 devices are detected. To detect HSG80 devices, run the following command: # **`scmonmgr detect -c hsg`**
`distribution`	Shows which servers are responsible for monitoring various objects.
`errors`	Shows any errors that are preventing an object from being monitored.
`help`	Prints help information.
`move`	Allows you to move the monitoring of an object from one server to another server.
`object`	Shows the data being retrieved by the monitor process for a given object.

Table 27–7 scmonmgr Commands

Command	Description
remove	Removes an object of the specified class and name (deletes its record from the SC database).

Table 27–8 describes the scmonmgr command options in alphabetical order.

Table 27–8 scmonmgr Command Options

Command	Description
-a	Specifies the name of the SANworks Management Appliance that monitors this object. In Version 2.6, this applies to HSV110 RAID systems.
-c	Specifies the object class(es) affected by the current scmonmgr command. Use the scmonmgr classes command to list all valid classes.
-d	Specifies whether debugging should be enabled for the scmonmgr detect command: • If -d 0 is specified, debugging is disabled. • If -d 1 is specified, debugging is enabled. If the -d option is not specified, debugging is disabled.
-i	Specifies an identifier for the object, as follows: • If the object is monitored through a network, this is the IP address of the object. • If the object is monitored through a Fibre Channel interconnect, this is the WWID of the object.
-l	Specifies the local name of an object.
-o	Specifies the name of the object affected by the current scmonmgr command.
-r	Specifies the number of the rack in which the object resides.
-s	Specifies the name of the node or domain that monitors an object.
-t	Specifies the type of terminal server. Valid types are DECserver732 or DECserver900.
-u	Specifies the number of the unit (position in the rack) in which the object resides.

The scmonmgr command is also described in the scmonmgr(8) reference page.

28

Using System Event Analyzer to Diagnose Node Problems

This chapter describes how to use System Event Analyzer (previously called Compaq Analyze) to diagnose, and recover from, hardware problems with HP AlphaServer SC nodes in an HP AlphaServer SC system.

These diagnostics will help you to determine the cause of a node hardware failure, or identify whether a node may be having problems. Most of the diagnostics examine the specified node and summarize any abnormalities found; several diagnostics suggest possible fixes. You can then determine the necessary action (if any) to quickly recover a failed node.

The diagnostics will not analyze software errors or a kernel panic. If an HP AlphaServer SC node is not responding because of a software problem with the kernel or any user processes, the diagnostics will not tell you what has happened — they can only diagnose hardware errors.

This chapter describes software that has been developed to maintain an HP AlphaServer SC system. The Tru64 UNIX operating system also provides the HP AlphaServer SC system administrator with various error detection and diagnosis facilities. Examples of such tools include `sysman`, `evmviewer`, `envconfig`, and so on. The HP AlphaServer SC software complements these Tru64 UNIX tools and, where necessary, supersedes their use.

The information in this chapter is organized as follows:

- Overview of Node Diagnostics (see Section 28.1 on page 28–2)
- Obtaining System Event Analyzer (see Section 28.2 on page 28–3)
- Installing System Event Analyzer (see Section 28.3 on page 28–3)
- Producing Consolidated Error Reports Using System Event Analyzer (see Section 28.4 on page 28–7)
- Using the sra diag Command on Individual Nodes (see Section 28.5 on page 28–10)
- Using the System Event Analyzer Command Line Interface (see Section 28.6 on page 28–14)
- Using the System Event Analyzer Web User Interface (see Section 28.7 on page 28–14)

- Managing the Size of the Error Log Files (see Section 28.8 on page 28–16)
- Checking the Status of the System Event Analyzer Processes (see Section 28.9 on page 28–19)
- Stopping the System Event Analyzer Processes (see Section 28.10 on page 28–19)
- Removing System Event Analyzer (see Section 28.11 on page 28–19)

28.1 Overview of Node Diagnostics

To check the status of a node, run the `rinfo` command, as described in Chapter 5.

To perform detailed node diagnostics, use System Event Analyzer, as described in this chapter.

System Event Analyzer is one of a suite of tools known as the Web-Based Enterprise Service (WEBES). The System Event Analyzer tool helps to identify errors and faults in an HP AlphaServer SC node. You can use this tool to constantly monitor an HP AlphaServer node and provide updates to a system administrator if a problem should arise. System Event Analyzer monitors various system files to learn about events in an HP AlphaServer node.

You can perform the diagnostics in several modes, as follows:

- Perform an analysis using `sra diag` and System Event Analyzer.

  This involves the following steps:

  a. Run the `sra diag` command to analyze the node.

  b. If errors are reported, examine the diagnostic report, which is recorded in the `/var/sra/diag/`*`node_name`*`.sra_diag_report` file created in step (a) above.

  c. If appropriate, examine the more detailed report from System Event Analyzer, which is recorded in the `/var/sra/diag/`*`node_name`*`.analyze_report` file.

  For more information on this type of diagnostic analysis, see Section 28.4 on page 28–7.

  In HP AlphaServer SC Version 2.6, you can choose how to use the System Event Analyzer functionality. You can use System Event Analyzer on one node (typically the management server) in your HP AlphaServer SC system to produce a single consolidated report for all (or optionally some) nodes in the system. The binary error log events from the nodes are copied to a single consolidated error log on one (host) node, and the report is then generated from the consolidated log by the copy of System Event Analyzer that is installed on the host node. Thus System Event Analyzer can provide a single consolidated report for a number of nodes across the system. For information on how to generate a report from a consolidated log, see Section 28.4 on page 28–7.

  You can also use the `sra diag` command and System Event Analyzer on each node to generate reports for the individual nodes.

If you intend to generate only consolidated reports for the entire HP AlphaServer SC system, then you only need to install System Event Analyzer on one node in the system (typically the management server).

- Perform real-time monitoring using System Event Analyzer

  Use the System Event Analyzer Web User Interface (WUI) to perform real-time monitoring of nodes that you consider critical.

  To use the System Event Analyzer WUI, the WEBES Director must be running on the nodes to be monitored, as described in Section 28.7 on page 28–14.

- Perform a partial analysis using the `sra diag` command

  If a node is halted, it is not possible to run System Event Analyzer on the node. However, the `sra diag` command can access some status information using the Remote Management Console (RMC) on the HP AlphaServer node.

28.2 Obtaining System Event Analyzer

System Event Analyzer is one component of the WEBES software. You can obtain WEBES from your local authorized service provider or your HP Customer Support representative. WEBES Version 4.2 is the minimum version supported on HP AlphaServer SC Version 2.6.

28.3 Installing System Event Analyzer

If you wish to install System Event Analyzer on all individual nodes in your HP AlphaServer SC system (and on a management server, if used), you must install System Event Analyzer on the management server and on the first node of each domain. Installing System Event Analyzer on the first node of a domain will install System Event Analyzer on every node in that domain. Only the `root` user can install and operate System Event Analyzer or any WEBES tool.

The HP *Web-Based Enterprise Services Installation Guide* is available at the following URL:
`http://h18023.www1.hp.com/support/svctools/webes/`

28.3.1 Installing System Event Analyzer on a Management Server

Note:

The `sra diag` command requires that System Event Analyzer be installed in the default directory.

If you have a clustered management server, installing System Event Analyzer on one management server will install it on both nodes in the management server cluster.

Installing System Event Analyzer

To install System Event Analyzer on a management server, perform the following steps:

1. To ensure a clean installation, remove any currently installed version of System Event Analyzer (previously called Compaq Analyze), as follows:
   a. Check to see if a version of System Event Analyzer is already installed, as follows:
   ```
   atlasms# setld -i | grep WEBES
   WEBESBASE420    installed    Compaq Web-Based Enterprise Service
   Suite V4.0
   ```
   b. If System Event Analyzer is already installed, remove it as shown in the following example:
   ```
   atlasms# setld -d -f WEBESBASE420
   ```
   Otherwise, go to step 2.
2. Install System Event Analyzer according to the instructions provided in the HP *Web-Based Enterprise Services Installation Guide*.
3. When you have finished installing System Event Analyzer on the management server, check whether System Event Analyzer is set to start automatically when the management server boots — this is the recommended configuration — and if necessary, configure System Event Analyzer to start when the management server boots, as shown in the following example:
   a. Run the following command:
   ```
   atlasms# /usr/sbin/webes_install_update
   ```
   b. When the main menu is displayed, choose option 4, as follows:
   ```
   Please choose from the following:

   1) Remove System Event Analyzer V4.2
   2) Remove Computer Crash Analysis Tool V5.1
   3) Remover all of Web-Based Enterprise Service Suite
   4) Start at Boot Time
   5) Customer Information
   6) System Information
   7) Service Obligation
   8) Start WEBES Director
   9) Stop WEBES Director
   10) Help
   11) Quit

   Choose from the above: [ ? ]:4
   WEBES is not currently configured to start at boot time. Should WEBES
    start at boot time on atlasms?
     [ yes/no ] [ ? ](default= yes):
   ```
   c. Accept the default of `yes`.
   d. When the main menu is displayed again, exit the WEBES Configuration Utility, as follows:
   ```
   Choice:  [ ? ]:11
   ```

4. If you chose not to start System Event Analyzer during the installation process, you can start it on the management server now, by using the following command:
```
atlasms# desta start
```

28.3.2 Installing System Event Analyzer on a Domain Member

In HP AlphaServer SC Version 2.6, the consolidated logging feature allows you to use System Event Analyzer on one node (typically the management server) to produce a consolidated report for all nodes in the HP AlphaServer SC system, without installing System Event Analyzer on individual nodes. HP strongly recommends that you take advantage of the consolidated logging capability; for more information, see Section 28.4 on page 28–7.

With consolidated logging, you no longer need to install System Event Analyzer on every node in your system; however, if you wish, you can install System Event Analyzer on each node in order to use the `sra diag` command.

Installing System Event Analyzer on a domain member will install System Event Analyzer on every member of that domain. Therefore, you must observe the following guidelines:

- Do not install System Event Analyzer on any domain member in an HP AlphaServer SC system until you have installed all other software, and all nodes have been added to the HP AlphaServer SC system.
- Install only System Event Analyzer — do not install any other tool from the WEBES kit. The other WEBES tools are not supported in an HP AlphaServer SC domain.
- Install System Event Analyzer in the default directory.

To install System Event Analyzer on each domain, perform the following steps on the first node of each domain (that is, Nodes 0, 32, 64, 96, and so on):

1. To ensure a clean installation, remove any currently installed version of System Event Analyzer (previously called Compaq Analyze), as follows:
   a. Check to see if a version of System Event Analyzer is already installed, as follows:
```
atlas0# setld -i | grep WEBES
WEBESBASE420    installed    Compaq Web-Based Enterprise Service
Suite V4.0
```
   b. If System Event Analyzer is already installed, remove it as shown in the following example:
```
atlas0# setld -d -f WEBESBASE420
```
      Otherwise, go to step 2.
2. Install System Event Analyzer according to the instructions provided in the HP *Web-Based Enterprise Services Installation Guide*.

3. When you have finished installing System Event Analyzer on the domain members, System Event Analyzer may be set up to automatically start at boot time on each member. HP recommends that System Event Analyzer should not be started automatically on the domain members.

   If necessary, disable the automatic startup of System Event Analyzer on the domain members, as shown in the following example:

   a. Run the following command:

   ```
   atlas0# /usr/sbin/webes_install_update
   ```

   b. When the main menu is displayed, choose option 4, as follows:

   ```
   Please choose from the following:

   1) Remove System Event Analyzer V4.2
   2) Remove Computer Crash Analysis Tool V5.1
   3) Remover all of Web-Based Enterprise Service Suite
   4) Start at Boot Time
   5) Customer Information
   6) System Information
   7) Service Obligation
   8) Start WEBES Director
   9) Stop WEBES Director
   10) Help
   11) Quit

   Choose from the above: [ ? ]:4
   ```

   c. When the *Start at Boot Time* menu is displayed, specify that System Event Analyzer should not start when the nodes boot, as follows:

   ```
   Please enter the nodes on which WEBES should start:

   1) Current node only (atlas0).
   2) All 32 candidate nodes.
   3) Selected nodes.
   4) No nodes.
   5) Help.
   6) Return.

   Choose from the above: [ ? ]:4
   ...updating registry on atlas0
   ...updating registry on atlas1
   ...updating registry on atlas2
   .
   .
   .
   ```

   d. When the main menu is displayed again, exit the WEBES Configuration Utility, as follows:

   ```
   Choice:  [ ? ]:11
   ```

28.4 Producing Consolidated Error Reports Using System Event Analyzer

When you install HP AlphaServer SC Version 2.6, all nodes in the HP AlphaServer SC system, including the management server, are configured to forward a copy of their binary error log events to a consolidated log on a host node (or management server cluster) in the system. By default, the management server (or Node 0 if no management server is used) is assumed to be the host node. If the host node is a management server in a management server cluster, both nodes are configured to receive events *via* the management server cluster alias for the entire HP AlphaServer SC system. It is possible to change this default setup using advanced options on the `scca_setup` command (see Section 28.4.1 on page 28–8 for more information).

The host node is configured to receive error log events from the entire HP AlphaServer SC system. By default, the consolidated log is created as `/var/adm/sc.binary.errlog`. This log will contain a copy of each record forwarded by the local `binlogd` daemons on each node. The `binary.errlog` file on each node is unchanged, and still contains the records specific to the node. On the management server itself, local binary error log events are copied directly to the `sc.binary.errlog`, as well as to the local `binary.errlog`. Due to the possible large number of records in the consolidated log, it is important to implement a periodic management scheme to maintain the log size (see Section 28.8 on page 28–16 for more information).

By using the consolidated logging facility, it is possible to install System Event Analyzer on the management server only, and to produce a report from the consolidated log to represent all nodes across the entire HP AlphaServer SC system.

Note:

In order to use System Event Analyzer to generate a consolidated report, you must have System Event Analyzer installed on the node that is the host for the consolidated log file. This is typically the management server if one is present. See Section 28.3 on page 28–3 for instructions on how to install System Event Analyzer.

To configure the system to produce a consolidated report, run the command shown in the following example on the node that is to be the host for the consolidated log. HP recommends that you use the management server (if one is present) as the host node:

Run the following command:

```
# /usr/bin/scca_setup [-f common-log]-C
```

This command updates the existing installation of System Event Analyzer (the `desta` daemon) on the host node to use the consolidated `/var/adm/sc.binary.errlog` file instead of the local `binary.errlog` file. The `desta` daemon is stopped while this change is made, and then restarted. If the management server is a cluster then all nodes of the management server cluster will be updated.

If `-f` *`common-log`* is specified, *`common-log`* will be used in place of the default `/var/adm/sc.binary.errlog`. The *`common-log`* name must be a fully qualified path name.

The system is now configured to report from the consolidated error log on one host node. If you wish to verify that the correct configuration is in place, run the following command:

```
# /usr/bin/scca_setup -v
```

You can now generate the consolidated report using the System Event Analyzer web user interface or command line interface on the host node.

If you want to change the settings on the host node so that System Event Analyzer on the node reverts to examining only the local binary error log, run the following command:

```
# /usr/bin/scca_setup -l /var/adm/binary.errlog -C
```

This command updates the `desta` daemon on the host node to use the `/var/adm/binary.errlog` file. The `desta` daemon is stopped while this change is made, and then restarted. If the host node is a clustered management server then all management server nodes are updated.

28.4.1 Advanced scca_setup Command Options

Other options are available with the `scca_setup` command, and if you wish, you can use these options to change the settings for the consolidated log setup.

Table 28–1 describes the options for the `scca_setup` command in alphabetical order.

Table 28–1 scca_setup Options

Option	Description
`-a`	Enables the `rbinlogd` daemon on the host node and creates an `/etc/rbinlog.conf` file and an `/etc/rbinlog.auth` file that lists all the nodes. Use this option to enable the `rbinlogd` daemon on the host node to accept incoming messages from the `binlogd` daemons on the other nodes in the HP AlphaServer SC system. The `scca_setup` command with the `-a` option must be run on the node that is hosting the consolidated log.

Table 28–1 scca_setup Options

Option	Description
`-C`	Updates an existing installation of System Event Analyzer (the `desta` daemon) to use the `/var/adm/sc.binary.errlog` file. Use this option to configure System Event Analyzer on the host node to use the consolidated log as the default log. The `scca_setup` command with the `-C` option must be run on the node that is hosting the consolidated log. The `desta` daemon is stopped while this change is made, and then restarted. If the host node is a clustered management server, all management server nodes are updated.
`-configure`	When run on the node that hosts the consolidated log, the `scca_setup` command with the `-configure` option configures the `rbinlogd` demon on the host node to accept log entries from all other nodes in the system (if the host node is a clustered management server, all management server nodes are similarly configured). When run on other nodes, the `-configure` option sets up the nodes to forward their logs to the node that is hosting the consolidated log. This command is typically performed on each node during installation, so it should not be necessary to use this option under normal circumstances.
`-d` *scrun-domain-list*	Can be used with the `-r` and `-R` options to specify the list of domains the `scca_setup` command is to run on. If you do not specify a list, the command runs on all nodes in the system. • Use the case-insensitive keyword `all` to run the `scca_setup` command on all domains in the system. • Use a list to specify a particular domain or domains. Domains can be specified using the domain number (for example, `-d 0`) or the domain name (for example, `-d atlasD0`).
`-f` *common-log*	Can be used with the `-a` and `-C` options to specify the fully qualified path name of the consolidated log. If no path name is specified, the default is `/var/adm/sc.binary.errlog`.
`-r|-m [`*management-station-host*`]`	Results in the `binlog.conf` file on a node being replaced with a new copy that includes a CDSL to the consolidated log on the host node. If you wish to preserve existing entries in the `binlog.conf` files, replace the `-r` option with the `-m` option to merge changes into the existing `binlog.conf` files. Use this option to configure a node to forward a copy of its binary log error events to the host node (*management-station-host*). If *management-station-host* is not specified, the management server is used by default. If you do not use the `-d` option to specify a list of nodes, the command runs on all nodes in the system.
`-R`	Results in the `binlogd` daemon on a node re-reading the `/etc/binlog.conf` file on the node. Run the `scca_setup` command with this option if you have previously used the `-r` or the `-u` option to change the `binlog.conf` file on one or more nodes. If you do not use the `-d` option to specify a list of nodes, the command runs on all nodes in the system.

Table 28–1 scca_setup Options

Option	Description
-S	When the scca_setup command is run with the -S option on a node other than the node that is hosting the consolidated log, it moves the /sbin/rc3.d/S*binlog startup/shutdown sequence (where * is a 2-digit number that determines where in the startup sequence during init level 3 that the rbinlogd daemon will be started) further out into the startup. When run on the host node, it performs the same task, and also links the /sbin/init.d/rbinlog start/stop script into the /sbin/rc3.d/S*rbinlog directory.
-u	Undoes the changes made by the -r and -m options and reverts a node back to its original state. Run the scca_setup command with this option if you want to stop a node from forwarding a copy of its binary error log events to a consolidated log.
-v	Verifies the links and status of the binlogd daemons on all of the nodes.

28.5 Using the sra diag Command on Individual Nodes

This section is organized as follows:

- Running the sra diag Command (see Section 28.5.1 on page 28–10)
- Diagnostics Performed by the sra diag Command (see Section 28.5.2 on page 28–11)
- Reviewing the Reports (see Section 28.5.3 on page 28–12)

28.5.1 Running the sra diag Command

The sra diag command examines the HP AlphaServer node using various SRM and RMC commands, and gathers as much data as possible about the state of the specified node(s). If System Event Analyzer has been installed, and you run the sra diag command with the -analyze yes and -rtde 60 options, the sra diag command provides further error and fault analysis.

Note:

The -analyze option is set to yes, and the -rtde option is set to 60, by default. You may omit these options.

The -analyze option controls whether System Event Analyzer is used or not. If System Event Analyzer is not installed, you should specify -analyze no. If a node is halted, the sra diag command can perform some checks without using System Event Analyzer. If a node is halted, you should use -analyze no so that the diagnostic does not complain that it cannot run System Event Analyzer.

The -rtde option controls whether System Event Analyzer uses old events in the binary error log as part of its analysis. By default, events occurring in the last 60 days are analyzed. If you have replaced a failed hardware component recently, you should specify a smaller value for -rtde so that events caused by the failed component are not used in the analysis. Alternatively, you can specify a larger value so that older events are analyzed.

You can use the sra diag command to generate node-specific reports. You can run the command for one node; or you can specify a range of nodes, which results in an individual report being generated for each node in the range.

Run the sra diag command as shown in the following examples:

- To run the sra diag command for a single node (for example, the second node), run the following command from the management server (if used) or Node 0:
  ```
  # sra diag -nodes atlas1
  ```
  Alternatively, you can explicitly call the default sra diag behavior, as follows:
  ```
  # sra diag -nodes atlas1 -analyze yes -rtde 60
  ```
- To run the sra diag command for multiple nodes (for example, the first six nodes), run the following command from the management server (if used) or Node 0:
  ```
  # sra diag -nodes 'atlas[0-5]'
  ```

After entering the sra diag command, you will be prompted for the root user password. While the sra diag command is running, a popup window displays progress information.

When all of the diagnostics have completed, the sra diag command summarizes the results in the /var/sra/diag/*node_name*.sra_diag_report text file. Examine the contents of this file, as described in Section 28.5.3 on page 28–12.

28.5.2 Diagnostics Performed by the sra diag Command

The following factors determine what diagnostics are performed:

- Is the node(s) at the operating system prompt?
- Is the node(s) a functioning member of the HP AlphaServer SC system?
- Has System Event Analyzer been installed?
- Has the proper root password been given?
- Is the node at single-user level?

The following example shows the sequence of events when you run the sra diag command for a single node that is at the operating system prompt:

1. Determine the current state of the node by accessing it through its console port using other sra commands.
2. The node is found to be running Tru64 UNIX.

3. Invoke the System Event Analyzer Command Line Interface (CLI) `wsea summ` command and save all output from this command. The `wsea summ` command reads the node's binary error log file and locates error events.
4. If the `wsea summ` command reports any error events, run the System Event Analyzer CLI `wsea filterlog` and `wsea analyze` commands. These commands determine the source and severity of, and suggest corrective actions for, any hardware faults on the node.
5. Connect to the node's RMC and check for errors related to the node's hardware.
6. When the diagnostics are complete, create an appropriate text file named `/var/sra/diag/`*`node_name`*`.sra_diag_report`.
7. If the `wsea analyze` command was executed, save its report in an appropriate text file named `/var/sra/diag/`*`node_name`*`.analyze_report`.

28.5.3 Reviewing the Reports

The diagnostics results are placed in the *`node_name`*`.sra_diag_report` file in the `/var/sra/diag` directory, where *`node_name`* is the name of the node being examined.

Note:

The diagnostic results file for a particular node is overwritten each time a new `sra diag` command is performed on that node.

The diagnostic results file has three basic sections, as follows:

- The first section is a header that displays the date, node name, and node type.
- The second section gives a brief summary of the findings.
- The third section of the file — the "Details" section — is different for each type of error or fault found, and may contain the following:
  - Observations of node state
  - Warnings
  - Fatal and non-fatal errors, or an indication that no errors were found
  - A summary of the problems found by System Event Analyzer

Example 28–1 displays an example diagnostic results file for a node (`atlas2`).

Example 28–1 Example Diagnostics Results File — Using System Event Analyzer

```
********************************************************************************
                          AlphaServer SC sra diag Report
********************************************************************************

Date/Time:                      Thu May 16 14:23:57 2002
Node Name:                      atlas2
Platform:                       ES40

_______________________________________________________________________________

Diagnostics Found: System Event Analyzer reports problems

_______________________________________________________________________________

Details:

Summary of events found in this node's binary error log
file (/var/adm/binary.errlog):

================== /var/adm/binary.errlog ================
        Qty    Type  Description
     ------  ------  -------------------------------------------------------------
          1     302  Tru64 UNIX Panic ASCII Message
          1     300  Tru64 UNIX Start-up ASCII Message
          1     660  UnCorrectable System Event
          1     110  Configuration Event
          1     310  Tru64 UNIX Time Stamp Message

    Total Entry Count: 5
     First Entry Date: Thu May 09 15:18:06 GMT 2002
      Last Entry Date: Thu May 09 19:00:32 GMT 2002

The script ran the following commands to analyze this node's
binary error log file (/var/adm/binary.errlog):
  1/ Interesting events were extracted as follows:
     wsea filterlog et=mchk,195,199 &rtde=60
  2/ Then the filtered file was analyzed by System Event Analyzer:
     wsea analyze

* System Event Analyzer reported the following:
    ---------- Problem Found: System uncorrectable DMA memory event detected.  at
Thu May 16 13:23:22 GMT 2002 ----------

   The full report from System Event Analyzer may be obtained as follows:
   1/ Log onto atlas2
   2/ cd  /var/sra/diag/
   3/ The report is in atlas2.analyze_report

--------------------------------------------------------------------------------
```

In this example, the `wsea summ` command found serious errors in the node's binary error log file, so the `wsea analyze` command was run to diagnose the problem. The `wsea analyze` command found one problem. The `Problem Found` line provides a summary of the information. Review the `/var/sra/diag/atlas2.analyze_report` file to see the full details.

28.6 Using the System Event Analyzer Command Line Interface

You can use the System Event Analyzer CLI on any node. You can use System Event Analyzer to analyze a node's binary error log in one of two ways:

- Log into the node and run System Event Analyzer on the node.
- Log into any node of the same domain. The binary error log is in the file `/var/adm/binary.errorlog`. This is a context-dependent symbolic link (CDSL) to each node's actual binary error log file. For example, the binary error log for Node 3 is actually contained in `/var/cluster/members/member4/adm/binary.errlog`. The System Event Analyzer CLI commands can process any named file. You can analyze the binary error log files of nodes that are currently halted.

28.7 Using the System Event Analyzer Web User Interface

To use the System Event Analyzer WUI, the WEBES Director must be running on the nodes to be monitored. As this adds a performance penalty to these nodes, you should only do this if you consider the nodes to be critical. For example, you might consider the management server (if used), and the first two members of each domain — for example, Nodes 0 and 1, Nodes 32 and 33, Nodes 64 and 65, and Nodes 96 and 97 in a 128-node system — to be critical nodes.

The information in this section is organized as follows:

- The WEBES Director (see Section 28.7.1 on page 28–14)
- Invoking the System Event Analyzer WUI (see Section 28.7.2 on page 28–16)

28.7.1 The WEBES Director

The WEBES Director (also known as *the desta Director* or, simply, *the Director*) is a daemon (`desta`) that runs in the background.

You can either configure the Director to start on the specified nodes each time they boot (see Section 28.7.1.1), or you can manually start the Director just before invoking the WUI (see Section 28.7.1.2).

28.7.1.1 Starting the Director at Boot Time

To configure the Director to start at boot time, perform the following steps on each domain:

1. Invoke the WEBES Interactive Configuration Utility, as follows:
   ```
   # /usr/sbin/webes_install_update
   ```
2. When the main menu is displayed, choose option 4, as follows:
   ```
   .
   .
   .
   4) Start at Boot Time
   .
   .
   .

   Choice:  [ ? ]:4
   ```
3. When the *Start at Boot Time* menu is displayed, specify that System Event Analyzer should start on the first two nodes in the domain at boot time, as follows:
   ```
   Please enter the nodes on which WEBES should start:

   1) Current node only (atlas0).
   2) All 32 candidate nodes.
   3) Selected nodes.
   4) Help.
   5) Return.

   Choose from the above: [ ? ]:3

   Enter a list of nodenames [? to see list of names]:atlas0 atlas1

   The following nodes were selected:
   atlas1 atlas0

   Is this list correct? [ yes/no ] [ ? ](default= yes):y
   ```
4. The main menu is displayed again. Select option 11 to exit the WEBES Configuration Utility, as follows:
   ```
   Choice:  [ ? ]:11
   ```
5. The Director is now set up to automatically start on the specified nodes at boot time, but is not currently running. You should now start the Director on these nodes, as described in Section 28.7.1.2.

28.7.1.2 Starting the Director Manually

To manually start the Director, use the `desta start` command. The following example shows how to start the Director on the first two nodes of each domain in a 128-node system:

```
# scrun -n 'atlas[0-1,32-33,64-65,96-97]' 'desta start'
```

28.7.2 Invoking the System Event Analyzer WUI

To invoke the System Event Analyzer WUI, perform the following steps:

1. Ensure that the Director is running on the nodes to be monitored (see Section 28.7.1).
2. Invoke a Web browser and specify a URL to access the node, as shown in the following examples (where `atlas` is an example system name):
   - To access the management server, specify the following URL:
     `http://atlasms:7902`
   - To access Node 0 when your browser is running on a node inside the HP AlphaServer SC system, specify the following URL:
     `http://atlas0:7902`
   - To access Node 0 when your browser is running on another system, specify the external network interface for Node 0; for example:
     `http://atlas0-ext1:7802`

Note:

You can specify the URL in any of the following ways:

- *localhost*`:7902`
- *nodename*`.`*domain*`:7902`
- *xxx*`.`*xxx*`.`*xxx*`.`*xxx*`:7902` (IP address)

Regardless of how you specify the URL, the System Event Analyzer WUI uses the node name (for example, `atlas0`) to identify the node.

28.8 Managing the Size of the Error Log Files

System Event Analyzer uses the historical data in the error log files to provide the system administrator with the most accurate diagnoses possible, when a true problem is detected by System Event Analyzer. Therefore, the operating system does not manage the size of the `binary.errlog` and `sc.binary.errlog` files.

Normally, a well-managed system will not produce excessively large log files, and you may choose to maintain the history and continuity of error logs. However, over time, each individual node's `binary.errlog` file, and the consolidated `sc.binary.errlog` file, will grow in size as the following entries are added: normal configuration updates, time stamps, shutdown events, Tru64 UNIX subsystem events, and hardware warnings or errors.

The binex+ utility, which is based on the Tru64 binex utility, is an HP AlphaServer SC system management facility that operates on the log files produced by the binary error log daemons binlogd and rbinlogd. It can be used to summarize the contents of an error log file, to copy a selected range of events to another file for remote analysis, to clean up corrupted areas of the file, or to split the error log file and archive older records.

HP recommends that you split the error logs on a regular basis to keep them at a manageable size. This will allow the sra diag command and System Event Analyzer to proceed more quickly. To split the error logs, use the HP AlphaServer SC binex+ command with the -s option, and specify the number of days of records you want to keep in the error log. Records older than the specified number of days will be moved out to an archive file, while the configuration data that System Event Analyzer uses will be retained in the original error log files.

To check the size of a binary.errlog file, run the following command:

```
# scrun -n all 'ls -l /var/cluster/members/member/adm/binary.errlog'
```

To split a log file on an individual node, perform the following steps on the node:

1. Run the following command:

   ```
   # /usr/sbin/binex+ -s /var/adm/archived.binary.errlog.mm-dd-yy -r 30d \
   /var/adm/binary.errlog > /var/adm/newlog
   ```

2. Stop the binlogd daemon and replace the original binary error log by running the following commands:

   ```
   # /sbin/init.d/binlog stop
   # desta stop
   # cp -p /var/adm/newlog /var/adm/binary.errlog
   ```

   In this example, all records older than 30 days are moved into the archived.binary.errlog.*mm-dd-yy* file. When you replace the binary.errlog with the reduced file /var/adm/newlog, all of the configuration records needed for System Event Analyzer to run properly are retained.

3. Restart the binlogd daemon, by running the following command:

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

Similarly, to split a consolidated log file, perform the following steps on the host node:

1. Run the following command:

   ```
   # /usr/sbin/binex+ -s /var/adm/archived.sc.binary.errlog.mm-dd-yy -r 30d \
   /var/adm/sc.binary.errlog > /var/adm/newlog
   ```

2. If the host node is a clustered management server (or Node 0, if a management server is not used), stop the binlogd and rbinlogd daemons by running the following commands:

   ```
   # CluCmd /sbin/init.d/binlog stop
   # CluCmd /sbin/init.d/rbinlog stop
   # desta stop
   ```

If the host node is a single management server, stop the binlogd and rbinlogd daemons by running the following commands:

```
# /sbin/init.d/rbinlog stop
# /sbin/init.d/rbinlog stop
# desta stop
```

3. Replace the original consolidated error log by running the following command:

```
# cp -p /var/adm/newlog /var/adm/sc.binary.errlog
```

In this example, all records older than 30 days are moved into the archived.sc.binary.errlog.*mm-dd-yy* file. When you replace the sc.binary.errlog with the reduced file /var/adm/newlog, all of the configuration records needed for System Event Analyzer to run properly are retained.

4. If the host node is a clustered management server (or Node 0, if a management server is not used), restart the binlogd and rbinlogd daemons by running the following commands:

```
# CluCmd /sbin/init.d/binlog start
# CluCmd /sbin/init.d/rbinlog start
# desta start
```

If the host node is a single management server, restart the binlogd and rbinlogd daemons daemon by running the following commands:

```
# /sbin/init.d/binlog start
# /sbin/init.d/rbinlog start
# desta start
```

You can use the cron(8) command to schedule the binex+ utility to run at regular intervals, as shown in the following example cron entry:

```
* * * 10 * /usr/sbin/binex+ -s /var/adm/archived.sc.binary.errlog.'date " \
+%m-%d-%y"' -r 30d /var/adm/sc.binary.errlog >/var/adm/newlog; \
   /sbin/init.d/rbinlog stop ; \
   /bin/rm /var/adm/sc.binary.errlog; \
   /bin/mv /var/adm/newlog /var/adm/sc.binary.errlog; \
   /sbin/init.d/rbinlog start ;\
   desta start
```

In this example, the binex+ utility is scheduled to run on the consolidated log every ten days; when it runs, it splits out and archives all records in the consolidated log that are older than thirty days.

See the binex+(8) reference page for more information on the syntax of the binex+ command.

28.9 Checking the Status of the System Event Analyzer Processes

To check the status of the Director process, use the `desta status` command.

For example, in a 128-node system with a management server, use the following commands:

```
atlasms# desta status
atlasms# scrun -n 'atlas[0-1,32-33,64-65,96-97]' 'desta status'
```

For more information about the Director process, see the *SEA User Guide*.

28.10 Stopping the System Event Analyzer Processes

If your system is configured as described in this chapter, the Director (`desta`) runs in the background — as a daemon process — on the management server (if used) and on the first two nodes of each domain. There may be times when this is not desired. To stop the Director process, use the `desta stop` command. For example, in a 128-node system with a management server, use the following commands:

```
atlasms# desta stop
atlasms# scrun -n 'atlas[0-1,32-33,64-65,96-97]' 'desta stop'
```

For more information about stopping the Director, see the *SEA User Guide*.

28.11 Removing System Event Analyzer

Note:

Removing System Event Analyzer from any node in a domain will remove System Event Analyzer from all nodes that are part of the same domain.

There are no special instructions for removing System Event Analyzer or WEBES from a management server or node in an HP AlphaServer SC system.

To remove System Event Analyzer, run the following command:

```
# setld -d -f WEBESBASE420
```

The `-f` option forces the subset to be deleted even if one or more of the nodes in the domain is down. The WEBES version is documented in the `/usr/opt/compaq/svctools/webes/release.txt` file.

29
Troubleshooting

This chapter describes solutions to problems that can arise during the day-to-day operation of an HP AlphaServer SC system. See also the "Known Problems" section of the *HP AlphaServer SC Release Notes*, and the "Troubleshooting" chapter of the *HP AlphaServer SC Installation Guide*.

HP provides a system check tool, `sys_check`, that gathers and analyzes configuration information for use in debugging or diagnosing system problems. The `sclogs` plug-in component has been created to support the use of the `sys_check` tool in HP AlphaServer SC systems. If you have a problem with your system that is not addressed in this chapter, or if the information and solutions provided here do not solve a problem with your system, you should install and run the `sys_check` tool as described in Section 29.22 on page 29–47 and provide the output to your HP Customer Support representative.

This chapter presents the following topics:

- Booting Nodes Without a License (see Section 29.1 on page 29–3)
- Shutdown Leaves Members Running (see Section 29.2 on page 29–3)
- Backing Up and Restoring Files (see Section 29.3 on page 29–3)
- CFS Problems (see Section 29.4 on page 29–5)
- Crashes and Hangs (see Section 29.5 on page 29–14)
- Network Problems (see Section 29.6 on page 29–19)
- RMS Problems (see Section 29.7 on page 29–25)
- Console Logger Problems (see Section 29.8 on page 29–28)
- Domain Member Fails and Domain Loses Quorum (see Section 29.9 on page 29–30)
- Kernel Crashes (see Section 29.10 on page 29–32)
- Console Messages (see Section 29.11 on page 29–33)

- Korn Shell Does Not Record True Path to Member-Specific Directories (see Section 29.12 on page 29–35)
- Pressing Ctrl/C Does Not Stop scrun Command (see Section 29.13 on page 29–36)
- Setting the HiPPI Tuning Parameters (see Section 29.14 on page 29–36)
- SSH Conflicts with sra shutdown -domain Command (see Section 29.15 on page 29–37)
- FORTRAN: How to Produce Core Files (see Section 29.16 on page 29–37)
- Checking the Status of the SRA Daemon (see Section 29.17 on page 29–37)
- Accessing the HP AlphaServer SC Interconnect Control Processor Directly (see Section 29.18 on page 29–38)
- SC Monitor Fails to Detect or Monitor HSG80 RAID Arrays (see Section 29.19 on page 29–38)
- Changing the Kernel Communications Rail (see Section 29.20 on page 29–40)
- SCFS/PFS File System Problems (see Section 29.21 on page 29–40)
- System Check Tool (see Section 29.22 on page 29–47)

29.1 Booting Nodes Without a License

You can boot a node that does not have a TruCluster Server license. The node joins the domain 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 node 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 node during an emergency.

29.2 Shutdown Leaves Members Running

On rare occasions, a domain shutdown (`shutdown -ch`) may fail to shut down one or more domain members. In this situation, you must complete the domain shutdown by shutting down all members.

Imagine a three-member domain where each member has one vote. During domain 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 domain shutdown never completes.

To avoid an impasse in situations like this, quorum checking is disabled at the start of the domain shutdown process. If a member fails to shut down during domain shutdown, it might appear to be a normally functioning domain 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 nodes that are still running:

- If the nodes are hung and not servicing commands from the console, halt the nodes and generate a crash dump.
- If the nodes are not hung, use the `/sbin/halt` command to halt the nodes.

29.3 Backing Up and Restoring Files

Back up and restore in a domain 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. You should back up the cluster disk immediately after installation, as described in Chapter 10 of the *HP AlphaServer SC Installation Guide*. You can use this backup to restore the cluster disk as detailed in Section 29.3.3 on page 29–4.

Make sure that all restore software you plan to use is available on the Tru64 UNIX disk of Node 0. Treat this disk as the emergency repair disk for the domain. If the domain loses the root domain, `cluster_root`, you can boot the initial domain member from the Tru64 UNIX disk and restore `cluster_root`.

The `bttape` utility is not supported in domains.

29.3.1 Suggestions for Files to Back Up

In addition to data files, you should regularly back up the following file systems:

- The domainwide root (/) file system.

  Use the same backup/restore methods that you use for user data. See also Section 29.4.2 on page 29–6 and Section 29.4.3 on page 29–8.

- The domainwide /usr file system.

  Use the same backup/restore methods that you use for user data.

- The domainwide /var file system.

  Use the same backup/restore methods that you use for user data.

  You must back up /var as a file system separate from /usr, because /usr and /var are separate AdvFS file domains.

29.3.2 Backing Up and Restoring the Management Server Root (/), /usr, and /var File Systems

Guidance on one possible method of backing up and subsequently restoring the root (/), /usr, and /var file systems on the management server is included in Chapter 10 of the *HP AlphaServer SC Installation Guide*. For further information on backup strategies in general, see Chapter 9 of the HP *Tru64 UNIX System Administration* manual.

29.3.3 Booting the Domain Using the Backup Cluster Disk

Note:

Use the procedure described in this section only if you have created a backup cluster disk as described in Chapter 10 of the *HP AlphaServer SC Installation Guide*.

If you did not create a backup cluster disk, follow the instructions in Section 29.4.2 on page 29–6 to recover the cluster root file system.

If the primary cluster disk fails, you can boot the domain using the backup cluster disk — use the cluster_root_dev major and minor numbers to specify the correct cluster_root device.

To use these attributes, shut down the domain 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 device(s). As other nodes boot into the domain, their member boot disk information is also updated.

To boot the domain using the backup cluster disk, perform the following steps:

1. Ensure that all domain members are shut down.
2. Boot member 1 interactively, specifying the device major and minor numbers of the backup cluster root partition. You should have noted the relevant device numbers for your backup cluster root partition when you created the backup cluster disk (see Chapter 10 of the *HP AlphaServer SC Installation Guide*).

   In the following example, the major and minor numbers of the backup `cluster_root` partition (`dsk5b`) are 19 and 221 respectively:

```
P00>>>b -fl ia
(boot dkb0.0.0.8.1 -flags ia)
block 0 of dkb0.0.0.8.1 is a valid boot block
reading 18 blocks from dkb0.0.0.8.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 2400
initializing HWRPB at 2000
initializing page table at 3ff58000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

UNIX boot - Thursday August 24, 2000

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
```

3. Boot the other domain members.

29.4 CFS Problems

This section describes the following cluster file system (CFS) problems:

- Specifying cluster_root at Boot Time (see Section 29.4.1 on page 29–6)
- Recovering the Cluster Root File System to a Disk Known to the Domain (see Section 29.4.2 on page 29–6)
- Recovering the Cluster Root File System to a New Disk (see Section 29.4.3 on page 29–8)
- Recovering When Both Boot Disks Fail (see Section 29.4.4 on page 29–12)
- Resolving AdvFS Domain Panics Due to Loss of Device Connectivity (see Section 29.4.5 on page 29–12)
- Forcibly Unmounting an AdvFS File System or Domain (see Section 29.4.6 on page 29–12)

- /var is Full (see Section 29.4.7 on page 29–13)

29.4.1 Specifying cluster_root at Boot Time

At boot time you can specify the device that the domain uses for mounting `cluster_root`, the cluster root file system. Use this feature only for disaster recovery, when you need to boot from the backup cluster disk. See Section 24.7.2 on page 24–46 for more information about booting from the backup cluster disk. To recover the cluster root file system when you do not have a backup cluster disk, see Section 29.4.2 on page 29–6 and Section 29.4.3 on page 29–8.

29.4.2 Recovering the Cluster Root File System to a Disk Known to the Domain

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 the RAID system is available to restore the file system to, and this disk was part of the domain 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 Tru64 UNIX disk is not available, install Tru64 UNIX on a disk that is local to the domain member. It must be the same version of Tru64 UNIX as that installed on the domain.
  - The member boot disk for this member (`dsk0a` in this example).
  - The device with the backup of cluster root.
- All members of the domain have been halted.

To restore the cluster root, do the following:

1. Boot the node with the base Tru64 UNIX disk. For the purposes of this procedure, we assume this node to be `atlas0`. When booting this node, you may need to adjust expected quorum votes (see Section 29.9 on page 29–30).
2. If this system's name for the device that will be the new cluster root is different to the name the domain had for that device, use the `dsfmgr -m` command to change the device name so that it matches the domain's name for the device.

For example, if the domain's name for the device that will be the new cluster root is dsk3b and the system's name for that device is dsk6b, rename the device with the following command:

```
# dsfmgr -m dsk6 dsk3
```

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/dsk3b 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, dsk3d:
   ```
   # addvol /dev/disk/dsk3d cluster_root
   ```
7. Mount the domain that will become the new cluster root:
   ```
   # mount cluster_root#root /mnt
   ```
8. Restore the 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/dsk3b
   # ln -s /dev/disk/dsk3d
   ```
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/dsk3b
    /dev/disk/dsk3b: block special (19/221)
    # file /dev/disk/dsk3d
    /dev/disk/dsk3d: block special (19/225)
    ```
11. Shut down the system and boot it interactively, specifying the device major and minor numbers of the new cluster root:
    ```
    P00>>>boot -fl ia
    (boot dka0.0.0.8.0 -flags ia)
    block 0 of dka0.0.0.8.0 is a valid boot block
    reading 19 blocks from dka0.0.0.8.0
    bootstrap code read in
    base = 6d4000, image_start = 0, image_bytes = 2600
    initializing HWRPB at 2000
    initializing page table at 7fff0000
    initializing machine state
    ```

```
setting affinity to the primary CPU
jumping to bootstrap code
.
.
.

Enter: <kernel_name> [option_1 ... option_n]
  or: ls [name]['help'] or: 'quit' to return to console
Press Return to boot 'vmunix'
# vmunix cfs:cluster_root_dev1_maj=19 \
cfs:cluster_root_dev1_min=221 cfs:cluster_root_dev2_maj=19 \
cfs:cluster_root_dev2_min=225
```

12. Boot the other domain members.

29.4.3 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 domain is complicated. Before you attempt it, try to find a disk that was already installed on the domain to serve as the new cluster boot disk, and follow the procedure in Section 29.4.2 on page 29–6.

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.
- No disk is available to which you can restore the file system — such a disk must be on the RAID system, and must have been part of the domain 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 Tru64 UNIX disk is not available, install Tru64 UNIX on a disk that is local to the domain member. Make sure that it is the same version of Tru64 UNIX as that installed on the domain.
  - The member boot disk for this member (`dsk0a` in this example).
  - The device with the cluster root backup.
  - The disk or disks for the new cluster root.
- All members of the domain have been halted.

To restore the cluster root, do the following:

1. Boot the node with the base Tru64 UNIX disk. For the purposes of this procedure, we assume this node to be `atlas0`. When booting this node, you may need to adjust expected quorum votes (see Section 29.9 on page 29–30).
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
   ```
   See the *HP AlphaServer SC Installation Guide* for more information about the default assignment of disks.
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 domain databases to the `/etc` directory of the base Tru64 UNIX disk:
   ```
   # 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 disk:
   ```
   # cd /mnt/cluster/members/member1/etc
   # cp dfs1.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/dsk0a
    ```
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 disk:
```
# 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, var_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 step 19.)

    For example:
```
# cd /etc/fdmns/root_domain
# rm *
# ln -s /dev/disk/dsk2a
# cd /etc/fdmns/usr_domain
# rm *
# ln -s /dev/disk/dsk2g
# cd /etc/fdmns/var_domain
# rm *
# ln -s /dev/disk/dsk2h
# cd /etc/fdmns/root1_domain
# rm *
# ln -s /dev/disk/dsk0a
# 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 domain 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 domain 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 step 26.

    For example:
```
# file /dev/disk/dsk5b
/dev/disk/dsk5b: block special (19/221)
# file /dev/disk/dsk8e
/dev/disk/dsk8e: block special (19/227)
```

26. Halt the system and boot it interactively, specifying the device major and minor numbers of the new cluster root:
```
P00>>>boot -fl ia
(boot dka0.0.0.8.0 -flags ia)
block 0 of dka0.0.0.8.0 is a valid boot block
reading 19 blocks from dka0.0.0.8.0
bootstrap code read in
base = 6d4000, image_start = 0, image_bytes = 2600
initializing HWRPB at 2000
initializing page table at 7fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
.
.
.

Enter: <kernel_name> [option_1 ... option_n]
or: ls [name]['help'] or: 'quit' to return to console
Press Return to boot 'vmunix'
# vmunix cfs:cluster_root_dev1_maj=19 \
cfs:cluster_root_dev1_min=221 cfs:cluster_root_dev2_maj=19 \
cfs:cluster_root_dev2_min=227
```

27. Boot the other domain members.

    If during boot you encounter errors with device files, run the `dsfmgr -v -F` command.

29.4.4 Recovering When Both Boot Disks Fail

If both boot disks fail on an HP AlphaServer ES40 or HP AlphaServer ES45, or if the (only) boot disk fails on an HP AlphaServer DS20L, delete and re-add the affected member as described in Section 21.6 and Section 21.7 respectively. If the problem node is the first member of a domain, rebuild the entire domain as described in Section 21.8.2.

29.4.5 Resolving AdvFS Domain Panics Due to Loss of Device Connectivity

An AdvFS domain can panic if one or more storage elements containing a domain or fileset becomes unavailable. The most likely cause of this problem is when a domain member is attached to private storage that is used in an AdvFS domain, and that member leaves the domain. A second possible cause is when a storage device has hardware trouble that causes it to become unavailable. In either case, because no domain 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 domain 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#mytests
Mounted On = /mnt/mytst
Server Name = atlas0
Server Status : OK
```

If you are able to restore connectivity to the device and return it to service, you can 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. For example:

```
# cfsmgr -a SERVER=atlas0 -d mytest_dmn
# cfsmgr -e
Domain or filesystem name = mytest_dmn#mytests
Mounted On = /mnt/mytst
Server Name = atlas0
Server Status : OK
```

29.4.6 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 5.1B includes the `cfsmgr -u` and the `cfsmgr -U` commands.

You can use the `cfsmgr -u` command to forcibly unmount an AdvFS file system or domain that is not being served by any domain member. The unmount is not performed if the file system or domain is being served.

The version of the command that you invoke 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 = atlas0
Server Status : Not Served
# cfsmgr -u /mnt/mytst
```

  If nested mounts on the file system are being unmounted, the forced unmount is not performed. Similarly, if nested mounts are 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.

  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.

- If the `cfsmgr -e` command indicates that the domain or file system is being served, use the `cfsmgr -U` command to forcibly unmount the entire domain:

```
# cfsmgr -e
Domain or filesystem name = mytest_dmn#mytests
Mounted On = /mnt/mytst
Server Name = atlas0
Server Status : OK
# cfsmgr -U /mnt/mytst
```

  Unlike the `-u` flag, `-U` applies only to an entire domain. You cannot unmount just one file system of a served domain; if you specify a file system, the entire domain is unmounted. If nested mounts are on any file system in the domain being 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 the `cfsmgr(8)` reference page. For more information about forcibly unmounting file systems, see Section 22.6.6 on page 22–18.

29.4.7 /var is Full

Do not allow `/var` to become full. If `/var` becomes full, the `msql3d` daemon will be unable to work, and you may have to restore the SC database from backup. The system may also be affected in other ways; for example, `sysman` will not operate correctly, and logins may be inhibited. See also Section 29.7.2 on page 29–27.

29.5 Crashes and Hangs

This section describes the following problems:

- Application Hangs (see Section 29.5.1 on page 29–14)
- Identifying and Booting Crashed Nodes (see Section 29.5.2 on page 29–17)
- Generating Crash Dumps from Responsive Domain Members (see Section 29.5.3 on page 29–17)
- Crashing Unresponsive Domain Members to Generate Crash Dumps (see Section 29.5.4 on page 29–17)
- LSM Hangs at Boot Time (see Section 29.5.5 on page 29–19)

29.5.1 Application Hangs

Use the following procedure to gather information about application hangs. This information should be submitted with the problem report when an application hang is suspected.

Determine whether the application has hung in user space or in a system call (kernel space), as follows:

```
# ps auxww| grep app_name
USER        PID %CPU %MEM    VSZ  RSS TTY      S    STARTED         TIME COMMAND
⋮
```

If the state field (*S*) is `U`, the application may have hung in a system call. If the state is `S` or `I`, the application may have hung in user space. See the `ps(1)` reference page for more details on process states.

29.5.1.1 Application Has Hung in User Space

If the application has hung in user space, use the `ladebug` debugger to provide a user space stack, and thread status if the application is multithreaded, as shown in the following example:

```
atlas1# ps auxww|grep rmsd
root     1052119  3.9  0.1 11.7M 5.9M ??       S      Jul 30     2:25.67 rmsd -f
root     1491288  0.0  0.0 2.27M 208K pts/0    S  + 14:46:46     0:00.01 grep rmsd

atlas1# ladebug -pid 1052119 `which rmsd`
Welcome to the Ladebug Debugger Version 65 (built Apr  4 2001 for
Compaq Tru64 UNIX)
------------------
object file name: /usr/sbin/rmsd
Reading symbolic information ...done
Attached to process id 1052119  ....

Interrupt (for process)

Stopping process localhost:1052119 (/usr/sbin/rmsd).
stopped at [<opaque> __poll(...) 0x3ff80136dc8]
```

```
(ladebug) show thread
  Thread   Name                      State           Substate      Policy          Pri
  ------   ----------------------    ------------    -----------   ------------    ---
>*      1  default thread            blocked         kern poll     SCHED_OTHER     19
       -1  manager thread            blk SCS                       SCHED_RR        19
       -2  null thread for VP 2      running VP 2                  null thread     -1
        2  <anonymous>               blocked         kern select   SCHED_OTHER     19
       -3  null thread for VP 3      running VP 3                  null thread     -1
        3  <anonymous>               blocked         kern poll     SCHED_OTHER     19

Information:  An <opaque> type was presented during execution of the previous
command.  For complete type information on this symbol, recompilation of the
program will be necessary.  Consult the compiler man pages for details on
producing full symbol table information using the -g (and -gall for cxx) flags.

(ladebug) where thread all
Stack trace for thread 1
>0  0x3ff80136dc8 in __poll(0x1400a9800, 0x2, 0x6978, 0x0, 0x0, 0x3ff8013ade4) in
/shlib/libc.so
#1  0x120027f00 in ((SingleServer*)0x14007d940)->SingleServer::
getCMD(sigmask=0x11fffbd18) "singleserver.cc":521
#2  0x1200277d4 in ((SingleServer*)0x14007d940)->SingleServer::
serve(sigmask=0x11fffbd18, housekeeper=0x1200213d4) "singleserver.cc":346
#3  0x12001f9d0 in main(argc=2, argv=0x11fffc018) "rmsd.cc":541
#4  0x12001bfd8 in __start(0x1400a9800, 0x2, 0x6978, 0x0, 0x0, 0x3ff8013ade4) in/
usr/sbin/rmsd

Stack trace for thread -1
#0  0x3ff8057d6f8 in __nxm_thread_block(0x8, 0x3d5bb0db, 0x1, 0x20000505b70,
0x6aeb7, 0x6aeb7) in /shlib/libpthread.so
#1  0x3ff805722e8 in UnknownProcedure240FromFile0(0x8, 0x3d5bb0db, 0x1,
0x20000505b70, 0x6aeb7, 0x6aeb7) in /shlib/libpthread.so
#2  0x3ff8058a4c8 in __thdBase(0x8, 0x3d5bb0db, 0x1, 0x20000505b70, 0x6aeb7,
0x6aeb7) in /shlib/libpthread.so

Stack trace for thread -2
#0  0x3ff8057d628 in __nxm_idle(0x0, 0x0, 0x20000a0f600, 0x1, 0x25, 0x20000a0fc40)
in /shlib/libpthread.so
#1  0x3ff8057bbe8 in __vpIdle(0x0, 0x0, 0x20000a0f600, 0x1, 0x25, 0x20000a0fc40)
in /shlib/libpthread.so
#2  0x3ff80575b58 in UnknownProcedure267FromFile0(0x0, 0x0, 0x20000a0f600, 0x1,
0x25, 0x20000a0fc40) in /shlib/libpthread.so
#3  0x3ff80575a70 in UnknownProcedure266FromFile0(0x0, 0x0, 0x20000a0f600, 0x1,
0x25, 0x20000a0fc40) in /shlib/libpthread.so
#4  0x3ff80572fa8 in UnknownProcedure242FromFile0(0x0, 0x0, 0x20000a0f600, 0x1,
0x25, 0x20000a0fc40) in /shlib/libpthread.so
#5  0x3ff8058a4c8 in __thdBase(0x0, 0x0, 0x20000a0f600, 0x1, 0x25, 0x20000a0fc40)
in /shlib/libpthread.so

Stack trace for thread 2
#0  0x3ff800d1ca8 in __select(0x1000, 0x20000f13aa0, 0x0, 0x0, 0x0, 0x30041818688)
in /shlib/libc.so
#1  0x3ff8016fbf4 in __select_nc(0x1000, 0x20000f13aa0, 0x0, 0x0, 0x0,
0x30041818688) in /shlib/libc.so
#2  0x3ff800e5720 in __svc_run(0x1000, 0x20000f13aa0, 0x0, 0x0, 0x0,
0x30041818688) in /shlib/libc.so
```

```
#3  0x300018186c0 in elan3_run_neterr_svc() "network_error.c":574
#4  0x120022d18 in networkErrorServerThread(param=0x0) "rmsd.cc":1625
#5  0x3ff8058a4c8 in __thdBase(0x1000, 0x20000f13aa0, 0x0, 0x0, 0x0,
0x30041818688) in /shlib/libpthread.so

Stack trace for thread -3
#0  0x3ff8057d628 in __nxm_idle(0x0, 0x0, 0x2000141f600, 0x1, 0x25, 0x2000141fc40)
in /shlib/libpthread.so
#1  0x3ff8057bbe8 in __vpIdle(0x0, 0x0, 0x2000141f600, 0x1, 0x25, 0x2000141fc40)
in /shlib/libpthread.so
#2  0x3ff80575b58 in UnknownProcedure267FromFile0(0x0, 0x0, 0x2000141f600, 0x1,
0x25, 0x2000141fc40) in /shlib/libpthread.so
#3  0x3ff80575a70 in UnknownProcedure266FromFile0(0x0, 0x0, 0x2000141f600, 0x1,
0x25, 0x2000141fc40) in /shlib/libpthread.so
#4  0x3ff80572fa8 in UnknownProcedure242FromFile0(0x0, 0x0, 0x2000141f600, 0x1,
0x25, 0x2000141fc40) in /shlib/libpthread.so
#5  0x3ff8058a4c8 in __thdBase(0x0, 0x0, 0x2000141f600, 0x1, 0x25, 0x2000141fc40)
in /shlib/libpthread.so

Stack trace for thread 3
#0  0x3ff80136dc8 in __poll(0x200019259e0, 0x0, 0x7530, 0x0, 0x0, 0x3ff801f98e4)
in /shlib/libc.so
#1  0x3ffbff869a4 in _rms_msleep(millisecs=30000) "OSF1.cc":308
#2  0x3ffbff17884 in _rms_sleep(secs=30) "utils.cc":145
#3  0x12002201c in ((NodeStats*)0x140095800)->NodeStats::statsMonitor()
"rmsd.cc":1333
#4  0x120033ad8 in ((Thread*)0x1400779c0)->Thread::start() "thread.cc":88
#5  0x120033934 in startFn(param=0x1400779c0) "thread.cc":29
#6  0x3ff8058a4c8 in __thdBase(0x200019259e0, 0x0, 0x7530, 0x0, 0x0,
0x3ff801f98e4) in /shlib/libpthread.so
```

29.5.1.2 Application Has Hung in Kernel Space

If the application is in the `U` state, repeat the `ps` command, and check that the CPU field is zero, or decreasing. Gather a kernel space stack, as follows:

```
# dbx -k /vmunix

dbx version 5.1
Type 'help' for help.

thread 0xfffffc00ffe18700 stopped at  [thread_block:3213 ,0xffffffff000b74c0]
Source not available

warning: Files compiled -g3: parameter values probably wrong
(dbx) set $pid=1048603
(dbx) tstack

....

(dbx) quit
```

29.5.2 Identifying and Booting Crashed Nodes

If the `sra info` command indicates that a node is halted, the node may have crashed. To check whether the node has crashed, perform the following tasks:

1. Check the node's console log file in the `/var/sra/logs` directory on the management server (or on Node 0, if not using a management server).
   For example, `/var/sra/logs/atlas5.log` is the console log file for the `atlas5` node, where `atlas` is an example system name.

   If the node had crashed, the reason for the crash will be logged in this file.
2. Try to boot the node by running the following command on either the management server (if used) or Node 0:
   ```
   # sra boot -nodes atlas5
   ```
   - If the node boots, the crash was caused by a software problem.
   - If the node does not boot, the crash may have been caused by a hardware problem.
3. If the node boots, check the crash dump log files in the `/var/adm/crash` directory. Crash files can be quite large and are generated on a per-node basis.

   For serious domain problems, crash dumps may be needed from all domain members. To get crash dumps from functioning members, use the `dumpsys` command to save a snapshot of the system memory to a dump file.

   See the HP *Tru64 UNIX System Administration* manual for more details on administering crash dump files.

29.5.3 Generating Crash Dumps from Responsive Domain Members

If a serious domain problem occurs, crash dumps might be needed from all domain 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 a crash dump, log in to each running domain member and run the `dumpsys` command. By default, `dumpsys` writes the dump to the member-specific directory `/var/adm/crash`.

For more information, see the `dumpsys(8)` reference page.

29.5.4 Crashing Unresponsive Domain Members to Generate Crash Dumps

You may be asked to deliberately crash a node that is unresponsive. This is so that HP can analyze the crash dump files that are produced from a crash. This section describes how to crash an HP AlphaServer ES45 or HP AlphaServer ES40 node.

Note:

For HP AlphaServer DS20L nodes, this functionality is not supported in HP AlphaServer SC Version 2.6.

To crash a node, perform the following steps:

1. Connect to the node's console, as shown in the following example:
   ```
   # sra -cl atlas2
   ```

   Note:

   Perform the remaining steps on the node's console.

2. Enter RMC mode by entering the following key sequence (do not enter any space or tab characters):
   Ctrl/[Ctrl/[rmc

   The RMC system displays the RMC> prompt.

3. Halt the node, as follows:
   ```
   RMC> halt in
   ```
   The node halts CPU 0 and returns to the SRM console prompt (P00>>>).

4. Halt the remaining CPUs, as follows:
   ```
   P00>>> halt 1
   P00>>> halt 2
   P00>>> halt 3
   ```

5. Crash the system, as follows:
   ```
   P00>>> crash
   ```

6. Enter RMC mode by entering the following key sequence (do not enter any space or tab characters):
   Ctrl/[Ctrl/[rmc

   The RMC system displays the RMC> prompt.

7. Deassert halt, as follows:
   ```
   RMC> halt out
   ```
   The node returns to the SRM console prompt (P00>>>).

8. Boot the node, as follows:
   ```
   P00>>> boot
   ```
   As the node boots, it creates the crash dump files.

If you are asked to generate multiple simultaneous crash dumps, use the crash script provided. For example, to generate simultaneous crash dumps for the first five nodes, run the following command:

```
# sra script -script crash -nodes 'atlas[0-4]' -width 5
```

The `-width` parameter is critical, and must be set to the number of simultaneous crash dumps required.

29.5.5 LSM Hangs at Boot Time

Normally, the `vold` command automatically configures disk devices that can be found by inspecting kernel disk drivers. These automatically configured disk devices are not stored in persistent configurations, but are regenerated from kernel tables after every reboot. Invoking the `vold` command with the `-x noautoconfig` option prevents the automatic configuration of disk devices, forcing the Logical Storage Manager to use only those disk devices listed in the `/etc/vol/volboot` file.

A node may sometimes hang at boot time while starting LSM services. To fix this problem, insert the `-x noautoconfig` option in the `vold` command in the `lsm-startup` script, as follows:

1. Save a copy of the current `/sbin/lsm-startup` file, as follows:
   ```
   # cp -p /sbin/lsm-startup /sbin/lsm-startup.orig
   ```
2. Edit the `/sbin/lsm-startup` file to update the `vold_opts` entry, as follows:
   Before: `vold_opts="$vold_opts -L"`
   After: `vold_opts="$vold_opts -L -x noautoconfig"`
3. Add the `rootdg` disks to the `/etc/vol/volboot` file, as follows:
   ```
   # voldctl add disk rootdg_disk_X
   # voldctl add disk rootdg_disk_Y
   ```

All nodes should now boot successfully.

29.6 Network Problems

This section describes potential networking problems in an HP AlphaServer SC domain and solutions to resolve them. This section is organized as follows:

- Access Problems (see Section 29.6.1 on page 29–20)
- NFS Problems (see Section 29.6.2 on page 29–22)
- Cluster Alias Problems (see Section 29.6.3 on page 29–24)
- Changes to TCP/IP Ephemeral Port Numbers (see Section 29.6.4 on page 29–25)

29.6.1 Access Problems

This section is organized as follows:

- Accessing the Cluster Alias from Outside the Domain (Section 29.6.1.1)
- Accessing External Networks from Externally Connected Members (Section 29.6.1.2)
- Accessing External Networks from Internally Connected Members (Section 29.6.1.3)
- Additional Access Checks (Section 29.6.1.4)

29.6.1.1 Accessing the Cluster Alias from Outside the Domain

Problem: Cannot `ping` the cluster alias from outside the domain.

Solution: Perform a general networking check (do you have the right address, and so on).

Problem: Cannot `telnet` to the cluster alias from outside the domain.

Solution: Check to see if `ping` will work. Check that `telnet` is configured correctly in the `/etc/clua_services` file. Services that require connections to the cluster alias must have `in_alias` specified.

Problem: Cannot `rlogin` or `rsh` to the cluster alias from outside the domain.

Solution: Check that `rlogin` is enabled in the `/etc/inetd.conf` file. Check to see if `telnet` will work. For `rsh` only: check also that ownership permission and contents of the `/.rhosts` file, and of the `.rhosts` file in the user's home area, are correct.

Problem: Cannot `ftp` to the cluster alias from outside the domain.

Solution: Check that `ftp` is enabled in the `/etc/inetd.conf` file. Check that `ftp` is configured correctly in the `/etc/clua_services` file — it should be specified as `in_multi`, and should not be specified as `in_noalias`.

29.6.1.2 Accessing External Networks from Externally Connected Members

Problem: Cannot `ping` external networks from externally connected members.

Solution: Perform a general networking check (do you have the right address, and so on). Also, check the default route configuration, using the information provided in Section 22.1 on page 22–2.

Problem: Cannot `telnet` to external networks from externally connected members.

Solution: Check that the service is configured correctly in the `/etc/clua_services` file. Check to see if any of the other services will work.

Problem: Cannot `rlogin` to external networks from externally connected members.

Solution: Check that the service is configured correctly in the `/etc/clua_services` file. Check to see if any of the other services will work.

Problem: Cannot `ftp` to external networks from externally connected members

Solution: Check that the service is configured correctly in the `/etc/clua_services` file. Check to see if any of the other services will work.

29.6.1.3 Accessing External Networks from Internally Connected Members

Problem: Cannot `ping` external networks from internally-only connected members.

Solution: The `ping` command will not work on a domain member that does not have an external connection. For more information, see Section 29.6.3.1 on page 29–24.

Problem: Cannot `telnet` to external networks from internally-only connected members.

Solution: Check that `telnet` is configured correctly in the `/etc/clua_services` file. Also, check the default route configuration, using the information provided in Section 22.1 on page 22–2, and check the preferred server cluster alias configuration, using the information provided in Section 19.15 on page 19–23.

Problem: Cannot `rlogin` or `rsh` to external networks from internally-only connected members.

Solution: Check the `shell` and `login` entries in the `/etc/clua_services` file.

Problem: Cannot `ftp` to external networks from internally-only connected members.

Solution: Check the `ftp` entry in the `/etc/clua_services` file.

29.6.1.4 Additional Access Checks

In addition to the checks mentioned in the previous sections, perform the following checks:

- Perform a thorough check of the default route configuration in all domains, using the information provided in Section 22.1 on page 22–2.
- Perform a thorough check of the preferred server cluster alias configuration on all domains, using the information provided in Section 19.15 on page 19–23.
- Check the `/etc/clua_services` file for services without `out_alias`.

  Append `out_alias` to such entries, and then reload the `/etc/clua_services` file, by running the following command:

  ```
  # cluamgr -f
  ```
- Check that YP and DNS are configured correctly.
- If you experience problems with a license manager, see Section 19.16 on page 19–29.
- Ensure that `/etc/hosts` has correct entries for the default cluster alias and domain members.

  At a minimum, ensure that `/etc/hosts` has the following:

  – IP address and name for the default cluster alias

- IP address and name for the preferred server cluster aliases

Note:

One requirement for the preferred server cluster aliases architecture is that the addresses must be on a common subnet. This common subnet can be on any network that is shared by the preferred server nodes in the FS domains and the importing nodes in the CS domains, for example, the `ee0` or `ee1` subnets.

- IP address and name for each domain member
- IP address and interface name associated with each member's cluster interconnect interface (`ics0`)
- IP address and interface name associated with each member's system interconnect (`eip0`)

- Ensure 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 domain 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 domain might reveal that the affected alias IP address maps to two or more different hardware addresses.

  If the domain is configured to log messages of severity `err`, search 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`*

  For more information on interpreting this message, see Section 29.6.3.3 on page 29–24.

After you have made the necessary corrections, restart network services on each node, using the following command:

```
atlasms# sra command -nodes all -command "rcinet restart"
```

29.6.2 NFS Problems

This section is organized as follows:

- Node Failure of Client to External NFS Server (Section 29.6.2.1)
- File-Locking Operations on NFS File Systems Hang Permanently (Section 29.6.2.2)

29.6.2.1 Node Failure of Client to External NFS Server

If a node that is acting as a client to an external NFS server fails and cannot be shut down and booted — for example, due to a permanent hardware error — access to the served file system will not be available. Due to a restriction in the current software release, the mount point remains busy and cannot be used by another node to mount the file system. To mount the file system *via* a different node, a new mount point must be used.

29.6.2.2 File-Locking Operations on NFS File Systems Hang Permanently

The `vi` command hangs when trying to open existing files on a file system that is NFS-mounted from a machine that is not part of the domain management network. The `vi` command hangs because it is attempting to obtain an exclusive lock on the file. The hang persists for many minutes; it may not be possible to use Ctrl/C to return to the command prompt.

The workaround is to ensure that the system that is NFS-serving the file system to a domain can resolve the internal domain member names (for example, `atlas0`) of the domain members that mount the NFS file system. The usual way of doing this is to use the internal domain member names as aliases for the address of the external interface on those nodes (for example, create an alias called `atlas0` for the `atlas0-ext1` external interface).

For example, domains `atlasD0` and `atlasD1` both NFS-mount the `/data` file system from the NFS server `dataserv`. The `/data` file system is being mounted by domain members `atlas0` and `atlas32`. These nodes have external interfaces `atlas0-ext1` and `atlas32-ext1` respectively. To avoid the `vi` hang problem, ensure that `dataserv` can resolve `atlas0` to `atlas0-ext1` and `atlas32` to `atlas32-ext1`.

This section describes three common ways of ensuring that the internal domain names can be resolved.

- `/etc/hosts`

  In the `/etc/hosts` file on `dataserv`, define `atlas0` as an alias for `atlas0-ext1`. In the `/etc/hosts` file on `dataserv`, define `atlas32` as an alias for `atlas32-ext1`.

  You must perform this action on every node that is NFS-serving file systems to the domain.

- NIS/YP

  If NIS/YP is in use, and is distributing a hosts table, put the alias definitions for `atlas0` and `atlas32` into this table.

- DNS

  If DNS is in use, and is distributing host address information, define `atlas0` and `atlas32` as aliases for their respective external interface entries.

Note:

If you choose either the NIS/YP option or the DNS option, ensure that `svc.conf` is configured so that hostname resolution checks locally (that is, `/etc/hosts`) before going to `bind` or `yp`. For more information, see the `svc.conf(4)` reference page.

29.6.3 Cluster Alias Problems

This section is organized as follows:

- Using the ping Command in a Domain (Section 29.6.3.1)
- Running routed in a Domain (Section 29.6.3.2)
- Incorrect Cabling Causes “Cluster Alias IP Address in Use” Warning (Section 29.6.3.3)

29.6.3.1 Using the ping Command in a Domain

The `ping` command will not work on a domain member that does not have an external connection.

The `ping` command uses the primitive ICMP (Internet Control Message Protocol) protocol. The `ping` command inserts the IP address of the interface used as the source IP address — it does not use the cluster alias. Therefore, `ping` responses are not seen by nodes with no external interface. The target node sees the ECHO_REQUEST, but cannot route the ECHO_RESPONSE to the `10.128.x.x` address.

29.6.3.2 Running routed in a Domain

Although it is technically possible to run `routed` in a domain, doing so can cause the loss of failover support in the event of a domain member failure. Running `routed` is considered a misconfiguration of the domain and generates console and Event Manager (EVM) warning messages.

The only supported router is `gated`. See also Section 19.15 on page 19–23.

29.6.3.3 Incorrect Cabling Causes “Cluster Alias IP Address in Use” Warning

You may see the following message:

```
Apr 23 15:48:58 atlas0 vmunix: arp: local IP address ww.xxx.yyy.zzz in use by
hardware address 00-50-8B-E3-21-D9
```

where the address *ww.xxx.yyy.zzz* is the cluster alias for `atlasD0`.

This may indicate that the Ethernet cables to `atlas0` were incorrectly cabled, resulting in `ee0` being placed on the external Ethernet network instead of its correct position on the internal management network. In such cases, both `ee0` and `ee1` of `atlas0` are on the external network and this causes the cluster alias code to print the above warning.

However, if the IP address in question is an alias belonging to the local domain, it is normal (as a consequence of the new preferred server cluster alias scheme) for other members of the domain to be using this address.

29.6.4 Changes to TCP/IP Ephemeral Port Numbers

Within HP AlphaServer SC domains, the range of well-known TCP/IP ports available for static use by user applications is different from the range used in TruCluster Server or Tru64 UNIX.

Certain applications that use well known TCP/IP port numbers may need to be configured with different port numbers.

Affected applications include PBS (Portable Batch System) for batch scheduling, and various Licence Management applications.

Using the PBS software as an example, the following ports are typically used:

```
15001 (tcp)
15002 (tcp)
15003(tcp, udp)
15004 (tcp)
```

The ports 15001, 15002, and so on are in the ephemeral range of ports on the HP AlphaServer SC system. As such, they are dynamically issued by the system. The system issues ephemeral ports within the range `ipport_userreserved_min` to `ipport_userreserved`.

On a normal Tru64 UNIX system, this range is from 1024 to 5000. On an HP AlphaServer SC system, these limits have been increased to 7500 and 65000 respectively, because of scalability issues with a shared port space (for example, a cluster alias for more than 32 nodes).

You can check the ephemeral range by running the `sysconfig -q inet` command. Affected applications should not try to use specific ports within the ephemeral range. Instead, they should be reconfigured to use ports either beneath `ipport_userreserved_min` or above `ipport_userreserved`.

29.7 RMS Problems

This section is organized as follows:

- RMS Core Files (see Section 29.7.1 on page 29–26)
- rmsquery Fails (see Section 29.7.2 on page 29–27)
- prun Fails with “Operation Would Block” Error (see Section 29.7.3 on page 29–27)
- Identifying the Causes of Load on msqld (see Section 29.7.4 on page 29–27)

- RMS May Generate “Hostname / IP address mismatch” Errors (see Section 29.7.5 on page 29–27)
- Management Server Reports rmsd Errors (see Section 29.7.6 on page 29–28)

29.7.1 RMS Core Files

By default, RMS will keep all core files. However, the system administrator may have configured RMS to automatically delete core files (see Section 5.5.9.3 on page 5–27), for the following reasons:

- The core files usually contain little diagnostic information, as production jobs are typically compiled with optimizations.
- Having thousands of useless core files scattered on local disks would lead to maintenance problems.

If core files are not being kept, and a process within a job fails, RMS generates core files as follows:

1. Kills off any remaining processes in the job.
2. Produces the following core file:
   `/local/core/rms/`*`resource_id`*`/core.`*`program.node.instance`*
3. Runs the Ladebug debugger on the core file and sends the back trace to `stderr`.
4. Deletes the core file and directory, as RMS frees the allocated resource.

Note:

To use the Ladebug debugger, you need license OSF-DEV. You can obtain this license by purchasing, for example, HP AlphaServer SC Development Software or Developer’s Toolkit for Tru64 UNIX.

If you are not licensed to use the Ladebug debugger, RMS will not print a back trace.

To diagnose a failing program, the programmer should perform the following tasks:

1. Compile the program with the `-g` flag, to specify that debug and symbolic information should be included.
2. Run the job as follows:
   a. Allocate a resource, using the `allocate` command.
   b. Run the job, using the `prun` command.

3. When the program fails, it produces a core file in the standard location. The `prun` command prints the path name of the core file.
4. The programmer can debug this core file and optionally copy it to a more permanent location.
5. When the programmer exits the `allocate` subshell, RMS deletes the core file and directory.

   To save core files from production runs — that is, when a job is run without using the `allocate` command in step 2 above — the programmer should run the job in a script that copies the core file to a permanent location.

29.7.2 rmsquery Fails

The `rmsquery` command may fail with the following error:

```
rmsquery: failed to add transaction log entry: Non unique value for unique index
```

This error indicates that the index data in the SC database has been corrupted — probably because `/var` became full while an update was in progress.

To recover from this situation, perform the following steps:

1. Drop the tables in question:
   ```
   # rmsquery "drop table resources"
   ```
2. Rebuild the tables as follows:
   ```
   # rmstbladm -u
   ```

29.7.3 prun Fails with “Operation Would Block” Error

The `prun` command may fail with the following error:

```
prun: operation would block
```

This error indicates that there is insufficient swap space on at least one of the nodes allocated by RMS to run the job. If you submit a job using the `prun` command, RMS will not start the job if any of the allocated nodes have less than 10% available swap space.

29.7.4 Identifying the Causes of Load on msqld

The command `msqladmin stats` can be used to identify the number of SC database queries per process name.

Running this command at intervals will determine if a particular node or daemon is generating a significant transaction load on `msqld`.

29.7.5 RMS May Generate “Hostname / IP address mismatch” Errors

RMS may generate `hostname / IP address mismatch` errors. This is probably a configuration problem related to the `/etc/hosts` file or DNS setup. Check the following on each domain and on the management server:

- The node in question has only one entry for each network interface in the `/etc/hosts` file.

- Each /etc/hosts entry is correct.
- The nslookup command either returns nothing for each interface on the node, or the IP address returned matches the IP address seen in the /etc/hosts file

See also Section 29.6 on page 29–19.

29.7.6 Management Server Reports rmsd Errors

The management server does not have an HP AlphaServer SC Elan adapter card. Because it cannot access the non-existent card, the rmsd daemon running on the management server will report problems similar to the following in the /var/log/rmsmhd.log file:

```
Jul 19 13:36:01 rmsmhd: server rmsmhd starting
Jul 19 13:48:30 rmsmhd: Error: failed to start peventproxy: failed to start
peventproxy -f
Jul 19 13:48:33 rmsd[atlasms]: Error: failed to open elan control device
```

In addition, if the rmsd daemon is stopped and you run the rinfo -s rmsd command, an error is displayed.

These errors are normal for nodes without an HP AlphaServer SC Elan adapter card, and can be ignored.

29.8 Console Logger Problems

This section is organized as follows:

- Port Not Connected Error (see Section 29.8.1 on page 29–28)
- CMF Daemon Reports connection.refused Event (see Section 29.8.2 on page 29–29)
- Console Output Wraps Strangely (see Section 29.8.3 on page 29–30)

29.8.1 Port Not Connected Error

When using the sra command, you may get a failure message similar to the following:

```
09:01:16 atlas2 info CMF-Port This node's port is not connected
09:01:16 atlas2 status:failed CMF-Port This node's port is not connected
```

This means that the cmfd daemon was unable to establish a connection to the terminal server port for this node (atlas2). This may be caused by either of the following scenarios:

- **Scenario 1:** Another user has used telnet to connect directly to the port before the cmfd daemon was able to capture the port.

  In this scenario, you can use the following command to force a logout:

  ```
  # sra ds_logout -node atlas2 -force yes
  ```

- **Scenario 2:** The terminal server that serves the console for atlas2 is not working.

  In this scenario, you must repair the terminal server.

In either case, the cmfd daemon will post an SC event similar to the following:

```
name: atlas2
class: cmfd
type: connection.failed or connection.refused
```

connection.refused indicates that the port is busy — that is, Scenario 1 above.

Check the cmfd log file (/var/sra/adm/log/cmfd/cmfd_*hostname_port*.log) for additional information.

29.8.2 CMF Daemon Reports connection.refused Event

This event is posted when the cmfd daemon receives an ECONNREFUSED signal when it attempts to connect to the terminal server port.

Check the cmfd log file (/var/sra/adm/log/cmfd/cmfd_*hostname_port*.log) for additional information.

You may see several such events, as the cmfd daemon continually attempts to connect to the terminal server port. Once a port is listed in the cmfd daemon's configuration, the cmfd daemon never stops trying to connect. However, it reduces the frequency of its connection attempts. You can control this frequency by using the sra edit command, as follows:

```
# sra edit
sra> sys
sys> edit system
Id        Description                                                Value
-------------------------------------------------------------------------
.
.
.
[15 ]   cmf reconnect wait time (seconds)                        60
[16 ]   cmf reconnect wait time (seconds) for failed ts          1800
.
.
.
-------------------------------------------------------------------------

Select attributes to edit, q to quit
eg. 1-5 10 15

edit? 15
cmf reconnect wait time (seconds)     [60]
new value? 3600

cmf reconnect wait time (seconds)     [300]
Correct? [y|n] y
sys> quit
sra> quit
```

You must then issue the following command, so that the updated value will take effect:

```
# /sbin/init.d/cmf update
```

Finally, you should investigate the problem, and take the appropriate action.

29.8.3 Console Output Wraps Strangely

Sometimes, when you press the Return key at a node's SRM prompt, you receive a linefeed only, instead of the expected carriage-return-and-linefeed pair. This causes the cursor on your terminal to move down to the next line, but not to the beginning of that line — similar to the following:

```
P00>>>
      P00>>>
            P00>>>
                  P00>>>
                        P00>>>
```

To fix this problem, you must issue a command similar to the following, where `atlas0` is the node whose SRM prompt is behaving in this way:

```
# sra ds_logout -nodes atlas0 -ts yes
```

Within a few minutes, the `cmfd` daemon will automatically reestablish the connection to the terminal server, and the standard carriage-return-and-linefeed behavior will be restored.

29.9 Domain Member Fails and Domain Loses Quorum

As long as a domain maintains quorum, you can use the `clu_quorum` command to adjust node votes and expected votes across the domain.

However, if a domain member loses quorum, all I/O is suspended and all network interfaces except the HP AlphaServer SC Interconnect interfaces are turned off.

Consider a domain that has lost one or more members due to hardware problems that prevent these members from being shut down and booted. Without these members, the domain has lost quorum, and its surviving members' expected votes and/or node votes settings are not realistic for the downsized domain. Having lost quorum, the domain hangs.

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

1. Shut down all members of the domain. Halt any unresponsive members as described in Section 29.5.4 on page 29–17.
2. Boot the first domain 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 1 (one) for the `cluster_expected_votes clubase` attribute.

   For example:

   ```
   P00>>>boot -fl ia
   (boot dka0.0.0.8.0 -flags ia)
   block 0 of dka0.0.0.8.0 is a valid boot block
   reading 19 blocks from dka0.0.0.8.0
   ```

```
bootstrap code read in
base = 6d4000, image_start = 0, image_bytes = 2600
initializing HWRPB at 2000
initializing page table at 7fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

UNIX boot - Wednesday August 01, 2001

Enter: <kernel_name> [option_1 ... option_n]
or: ls [name]['help'] or: 'quit' to return to console
Press Return to boot 'vmunix'
# vmunix clubase:cluster_expected_votes=1
```

3. Interactively boot all of the other nodes in the domain, as described in step 2.
4. Once the domain is up and stable, you can temporarily fix the configuration of votes in the domain until the broken hardware is repaired or replaced, by running the following command on the first domain member:

   # **clu_quorum -f -e *lower_expected_votes_value***

   This command lowers 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.

   Ignore the warnings about being unable to access the boot partitions of down members. The `clu_quorum -f` command will not be able to access a down member's `/etc/sysconfigtab` file; therefore, it will report an appropriate warning message. This happens because the down member's boot disk is on a bus private to that member.

   To resolve quorum problems involving a down member, boot that member interactively, setting `cluster_expected_votes` to a value that allows the member to join the domain. When it joins, use the `clu_quorum` command to correct vote settings as suggested in this section.

   Note:

   When editing member `sysconfigtab` files, remember that all members must specify the same number of expected votes, and that expected votes must be the total number of node votes in the domain.

   Finally, when changing the `cluster_expected_votes` attribute in the members' `/etc/sysconfigtab` files, you must make sure that:

   - The value is the same on each domain member and it reflects the total number of node votes supplied by each member.
   - The `cluster_expected_votes` attribute in the `/etc/sysconfigtab.cluster` domainwide file has the same value. If the value of the attribute in the `/etc/sysconfigtab.cluster` does not match that in the member-specific `sysconfigtab` files, the member-specific value may be erroneously changed upon the next shutdown and boot.

Eventually, once the hardware on the faulty node is fixed, boot the repaired node interactively using the amended expected votes value. When the node has booted, return the cluster quorum configuration to the original status using the `clu_quorum` command.

29.10 Kernel Crashes

The output from a kernel memory fault is similar to the following:

```
20/Oct/1999 04:48:01 trap: invalid memory read access from kernel mode
20/Oct/1999 04:48:01
20/Oct/1999 04:48:01 faulting virtual address:     0x00000085000000c0
20/Oct/1999 04:48:01 pc of faulting instruction:   0xffffffff007fd970
20/Oct/1999 04:48:01 ra contents at time of fault: 0xffffffff007fad78
20/Oct/1999 04:48:10 sp contents at time of fault: 0xfffffe054472f890
20/Oct/1999 04:48:10
20/Oct/1999 04:48:10 panic (cpu 1): kernel memory fault
```

If you experience a kernel crash, include the following information with your problem report:

- The `panic()` string
- The crash-data files

  If there is no crash-data file, send the output of the following commands:
  `#` **`kdbx -k /vmunix`** (or `/genvmunix`, whichever was booted at the time of the crash)
  `(kdbx)` ***`ra`*****`/i`**
  `(kdbx)` ***`pc`*****`/i`**

  where *`ra`* and *`pc`* are the values printed on the console
- The console logs for the crashed and any related system
- Data from the `vmzcore`/`vmunix` or the files themselves

If a system dumped to memory and not to disk, set BOOT_RESET to `off` at the console before booting up the machine again or the crash dump will be lost — this usually only happens if the machine crashed early in the boot sequence.

Note:

If the kernel was overwritten while the node was up and before it crashed, and there is no copy of the old kernel, the crash-data file will not be useful.

If the crash-data is incorrect, you can manually generate the proper crash-data file by executing the following command as the `root` user:

```
# crashdc propervmunix vmzcore.n > crash-data.new.n
```

29.11 Console Messages

Many messages are printed to console during both normal and abnormal operations. You can see console messages in a number of ways:

- Use `sra [console] -c` (or `-cl`) to connect to the console.
- Use `sra [console] -m` (or `-ml`) to monitor output from the console.
- Output from the console is written to the `/var/sra/cmf.dated/`*`date`*`/`*`nodename`*`.log` file.
- Messages can be written to the `/var/adm/messages` file.

This section describes a number of HP AlphaServer SC console messages.

Message Text:

```
18/Mar/2002 08:20:03 elan0: nodeid=29 level=5 numnodes=512
18/Mar/2002 08:20:03 elan0: waiting for network position to be found
18/Mar/2002 08:20:03 elan0: nodeid=29 level=5 numnodes=512
18/Mar/2002 08:20:03 elan0: network position found at nodeid 29
.
.
.
18/Mar/2002 08:20:04 elan0: New Nodeset [29]
.
.
.
18/Mar/2002 08:20:04 elan0: RESTART REQUEST from [28-28][30-31]
18/Mar/2002 08:20:04 elan0: RESTART REQUEST from [16-27]
18/Mar/2002 08:20:04 elan0: RESTART REQUEST from [0-15][32-63]
18/Mar/2002 08:20:04 elan0: RESTART REQUEST from [64-255]
.
.
.
18/Mar/2002 08:20:05 elan0: Online      [256-511]
18/Mar/2002 08:20:06 elan0: Online      [28-28][30-31]
18/Mar/2002 08:20:06 elan0: New Nodeset [28-31]
18/Mar/2002 08:20:06 elan0: Online      [16-27]
18/Mar/2002 08:20:06 elan0: New Nodeset [16-31]
.
.
.
18/Mar/2002 08:20:07 elan0: Online      [0-15][32-63]
18/Mar/2002 08:20:07 elan0: New Nodeset [0-63]
.
.
.
18/Mar/2002 08:20:08 elan0: Online      [64-255]
18/Mar/2002 08:20:08 elan0: New Nodeset [0-255]
```

Description:
These are informational messages from the Elan driver describing the nodes that it thinks are active (that is, that are connected to the network). This message is normal and is printed when nodes connect or disconnect from the network. The above example shows the output on Node 29 in a 256-node system, when Node 29 is booted.

Message Text:

```
18/Mar/2002 08:20:07 elan0: Online     [29]
18/Mar/2002 08:20:07 elan0: New Nodeset [0-255]
18/Mar/2002 08:20:07 ics_elan: seticsinfo: [elan node 29] <=> [ics node 30]
18/Mar/2002 08:20:15 CNX MGR: Join operation complete
18/Mar/2002 08:20:15 CNX MGR: membership configuration index: 34 (33 additions, 1
removals)
18/Mar/2002 08:20:15 CNX MGR: Node atlas29 30 incarn 0x45002 csid 0x2001e has been
added to the cluster
18/Mar/2002 08:20:15 kch: suspending activity
18/Mar/2002 08:20:18 dlm: suspending lock activity
18/Mar/2002 08:20:18 dlm: resuming lock activity
18/Mar/2002 08:20:18 kch: resuming activity
```

Description:

These are informational messages from the Elan driver describing the nodes that it thinks are active (that is, that are connected to the network). This message is normal and is printed when nodes connect or disconnect from the network. The above example shows the output on Node 3 in a 256-node system, when Node 29 is booted.

Message Text:

```
18/Mar/2002 08:18:28 kch: suspending activity
18/Mar/2002 08:18:28 dlm: suspending lock activity
18/Mar/2002 08:18:28 CNX MGR: Reconfig operation complete
18/Mar/2002 08:18:30 CNX MGR: membership configuration index: 33 (32 additions, 1
removals)
18/Mar/2002 08:18:30 ics_elan: llnodedown: ics node 30 going down
18/Mar/2002 08:18:30 CNX MGR: Node atlas29 30 incarn 0xbde0f csid 0x1001e has been
removed from the cluster
18/Mar/2002 08:18:30 CLSM Rebuild: starting...
18/Mar/2002 08:18:30 dlm: resuming lock activity
18/Mar/2002 08:18:30 kch: resuming activity
18/Mar/2002 08:18:34 clua: reconfiguring for member 30 down
18/Mar/2002 08:18:34 CLSM Rebuild: initiated
18/Mar/2002 08:18:34 CLSM Rebuild: completed
18/Mar/2002 08:18:34 CLSM Rebuild: done.
18/Mar/2002 08:18:39 elan0: Offline    [29]
18/Mar/2002 08:18:39 elan0: New Nodeset [0-28,30-255]
```

Description:

These are informational messages from the domain subsystems as they reconfigure for a node dropping out of a domain. These messages are normal. The above example shows the output on Node 3 in a 256-node system, when Node 29 has dropped out of the domain.

Message Text:

```
nodestatus: Warning: Can't connect to MSQL server on rmshost:
retrying ...
```

Description:

`nodestatus` is responsible for updating the `runlevel` field in the `nodes` table of the SC database. This error occurs when the `msql3d` daemon on the RMS master node (Node 0) is not running. You can restart `msql3d` on the RMS master node with the following command:

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

Message Text:

```
nodestatus: Error: can't force already running nodestatus (pid 3146589) to exit
```

Description:

This is an abnormal condition. If the message is repeating, the boot process is being held up. Connect to the console and enter Ctrl/C. This allows the boot process to continue. If this occurs more than once, run the following command:

```
# mv /sbin/init.d/nodestatus /sbin/init.d/nodestatus.disabled
```

Message Text:

```
elan0: stray interrupt
```

Description:

These messages are benign. The cause of the interrupt was handled by another kernel thread in the interim, leaving no work to be completed when the interrupt was eventually serviced.

29.12 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 HP AlphaServer SC uses CDSLs to maintain member-specific directories in a domainwide 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
```

29.13 Pressing Ctrl/C Does Not Stop scrun Command

As described in Section 12.4 on page 12–7, pressing Ctrl/C should stop the `scrun` command. If a node goes down while a command is running, pressing Ctrl/C twice may not stop the `scrun` command. You must press Ctrl/C a third time to disconnect `scrun` from its daemons.

29.14 Setting the HiPPI Tuning Parameters

If you fail to set the HiPPI tuning parameters, you may experience performance problems because of Direct Memory Access (DMA) restarts, or machine checks because of excessive DMA problems. A HiPPI card will retain its tuning parameters, even if moved to a new host.

For optimal performance over HiPPI, perform the following steps:

1. Check the HiPPI card parameters before tuning, as follows:

```
atlas1# esstune -p
Driver RunCode Tuning Parameters
        conRetryReg          (-c)   0x5000     20480
        conRetryTmrReg       (-t)    0x100       256
        conTmoutReg          (-o) 0x500000   5242880
        statTmrReg           (-s)  0xf4240   1000000
        intrTmrReg           (-i)     0x20        32
        txDataMvTimeoutReg   (-x) 0x500000   5242880
        rxDataMvTimeoutReg   (-r)  0x80000    524288
        pciStateReg          (-h)        0
           Minimum DMA 0x0,   DMA write max DMA 0x0,  DMA read max DMA 0x0
        dmaWriteState        (-w)     0x80     Threshold 8
        dmaReadState         (-d)     0x80     Threshold 8
        driverParam                      0   short fp network switched
```

2. Set the HiPPI tuning parameters, as follows:

```
atlas1# setld -v HIPRC222
HIPPI/PCI NIC Tuning (HIPRC222)
Turn off NIC hip0
Tuning NIC hip0
Turn on NIC hip0
```

3. Check the new parameters, as follows:

```
atlas1# esstune -p
Driver RunCode Tuning Parameters
        conRetryReg          (-c)   0x5000     20480
        conRetryTmrReg       (-t)    0x100       256
        conTmoutReg          (-o) 0x500000   5242880
        statTmrReg           (-s)  0xf4240   1000000
        intrTmrReg           (-i)     0x20        32
        txDataMvTimeoutReg   (-x) 0x989680  10000000
        rxDataMvTimeoutReg   (-r) 0x989680  10000000
        pciStateReg          (-h)     0xdc
           Minimum DMA 0x0,   DMA write max DMA 0x6,  DMA read max DMA 0x7
        dmaWriteState        (-w)     0x80     Threshold 8
        dmaReadState         (-d)     0x80     Threshold 8
        driverParam                      0   short fp network switched
```

29.15 SSH Conflicts with sra shutdown -domain Command

When Secure Shell (SSH) software is installed onto an HP AlphaServer SC system and its default configuration is modified so that it replaces the `r*` commands (that is, `rlogin`, `rsh`, and so on), the `sra shutdown -domain` *N* command ceases to work correctly, because a password is requested for every `rsh` connection made by the underlying `shutdown -ch` command.

(If SSH is installed on a system and its default settings are not modified, the above problem does not occur, as SSH deals only with incoming SSH connections and ignores all `r*` commands.)

To correct this problem, edit the `/etc/ssh2/ssh2_config` file as follows:

- Before:
  ```
  enforceSecureRUtils yes
  ```
- After:
  ```
  enforceSecureRUtils no
  ```

For more information about SSH, see Chapter 26.

29.16 FORTRAN: How to Produce Core Files

Stack traces and core files are needed to debug SEQV-type problems. The FORTRAN library does not produce these by default. To produce these files, set the `decfort_dump_flag` environment variable, as follows:

- If using C shell, run the following command:
  ```
  % setenv decfort_dump_flag y
  ```
- If not using C shell, run the following command:
  ```
  $ decfort_dump_flag=y; export decfort_dump_flag
  ```

29.17 Checking the Status of the SRA Daemon

The `srad_info` command provides simple diagnostic information about the current state of the SRA daemons.

The syntax of this command is as follows:

```
sra srad_info [-system yes|no] [-domains <domains>|all] [-width <width>]
```

where

- `-system` specifies whether to check the `System` daemon (the default is `-system yes`)
- `-domains` specifies which domain daemons to check (the default is `-domains all`)
- `-width` specifies the number of nodes to target in parallel (the default is `-width 32`)

The srad_info command displays the current state of each specified daemon, where the state is one of the following:

- Down
  The daemon is not running, or the srad_info command is unable to connect to the daemon.
- Idle
  The daemon is running, but the scheduler is not active.
- Running
  The daemon is running, and the scheduler is active.

  An additional information line displays the number of seconds since the daemon last recorded any scheduler activity.

29.18 Accessing the HP AlphaServer SC Interconnect Control Processor Directly

The HP AlphaServer SC Interconnect Control Processor may fail to respond to the telnet command, so that it appears that the card is non-functional. However, in this situation, the controller card is still responsive to jtest, and swmgr is still maintaining contact and gathering statistics. Should this situation arise, you can access the controller as follows from the management server:

```
atlasms# cd /usr/opt/qswdiags/bin
atlasms# ./jtest QR0N00
```

where QR0N00 is the name of the appropriate switch.

You will then be presented with the familiar jtest menu from which you can perform various actions, including rebooting the HP AlphaServer SC Interconnect Control Processor.

29.19 SC Monitor Fails to Detect or Monitor HSG80 RAID Arrays

If the SC Monitor system fails to detect or monitor an HSG80 RAID system, it is usually because the fabric connecting the node with the HSG80 RAID system is disconnected or because the HSG80 RAID system is powered off or faulty. However, the HSG80 might also be running a diagnostic or utility program.

You can determine the situation as follows:

1. Log into the node as the root user.
2. Use the hwmgr command to see whether the HSG80 RAID system is seen by the host. The following example shows that one or more HSG80s are visible to the system (dsk6c, dsk7c, scp0 are the devices).

   ```
   atlas0# hwmgr -v dev
   ```

```
HWID: Device Name          Mfg      Model            Location
-------------------------------------------------------------------------
   6: /dev/dmapi/dmapi
   7: /dev/scp_scsi
   8: /dev/kevm
  35: /dev/disk/floppy0c             3.5in floppy     fdi0-unit-0
  70: /dev/disk/dsk0c      COMPAQ    BD018635C4       bus-0-targ-0-lun-0
  75: /dev/disk/dsk5c      COMPAQ    BD018635C4       bus-0-targ-5-lun-0
  76: /dev/disk/dsk6c      DEC       HSG80            IDENTIFIER=1
  77: /dev/disk/dsk7c      DEC       HSG80            IDENTIFIER=2
  80: /dev/disk/cdrom0c    COMPAQ    CRD-8402B        bus-3-targ-0-lun-0
  86: /dev/disk/dsk15c     COMPAQ    BD018635C4       bus-5-targ-5-lun-0
  87: /dev/cport/scp0                HSG80CCL         bus-2-targ-0-lun-0
```

If no HSG80 devices are seen, there is a problem in the node's fibre adapter, the fibre fabric, or the HSG80 RAID system. In addition, configuration parameters (such as switch zoning or access paths) may prevent the node and the HSG80 from communicating with each other.

3. Use the `hsxterm5` program to connect to the devices. You should step through all of the HSG80 devices shown by `hwmgr`. Usually a single HSG80 is shown as several devices. You can determine which device belongs to which HSG80 using the `show this` command, as shown in the following example:

```
atlas0# /usr/lbin/hsxterm5 -F dsk6c "show this"
Controller:
        HSG80 ZG03200632 Software V85G-0, Hardware  E12
        NODE_ID          = 5000-1FE1-0009-5160
        ALLOCATION_CLASS = 0
        SCSI_VERSION     = SCSI-3
        .
        .
        .
```

4. If the `show this` command works, the SC Monitor system should be able to communicate with the HSG80 RAID system. You can trigger the SC Monitor to scan for HSG80 RAID systems as follows:

```
atlas0# /sbin/init.d/scmon reload
```

   If several users attempt to connect to the HSG80 RAID system, you can get an error similar to that shown in the following example:

```
atlas0# /usr/lbin/hsxterm5 -F dsk6c "show this"
ScsiExecCli Failed
```

   This can happen if the SC Monitor on another node is connected to the HSG80. You can repeat the command.

5. If the HSG80 RAID system is running a diagnostic or utility program, that program will not recognize the `show this` command, as shown in the following example:

```
atlas0# /usr/lbin/hsxterm5 -f dsk6c "show this"
          ^
Command syntax error at or near here
FMU>
```

6. The HSG80 RAID system is running the FMU utility. You can force the FMU to exit, as follows:
   ```
   atlas0# /usr/lbin/hsxterm5 -f dsk6c "exit"
   ```
7. If the HSG80 RAID system was running a diagnostic or utility, you should trigger the SC Monitor system to rescan the HSG80 as follows:
   ```
   atlas0# /sbin/init.d/scmon reload
   ```
8. Wait 10 minutes, and then check if new HSG80 RAID systems are detected using the following commands:
   ```
   atlas0# scevent -f '[class hsg] and [age < 20m]'
   atlas0# scmonmgr distribution -c
   ```

29.20 Changing the Kernel Communications Rail

WARNING:

This procedure is documented for emergency situations only, and should only be used under such special circumstances. The HP AlphaServer SC system should be restored to its original condition once the emergency has passed.

As a result of prudent PCI card placement, and suitable default configuration by SRA, rail usage in a multirail HP AlphaServer SC system is automatically configured for optimal performance.

Therefore, cluster/kernel communication will operate over a nominated rail (depending on the HP AlphaServer SC node type), and the second rail will be available for use by parallel applications.

If you need to temporarily boot a machine such that cluster communication takes place over a different rail, use one of the following commands:

- To boot off the first rail, run the following command:
  ```
  # sra boot -nodes all -file vmunix -iarg 'ep3:ep_cluster_rail=0'
  ```
- To boot off the second rail, run the following command:
  ```
  # sra boot -nodes all -file vmunix -iarg 'ep3:ep_cluster_rail=1'
  ```

29.21 SCFS/PFS File System Problems

The information in this section is organized as follows:

- Mount State for Domain Is Unknown (see Section 29.21.1 on page 29–41)
- PFS File System Cannot be Deleted if Underlying SCFS File System is Broken (see Section 29.21.2 on page 29–42)

- Mount State Is mounted-busy (see Section 29.21.3 on page 29–43)
- PFS Mount State Is mounted-partial (see Section 29.21.4 on page 29–44)
- Mount State Remains unknown After Reboot (see Section 29.21.5 on page 29–45)

29.21.1 Mount State for Domain Is Unknown

The `unknown` mount status is used whenever the `scmountd` daemon cannot communicate with the `srad` daemon on a domain. The `srad` daemon also runs file system management scripts. If the `scmountd` daemon cannot invoke a script because the `srad` daemon is not responding or because a script fails to complete normally, it marks all file systems on that domain as being in the `unknown` mount state.

The usual reason why the `scmountd` daemon cannot communicate with the `srad` daemon on a domain is because the domain is shut down. However, if the domain appears to be operating normally, you should perform the following steps to restore normal operation:

1. Run the `scfsmgr sync` command, and use the `scfsmgr status` command to determine when the synchronization has completed. If the mount status of the file systems remains `unknown`, go to step 2.
2. Use the `sra srad_info` command to determine whether the `srad` daemon is responding. If the daemon is not responding, use the `caa_stat` command to determine whether the `SC15srad` service is online. If the `SC15srad` service is not online, use the `caa_start` command to place it online. Repeat the `scfsmgr sync` command.
3. If the `srad` daemon is running, run the `scfsmgr status` command when the `scfsmgr sync` command has completed. If the `srad` daemon is not responding, the output is similar to the following:
   ```
   Domain: atlasD1 (1)  state: not-responding  command: state: idle name:
   (353) ; timer: not set
   ```
   Restart the `srad` daemon. If it remains unresponsive, contact your HP Customer Support representative.
4. If the `srad` daemon starts a script successfully, but the script fails to run normally, the output from the `scfsmgr status` command is similar to the following:
   ```
   Domain: atlasD1 (1)  state: timeout  command: state: idle name:  (353);
   timer: not set
   ```
   Search the `/var/sra/adm/log/scmountd/srad.log` and `/var/sra/adm/log/scmountd/fsmgrScripts.log` files for any errors that might account for the failure to complete. Report any errors in the `/var/sra/adm/log/scmountd/srad.log` file to your HP Customer Support representative.

29.21.2 PFS File System Cannot be Deleted if Underlying SCFS File System is Broken

A PFS file system cannot be deleted until all underlying component file systems are mounted by at least one domain. If one of the underlying SCFS file systems cannot be mounted (for example, because the underlying AdvFS domain has been deleted), the `pfsmgr delete` command cannot be used to delete the PFS.

The process described in this section can be used to delete the PFS manually. The example assumes the PFS set is called `pdata`, mounted at `/pdata` with components `/c1/a`, `/c1/b`, `/c2/c`, `/c2/b`, where `/c1` and `/c2` are SCFS file systems. The example assumes that `/c2` is the broken SCFS and that you cannot mount it.

1. Get the PFS set name for the `/pdata` mountpoint as follows:
```
# rmsquery "select pfs_set from sc_pfs where mountpoint='/pdata'"
pdata
#
```
2. Get the names of the PFS components as follows:
```
# rmsquery "select component_fs_path from sc_pfs_components where \
pfs_set='pdata'"
/c1/a
/c1/b
/c2/c
/c2/d
#
```
3. Place the underlying SCFS file systems online as follows:
```
# scfsmgr online /c1
```
4. When the file system is mounted, delete the contents of the component directories listed from step 2. In this example, this is `/c1/a` and `/c1/b`. You cannot delete `/c2/c` or `/c2/d`, as `/c2` is not mountable.
5. Delete records related to the PFS from the database as follows:
```
# rsmquery "delete from sc_pfs_filesystems where pfs_set='pdata'"
# rsmquery "delete from sc_pfs_components  where pfs_set='pdata'"
# rsmquery "delete from sc_pfs  where pfs_set='pdata'"
# rsmquery "delete from sc_pfs_mount  where pfs_set='pdata'"
```

The PFS file system is now deleted. Later, if you manage to recreate the broken SCFS filesystem, make sure that all component directories are empty before attempting to recreate the PFS file system.

29.21.3 Mount State Is mounted-busy

An SCFS or PFS file system will remain mounted even if it is offline (`mounted-busy`), for the following reasons:

- A PFS file system is being used by application programs — it cannot be unmounted until the applications stop using the file system.
- An online PFS file system is using an SCFS file system — the SCFS file system cannot be unmounted until the PFS file system unmounts.
- An SCFS file system is being used by application programs — it cannot be unmounted until the applications stop using the file system.
- Compute-Serving (CS) domains still have an SCFS file system mounted — the File-Serving (FS) domain will not unmount until all CS domains have completed their unmount operations.

Use the `fuser` command to find application programs that are using file systems. The `fuser` command does not show whether a PFS file system is using an SCFS file system — use the `scfsmgr show` command to show the PFS file systems that are based on a given SCFS file system. For more information about the `fuser` command, see the `fuser(8)` reference page.

When a file system fails to unmount, an event is posted. To retrieve such events for the last two days, run the following command:

```
# scevent -f '[age < 2d] and [type unmount.failed]'
08/06/02 11:52:02       atlasD2       scfs     unmount.failed      Cannot unmount
/pfs/pfs0/a: PFS may be mounted
08/07/02 11:33:15       atlas32       scfs     unmount.failed      Unmount /f1
failed: /f1: Device busy
08/07/02 11:34:08       atlasD0       scfs     unmount.failed      Cannot unmount
/f1: CS domain(s) have not unmounted
```

In this example output:

- The first event (Event 1) indicates that a PFS file system remains mounted — the component file system `/pfs/pfs0/a` cannot be unmounted.
- The second event (Event 2) indicates that an application is using an SCFS file system.
- The third event (Event 3) indicates that an FS domain cannot unmount an SCFS file system because a CS domain still has `/f1` mounted. Use the `scfsmgr show` command to see which CS domains are still mounting `/f1`. In addition, you could infer from Event 2 that `atlasD1` is the CS domain that still has `/f1` mounted.

29.21.4 PFS Mount State Is mounted-partial

PFS file systems must be mounted by each node. This is in contrast to other file systems, which are visible to all members of a domain once the file system is mounted by any member.

The `mounted-partial` mount status indicates that the PFS file system is mounted by some members of the domain, but other members of the domain have failed to mount the PFS file system. A shut-down member is not considered to have failed to mount — the `mounted-partial` mount status is only used for members that are operating normally but have failed to mount.

To see why the mount failed, review the PFS events for the period in which the mount attempt was made. For example, if the `pfsmgr online` (or `scfsmgr sync`) command had been run within the last ten minutes, the following command will retrieve appropriate events:

```
# scevent -f '[age < 10m] and [class pfs] and [severity ge warning]'
08/13/02 17:24:34        atlasD7          pfs        mount.failed          Mount of
/pfs/pfs0 failed on atlas[226-227]
08/13/02 17:24:34        atlasD7          pfs        script.error          scrun
failed: atlas[226-227]
```

In this example output:

- The first event indicates that the mount of `/pfs/pfs0` failed on `atlas[226-227]`.
- The second event explains that the `scrun` command failed. This is the reason why the mount failed: the file-system management system was unable to use the `scrun` command to dispatch the mount request to all members of the domain.

Correct the `scrun` problem (try to restart the `gxclusterd`, `gxmgmtd`, and `gxnoded` daemons) and then use the `scfsmgr sync` command to trigger another attempt to mount the `/pfs/pfs0` file system.

If the `scrun` command is not responsible for the failure, you must examine the `atlasD7` log files. For example, run the following command to retrieve the PFS log file for `atlas226`:

```
# scrun -n atlas226 tail -n 1 /var/sra/adm/log/scmountd/pfsmgr.atlas226.log
atlasD7: Wed Aug 13 17:27:34 IST 2002: mount_pfs.tcl
/usr/sbin/mount_pfs /comp/pfs0/a /pfs/pfs0:
File system atlasD0:/comp/pfs0/a has invalid
protection 0777: 0700 expected
```

In this example, the component file system has an invalid protection mask. Since the `pfsmgr create` command sets the file-system protection to 700, someone must have changed the protection of the file system (not the protection on the mount point, but the protection of the file system mounted on the mount point). Correct the protections and use the `scfsmgr sync` command to trigger another attempt to mount the PFS.

29.21.5 Mount State Remains unknown After Reboot

After a reboot of the complete HP AlphaServer SC system, the mount states of file systems may remain `unknown`. To resolve this, run the `scfsmgr sync` command as follows:

```
# scfsmgr sync
```

29.21.6 SCFS File Systems Not Mounted by Newly Created FS Domain

If there are problems with exporting file systems from an FS domain to other domains, it may be because the FS domain is missing from the SCFS_SRV_DOMS variable in the `/etc/rc.config.common` file on the management server or on a domain. The most common reason for this is that a CS domain was converted to an FS domain, but the new FS domain was not added to the SCFS_SRV_DOMS variable. In this case, commands such as `scfsmgr sync` and `scfsmgr online` will fail to trigger the correct response.

To determine whether the FS domain is missing from the SCFS_SRV_DOMS variable on the management server (if present), run the command shown in the following example:

```
atlasms# rcmgr get SCFS_SRV_DOMS
```

In the above example, `atlasD0` is the only FS domain in the SCFS_SRV_DOMS variable on the management sever. To add the missing FS domain (for example, `atlasD1`) on the management server, run the command shown in the following example:

```
atlasms# rcmgr -c set SCFS_SRV_DOMS atlasD0, atlasD1
```

To determine whether an FS domain is missing from the SCFS_SRV_DOMS variable on any of the domains, run the command shown in the following example (where `atlas` is an example system name) on the management server:

```
atlasms# scrun -d all rcmgr get SCFS_SRV_DOMS
atlasD0: atlasD0
atlasD1: atlasD0
atlasD2: atlasD0
atlasD3: atlasD0
.
.
.
```

In the above example, `atlasD0` is the only FS domain in the SCFS_SRV_DOMS variable on each domain. To add a missing FS domain (for example, `atlasD1`) on all domains, run the command shown in the following example:

```
atlasms# scrun -d all rcmgr -c set SCFS_SRV_DOMS atlasD0, atlasD1
```

When the FS domain has been added to the SCFS_SRV_DOMS variable on the management server and on all domains, you must restart the `scmountd` daemon on the management server (if present) or on Node 0, as shown in the following example:

```
atlasms# /sbin/init.d/scmountd restart
scmountd: Stopping the daemon
scmountd: Starting the daemon
```

Run the scfsmgr scan command on any node to find the disks and file systems on the new domain. The scfsmgr scan command scans all domains, and may take a long time to run on a large system.

29.21.7 SCFS: scfsmgr May Fail to Create an Entry for a Disk

The scfsmgr command may fail to create an entry for a disk that already exists in the sc_advfs_vols table.

For example:

```
# scfsmgr create test_dom_2#test_fset_2 /test_fset_2 atlasD0 rw atlas1 FAST \
root system 755 /dev/disk/dsk10c
File domain, file set and mount point created
scfs record added to sc_scfs
/etc/exports updated
Error updating sc_advfs_vols DB: libscdb.so: Could not insert duplicate record:
Non unique value for unique index
```

The workaround is to remove the entries from the sc_advfs_vols and sc_advfs_filesets tables, as follows:

1. Run the following rmsquery command on the sc_advfs_vols table:

   ```
   atlas0 # rmsquery -v "select * from sc_advfs_vols where name='dsk10c'"
   clu_domain  name    advfs_domain    type
   ------------------------------------------
   atlasD0    dsk10c  test_dom_1      DISK
   ```

   dsk10c already exists in this table and therefore needs to be removed.

2. Remove the entry from the sc_advfs_vols table, as follows:

   ```
   atlas0 # rmsquery -v "delete from sc_advfs_vols where name='dsk10c'"
   ```

3. Run the following rmsquery on the sc_advfs_filesets table:

   ```
   atlas0 # rmsquery -v "select * from sc_advfs_filesets where \
   advfs_domain='test_dom_1'"
   clu_domain  advfs_domain    fset_name
   ------------------------------------------
   atlasD0    test_dom_1      test_fset_1
   ```

   The entry for test_dom_1 needs to be removed from this table.

4. Remove the entry from the sc_advfs_vols table, as follows:

   ```
   atlas0 # rmsquery -v "delete from sc_advfs_filesets where \
   advfs_domain='test_dom_1'"
   ```

29.22 System Check Tool

HP provides a system check tool, `sys_check`, that gathers configuration data and analyzes information for use in debugging and diagnosing system problems. This section describes how to install and run this tool, and is organized as follows:

- System Check Tool Overview (see Section 29.22.1 on page 29–47)
- Installing the System Check Tool (see Section 29.22.2 on page 29–48)
- Running the System Check Tool (see Section 29.22.3 on page 29–48)

29.22.1 System Check Tool Overview

The `sys_check` tool creates a configuration report that is stored in an HTML file that can easily be browsed and transported. When invoked with the `-escalate` option, the `sys_check` tool also collects system log and crash files and places them in a `.tar` archive that you can send to your HP Customer Support representative to assist in diagnosing system problems.

Note:

The `sys_check` tool does not change any system parameters or configurations.

The task of creating the configuration report can take up to several hours on an HP AlphaServer SC system. To address this problem, `sys_check` components can now be enabled or disabled during a `sys_check` run. This feature allows you to bypass the creation of the configuration report by disabling all `sys_check` components or to create a sparse configuration report by enabling just a few `sys_check` components.

When the `sys_check` command is used with only the `sclogs` component, all that happens is that the system log and crash files (including the `sra` and `rms` log files) are gathered and placed in a `.tar` archive. This task takes only a few minutes. For more information on using the `sclogs` plug-in component with the `sys_check` tool, see Section 29.22.3 on page 29–48.

The `sys_check` tool runs on all supported versions of the Tru64 UNIX operating system.

Although Version 126 of the `sys_check` tool is included in the base Tru64 UNIX Version 5.1B operating system, newer versions of the tool have been released and Version 126 is no longer supported. Therefore, you must download the latest version of the `sys_check` tool from the `http://h30097.www3.hp.com/sys_check/` Website, as described in Section 29.22 on page 29–47.

As the `sys_check` tool is updated frequently, you should also check the Website regularly for newer versions.

29.22.2 Installing the System Check Tool

To download and install the sys_check kit and the sclogs script, perform the following steps:

1. Log in as root and access the sys_check Website at:
   http://h30097.www3.hp.com/sys_check/
2. Download the kit by clicking on the Download tab, then clicking on the Download sys_check V*nnn* kit link (where *nnn* is the most recent version number).
3. When the kit has finished downloading, unpack the kit by running the following command:
   # tar xvf sys_check.tar
4. You must install the kit on each domain in your HP AlphaServer SC system; when you install the kit on one domain member, the kit is installed on all members of that domain. Run the following command on the first member of each domain:
   # setld -l SYSCHECK132.kit
5. You now need to download the sclogs plug-in file. Access the sys_check Website again, and click on the Download tab.
6. If necessary, scroll down the page to find the Download sclogs plug-in link. Click on this link, and save the sclogs file (without an extension) in the /usr/lib/sys_check_isv directory.
7. Copy the sclogs plug-in file to the /usr/lib/sys_check_isv directory of the first node in each domain. This directory is shared by all members of the domain.

29.22.3 Running the System Check Tool

To collect system log and crash files (including the sra and rms log files) without producing a configuration report, run the sys_check -escalate command with only the sclogs component as follows:

 # sys_check -escalate sclogs

Note:

In most cases, it is sufficient to run the system check with only the sclogs component, as described here. This task only takes a few minutes.

Running the entire system check (that is, running the sys_check command without specifying any components, which implies that all components are enabled) may take a number of hours on an HP AlphaServer SC system. See the sys_check(8) reference page for more information on enabling and disabling individual sys_check components.

The log and crash files are stored in the following files:

- `escalate.tar` — contains all the log files
- `escalate_vmunix.xx.gz` — contains the `vmunix` crash file, if it exists
- `escalate_vmcore.xx.gz` — contains the `vmcore` crash file, if it exists

You will need to supply these three files to your HP Customer Support representative when reporting a problem.

For more information on using the sys_check tool, see the `sys_check(8)` reference page.

Part 4:

Appendixes

A

Cluster Events

Cluster events are Event Manager (EVM) events that are posted on behalf of the domain, 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 file system (CFS) to indicate that a
   filesystem has been mounted in the cluster, or that a file system for which
   this node is the server has been relocated or failed over.
```

For a description of EVM priorities, see the `EvmEvent(5)` reference page. For more information on event management, see the `EVM(5)` reference page.

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 shutting down and booting each member individually.

For more information about `rc.config`, see Section 21.1 on page 21–2.

Table B–1 Cluster Configuration Variables

Variable	Description
ALIASD_NONIFF	Specifies which network interfaces should not be configured for NIFF monitoring. HP AlphaServer SC Version 2.6 disables NIFF monitoring on the `eip0` interface, by default.
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 on which the HP AlphaServer SC software is based.
CLUSTER_NET	Specifies the name of the system's primary network interface.
SC_CLUSTER	Specifies that this is an HP AlphaServer SC cluster.
SC_MOUNT_OPTIONS	Specifies the options used (default `-o server_only`) when mounting local file systems (`tmp` and `local`).
SC_MS	Specifies the name of the management server (if used) or Node 0 (if not using a management server).
SC_USE_ALT_BOOT	Set if the alternate boot disk is in use.
SCFS_SRV_DOMS	Lists the SCFS File-Server domains.
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

SC Daemons

This appendix lists the daemons that run in an HP AlphaServer SC system, and the daemons that are not supported in an HP AlphaServer SC system.

The information in this appendix is organized as follows:

- HP AlphaServer SC Daemons
- LSF Daemons
- RMS Daemons
- Domain Daemons
- Tru64 UNIX Daemons
- Daemons Not Supported in an HP AlphaServer SC System

C.1 HP AlphaServer SC Daemons

Table C–1 lists the HP AlphaServer SC daemons.

Table C–1 HP AlphaServer SC Daemons

Name	Description
`cmfd`	The console logger daemon
`gxclusterd`	This `scrun` daemon is the domain daemon. There are two copies running on each domain, one each on member 1 and member 2.
`gxmgmtd`	This `scrun` daemon is the management daemon. The daemon runs on the management server. If a clustered management server is used, there is a copy of the daemon on each member of the cluster, but the daemon is active on only one member.
`gxnoded`	This `scrun` daemon is the node daemon. There is one copy of this daemon on each node.
`rbinlogd`	Remote binary event-log daemon
`scalertd`	HP AlphaServer SC event monitoring daemon
`scmond`	HP AlphaServer SC hardware monitoring daemon
`scmountd`	This daemon manages the SCFS and PFS file systems. This daemon runs on the management server (if any) or on Node 0 (if no management server is used).
`srad`	HP AlphaServer SC install daemon

C.2 LSF Daemons

Table C–2 lists the LSF daemons.

Table C–2 LSF Daemons

Name	Description
`elim`	External Load Information Manager daemon
`lim`	Load Information Manager daemon
`mbatchd`	Master Batch daemon
`sbatchd`	Slave Batch daemon
`topd`	Topology daemon

C.3 RMS Daemons

Table C–3 lists the RMS daemons.

Table C–3 RMS Daemons

Name	Description
`eventmgr`	RMS event manager daemon
`mmanager`	RMS machine manager daemon
`msql3d`	RMS daemon
`pmanager`	RMS partition manager daemon
`rmsd`	Loads and schedules the processes that constitute a job's processes on a particular node
`rmsmhd`	Monitors the status of the `rmsd` daemon
`swmgr`	HP AlphaServer SC Interconnect switch manager
`tlogmgr`	RMS transaction logger

C.4 Domain Daemons

Table C–4 lists the domain daemons.

Table C–4 Domain Daemons

Name	Description
`aliasd`	Cluster alias daemon, runs on each domain member except on members that do not have external interfaces and are members of a CS domain. Used to create a member-specific `/etc/gated.conf.member`*N* configuration file, and to start the `gated` daemon. Supports only the Routing Information Protocol (RIP).
`caad`	The CAA daemon
`cfsd`	The load monitoring daemon that can monitor, report on, and respond to file-system-related member and domain activity
`clu_wall`	Runs on each domain member to receive `wall -c` messages
`gated`	The gateway routing daemon
`niffd`	Monitors the network interfaces in the domain

C.5 Tru64 UNIX Daemons

Table C–5 lists the Tru64 UNIX daemons.

Table C–5 Tru64 UNIX Daemons

Name	Description
auditd	An audit daemon, runs on each domain member
autofsd	The AutoFS or automount daemon
binlogd	Binary event-log daemon
cron	The system clock daemon
desta	System Event Analyzer daemon
evmchmgr	Event Manager channel manager
evmd	Event Manager daemon
evmlogger	Event Manager logger daemon
inetd	The Internet server daemon
joind	BOOTP and DHCP server daemon
kloadsrv	The kernel load server daemon
mountd	The mount daemon
named	Internet Domain Name Server (DNS) or Berkeley Internet Name Daemon (BIND)
nfsd	NFS daemons
nfsiod	The local NFS-compatible asynchronous I/O daemon
pmgrd	The Performance Manager metrics server daemon
portmap	Maps DARPA ports to RPC program numbers
rlogind	The remote login server
rpc.lockd	NFS lock manager daemon
rpc.statd	NFS lock status monitoring daemon
rpc.yppasswdd	NIS single-instance daemon
sendmail	Internet mail daemon
syslogd	System message logger daemon

Table C–5 Tru64 UNIX Daemons

Name	Description
`xntpd`	The NTP daemon
`ypbind`	NIS server process
`ypserv`	NIS binder process
`ypxfrd`	NIS multi-instance daemon, active on all members

C.6 Daemons Not Supported in an HP AlphaServer SC System

The following daemons are not supported in an HP AlphaServer SC system:

- `ogated`
- `routed`
- `rwhod`
- The DHCP server daemon is not supported except for use by RIS.
- Do not use the `timed` daemon to synchronize the time.
- The `cfsd` daemon can be used in CFS file systems, but cannot be used in SCFS file systems.

D

Example Output

This appendix is organized as follows:

- Sample Output from sra delete_member Command (see Section D.1 on page D–1)
- Sample clu_delete_member Log File (see Section D.2 on page D–2)

D.1 Sample Output from sra delete_member Command

Each time you run `sra delete_member`, it displays output on screen and writes log messages to the `/cluster/admin/clu_delete_member.log` file.

Example D–1 shows sample screen output for the `sra delete_member` command.

Example D–1 sra delete_member Output

```
atlasms# sra delete_member -nodes atlas186

This command will remove specified nodes from a cluster
Note: Any member specific files will be lost

Confirm delete_member for nodes atlas186 [yes]:
11:59:47  Command   53 (del_member) atlasD5 : atlas186 -- <Unallocated>
11:59:57  Command   53 (del_member) atlasD5 : atlas186 -- <Allocated>
11:59:57     Node                           atlas186 -- <member_delete> Working
12:01:17  Command   53 (del_member) atlasD5 : atlas186 -- <Success>
12:01:17     Node                           atlas186 -- <Complete:member_delete>
Finished


Command has finished:

        Command   53 (del_member) atlasD5 : atlas186 -- <Success>

*** Node States *** Completed: atlas186

atlasms#
```

D.2 Sample clu_delete_member Log File

Example D–2 is a sample `clu_delete_member` log file.

Example D–2 clu_delete_member Log File

```
clu_delete_member on 'atlas160' begin logging at Tue Aug  6 11:59:50 EDT 2002
------------------------------------------------------------------------
ls: /etc/fdmns/root27_domain not found

Deleting member disk boot partition files
root27_domain#root on /cluster/admin/tmp/boot_partition.543002: No such domain,
fileset or mount directory

Warning: clu_delete_member: Cannot remove boot files from member : /cluster/admin/
tmp/boot_partition.543002

Warning: clu_delete_member: Cannot remove Domain: root27_domain

Removing deleted member entries from shared configuration files
  Removing cluster interconnect interface 'atlas186-ics0' from /.rhosts
  Removing cluster member hostname atlas186 from /etc/cfgmgr.auth.
  Configuring Network Time Protocol for new member
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas160'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas161'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas162'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas163'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas164'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas165'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas166'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas167'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas168'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas169'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas170'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas171'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas172'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas173'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas174'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas175'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas176'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas177'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas178'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas179'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas180'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas181'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas182'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas183'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas184'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas185'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas187'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas188'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas189'
```

```
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas190'
  Deleting interface 'atlas186-ics0' as an NTP peer to member 'atlas191'
usage: hostid [hexnum or internet address]
Clusterizing mail...
Saving original /var/adm/sendmail/sendmail.cf as /var/adm/sendmail/
sendmail.cf.cluster.sav
Saving original /var/adm/sendmail/atlasD5.m4 as /var/adm/sendmail/
atlasD5.m4.cluster.sav

Restarting sendmail on cluster member atlas160...

Restarting sendmail on cluster member atlas161...
SMTP Mail Service started.
Permission denied.

Restarting sendmail on cluster member atlas162...
Permission denied.

Restarting sendmail on cluster member atlas163...

Restarting sendmail on cluster member atlas164...
Permission denied.
Permission denied.

Restarting sendmail on cluster member atlas165...
Permission denied.

Restarting sendmail on cluster member atlas166...

Restarting sendmail on cluster member atlas167...
Permission denied.
Permission denied.

Restarting sendmail on cluster member atlas168...
Permission denied.

Restarting sendmail on cluster member atlas169...
Permission denied.

Restarting sendmail on cluster member atlas170...
Permission denied.

Restarting sendmail on cluster member atlas171...
Permission denied.

Restarting sendmail on cluster member atlas172...
Permission denied.

Restarting sendmail on cluster member atlas173...
Permission denied.

Restarting sendmail on cluster member atlas174...
Permission denied.
```

Sample clu_delete_member Log File

```
Restarting sendmail on cluster member atlas175...
Permission denied.

Restarting sendmail on cluster member atlas176...

Restarting sendmail on cluster member atlas177...
Permission denied.
Permission denied.

Restarting sendmail on cluster member atlas178...
Permission denied.

Restarting sendmail on cluster member atlas179...
Permission denied.

Restarting sendmail on cluster member atlas180...
Permission denied.

Restarting sendmail on cluster member atlas181...
Permission denied.

Restarting sendmail on cluster member atlas182...
Permission denied.

Restarting sendmail on cluster member atlas183...
Permission denied.

Restarting sendmail on cluster member atlas184...
Permission denied.

Restarting sendmail on cluster member atlas185...
Permission denied.

Restarting sendmail on cluster member atlas187...
Permission denied.

Restarting sendmail on cluster member atlas188...
Permission denied.

Restarting sendmail on cluster member atlas189...
Permission denied.

Restarting sendmail on cluster member atlas190...
Permission denied.

Restarting sendmail on cluster member atlas191...
Changes to mail configuration complete
Permission denied.

Deleting Member Specific Directories
  Deleting: /cluster/members/member27/
  Deleting: /usr/cluster/members/member27/
```

```
  Deleting: /var/cluster/members/member27/

Initial cluster deletion successful, member '27' can no longer jointhe cluster.
Deletion continuing with cleanup.

Warning: clu_delete_member was unable to determine the number of
votes contributed to the cluster by this member and will NOT automatically
adjusted expected votes. You may run the clu_quorum command to manually
adjust expected votes after the member deletion completes. See the
TruCluster Cluster Administration manual for more information on deleting
a cluster member.

clu_delete_member: The deletion of cluster member '27' completed successfully.
------------------------------------------------------------------------
clu_delete_member on 'Tue Aug  6 12:01:13 EDT 2002' end logging at atlas160
```

Index

A

Abbreviations, xxxviii

Accounting Services, 21–43

ACLs (Access Control Lists), 7–27

Active/Standby Mode, 6–13

AdvFS (Advanced File System), 24–37

Application Hangs, 29–14

authcap Database, 26–3

B

Backing Up Files, 24–45

Berkeley Internet Name Domain
    See DNS/BIND

BIND
    See DNS/BIND

binex+ utility, 28–16

Boot Disks, 29–12
    Alternate Boot Disk, 2–8, 2–9
    Backup Boot Disk, 2–12
    Creating, 2–14
    Managing, 2–7

BOOT_RESET Console Variable, 2–5

Booting
    See Domain Members, Booting

C

CAA
    Balancing Application Resources, 23–15
    caad, 23–26
    Checking Resource Status, 23–3
    Considerations for Startup and Shutdown, 23–25
    Managing the CAA Daemon (caad), 23–26
    Managing with SysMan Menu, 23–21
    Network, Tape, and Media Changer Resources, 23–20
    Registering and Unregistering Resources, 23–18
    Relocating Applications, 23–11
    Reporting Resource Availability Measurements, 23–9
    Starting and Stopping Application Resources, 23–13
    Troubleshooting, 23–29
    Using EVM to View CAA Events, 23–27

CAA Failover Capability
    CMF, 14–17
    RMS, 5–74

CDFS File Systems, 24–47

CD-ROM, 24–15

CDSL (Context-Dependent Symbolic Link)
    Creating, 24–5
    Exporting and Mounting, 24–6
    Kernel Builds, 24–6
    Maintaining, 24–6
    Overview, 24–4

CFS (Cluster File System)
    Block Devices, 24–36
    Cache Coherency, 24–36
    Direct I/O, 24–27
    Mounting CFS File Systems, 24–15
    Optimizing, 24–24
    Overview, 1–21, 24–1

Partitioning File Systems, 24–34
Cluster
See Domain
Cluster Alias
Changing IP Address, 19–18
Changing IP Name, 19–15
Cluster Alias and NFS, 19–19
Cluster Application Availability, 19–20
Configuration Files, 19–5
Default Cluster Alias, 19–2
Deleting, 19–13
Features, 19–2
Leaving, 19–12
Modifying, 19–11
Modifying Domainwide Port Space, 19–14
Monitoring, 19–12
Planning, 19–6
Properties, 19–2
Specifying and Joining, 19–9
Troubleshooting, 29–24
Cluster Aliases
Preferred Server, 19–23
Cluster File System
See CFS
Cluster Member
See Domain Member
Cluster Quorum, 20–5
CMF
See Console Logger
Code Examples, xliv
Commands
addvol, 24–38
clu_quorum, 20–10
cluamgr, 19–2
cmf, 14–13
pfsmgr, 8–13
rcmgr, 21–2
ris, 26–2
rmvol, 24–38
scalertmgr, 9–13
scca_setup, 28–7
scevent, 9–9
scfsmgr, 7–8
scload, 11–1
scmonmgr, 27–16
scpvis, 11–1
scrun, 12–1
scviewer, 9–9, 10–2
setld, 26–2
sra, 16–1
sra dbcheck, 3–13
sra diag, 28–11
sra edit, 16–22
sra-display, 16–39
SSH (Secure Shell), 26–10
sysconfig, 26–2
Configuration Variables, 21–2, B–1
Connection Manager
Monitoring, 20–12
Overview, 20–2
Panics, 20–13
Console Logger, 14–2
Backing Up or Deleting Console Log Files, 14–15
CAA Failover Capability, 14–17
Changing CMF Host, 14–21
Changing CMF Port Number, 14–17
Configurable CMF Information, 14–4
Configuration and Output Files, 14–5
Log Files, 14–8, 15–4
Starting and Stopping, 14–13
Troubleshooting, 29–28
Console Messages, 29–33
Console Network, 1–12, 1–15, 14–2
See also Console Logger
Context-Dependent Symbolic Link
See CDSL
Cookies, 3–12
Core Files, FORTRAN, 29–37
Crash Dumps, 15–4, 29–17
Critical Voting Member, 2–17
CS Domain, 1–14

D

Daemons
Domain, C–3
HP AlphaServer SC, C–2
LSF, C–2
Not Supported, C–5
RMS, C–3
SSH (Secure Shell), 26–11

System Event Analyzer (desta), 28–14, 28–19
Tru64 UNIX, C–4
Database
See SC Database
Default Route
Configuring, 22–2
Device Request Dispatcher (DRD), 1–23
Devices
Device Request Dispatcher Utility (drdmgr), 24–8
Device Special File Management Utility (dsfmgr), 24–8
Hardware Management Utility (hwmgr), 24–8
Managing, 24–7
Overview, 1–26
DHCP (Dynamic Host Configuration Protocol), 17–6
Diagnostics
See HP AlphaServer SC Node Diagnostics
Diskettes, 24–15
DNS/BIND, 17–6, 22–8
Documentation
Conventions, xlii
Online, xliv
Domain
Command Differences, 17–4
Commands and Utilities, 17–2
Configuration Tools, 18–3
Converting a CS Domain into an FS Domain, 21–12
Daemons, C–3
Events, A–1
Managing Multiple Domains, 12–1
Moving to Another IP Subnet, 21–18
Overview, 1–13
Recovering Cluster Root File System, 29–6
Domain Members
Adding After Installation, 21–6
Adding Deleted Member Back into Domain, 21–17
Booting, 2–1, 29–3
Connecting to, 14–15
Deleting, 21–15, D–1
Halting, 2–19
Monitoring Console Output, 14–16
Not Bootable, 2–6
Powering Off or On, 2–19
Rebooting, 2–6
Reinstalling, 21–17
Resetting, 2–19
Shutting Down, 2–15
Single-User Mode, 2–5
drdmgr, 24–8
dsfmgr, 24–8
DVD-ROM, 24–15

E

edauth Command, 26–3
Elan Adapter, 1–12
Ethernet Card, Changing, 21–36
Event Management, 17–9
Events
Categories, 9–3
Classes, 9–3
Cluster, A–1
Domain, A–1
Event Handler Scripts, 9–18
Event Handlers, 9–16
Examples, 9–10
Filter Syntax, 9–6
Log Files, 15–4
Notification of, 9–13
Overview, 9–2
SC Monitor, 27–4
Severities, 9–6
Viewing, 9–9
Examples
Code, xliv
External Network, 1–18
External Storage
See Storage, Global

F

FAST Mode, 6–4
Fibre Channel
See Storage, System
File Servers, Locating and Migrating, 24–24
File System Overview, 6–1

Floppy Disks
See Diskettes
FS Domain, 1–14, 7–2
fstab File, 13–3, 22–12, 24–16

G

gated Daemon, 19–8
Graphics Consoles, 1–13

H

HiPPI, Setting Tuning Parameters, 29–36
HP AlphaServer SC Interconnect, 1–12, 1–16
HP AlphaServer SC Node Diagnostics
Checking Status of System Event Analyzer Processes, 28–19
Installing System Event Analyzer, 28–3
Managing Log File, 28–16
Overview, 28–1
Removing System Event Analyzer, 28–19
Stopping System Event Analyzer, 28–19
System Event Analyzer Command Line Interface, 28–14
System Event Analyzer Overview, 28–2
System Event Analyzer Web User Interface, 28–14
HP AlphaServer SC Nodes, 1–12
Crashed, 29–17
See also HP AlphaServer SC Node Diagnostics
HP AlphaServer SC System Components, 1–3
hwmgr, 24–8

I

inetd Configuration, 22–25
Internal Storage
See Storage, Local
Ioctl
See PFS
IP Addresses, Table Of, 1–10

K

Kernel
Attributes, 21–3
Troubleshooting, 29–32, 29–40
Updating After Domain Creation, 21–36
Korn Shell, 29–35
KVM Switch, 1–13

L

Layered Applications, 21–42
License
Booting Without, 29–3
Managers, 19–29
Managing, 21–34
Load Sharing Facility
See LSF
Local Disks, 1–15
Log Files, 5–73, 14–8, 15–6, 28–16
LSF (Load Sharing Facility), 4–1
Checking the Configuration, 4–9
Commands, 4–4
Compatibility with Earlier Releases, 4–3
Configuration Notes, 4–10
Customizing Job Control Actions, 4–9
Daemons, C–2
Default Queue, 4–11
DEFAULT_EXTSCHED Parameter, 4–14
Directory Structure, 4–2
Installing, 4–2
Job Slot Limit, 4–10
Job Termination Reasons, 4–20
Known Problems or Limitations, 4–29
Licensing, 4–18
Log Files, 15–3
lsf.conf File, 4–26
LSF/HPC Extensions, 4–24
Maintaining a Pool of Nodes in Reserve, 4–19
MANDATORY_EXTSCHED Parameter, 4–15
Operating, 4–16
Remote LSF Adapter (RLA), 4–17
Running Jobs on Any Host Type, 4–17
Running Pre-Execution Programs, 4–19
Setting Dedicated LSF Partitions, 4–9
Setting Up Virtual Hosts, 4–5

Shutting Down the LSF Daemons, 4–7
Starting the LSF Daemons, 4–5
Topology Scheduler, 4–11
User Information for Interactive Batch Jobs, 4–24
User-Level Checkpointing Limitations, 4–26
Using NFS to Share Configuration Information, 4–4

LSM (Logical Storage Management)
Configuring for a Domain, 25–4
Creating an LSM Volume with a Single Striped Plex, 25–4
Overview, 25–2
Storage Connectivity, 25–3
Troubleshooting, 29–19

M

Mail, 17–6, 22–22

Management Network, 1–12, 1–16

Management Server, 1–18

member_fstab File, 13–3, 22–12, 24–16

Multiple-Bus Failover Mode, 6–15

N

Network
Changing Ethernet Card, 21–36
Configuring, 22–7
Console, 1–12
Default Route Configuration, 22–2
External, 1–18
HP AlphaServer SC Interconnect, 1–12
IP Routers, 22–6
Troubleshooting, 29–19

Network Adapters, Supported, xliv

NFS (Network File System)
Configuring, 22–11
Troubleshooting, 29–23

NIFF (Network Interface Failure Finder), 17–7

NIS (Network Information Service), 22–21

Node Types, Supported, xliv

NTP (Network Time Protocol), 22–9

P

Panics, 20–13, 29–12

Parallel File System
See PFS

Performance Visualizer
See SC Performance Visualizer

PFS (Parallel File System), 1–25
Attributes, 8–3
Backing up, 8–12
Checking, 8–11
Creating, 8–7
Exporting, 8–12
Increasing the Capacity of, 8–10
Ioctl Calls, 8–25
Log Files, 15–7
Managing, 8–7, 8–21
Mounting, 8–7
Optimizing, 8–23
Overview, 6–5, 8–2
pfsmgr Command, 8–13
Planning, 8–5
SC Database Tables, 8–29
Storage Capacity, 8–4
Structure, 8–4
Troubleshooting, 29–40
Upgrading, 8–11
Using, 8–22

Preferred Server
Cluster Aliases, 19–23

Printing, 17–7

Processor Sets, 5–60

pstartup Script, 5–74

Q

Quotas, 24–39
Support in SCFS File Systems, 7–22

R

RAID, 6–12
See Storage, System

Remote Access, 21–5

Reset Button, 14–9

Resource Management System
See RMS
Restoring
Booting Using the Backup Cluster Disk, 24–46, 29–4
Files, 24–45, 29–3
RMS
Processor Sets, 5–60
Running User-Defined Scripts, 5–34
RMS (Resource Management System)
Accounting, 5–3
CAA Failover Capability, 5–74
Concepts, 5–2
Core File Management, 5–25
Daemons, C–3
Event Handler Scripts, 9–18
Event Handlers, 9–16
Exit Timeout Management, 5–24
Idle Timeout, 5–24
Jobs, *See* RMS Jobs
Log Files, 5–73, 15–3
Memory Limits, 5–47
Monitoring, 5–6
Nodes, *See* RMS Nodes
Overview, 1–24, 5–1
Partition Queue Depth, 5–59
Partitions, *See* RMS Partitions
rcontrol Command, 5–8
Resources, *See* RMS Resources
rinfo Command, 5–6
rmsquery Command, 5–8
Servers and Daemons, 5–67
Site-Specific Modifications, 5–74
Specifying Configurations, 5–10
Starting Manually, 5–71
Stopping, 5–69
Stopping and Starting Servers, 5–72
Switch Manager, 5–72
Tasks, 5–3
Time Limits, 5–55
Timesliced Gang Scheduling, 5–56
Troubleshooting, 29–26
Useful SQL Commands, 5–76
RMS Jobs
Concepts, 5–16
Effect of Node and Partition Transitions, 5–28
Running as Root, 5–23
Viewing, 5–17
RMS Nodes
Booting, 5–63
Configuring In and Out, 5–61
Draining, 5–66
Node Failure, 5–64
Shutting Down, 5–63
Status, 5–64
Transitions, 5–28
RMS Partitions
Creating, 5–9
Deleting, 5–15
Managing, 5–8
Reloading, 5–13
Starting, 5–12
Status, 5–65
Stopping, 5–13
Transitions, 5–28
RMS Resources
Concepts, 5–16
Controlling Usage, 5–45
Effect of Node and Partition Transitions, 5–28
Killing, 5–21
Priorities, 5–46
Suspending, 5–20
Viewing, 5–17
routed, Not Supported, 29–24
Routing, 22–6
RSH, 26–2

S

SANworks Management Appliance
See Storage, System
SC Database
Archiving, 3–4
Backing Up, 3–2
Checking Integrity, 3–13
Deleting, 3–10
Managing, 3–1
Purging, 3–4
Restoring, 3–7
SC Monitor, 27–1
Attributes, 27–6
Distributing, 27–9
Events, 27–4
Hardware Components Managed by, 27–2
Managing, 27–6
Managing Impact, 27–13
Monitoring, 27–14
scmonmgr Command, 27–16

Specifying Hardware Components, 27–7
Viewing Properties, 27–15

SC Performance Visualizer, 11–1

SC Viewer, 10–1
Icons, 10–4
Invoking, 10–2
Menus, 10–3
Properties Pane, 10–11
Tabs, 10–9

SCFS, 1–25, 6–3
Configuration, 7–3
Creating, 7–7
Log Files, 15–7
Monitoring and Correcting, 7–28
Overview, 7–2
SC Database Tables, 7–35
scfsmgr Command, 7–8
SysMan Menu, 7–27
Troubleshooting, 29–40
Tuning, 7–32

Security, 3–12, 17–8, 26–3

Shutdown Grace Period, 2–16

Shutting Down
See Domain Members, Shutting Down

Single-Rail and Dual-Rail Configurations, 1–17, 5–76

sra Command
Description, 16–5, 16–8, 16–10, 16–13, 16–18, 16–20, 16–21, 16–22
Options, 16–12
Overview, 16–2
Syntax, 16–4

SRA Daemon, Checking Status, 29–37

sra dbcheck Command, 3–13

sra diag
Log Files, 15–6

sra edit Command
Node Submenu, 16–24
Overview, 16–22
System Submenu, 16–30
Usage, 16–23

sra_clu_min Script, 24–17

sra_orphans Script, 5–34

sra-display Command, 16–39

SSH (Secure Shell), 26–4
Commands, 26–10
Daemon, 26–11
Sample Configuration Files, 26–4
Troubleshooting, 29–37

Start Up Scripts, 24–16

Storage
Global, 1–20, 6–11
Local, 1–20, 6–10
Overview, 6–1, 6–9
Physical, 1–19
System, 1–21, 6–12
Third-Party, 24–13

Stride, 8–3

Stripe, 8–3

Supported Node Types, xliv

Swap Space, 21–37

sysman Command, 13–2, 18–2

SysMan Menu, 18–4

System Activity, Monitoring, 1–27

System Check Tool, 29–47

System Event Analyzer
See HP AlphaServer SC Node Diagnostics

System Firmware, Updating, 21–34

System Management, 17–9

T

TCP/IP Ephemeral Port Numbers, 29–25

Terminal Server, 1–12, 14–9
Changing Password, 14–12
Configuring for New Members, 14–12
Configuring Ports, 14–9, 14–10, 14–12
Connecting To, 14–16
Logging Out Ports, 14–14
Reconfiguring or Replacing, 14–9
User Communication, 14–14

Time Synchronization, Managing, 22–9

Troubleshooting, 29–1

U

UBC Mode, 7–4

UNIX Accounting, 21–44

User Administration
- Adding Local Users, 13–2
- Configuring Enhanced Security, 26–3
- Managing Home Directories, 13–3
- Managing Local Users, 13–3
- Overview, 13–1
- Removing Local Users, 13–2

V

verify Command, 24–48

Votes, 20–2

W

WEBES (Web-Based Enterprise Service), 28–2

X

X Window Applications
- Displaying Remotely, 22–25