IRIS FailSafe™ Administrator's Guide

Document Number 007-3109-001

CONTRIBUTORS

Written by Carolyn Curtis
Illustrated by Dan Young, Dany Galgani, Cheri Brown, and Carolyn Curtis
Edited by Christina Cary
Production by Derrald Vogt
Engineering contributions by Wei Hu, Michael Nishimoto, Paddy Sreenivasan, Gilberto Arnaiz, Ravi Chhabria, Ewan McKissock, and Raj Das
Cover design and illustration by Rob Aguilar, Rikk Carey, Dean Hodgkinson, Erik Lindholm, and Kay Maitz

IRIS FailSafe™ Administrator’s Guide
Document Number 007-3109-001

Contents

List of Figures vii

List of Tables xi

About This Guide xiii

Audience xiv

Structure of This Document xiv

Related Documentation xv

Conventions xvi

1. Features and Capabilities of the IRIS FailSafe System 1

High Availability, Dual-Active Operation, and Active/Standby Operation 2

System Requirements for Failover and Takeback 3

- The Failover Process 3
- IRIS FailSafe System Components 5
- IRIS FailSafe System Software 7

Handling Points of Failure 11

- Server 12
- Network Adapter 12
- Storage Disk or Disk Controller 15
- Service 16

2. Administering the IRIS FailSafe System 17

Using IRIS FailSafe Commands 17

- IRIS FailSafe Commands 18
- IRIS FailSafe Files and Directories 19

Getting the Current Network State 21

Getting the Current Node State 21

Performing Controlled Failback 21

Changing Cluster Configuration 22

Changing Cluster Configuration When *ha.conf* Is Not Being Modified 22

Changing Cluster Configuration When *ha.conf* Is Being Modified 25

Starting the IRIS FailSafe System 25

Stopping the IRIS FailSafe System 26

3. Configuring the IRIS FailSafe System 27

Configuring the Networks 27

Configuring Networks for a Dual-Active Configuration 27

Configuring Networks for an Active/Standby Configuration 32

Testing the Networks 36

Configuring the Netscape Server Option 37

Configuring a Netscape Server for a Dual-Active Configuration 38

Configuring a Netscape Server for a Dual-Active Configuration Serving the Same Document Root 42

Configuring a Netscape Server for an Active/Standby Configuration 43

Web Server Usage Accounting 45

Configuring the NFS Server Option 45

Configuring NFS for a Dual-Active Configuration 45

Configuring NFS for an Active/Standby Configuration 47

Configuring Shared Filesystems 48

4. Creating the Configuration File 51

Preparing for Configuration 52

Editing the Blocks of the Configuration File 54

System Configuration Block 55

Node Block 56

Application Classes Block 64

Disks/Filesystem Block 66

Optional NFS Blocks 69

Optional Web Server Blocks 71

Saving the Configuration File 72

5. **Testing the Configuration** 73

Testing the Configuration File 73

Testing Filesystems 73

Testing Public Network Interfaces 74

Testing the NFS Configuration 75

Testing Web Server Configuration 76

Testing System Behavior 76

A. **Setting Up an IRIS FailSafe System With CHALLENGE S Servers** 79

Setting Up the Component Systems 79

Cabling the Private and Public Networks 84

Cabling the Private (Heartbeat) Connection Between the Two Nodes 84

Cabling to the Public Network 85

Setting Up the IRIS FailSafe Serial Connection 87

Cabling Two CHALLENGE S Servers 88

Cabling a CHALLENGE S and a Larger Server 94

Testing the Installed IRIS FailSafe Hardware 96

Testing the Serial Connection 98

Installing the IRIS FailSafe Software 98

Installing System Software 100

Installing IRIS FailSafe Software 100

Setting *nvram* Parameters 101

Installing the Web Server Option 102

Installing the NFS Option 102

Modifying and Testing the Configuration File 102

Installing the Disk Plexing License 103

Cloning the Disk 103

B. **ha_cfgverify Error Messages** 105

C. **Keywords** 119

D. **Sample Configuration Files** 123

ha.conf.nfs_dual_active 123

ha.conf.web_active_standby 136

E. System Maintenance and Troubleshooting 147

Monitoring the Serial Connection 147

Troubleshooting System Problems 148

- Duplicate SCSI IDs 149
- Volumes That Have Not Been Mounted 149
- Netscape Server Warning Messages at Startup 150
- Inability to Access a Node via the Network 151
- Trouble Mounting Shared Filesystems 151
- Trouble Accessing the Network Interfaces on a Cluster Node 152
- Inability to Access a Shared Filesystem Over NFS 152
- Web Server Not Responding 153
- IRIS Failsafe System Does Not Start 154
- Local Monitor Failures 154
- Recovery Script Failures 155
- Replacing the Private Network 156
- Errors Logged to */var/adm/SYSLOG* 156

Replacing Batteries in the Remote Power Control Unit 159

Index 161

List of Figures

Figure 1-1 IRIS FailSafe System Components 5

Figure 1-2 IRIS FailSafe System Software Architecture 9

Figure 1-3 IRIS FailSafe System Cluster States and Transitions 10

Figure 1-4 Network Adapter or IP Address Takeover: Dual-Active 14

Figure 1-5 Network Adapter or IP Address Takeover: Active/Standby 14

Figure 1-6 Disk Storage Takeover 15

Figure 3-1 Interfaces and Networks to Configure for the Dual-Active System (Example) 27

Figure 3-2 Interfaces and Networks to Configure for the Active/Standby System (Example) 32

Figure 3-3 Dual-Active Configuration: Netscape Failover Example 39

Figure 3-4 Dual-Active Configuration Serving the Same Document Root: Netscape Failover Example 42

Figure 3-5 Active/Standby Configuration: Netscape Failover Example 43

Figure 3-6 Dual-Active Configuration: NFS Failover Example 46

Figure 3-7 Active/Standby Configuration: Netscape Failover Example 48

Figure 4-1 Configuration File Naming Conventions 53

Figure 4-2 System Configuration Block 55

Figure 4-3 Dual-Active Configuration File Block: First Node 57

Figure 4-4 Dual-Active Configuration File Block: Second Node 59

Figure 4-5 Active/Standby Configuration File Block: Primary Node 61

Figure 4-6 Active/Standby Configuration File Block: Secondary Node 62

Figure 4-7 Dual-Active Configuration File Block: Application Classes 64

Figure 4-8 NFS Active / Standby Configuration File Block: Application Classes 65

Figure 4-9 Web Server Active / Standby Configuration File Block: Application Classes 66

Figure 4-10 Dual-Active NFS Configuration File Block: Disk / Filesystem 67

Figure 4-11 Active / Standby Web Configuration File Block: Disk / Filesystem Configuration 68

Figure 4-12 NFS Dual-Active NFS Configuration File Block: Exported Filesystems 69

Figure 4-13 NFS Action and Action Timer Blocks 70

Figure 4-14 Web Server Active / Standby Configuration: Web Servers Block 71

Figure 4-15 Web Server Active / Standby Configuration File Blocks: Action and Action Timer 72

Figure A-1 Basic IRIS FailSafe Cabling Scheme 80

Figure A-2 IRIS FailSafe Cabling With Two Public Networks (Dual-Active Configuration) 81

Figure A-3 CHALLENGE S Server Ports: Private and Public Network Cabling 82

Figure A-4 Serial and Power Connections: Two CHALLENGE S Servers 83

Figure A-5 Connecting an Ethernet 10-BASE T Cable to the CHALLENGE S Server (S-100 and S-150 Models) 84

Figure A-6 Connecting an Ethernet AUI Cable to the CHALLENGE S Server 85

Figure A-7 Ethernet AUI cable 86

Figure A-8 Serial and Power Connections: Two CHALLENGE S Servers 88

Figure A-9 Remote Power Control Unit: Connector Panel 89

Figure A-10 Cabling the Serial Port on the Remote Power Control Unit 90

Figure A-11 Connecting the Serial Cable to the CHALLENGE S Server 91

Figure A-12 Cabling the Power Control Port 92

Figure A-13 Cabling the Power Control Unit 92

Figure A-14 Serial Connection: CHALLENGE S Server and Larger CHALLENGE Server 95

List of Tables

Table 1-1 IRIS FailSafe Processes 8
Table 1-2 IRIS FailSafe Scripts 8
Table 1-3 Cluster States 10
Table 2-1 IRIS FailSafe Executables 18
Table 2-2 ha_admin Command Parameters 19

About This Guide

The Silicon Graphics® IRIS FailSafe™ product provides a general facility for highly available services. The IRIS FailSafe system is based on two CHALLENGE® servers, each offering services, such as NFS™ and Netscape Communications Server™ (Netsite®). While running these services, the servers can also run database or other application software. Storage devices are physically attached to the two nodes in the system, but are owned and accessed by one node at a time.

The Silicon Graphics IRIS FailSafe system consists of the following hardware:

- two CHALLENGE XL, L, DM, or S servers, in any combination
- shared storage:
  - optional external peripheral enclosure for SCSI storage devices: CHALLENGE Vault XL, Vault L, or Vault DM
  - CHALLENGE RAID deskside or rackmount storage system; each chassis assembly has two storage-control processors (SPs) and at least five disk modules, caching enabled
- Ethernet or FDDI networking adapters and facilities
- required hardware upgrades and cables

The software for the IRIS FailSafe system consists of IRIX™ 5.3 with XFS, IRIX patches, IRIS FailSafe software, FDDI software (if FDDI networking is used), and software for the optional CHALLENGE RAID storage system.

Besides the basic configuration, the IRIS FailSafe product is available with an NFS or Web server option. Optional software includes NFS and Netsite.

IRIS Failsafe enhances the Silicon Graphics Oracle Parallel Server™ (OPS) by providing IP failover in an OPS hardware configuration. However, the two products are not merged administratively, so different tools are required to maintain a combined system.

Audience

This guide is written for the person who administers the IRIS FailSafe system. The IRIS FailSafe administrator must be familiar with the operation of CHALLENGE servers, as well as optional Vault storage systems or the CHALLENGE RAID storage system, whichever is used in the IRIS FailSafe configuration. Good knowledge of XLV and XFS is also required.

Structure of This Document

This guide contains the following chapters and appendices:

- Chapter 1, "Features and Capabilities of the IRIS FailSafe System," introduces the components of the IRIS FailSafe system and explains its hardware and software architecture. This chapter also describes cluster states and how failover works.
- Chapter 2, "Administering the IRIS FailSafe System," explains how to perform basic system tasks, such as adding filesystems to existing configurations, starting and shutting down the IRIS FailSafe system, and getting current status.
- Chapter 3, "Configuring the IRIS FailSafe System,"describes how to configure the IRIS FailSafe software.
- Chapter 4, "Creating the Configuration File," explains how to edit sample configuration files to create or modify an IRIS FailSafe system.
- Chapter 5, "Testing the Configuration," describes how to test the newly configured IRIS FailSafe system.
- Appendix A, "Setting Up an IRIS FailSafe System With CHALLENGE S Servers," explains how to cable the CHALLENGE S servers in an IRIS FailSafe system.

- Appendix B, “ha_cfgverify Error Messages,” lists error messages that can appear as output of the *ha_cfverify* command and gives an explanation of each one.
- Appendix C, “Keywords,” lists keywords used in the configuration file *ha.conf*.
- Appendix D, “Sample Configuration Files,” gives the full code for the two configuration files referenced in Chapter 4.
- Appendix E, “System Maintenance and Troubleshooting,” explains how to use system tools for maintenance, how to change cluster configuration, and how to troubleshoot system problems.

An index completes this guide.

Related Documentation

Besides this guide, other documentation for the IRIS FailSafe system includes

- *CHALLENGE S Server Owner’s Guide* (007-2314-003)
- *Deskside POWER CHALLENGE and CHALLENGE L Owner’s Guide* (007-1732-040)
- *NFS Administration Guide* (007-0850-070)
- *Getting Started with XFS Filesystems* (007-2549-001)
- *CHALLENGE RAID Owner’s Guide* (007-2532-00x; included with optional CHALLENGE RAID)
- *CHALLENGE Vault XL and SCSIBox 2 Owner’s Guide* (007-1762-030; included with optional CHALLENGE Vault XL)
- *CHALLENGE Vault L Owner’s Guide* (007-2443-001; included with optional CHALLENGE Vault L)
- *CHALLENGE Vault M Owner’s Guide* (007-2155-001; included with optional CHALLENGE Vault M)
- *Netscape Commerce and Communications Servers Administrator’s Guide* (007-2909-001)

Conventions

These type conventions and symbols are used in this guide:

Helvetica Bold Hardware labels

Italics Executable names, filenames, IRIX commands, manual or book titles, new terms, program variables, tools, utilities, variable command-line arguments, variable coordinates, and variables to be supplied by the user in examples, code, and syntax statements

`Fixed-width type`
Error messages, prompts, and onscreen text

`Bold fixed-width type`
User input, including keyboard keys (printing and nonprinting); literals supplied by the user in examples, code, and syntax statements (*see also* <>)

“” (Double quotation marks) Onscreen menu items and references in text to document section titles

[] (Brackets) Surrounding optional syntax statement arguments

<> (Angle brackets) Surrounding nonprinting keyboard keys, for example, <Esc>, <Ctrl-D>

Chapter 1

Features and Capabilities of the IRIS FailSafe System

The Silicon Graphics IRIS FailSafe system consists of a cluster of two CHALLENGE servers, which provides highly available services utilizing shared resources. These shared resources are owned and accessed by one server at a time.

The IRIS FailSafe system supports *fast failover*: if a component—network adapter, disk, disk controller, service, or server itself—fails, the service quickly (although not instantaneously) resumes because the second server in the system has gracefully shut down the failed system and taken over its services. To clients, the services are indistinguishable from the original services before failure occurred.

The IRIS FailSafe system also supports *takeback*: after the failed server has been restored, it automatically takes its services back from the surviving server and resumes its duties.

Besides the software that handles failover and takeback, the IRIS FailSafe system includes tools for system administration. These tools are described in Chapter 2, “Administering the IRIS FailSafe System.”

This chapter explains

- high availability, dual-active operation, and active/standby operation
- system requirements for failover and takeback
- handling points of failure

High Availability, Dual-Active Operation, and Active/Standby Operation

In a high-availability system, each node serves as backup for the other, sharing resources. Unlike the backup node in a mainframe-based fault-tolerant (continuous availability) system, which serves purely as redundant hardware for backup in case of failure, the resources of each node in a high-availability system can be used during normal operation.

Each IRIS FailSafe service is assigned one node as its primary owner and the other node as its secondary owner. Two configurations are possible:

- *Dual-active*: IRIS FailSafe services run concurrently on both nodes. Each node is primary owner of one or more IRIS FailSafe services; the other node serves as secondary node for those services. For example, for NFS, both nodes can be exporting different filesystems.

  In a cluster in this configuration, the primary node is the node on which a critical server process is currently active. The secondary node is the node on which a server process is restarted in case of failure on the primary node.

  This configuration requires three network adapters per server (one for the private network between the two nodes, two for public network or networks).

- *Active/standby*: Services run on one node—the primary node—which is the primary owner of all the services in the system. The other (secondary) node is the secondary owner of all services. After failover, the services run on the secondary node. In this case, the secondary node is a hot standby for failover purposes only. The resources of the secondary node are dedicated to standby use and are not available for normal operation.

  This configuration requires two adapters per server (one private, one public).

In either configuration, the IRIS FailSafe system software is installed on both nodes.

System Requirements for Failover and Takeback

In a high-availability system, if any part of a node fails, its highly available services are quickly restarted on the other node. The service user sees only a brief interruption of services and remains unaware that another server has taken them over. When the failed system is repaired and the recovered server resumes its services, the user is again unaware of any change in services.

For this minimal interruption of services to happen, hardware components of the system must be redundant (servers and network adapters) or transferrable (disk storage) and system software must include monitoring, failover (takeover, giveover), and takeback / giveback scripts.

For failover and takeback, the system must have

- redundant hardware: two servers, two or three network adapters per server, shared storage
- ability to save and transfer the "state" of the service so that it can resume where it left off when it is restarted
- software for coordinating control of the failover and takeback processes

This section describes how IRIS FailSafe meets these requirements:

- the failover process
- IRIS FailSafe system components
- IRIS FailSafe system software

The Failover Process

When one node fails, the surviving node

- using the serial connection between the two nodes, shuts down the failed node to prevent corruption of data
- takes over the public IP address of the failed node
- takes over the public MAC (Media Access Control) address of the failed node

- takes over the shared disks
- starts offering the services using its own resources

To move the state of a service from one node to the other, the ownership of shared filesystems must be changed. For NFS services, exported filesystems are moved. For Web services, HTML documents are offered by the backup node.

IRIS FailSafe supports the following three-step process:

1. Takeover: If node A detects that node B has failed, A executes a script that enables it to take over all services running on node B.
2. Giveaway: Node B executes a script that relinquishes all services to the surviving node A. This process is executed when node B is to leave the cluster.
3. Reboot: The surviving node, A, reboots the failed node, B.

The Takeback Process

After the failed node B is repaired, these reintegration processes occur:

1. Check: Node B checks to see what services it is running and what services node A is running. If node A has taken over node B's resources, the next two steps occur.
2. Giveback: Node A executes a script that relinquishes to the restored node all services included in the IRIS FailSafe system for which it is the backup node.
3. Takeback: The restored node B executes a script that enables it to take back all services included in the IRIS FailSafe system for which it is the primary owner (for example, the NFS filesystem and the filesystem containing the HTML [Web] documents).

If node B comes up and cannot rejoin the cluster, it remains in standby state.

‘When a failed node is repaired and rebooted, it is automatically reintegrated into the cluster and resumes servicing requests. This model supports planned outages for equipment and software upgrades.

IRIS FailSafe System Components

Figure 1-1 diagrams IRIS FailSafe system configuration. In this diagram, the second network adapter on each node is required for the dual-active configuration and not required for the active/standby configuration.

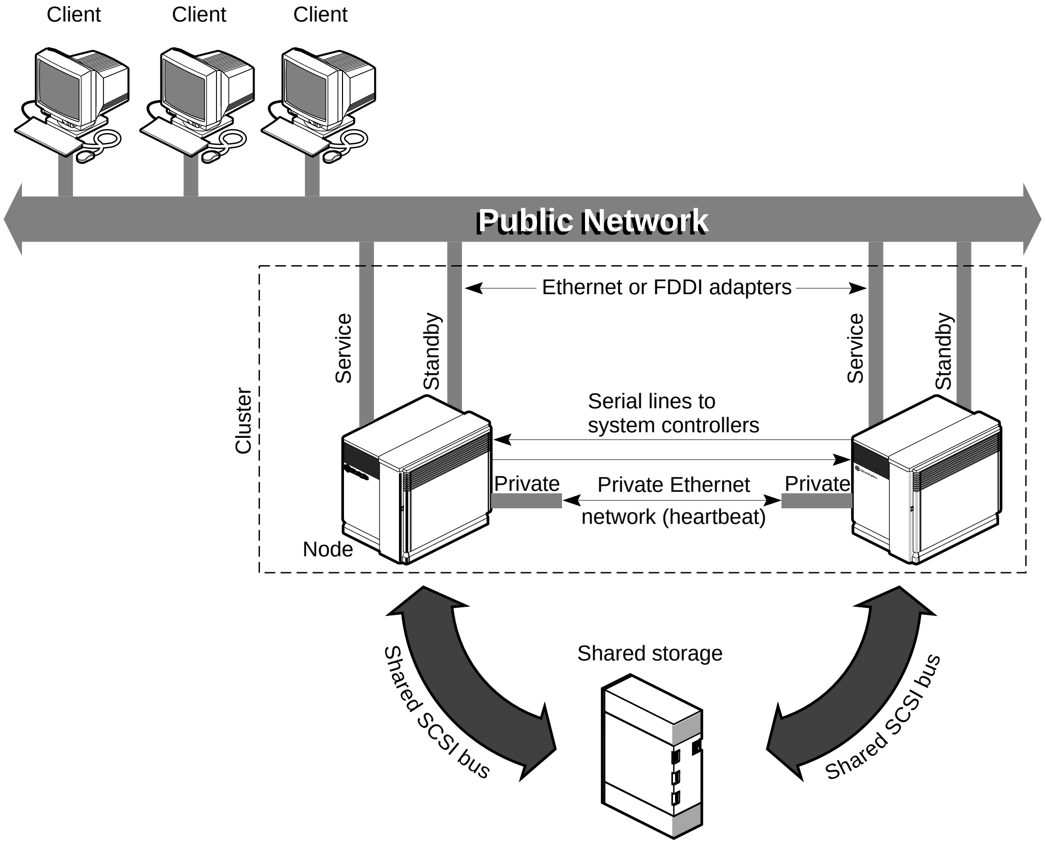


Figure 1-1 IRIS FailSafe System Components

Note these system components:

- two CHALLENGE servers: the *nodes*
- one interface to the private (Ethernet) network

  One Ethernet network adapter on each node is required for the private *heartbeat* connection, by which each node monitors the state of the other's IRIS FailSafe processes. The IRIS FailSafe software also uses this connection to pass control messages between nodes. Private for security reasons, these network adapters have distinct IP addresses.
- one or two interfaces to the public network

  The public network adapter(s) attached to each node connect to the public network, which links the cluster to clients. The service network adapters service client requests, such as NFS file operations. Their IP addresses are known to clients. The standby network adapters are backup adapters for servicing requests when a service adapter fails.

  For dual-active operation, two network adapters of the same type (Ethernet or FDDI) are required per node. On each node, one network adapter serves as the primary adapter; both primary adapters are used for normal operation. The second network adapters function as backup in case the primary adapters of the other node fail.

  If two networks are used, the primary adapter of one node and the second adapter of the other node must be on the same network (and vice versa) for failover to work.

  For active/standby operation, one public network adapter is required on each node. One node and its network adapter are primary; the other node and its network adapter are secondary (backup).

  Note: Although the cluster is defined in Figure 1-1 as nodes and the network adapters that are included in the server chassis, the network adapters are also considered separate possible points of failure, as discussed in "Handling Points of Failure," later in this chapter.
- *storage and SCSI bus* shared by the nodes in the cluster

  The nodes in the IRIS FailSafe system share disk storage over a shared fast and wide SCSI bus. The bus is shared so that either node can take over the disks in case of failure. The IRIS FailSafe system requires RAID storage, mirrored XLV volumes, or both.

- a *serial line* from each server to the other's Remote System Control port (for the CHALLENGE S server, the Silicon Graphics remote power control unit)

  A surviving node uses this line to reboot (or, optionally, power down) the failed node during takeover. This procedure ensures that the failed node is not accessing the disks when the surviving node takes them over.

IRIS FailSafe System Software

For services to remain available in a high-availability system, these software capabilities are required:

- Each node must monitor the other to determine its liveliness, that is, to detect CPU, network, disk, or IRIS FailSafe system software failure.
- Each node must monitor the liveliness of the highly available services running on both nodes, detecting any crash or hang.
- If a node fails, the surviving node must be able to
  - shut down the failed node
  - take over the other node's public network address and disk storage
  - restart any failed service
- When a failed node is coming back up, it must be able to determine if the other node is running:
  - If the other node is not running, or if the private network or network adapters on the private network are not functioning, the recovering node will not take over cluster services. Instead, it remains in standby state and sends notification that it could not rejoin the cluster.
  - If the other node is running, the recovered node must be able to take back its services from the surviving node and reintegrate itself into the IRIS FailSafe system
- A node that has taken over services in a failover (surviving node) must be able to relinquish these services when the failed node comes back up.

In addition, for administrative purposes, each node must be able to manually force failover or automatic reintegration and allow for customized modules for monitoring, restarting, takeover, and reintegration.

These capabilities are provided by a number of IRIS FailSafe processes and scripts, as summarized in Table 1-1and Table 1-2.

Table 1-1 IRIS FailSafe Processes

Process	Executed By	Purpose
ha_nc	Each node	Keeps track of the state of this node in the cluster.
ha_appmon	Each node	Monitors its own and other node's services.Generates and monitors the heartbeat messages that the cluster nodes exchange. Executes scripts.
ha_killd	Each node	Monitors the serial connection to the other node and provides power-cycling capability.

Table 1-2 IRIS FailSafe Scripts

Script	Executed By	Purpose
lmon	Each node	This script (one for each service) monitors a service on this node.
rmon	Each node	This script (one for each service) monitors a service on the other node.
takeover	Surviving node	Surviving node takes over failed node's IRIS FailSafe services.
giveaway	Failed node	Failed node gives up its services to the other node.
takeback	Failed (recovered) node	Recovered node takes back its services and reintegrates itself into system.
giveback	Surviving node	Surviving node relinquishes recovered node's services.

Figure 1-2 diagrams IRIS FailSafe software architecture.

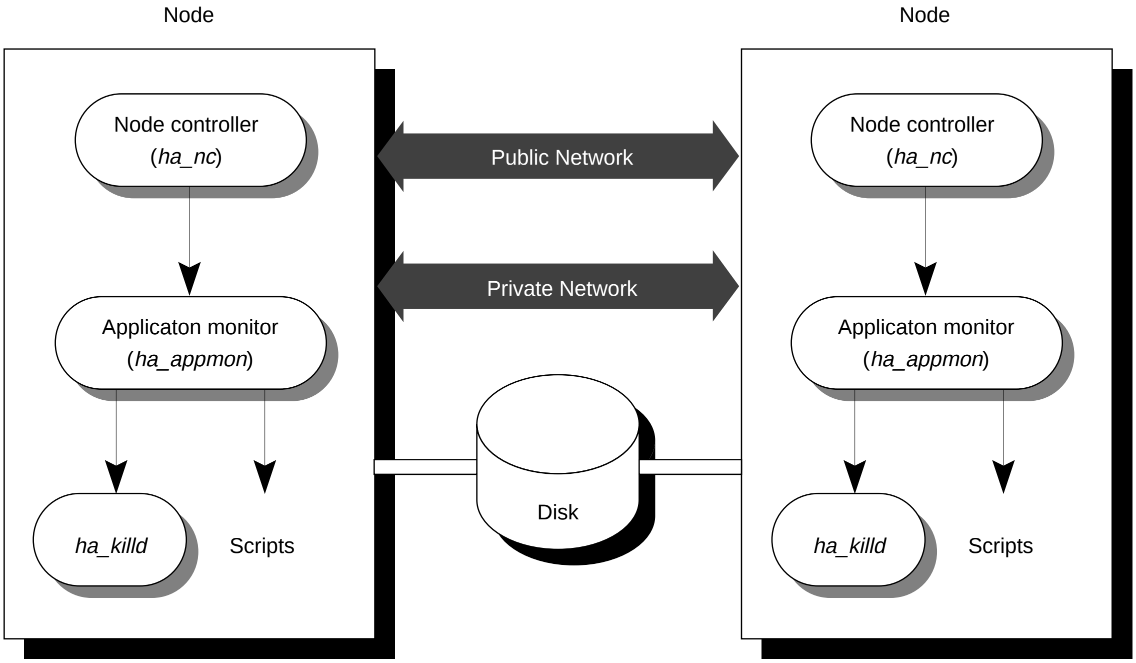

Node
Node
Node controller (*ha_nc*)
Node controller (*ha_nc*)
Public Network
Private Network
Applicaton monitor (*ha_appmon*)
Applicaton monitor (*ha_appmon*)
Disk
ha_killd
Scripts
ha_killd
Scripts

Figure 1-2 IRIS FailSafe System Software Architecture

The rest of this section explains the roles of the node controller process *ha_nc* and the application monitor process *ha_appmon* in detecting failure.

Node Controller

The node controller process *ha_nc* evaluates the current cluster state. Table 1-3 summarizes the cluster states.

Table 1-3 Cluster States

Cluster State	Definition
joining	Server is coming up and joining the cluster. The node controller should never remain in this state for more than two or three minutes.
normal	Actively processing cluster services.
degraded	Surviving node in the cluster.
standby	Local error has been detected; this node has stopped monitoring the other node in the cluster and is no longer part of the cluster. If a node cannot rejoin the cluster during the joining phase, it moves to this state.
error	Crashed or hung.

Figure 1-3 diagrams the cluster states and the events that govern them.

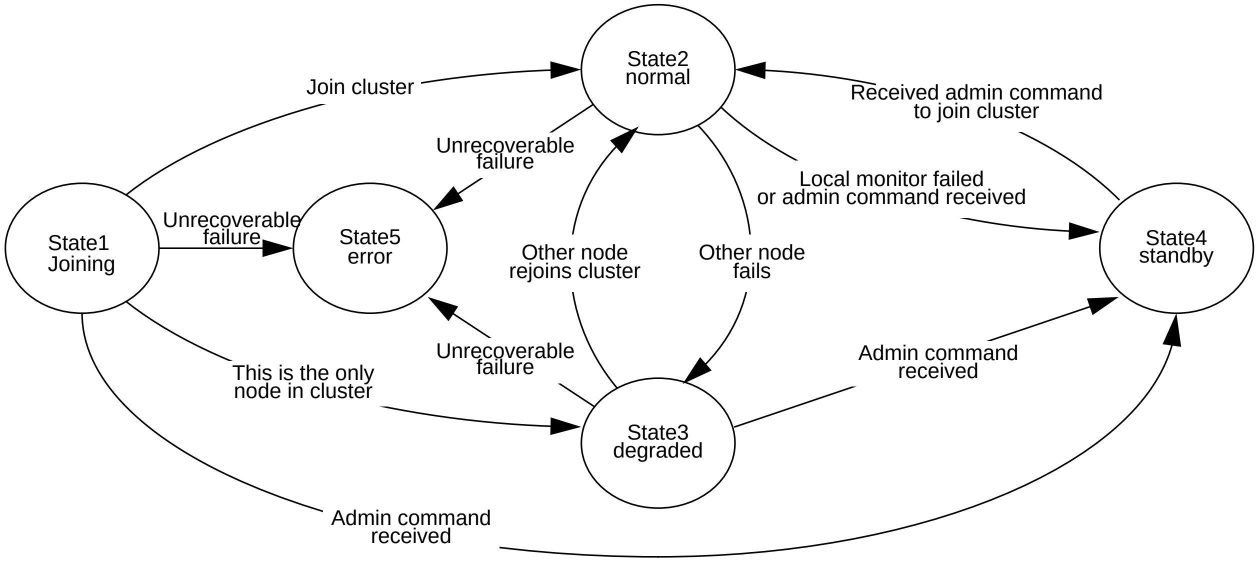


Figure 1-3 IRIS FailSafe System Cluster States and Transitions

Application Monitor

On each node, the application monitor process *ha_appmon* monitors all services on both nodes and reports any abnormalities to the node controller. This process executes two scripts for each service:

- local monitor script (*lmon*) for monitoring the service on the local node
- remote monitor script (*rmon*) for monitoring the service on the other node in the cluster

In addition, the two application monitor processes exchange heartbeat messages. During state transitions (takeover, giveaway, giveback, takeback), it also executes the actions taken.

If you add a service that you want included in IRIS FailSafe system failover, you must write scripts to handle failover and takeback. Depending on your service, you may need to provide one or more of the scripts listed in Table 1-1. The NFS and Web server options include additional scripts (*nfs*, *ha_nfs_lmon*, *ha_nfs_rmon*, *ha_web_lmon*, and *ha_web_rmon*). Finally, you must update the *ha.conf* file to register your scripts with the IRIS FailSafe software.

Handling Points of Failure

The goal of the IRIS FailSafe system is to make a pair of servers highly available to clients over the network. To remain available, a network-based service must be able to survive failures in the

- server
- network adapter
- storage disk or disk controller
- service

The rest of this section explains how the IRIS FailSafe system handles or eliminates these points of failure.

Caution: The IRIS FailSafe system is designed to survive a single point of failure. Therefore, when a system component fails, it must be restarted, repaired, or replaced as soon as possible to avoid the possibility of dual failure.

Server

If a server crashes or hangs (for example, due to a parity error or bus error), it will not respond to the heartbeat message sent by *ha_appmon* on the other node. The other node takes over the failed node's services after resetting the failed node.

If a server fails, the network adapters, access to storage, and services also become unavailable. See the succeeding sections for descriptions of how the IRIS FailSafe system handles or eliminates these points of failure.

Network Adapter

Attached to each node in the IRIS FailSafe system are one or two public network adapters connected to the public network. These network adapters are configured as follows:

- If there are two public network adapters per node (dual-active configuration only), the first is configured as the primary interface for a particular service. The second network adapter is configured as secondary on each node.

  If the primary network adapter on one node becomes unavailable because it or the server itself has failed, the other node reconfigures its own secondary adapter as a second primary adapter by giving it the IP address of the failed network adapter. The other node also reconfigures the system to accommodate this change. Thus, all client requests to the address of the failed network adapter are redirected to the surviving node.

- If there is only one public network adapter per node (active/standby configuration), the public network adapter on the primary node is configured as primary and the network adapter on the secondary node is configured as secondary.

  If the primary network adapter (or the primary node) fails, the secondary node, which functions as a hot standby, assigns the IP address of the primary node's network adapter to its own network adapter and reconfigures the system to accommodate this change.

  In a dual-active configuration with only one public network adapter per node, the public network adapters are configured according to how the services are owned in the cluster. The network adapter on a node is configured as primary if the node is the primary owner of the service; it is configured as secondary if the node is the secondary (backup) owner of the service.

Note: For systems using Ethernet for the public network, the IRIS FailSafe software uses *re-arp* to program a surviving network adapter to the IP address of a failed network adapter. To support client systems that do not implement the full IP *re-arp* protocol, the IRIS FailSafe software can also use *re-mac*, which changes the hardware Ethernet address of the surviving network adapter to that of the failed network adapter.

Figure 1-4 and Figure 1-5 diagram network adapter or IP address takeover.

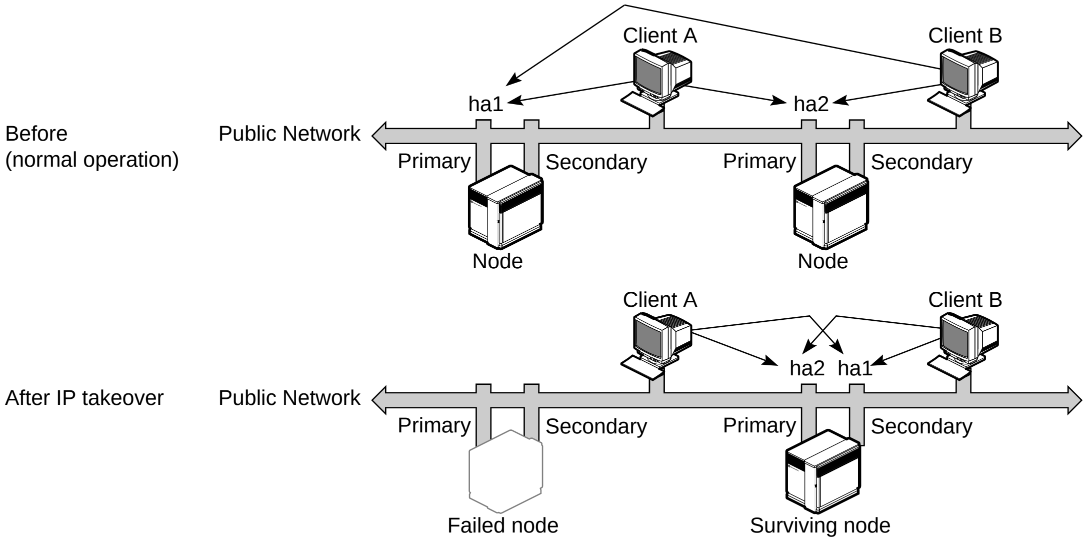


Figure 1-4 Network Adapter or IP Address Takeover: Dual-Active

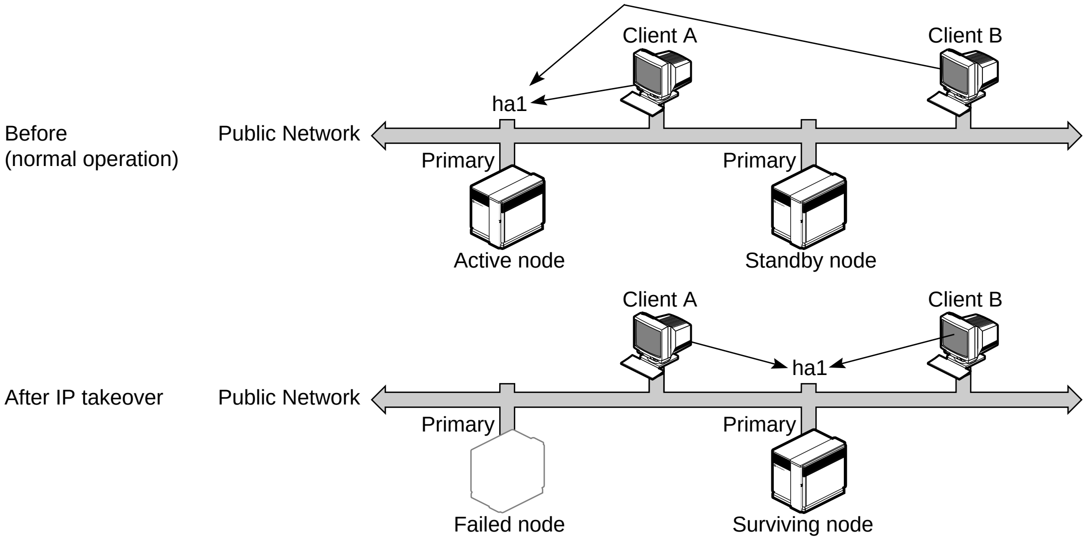


Figure 1-5 Network Adapter or IP Address Takeover: Active/Standby

Storage Disk or Disk Controller

The IRIS FailSafe system includes shared SCSI-based storage in the form of one or more CHALLENGE RAID storage systems, CHALLENGE Vaults with plexed disks, or both.

Note: CHALLENGE S systems must use the fast and wide SCSI bus.

If a disk fails, the storage system is equipped to keep services available through its own capabilities; no participation of the IRIS FailSafe system software is required.

If a disk controller fails, the IRIS FailSafe system software initiates the failover process. Figure 1-6 diagrams disk storage takeover.

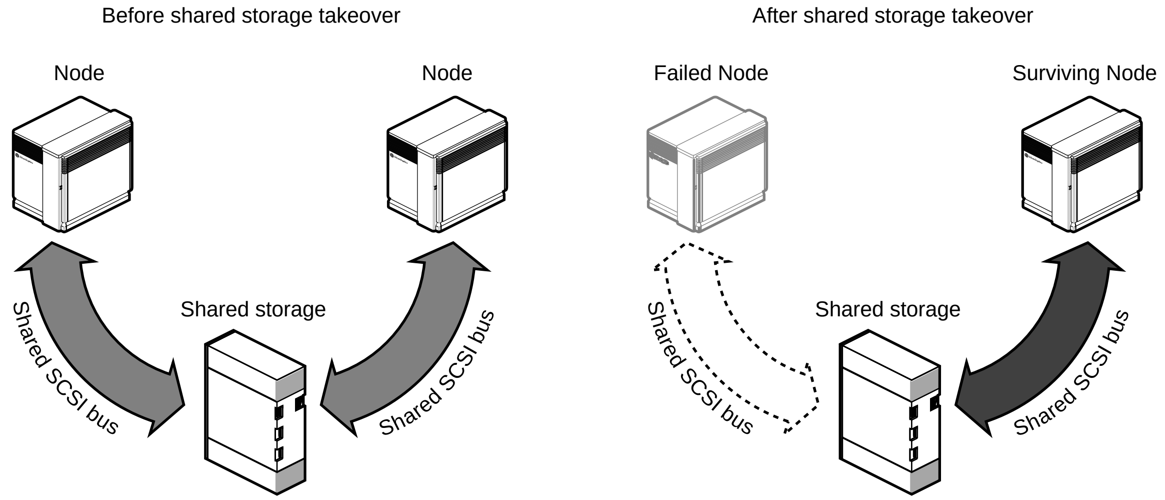


Figure 1-6 Disk Storage Takeover

The surviving node takes over the shared disks and recovers the filesystem. This process is expedited by the XFS journaled filesystem, which supports fast recovery because it does not require fsck'ing.

For more information on storage systems, consult the manuals listed in the introduction to this guide.

Service

The shared storage system makes it possible for the IRIS FailSafe system to support failover for services included in the IRIS FailSafe configuration.

The node that is the primary owner of a service writes application data to the shared storage system. If this node becomes unavailable, the surviving node reads from the shared storage system to determine the state of the service, so that it can continue the service with the same context as that of the failed node. Thus, when a service or any of the previously discussed system components fails, the surviving node takes over the service so that it appears to restart after a brief interruption.

The IRIS FailSafe software provides a framework for building high-availability services, with application-specific monitoring only for NFS and Web services. Other applications must provide facilities for saving and updating data to the shared storage system. Except for NFS locking, NFS and Web servers are stateless, with nothing to checkpoint. The IRIS FailSafe software supports failure of NFS locks. So-called stateful services require keeping track of the service’s state, which must be saved to disk.

Chapter 2

Administering the IRIS FailSafe System

This chapter explains

- using IRIS FailSafe commands
- getting the current network state
- getting the current node state
- performing controlled failback
- changing cluster configuration
- starting the IRIS FailSafe system
- stopping the IRIS FailSafe system

Using IRIS FailSafe Commands

The IRIS FailSafe software includes the command line interface *ha_admin*. Use this command to

- show the state of a node
- set and show timeouts
- perform a manual cluster shutdown or manually detach a node from the cluster or reintegrate it into the cluster
- debug the cluster

Note: This command interacts using remote procedure calls (RPCs) as explained in the *IRIX Network Programming Guide* (007-0810-050).

All messages from scripts and from the IRIS FailSafe daemons go into the */var/adm/SYSLOG* file.

IRIS FailSafe Commands

The IRIS FailSafe system uses the executables summarized in Table 2-1.

Table 2-1 IRIS FailSafe Executables

Command	Use	Reference
ha_admin	Cluster administration and information.	Later in this section
ha_appmon	Application monitor.	"Application Monitor" in Chapter 1
ha_killd	Serial-line monitor process; the reset device control daemon.	
ha_cfgcksum	Computes a checksum for the configuration file. Because *ha.conf* is kept on two systems, the IRIS FailSafe software uses a checksum on the files to make sure they are identical.	
ha_cfginfo	Extracts information from the configuration file; this command is required for adding scripts for new applications	
ha_cfgverify	Verifies the information in the *ha.conf* file, cross-checking it against other configuration information in the system.	"Saving the Configuration File" in Chapter 4; messages in Appendix B
ha_exec	Used internally by the IRIS FailSafe software only.	
ha_nc	Node controller.	"Node Controller" in Chapter 1
ha_spng	Tests the serial link between the two node.	
macconfig	Displays and changes the MAC address of a network interface.	"Setting re-mac Parameters" in Chapter 4

Table 2-2 summarizes parameters for the *ha_admin* command.

Table 2-2 ha_admin Command Parameters

Parameter	Use
-fr [servername]	Reintegrates a failed node into the cluster (“force” option)
-fs [servername]	Moves the specified node from degraded state to standby state (“force” option)
-i	Displays the state of the specified node (controller); see “Node Controller” in Chapter 1 for details
-m start [servername]	Starts monitoring the serial connection to the other node
-m stop [servername]	Stops monitoring the serial connection to the other node
-q	Shuts down the specified node, whether or not the other node is in normal state
-r [servername]	Reintegrates the specified node, which is in standby state, into the cluster
-s [servername]	Changes the specified node from normal to standby state so that you can remove it from the cluster
-x	Switches the heartbeat and IRIS FailSafe internode messages from the public network to the private network

IRIS FailSafe Files and Directories

The */var/ha* directory is the default location for these files and directories:

- *actions*: contains the top-level scripts
  - *giveaway*
  - *giveback*
  - *takeback*
  - *takeover*

  The directory *actions* also contains the scripts *mail* and *kill*.

- *actions.d*: contains the scripts for giveaway, giveback, kill, takeback, and takeover**.**

  Each directory contains links to scripts in the *resources* directory. The links are used to specify the order in which the scripts are to be executed.
- *ha.conf*: configuration file, once it is created
- *resources*: this subdirectory contains the actual scripts, as well as information on all system resources, such as filesystems, interfaces, and options (NFS and Web server).

  For example, the order in which the system resources are taken over is imposed by the order in *actions.d/takeover*. This file uses markers such as S100, S200 S600, and so on.

  The script that enforces this order is in the *actions* directory. For example, the script *takeover* executes all scripts in *actions.d/takeover* in lexical order.

  The scripts in the *actions* directory execute the appropriate scripts in the *resources* directory in the lexical order imposed by their links in the *actions.d* directory.

The */var/ha* directory also contains scripts for options:

- *ha_nfs_lmon*: NFS local monitor script
- *ha_nfs_rmon*: NFS remote monitor script
- *ha_web_lmon*: Web server local monitor script
- *ha_web_rmon*: Web server remote monitor script

  A local and remote monitor script must be present for each service that you want included in the IRIS FailSafe system.

If you change the location of any of these files, you must also change the configuration file accordingly.

Getting the Current Network State

To display the network state, use **netstat**(1M) with the **-i** flag. For information on this command, see its reference page.

Getting the Current Node State

To display the state of a node (controller), enter

```
ha_admin -i nodename
```

A possible return might be

```
ha_admin: Node controller state normal
```

Table 1-3 in Chapter 1 explains the possible returns from this command.

Performing Controlled Failback

By default, the IRIS FailSafe software restarts a failed node when the surviving node detects that the other node has failed. This action reboots the failed node. After the node comes back up, it automatically reclaims its resources. For an IRIS FailSafe system with the NFS option, for example, the node would take back its disks, mount them, and then re-export its filesystems. Having a failed node rejoin the cluster automatically after restarting is desirable because most failures are due to transitory software problems that are cleared after a system reboot.

In some cases, however, an administrator may wish to configure the cluster so that a failed node does not automatically rejoin the cluster after it restarts after a failure. Instead, the administrator may wish to reintegrate a node manually. In this case, you should do the following:

1. After the cluster is in normal state (both nodes active), run on both nodes of a dual-active configuration

   ```
   chkconfig failsafe off
   ```

   Note: In an active / standby configuration run this command only on the active node.

This command prevents each node from starting IRIS FailSafe on reboot.

2. If a failure occurs, verify the state of the failed node. If it is to be reintegrated into the cluster, run

   ```
   chkconfig failsafe on
   ```

3. Start up the IRIS FailSafe system:

   ```
   /etc/init.d/failsafe start
   ```

4. After the node comes up and the cluster reaches normal state, run

   ```
   chkconfig failsafe off
   ```

To reconfigure the cluster so that failed nodes are always reintegrated automatically, run *chkconfig failsafe on* on both nodes.

Changing Cluster Configuration

This section explains how to change cluster configuration. Two situations call for different procedures:

- changing cluster configuration when *ha.conf* is not being modified
- changing cluster configuration when *ha.conf* is being modified

Changing Cluster Configuration When *ha.conf* Is Not Being Modified

This section explains how to use *ha_admin* parameters for

- removing a node in normal state
- removing a node in degraded state
- reintegrating a node in standby state into a cluster
- shutting down a node

Removing a Node in Normal State

To remove a node from a cluster, use the “standby” parameter (**-s**):

```
ha_admin -s servername
```

This command returns

```
ha_admin: <servername> successfully moved to standby
```

This command puts the named server in standby state so that you can upgrade server hardware or software.

Note: High-availability services such as NFS and Web are no longer highly available from this point on. Only one node is providing service.

Removing a Node in Degraded State

To move a node that is in degraded state (providing all cluster services) into standby state, use the “force” option (**-f**) with the “stop” parameter (**-s**) to remove it from the cluster.

```
# ha_admin -fs servername
```

This command returns

```
ha_admin: <servername> successfully moved to standby
```

Note: No high-availability services are available, since neither node is providing service.

Reintegrating a Node in Standby State Into a Cluster

After software or hardware has been upgraded or repaired, you can add or reintegrate a node that is in standby state into a cluster. The reintegrating node must have gone into standby state because

- the node was removed from the cluster with *ha_admin -s <servername>*
- the node booted up and the initial cluster rejoin failed

To reintegrate a node in standby state, use the “reintegrate” (**-r**) option:

```
# ha_admin -r servername
```

This command returns

```
ha_admin: <servername> successfully reintegrated
```

If the reintegrating node is joining a cluster that has a node in the degraded state, the cluster’s high-availability services become highly available again. If the reintegrating node is joining a cluster with a node in the standby state, the cluster’s high-availability services are available, but not highly available.

If a node has moved to standby state because of local monitor failure, reintegration is accomplished using *ha_admin* with the “force” option (**-f**)

This command returns

```
ha_admin: <servername> successfully reintegrated
```

Shutting Down a Node

To shut down the node which is a part of a cluster, use the **-q** option. If the other node is in normal state, it takes over this node’s resources and provides all the high-availability services (moves to degraded state).

```
# ha_admin -q servername
```

This command returns

```
ha_admin: <servername> successfully moved to standby
```

If the node being shut down is in standby state (the other node is in degraded state), this command has no effect and returns

```
ha_admin: error = Invalid argument
The state transition is not valid or -f option is needed
```

If the node being shut down is in degraded state (the other node is in standby state), this command stops all high-availability services and returns

```
ha_admin: <servername> successfully moved to standby
```

Changing Cluster Configuration When *ha.conf* Is Being Modified

If you must change the configuration file *ha.conf*, follow these steps to update the cluster:

1. Put one of the nodes in standby state:

   ```
   ha_admin -s servername
   ha_admin: <servername> successfully moved to standby
   ```

   At this point, high-availability services such as NFS and Web are no longer highly available.

2. Perform the necessary hardware and software modifications to the node. Change the configuration file.

3. Put the other node into standby state with

   ```
   ha_admin -fs servername
   ha_admin: <servername> successfully moved to standby
   ```

   At this point, no high-availability services are available.

4. Update the configuration file so that it is identical to the configuration file on the first node.

5. Reintegrate the nodes one at a time using

   ```
   # ha_admin -r servername
   ha_admin: <servername> successfully reintegrated
   ```

   At this point, high-availability services automatically become highly available again.

Starting the IRIS FailSafe System

To start the IRIS FailSafe system, enter

```
chkconfig failsafe on
```

Then either reboot, or enter

```
/etc/init.d/failsafe start
```

Repeat the process for the second server.

Stopping the IRIS FailSafe System

To start the IRIS FailSafe system, enter

```
/etc/init.d/failsafe stop
```

and wait for the command to finish. Repeat the process for the second server.

Chapter 3

Configuring the IRIS FailSafe System

This chapter explains how to set up the software for an IRIS FailSafe system. The process consists of:

- configuring the networks
- configuring the Netscape server option
- configuring the NFS server option
- configuring shared filesystems

Configuring the Networks

This section explains how to configure the networks for the dual-active and active/standby configurations.

Configuring Networks for a Dual-Active Configuration

Figure 3-1 diagrams the interfaces and networks to configure.

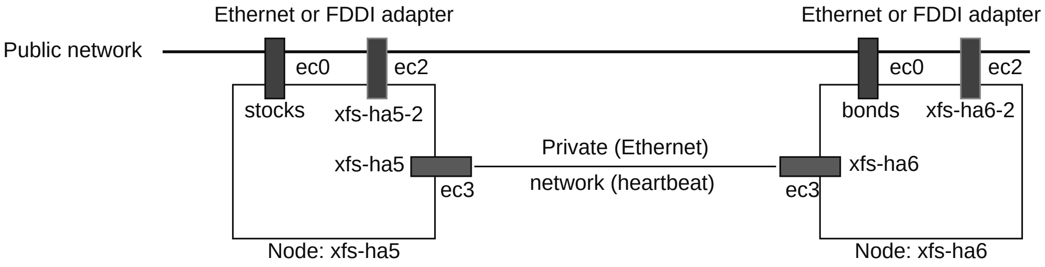


Figure 3-1 Interfaces and Networks to Configure for the Dual-Active System (Example)

Note: In a dual-active system, neither node is primary. Each node serves as primary node for a distinct set of highly available services.

The interfaces and networks are

- private (Ethernet)

  One Ethernet network adapter on each node is required for the private *heartbeat* connection, by which each node monitors the state of the other. Private for security reasons, this network's adapters have distinct IP addresses (one for each adapter).

- public (two for each node, either all Ethernet or all FDDI)

  The network adapters attached to each node connect to the public network, which links the cluster to clients. The service network adapters service client requests, such as for NFS file operations. Their IP addresses are known to clients. The standby network adapters are backup adapters for servicing requests when a service adapter fails.

  On each node, one network adapter serves as the primary adapter; both primary adapters are used for normal operation. The other network adapter in each node functions as backup in case the primary adapter on the other node fails.

Before you begin configuring the networks, have ready the following information:

- Hostnames for the two nodes in the cluster (*xfs-ha5* and *xfs-ha6* in Figure 3-1)
- For each node, network names for
  - private (heartbeat) (by convention, *hostname*; *xfs-ha5* and *xfs-ha6* in Figure 3-1)

    Note: Use the hostname for these network adapters; for example, *xfs-ha5* and *xfs-ha6*. These addresses remain constant, whereas the public network addresses can change in failover situations. When you are naming entities in the IRIS FailSafe system, avoid using the configuration file keywords; see Appendix C, "Keywords," for an alphabetical list.

  - primary public interface (by convention, *interfacename*; *stocks* and *bonds* in Figure 3-1)

  - secondary public (by convention, *interfacename-2*; *xfs-ha5-2* and *xfs-ha6-2* in Figure 3-1)

- For each node, IP addresses for all these interfaces

  Note: If client workstations or servers in your network do not support the full IP *re-arp* protocol, use *macconfig* to obtain the MAC (Media Access Control) addresses for each network interface on each node. You use this in adapting the configuration file later in the installation process. MAC address failover is necessary for supporting NFS for certain PC clients. MAC address failover is supported on Ethernet networks only, and not on FDDI networks.

To configure the public and private networks, follow these steps:

1. To set the */etc/sys_id* and the hostname, enter

   ```
   echo yourhostname > /etc/sys_id
   hostname -s yourhostname
   ```

   For example:

   ```
   echo xfs-ha5 > /etc/sys_id
   hostname -s xfs-ha5
   ```

2. Add to */etc/hosts* the

   - private interface name for this server
   - primary and secondary interface names for this server
   - private interface name for the other server
   - primary and secondary interface names for the other server

   For example:

   ```
   # IP address-hostname database (see hosts(4) for more information).

   # This entry must be present or the system will not work.
   127.0.0.1 localhost

   192.48.165.94 stocks
   192.48.165.95 xfs-ha5-2
   192.48.165.92 bonds
   192.48.165.93 xfs-ha6-2
   197.50.50.11 xfs-ha6.company.com xfs-ha6
   197.50.50.22 xfs-ha5.company.com xfs-ha5
   ```

In the six lines above,

- 192.48.165.94 stocks is the primary node's public-network active interface IP address
- 192.48.165.95 xfs-ha5-2 is the primary node's public-network standby interface IP address
- 192.48.165.92 bonds is the secondary node's public-network active interface IP address
- 192.48.165.93 xfs-ha6-2 is the secondary node's public-network standby interface IP address
- 197.50.50.11 xfs-ha6.company.com xfs-ha6 is the primary node's private-network heartbeat IP address
- 197.50.50.22 xfs-ha5.company.com xfs-ha5 is the secondary node's private-network heartbeat IP address

3. Update */etc/config/netif.options* as follows, referring to Figure 3-1 for information on the use of the interfaces:

```
# Append the interface name and remove the leading : to override
# the primary interface selection.

if1name=ec3

# To override the primary interface address, change the value part
# and remove the leading : character.

if1addr=$HOSTNAME

# To override the name and/or address of the first gateway interface,
# change the value part and remove the leading : character.

if2name=
if2addr=

# If this host has more than 2 interfaces, you must define values for
# if3name (and if4name if appropriate). Change if3addr (and if4addr) to
# the appropriate names in /etc/hosts if your site has different naming
# conventions.

if3name=ec2
if3addr=$HOSTNAME-2
```

```
: if4name=
: if4addr=gate3-$HOSTNAME
# If this host has more than 8 network interfaces, set the number of
# interfaces that the network startup script will configure.

: if_num=8
```

Note: In this file, you configure the private (heartbeat) interface and the secondary public interface, but not the primary public interface. The IRIS FailSafe software configures the active public interface.

4. Set */etc/config/routed.options* so that the routes are not shown over the private network:

   ```
   -h -q
   ```

   This option is required for IRIS FailSafe to function correctly.

5. Reboot the system to put the network configuration into effect.

6. To configure your e-mail interface so that you can receive notification of cluster transitions, set up an e-mail alias on each node that includes a user on each server and at least one user outside the IRIS FailSafe cluster.

7. Notice that the IRIS FailSafe system is *chkconfig*’d off by default:

   ```
       Flag                 State
       ====                 =====

       autoconfig_ipaddress off
       automount            off
       gated                off
       failsafe             off
       lockd                on
       mrouted              off
       named                off
       ns_httpd             off
       network              on
       nfs                  on
       rarpd                off
       routed               on
       rtnetd               off
       rwhod                off
       timed                on
       timeslave            off
       verbose              off
   ```

```
vswap               off
xlv                 on
yp                  off
ypmaster            off
ypserv              off
```

Caution: Note that *yp* is off. Do not run *yp*.

Configuring Networks for an Active/Standby Configuration

Figure 3-2 diagrams the interfaces and networks to configure.

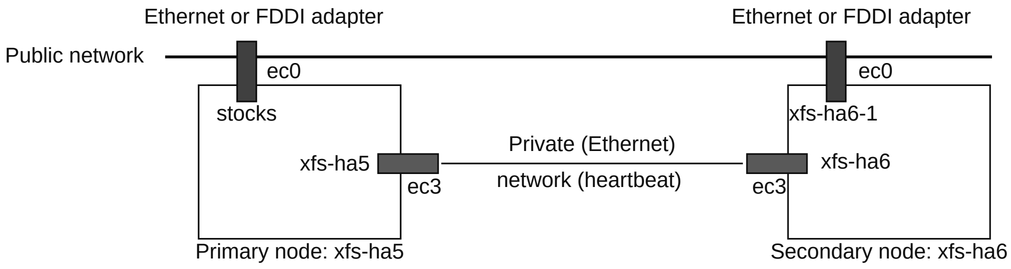


Figure 3-2 Interfaces and Networks to Configure for the Active/Standby System (Example)

The interfaces and networks are

- private (Ethernet): see the description at “Configuring Networks for a Dual-Active Configuration” earlier in this chapter for details
- public (one for each node, either both Ethernet or both FDDI)

  The network adapter attached to each node connects to the public network, which links the cluster to clients. The service network adapters service client requests, such as for NFS file operations. Their IP addresses are known to clients. The network adapter at the secondary node is a backup adapter for servicing requests when the primary node fails.

Before you begin configuring the networks, have ready the following information:

- Hostnames for the two nodes in the cluster
- Network names for
  - public interface for the primary node (by convention, *interfacename*; *stocks* in Figure 3-2)
  - public interface for the secondary (backup) node (by convention, *hostname-2*; *xfs-ha6-2* in Figure 3-2)
  - private (heartbeat) for both nodes (by convention, *hostname*; *xfs-ha5* and *xfs-ha6* in Figure 3-2)

  Note: Use the hostname for these network adapters; for example, *xfs-ha5* and *xfs-ha6*. These addresses remain constant, whereas the public network addresses can change in failover situations. When you are naming entities in the IRIS FailSafe system, avoid using the configuration file keywords; see Appendix C, “Keywords,” for an alphabetical list.
- For each node, IP addresses for all its interfaces

Note: If client workstations or servers in your network do not support the full IP *re-arp* protocol, use *macconfig* to obtain the MAC addresses for each network interface on each node. You use this in adapting the configuration file later in the installation process. MAC (Media Access Control) address failover is necessary for supporting NFS for certain PC clients. MAC address failover is supported on Ethernet networks only, and not on FDDI networks.

To configure the public and private networks, follow these steps:

1. To set the */etc/sys_id* and the hostname, enter

   ```
   echo yourhostname > /etc/sys_id
   hostname -s yourhostname
   ```

   For example:

   ```
   echo xfs-ha5 > /etc/sys_id
   hostname -s xfs-ha5
   ```

2. Add to */etc/hosts* the
   - private interface name for each server
   - public interface name for each server

   For example:

   ```
   # This entry must be present or the system will not work.
   127.0.0.1 localhost

   192.48.165.94 stocks
   192.48.165.93 xfs-ha6-2
   197.50.50.11 xfs-ha6.company.com xfs-ha6
   197.50.50.22 xfs-ha5.company.com xfs-ha5
   ```

   In the six lines above,
   - `192.48.165.94 stocks` is the primary node's public network interface IP address
   - `192.48.165.93 xfs-ha6-2` is the secondary node's public network interface IP address
   - `197.50.50.11 xfs-ha6.company.com xfs-ha6` is the primary node's private network heartbeat IP address
   - `197.50.50.22 xfs-ha5.company.com xfs-ha5` is the secondary node's private network heartbeat IP address

3. Update */etc/config/netif.options* as follows, referring to Figure 3-1 for information on the use of the interfaces:

   ```
   # Append the interface name and remove the leading : to override
   # the primary interface selection.

   if1name=ec3

   # To override the primary interface address, change the value part
   # and remove the leading : character.

   if1addr=$HOSTNAME

   # To override the name and/or address of the first gateway interface,
   # change the value part and remove the leading : character.

   if2name=
   if2addr=
   ```

```
: if4name=
: if4addr=gate3-$HOSTNAME
# If this host has more than 8 network interfaces, set the number of
# interfaces that the network startup script will configure.

: if_num=8
```

In this file, you configure the private (heartbeat) interface, but not the active public interface. The IRIS FailSafe software configures the active public interface. Since this configuration is active / standby, these are the only two interfaces.

Note: For an active / standby configuration, you must explicitly put in an entry for *if2addr/if2name* in the primary node's *netif.options* file so that it is not configured by default as gate-$HOSTNAME.

4. Set the backup node's *netif.options* file as follows:

```
if1name=ec3
if1addr=$HOSTNAME
if2name=ec0
if2addr=$HOSTNAME-2
```

In this file, you can configure the public interface and private interface (heartbeat) on the secondary node. You should not configure the public interface on the primary node; the IRIS FailSafe software configures the primary node's public interface.

Note: If you clone disks and do not configure the servers separately, you must set addresses separately for the second node as part of the adaptation of the disk.

5. Set */etc/config/routed.options* so that the routes are not shown over the private network:

```
-h -q
```

This option is required for IRIS FailSafe to function correctly.

6. Reboot both servers to put the network configuration into effect.

7. To configure an e-mail interface to receive notification of cluster transitions, set up an e-mail alias that includes a user on each server and at least one user outside the IRIS FailSafe cluster.

8. Notice that the IRIS FailSafe system is *chkconfig*'d off by default:

```
Flag                 State
====                 =====

autoconfig_ipaddress off
automount            off
gated                off
failsafe             off
lockd                on
mrouted              off
named                off
ns_httpd             off
network              on
nfs                  on
rarpd                off
routed               on
rtnetd               off
rwhod                off
timed                on
timeslave            off
verbose              off
vswap                off
xlv                  on
yp                   off
ypmaster             off
ypserv               off
```

Caution: Note that *yp* is off. Do not run *yp*.

Testing the Networks

For either type of IRIS FailSafe configuration, test the networks by following these steps:

1. To test the private (heartbeat) network from the primary server, enter

```
/usr/etc/ping xfs-ha6
```

where xfs-ha6 is the private IP address of the secondary server.

Typical ping output should appear, for example:

```
PING xfs-ha6.engr.sgi.com (192.48.165.94): 56 data bytes
64 bytes from 192.48.165.94: icmp_seq=0 ttl=254 time=3 ms
64 bytes from 192.48.165.94: icmp_seq=1 ttl=254 time=2 ms
64 bytes from 192.48.165.94: icmp_seq=2 ttl=254 time=2 ms
```

2. To test the private (heartbeat) network from the second server (xfs-ha6) in a dual-active configuration, enter, for example:

   ```
   /usr/etc/ping  xfs-ha5
   ```

3. To test the public network connection from the first server (xfs-ha5) in a dual-active configuration, the IP address (for example, *stocks*) must be *ifconfig*'d up. From another workstation or server on the public network, enter

   ```
   /usr/etc/ping stocks
   ```

4. After you test the public network connection, *ifconfigure* this interface down.
5. If the other node is attached to the same public network, enter

   ```
   /usr/etc/ping xfs-ha5-2
   ```

6. Repeat this *ping* process for the interface of the other server.

Configuring the Netscape Server Option

IRIS FailSafe provides failover protection for Netscape server

- documents (HTML pages), stored in the document root
- server configuration information, stored in the server root
- accounting files, also stored in the server root

Configuring the software varies for the IRIS FailSafe dual-active and active/standby configurations and is explained in separate sections.

Note: The IRIS FailSafe software supports the Netscape Communications Server. For another Web server, you must modify the recovery script *webserver* in the */var/ha/resources* directory.

Configuring a Netscape Server for a Dual-Active Configuration

You must configure two different Web servers on each node in an IRIS FailSafe dual-active system. Each Web server must have a different document root, for example, *stocks* serving stocks pages and *bonds* serving bonds pages.

In order for the Web server to distinguish requests to the two Web servers, each must have a unique IP address. In turn, each Netscape server has its own configuration information in a separate server root directory. In normal operation, only one server is active on each node. However, you must configure for two servers to cover the case in which a node fails and both Web servers run on the surviving node. This section explains the process, giving naming conventions.

Figure 3-3 diagrams shared storage and failover for Netscape on a dual-active IRIS FailSafe configuration.

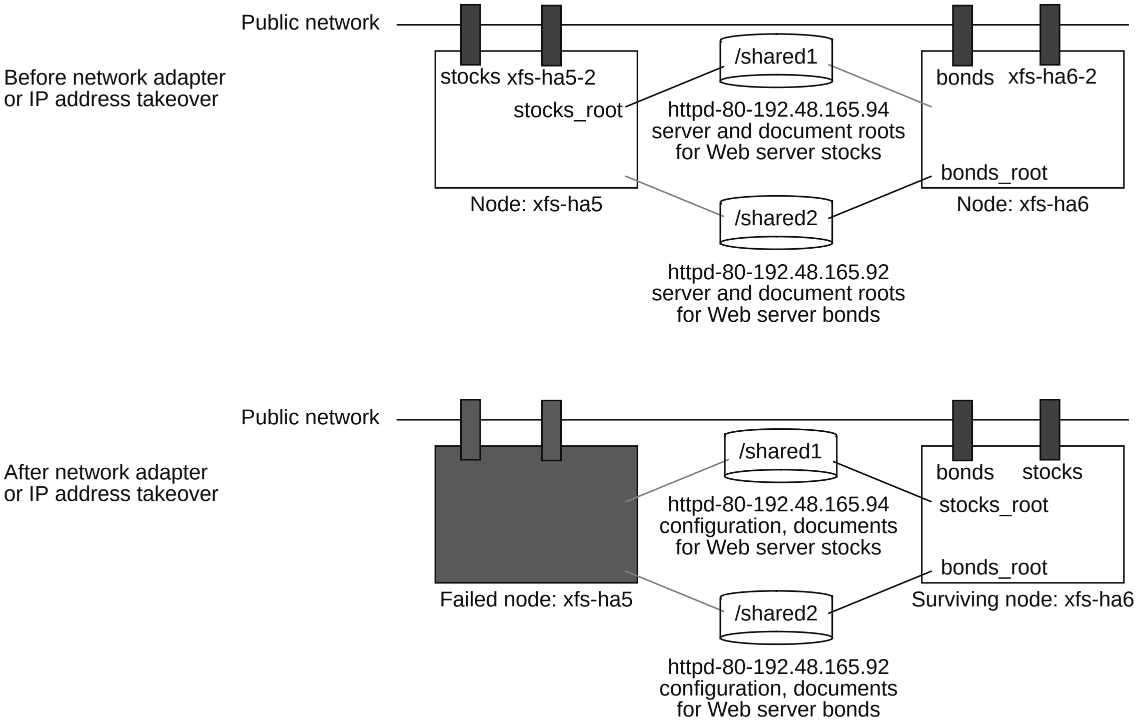


Figure 3-3 Dual-Active Configuration: Netscape Failover Example

When you install a Netscape server, you specify the root directory in which this information is stored. The default is */usr/ns-home*. Each server has a subdirectory under this directory, corresponding to its IP address; for example, the server root directory for the bonds Web server is */usr/ns-home/httpd-80.192.48.165.92*.

In a degraded configuration with both Web servers on the surviving node, this arrangement ensures that all requests to *stocks* are sent to its server, even though it is running on the same node as *bonds*. The two Web servers on each node must have different document roots. Each Web server's document root is on a separate shared filesystem. For example, *stocks* normally uses */shared1*, and *bonds* normally uses */shared2*.

If the Web server fails on one node, the surviving node takes over the IP address and the shared disk containing the failed node's Web server root and document root, and runs the appropriate */etc/init.d/ns_httpd* startup script. This script starts all Web servers, including those that are already running (such as the Web server on the surviving node). Ignore the warnings that these Web servers are already up.

To configure the Netscape Communications Server for an dual-active configuration, follow these steps:

1. Because Netscape server installation requires that the interface be accessible from a Netscape browser, *ifconfig* the interface to the public network up.
2. Make sure both interfaces are configured, as explained in "Configuring Networks for a Dual-Active Configuration," earlier in this chapter.
3. To install the first Web server (*stocks*), enter

   ```
   cd /usr/netscape/httpd/install; ./ns-setup
   ```
4. Start the Netscape browser on a workstation and open the Netscape server's configuration page.
5. Click the *Server Config* button and enter information according to the prompts. For example, for the system diagrammed in Figure 3-5, the server information is as follows:
   - name (*servername.domainname*): *stocks.companyname.com*
   - IP address for *stocks*: 192.48.165.94
   - accessible on: port *80* (the default port)

     Check */etc/services* to make sure the port you want is not already in use. If you choose the default port, the URL to your home page will be *http://.www.servername.domainname*. If you choose a different port, the URL to your home page will be *http:./.www.servername.domainname:portnumber*.
   - installed to directory */usr/ns-home*
   - errors recorded to a file in the server root

6. Click the *Document Config* button and enter information according to the prompts. Entries for the system diagrammed in Figure 3-4 are as follows:
   - server looks for documents in */usr/ns-home/stocks_root*
   - server looks for the documents *index.html* and *home.html* in directories
7. Click the *Admin Config* button and enter information according to the prompts. Exit the page.
8. Copy the contents of the server's root (*/usr/ns-home/httpd-80.192.48.165.94*) and document root (*/usr/hs-home/stocks_root*) to the shared filesystem. Replace the original directory with symbolic links to the directories.
9. To install the second Web server (*bonds*) on this node, enter

   ```
   cd /usr/netscape/httpd/install; ./ns-setup
   ```
10. Click the *Server Config* button; enter information for the second Web server (*bonds*):
    - name: make sure it matches the name of the second server; for example, *bonds.companyname.com*
    - IP address: make sure it is different from the IP address for the first Web server
    - all other parameters are the same as for the first Web server
11. Click the *Document Config* button; enter information for the second Web server (*bonds*):
    - server looks for documents in */usr/ns-home/bonds_root*
    - server looks for the documents *index.html* and *home.html* in directories
12. Click the *Admin Config* button and enter information according to the prompts, as for the first Web server.
13. Copy the contents of the server and document root directories to the shared filesystem used by *bonds*. Replace the original directories with symbolic links.

Configuring a Netscape Server for a Dual-Active Configuration Serving the Same Document Root

A special case of the dual-active configuration is one in which the two servers serve the same set of documents. Figure 3-4 diagrams failover for the Netscape server on this configuration.

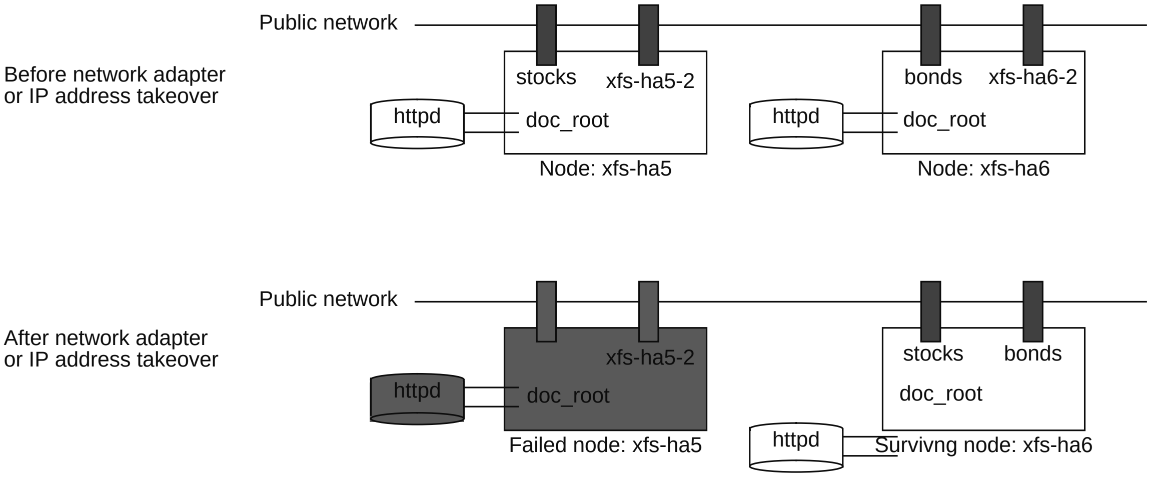


Figure 3-4 Dual-Active Configuration Serving the Same Document Root: Netscape Failover Example

Notice that both filesystems have the same document root, which is replicated across the two nodes.

You must make two copies of the document root and configuration information, because a filesystem cannot be mounted simultaneously on both nodes. Because the information is not shared in this case, the copies do not need to be on a shared disk.

Since each node serves the same documents, it is unnecessary to distinguish requests for the different servers. Thus, you do not need to specify a server's IP address during Netscape server configuration. A single server can handle all Web requests, even in case of failover.

Configuring a Netscape Server for an Active/Standby Configuration

Figure 3-5 diagrams shared storage and failover for Netscape servers on an active/standby IRIS FailSafe configuration.

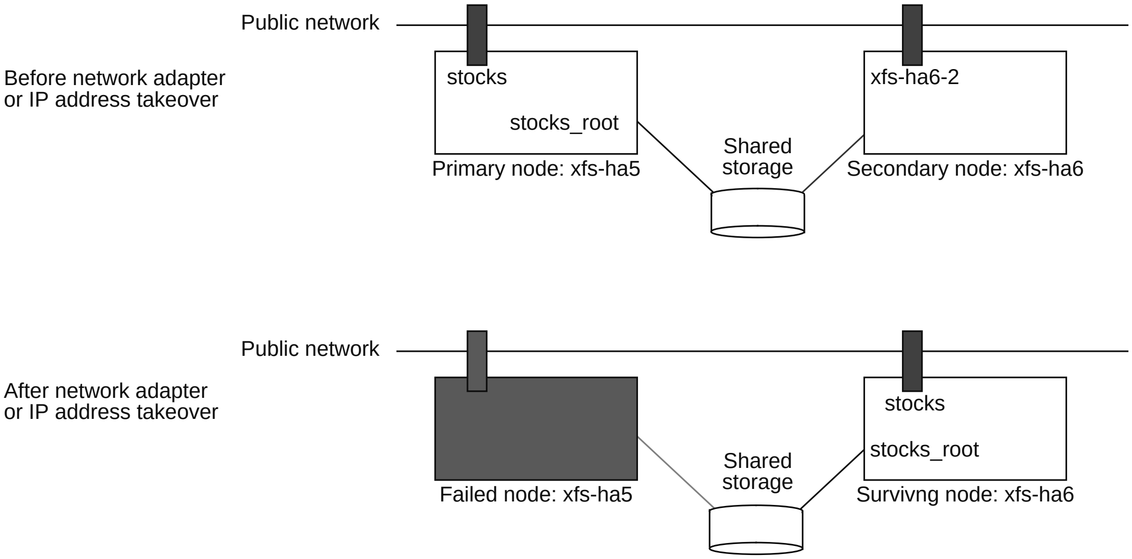


Figure 3-5 Active/Standby Configuration: Netscape Failover Example

For failover on an active/standby IRIS FailSafe system, the server root directory should be on the shared filesystem. If the secondary node detects that the primary server is not responding, it restarts the primary node and takes over the shared disk, bringing over the state of the Web server. The surviving node takes over the failed node’s service IP address.

The Netscape server has an installation form accessible through the browser.

1. Because Netscape installation requires that the interface be accessible from a Netscape browser, *ifconfig* the interface to the public network up:
2. To configure the Netscape server for an active/standby configuration, enter

   ```
   cd /usr/netscape/httpd/install; ./ns-setup
   ```

3. Start the Netscape browser on a workstation and open the Netscape server's configuration page.
4. Click the *Server Config* button and enter information according to the prompts. For example, for the system diagrammed in Figure 3-4, the server information is as follows:
    - name (*servername.domainname*): *stocks.companyname.com*
    - accessible on: port *80* (the default port)

      Check */etc/services* to make sure the port you want is not already in use. If you choose the default port, the URL to your home page will be *http:./.www.servername.domainname*. If you choose a different port, the URL to your home page will be *http://.www.servername.domainname:portnumber.*

      Note: To configure multiple Web servers, use the same IP address and different port numbers.
    - installed to directory */usr/ns-home*
    - errors recorded to a file in the server root
5. Click the *Document Config* button and enter information according to the prompts. Entries for the system diagrammed in Figure 3-4 are as follows:
    - server looks for documents in */usr/ns-home/stocks_root* (or use a symbolic link to a directory on the shared filesystem)
    - server looks for the documents *index.html* and *home.html* in directories
6. Click the *Admin Config* button and enter information according to the prompts.
7. Copy the contents of the server and document root directories to the shared filesystem. Replace the original directories with symbolic links to the directories on the shared filesystem.

Web Server Usage Accounting

Most Web sites run Web usage accounting. One common way of accounting for Web usage is to parse the access log files. These files are normally held within the *httpd* configuration directory tree (for example, */usr/ns-home/httpd-80/logs*).

If Web accounting is to be accurate, the log files must be failed over at the same time as the main Web service. If your Web configuration files are on the shared file system, they are automatically failed over, because the logs are in a subdirectory within the Web configuration directory.

Note that the log file also records accesses made by the IRIS FailSafe software monitors. Therefore, you must subtract them from the total number of hits to get an accurate count. An easy way to do this is to eliminate all accesses made from the other node in the cluster. However, doing so also eliminates accesses made by any users on the other node in the cluster. Because these users are servers, removing these accesses should present no serious problems.

Configuring the NFS Server Option

IRIS FailSafe provides failover protection for NFS server:

- filesystems: all parts, including files and directories
- file-locking information, as supported by *rpc.lockd/rpc.statd*

Software configurations differ for the two possible IRIS FailSafe systems, dual-active and active / standby. They are explained in separate sections.

Configuring NFS for a Dual-Active Configuration

In an IRIS FailSafe dual-active system, both NFS server nodes export NFS filesystems. Each node also provides backup service if the other node fails.

In order for the IRIS FailSafe system to distinguish requests destined for the two nodes, each node must have two public network adapters. Consequently, each server node is assigned two public interface names with corresponding IP addresses. This section explains the process, giving naming conventions.

Figure 3-6 diagrams the shared storage and failover process for NFS in a dual-active IRIS FailSafe configuration. Note that in this system, neither node can be considered primary or secondary. Rather, each is primary and secondary, depending on the specific NFS filesystem.

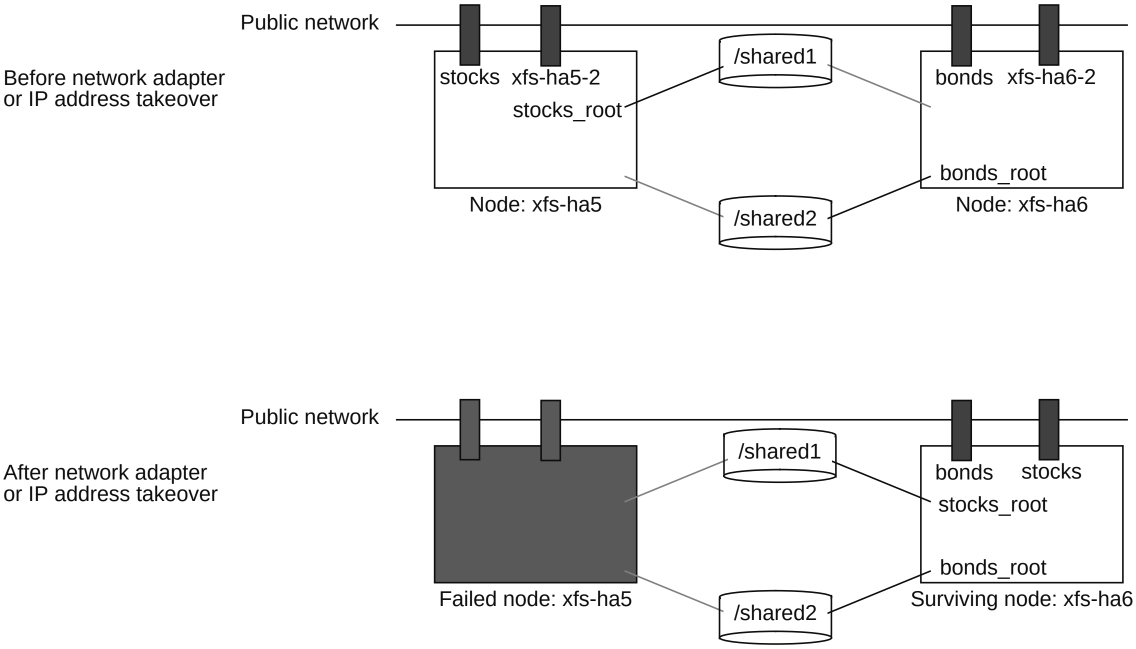


Figure 3-6 Dual-Active Configuration: NFS Failover Example

When you set up the various IRIS FailSafe NFS filesystems on each server, make sure that they use different mount points. For example, do not choose the same mount point */shared* if each node is primary to its own filesystem.

In the degraded state, in which all NFS filesystems are owned by the surviving node, that node must be able to concurrently service requests for all NFS filesystems. If the mount point directories are different, such as */shared1* and */shared2,* the surviving node can ensure that all NFS requests sent to *stocks* are handled properly, even though *bonds* is being offered by the same node.

When the NFS server on one node fails, the surviving node takes over NFS file-locking information from the failed node. IRIS FailSafe stores the NFS locking state for each node in a single directory on one shared filesystem. Consequently, a dual-active configuration contains two such directories, one for each node.

For example, if *stocks* is primary to the NFS filesystem */shared1* and *bonds* is primary to the NFS filesystems */shared2* and */shared3*, the two directories */shared1/statmon* and */shared2/statmon* should be created. The directory */shared1/statmon* stores NFS lock information that allows *bonds* to recover NFS locks for the */shared1* filesystem if *stocks* fails. The directory */shared2/statmon* stores NFS lock information that allows *stocks* to recover NFS locks for */shared2* and */shared3* if *bonds* fails.

Configuring NFS for an Active/Standby Configuration

In an IRIS FailSafe active / standby system, one NFS server node is configured to offer NFS exported filesystems. The other node provides backup service if the first node fails. In this configuration, each node requires only one public network adapter. Thus, one public interface name with corresponding IP address is assigned to each server. Figure 3-7 diagrams shared storage and failover for NFS in an active / standby IRIS FailSafe configuration.

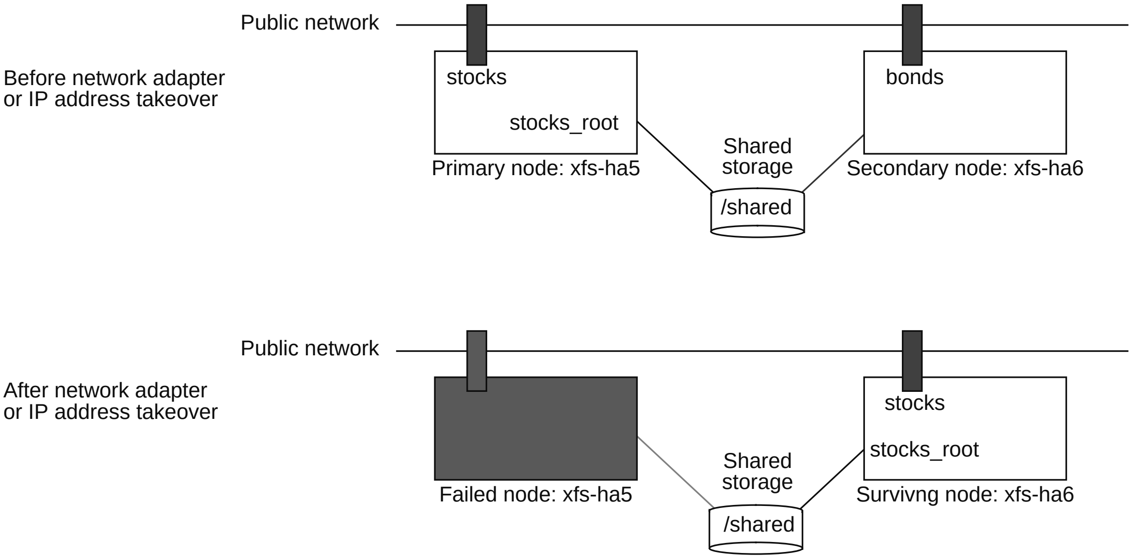


Figure 3-7 Active / Standby Configuration: Netscape Failover Example

The configuration process for the active / standby system is identical to that for the dual-active configuration, except that only one server normally offers NFS service in the active / standby configuration.

Configuring Shared Filesystems

The IRIS FailSafe software supports the automatic failover of filesystems on shared disk storage. A shared filesystem is a filesystem that is created on disks (either in CHALLENGE Vault or CHALLENGE RAID storage systems) that are shared between two CHALLENGE servers.

You can work with filesystems and volumes on shared disks as you would work with other disks. You can create volumes, create filesystems, mount filesystems, and son on, from either node. For a complete discussion on the use of the XFS filesystem and XLV volumes, consult the *Disks and Filesystems volume of the IRIX Admin Guide* (007-2825-001).

The following are special issues that you need to be aware of when you are working with shared filesystems in an IRIS FailSafe cluster:

1. You can access a shared filesystem from either node. However, you must not simultaneously mount the same filesystem from both nodes. Doing so causes data corruption.Therefore, before you mount or otherwise use a shared filesystem, make sure that it is not being used by the other node.
2. All shared filesystems must be created on XLV volumes.
3. Work with the volumes on a shared disk from only one node in the cluster. Each XLV volume records the "owning node," which is initially the node from which the volume was created. During a failover, the new owner of the volume changes the "owning node" field of the volume.

   In an IRIS FailSafe configuration, the default behavior of *xlv_assemble*(1M) is to assemble only volumes that are owned by the node. Note that this behavior is different from earlier releases of XLV that used to assemble all available volumes, whether or not they are owned by the node.

   You can initially create all the volumes on one node and then selectively change the "owning node" to the other node afterwards using *xlv_mgr*.
4. XLV allows multiple volumes to be created on the same physical disk.In an IRIS FailSafe environment, if you create more than one volume on a single disk, they must all be owned by the same node. For example, if a disk has two partitions that are part of two XLV volumes, both XLV volumes must be owned by the same node.
5. For reliability, the shared filesystem in an IRIS FailSafe cluster must be created on either mirrored disks (via the XLV plexing software), or on the CHALLENGE RAID storage system. For details on configuring RAID, see the *CHALLENGE RAID Owner's Guide* (007-2532-00x).

Chapter 4

Creating the Configuration File

The IRIS FailSafe system uses a configuration file, *ha.conf*, to determine system resources, such as the primary and secondary (active and standby) server names, network interface names and addresses, and shared storage and filesystem parameters. (For more information on the configuration file, see the *ha.conf* reference page.)

The IRIS FailSafe system software includes several versions of the configuration file for various configurations: dual-active Web server cluster, active / standby Web Server cluster, dual-active NFS, active / standby NFS, and a combination of active / standby Web Server and dual-active NFS. These files are in */var/ha/samples*.

Choose one of these example configuration files, adapt it for your IRIS FailSafe system, and save it as */var/ha/ha.conf*. The IRIS FailSafe system uses the *ha.conf* file to determine system resources, such as the primary and secondary (active and standby) server names, network interface names and addresses, and shared storage and filesystem parameters.

This chapter explains how to adapt a version of the configuration file and customize it for your IRIS FailSafe system. This process involves

- preparing for configuration
- editing the blocks of the configuration file
- saving the configuration file

Preparing for Configuration

From */var/ha/samples*, select the sample configuration file that best corresponds to your configuration.

Before you begin modifying a sample configuration file, have ready the following information:

- E-mail address where the IRIS FailSafe software should send e-mail messages when failures occur
- For each node:
  - hostname
  - IP address of each public network interface
  - IP address of the private network interface
  - for nodes with two public network interfaces, determination of which is active (primary) and which is standby (secondary) on each node
- The hostname of the node on which the Web server is running, if present
- The names of the server's shared volumes

Figure 4-1 shows the correspondence between the IRIS FailSafe system hardware and the naming conventions used in two configuration file examples, *ha.conf.nfs_dual_active* and *ha.conf.web_active_standby.*

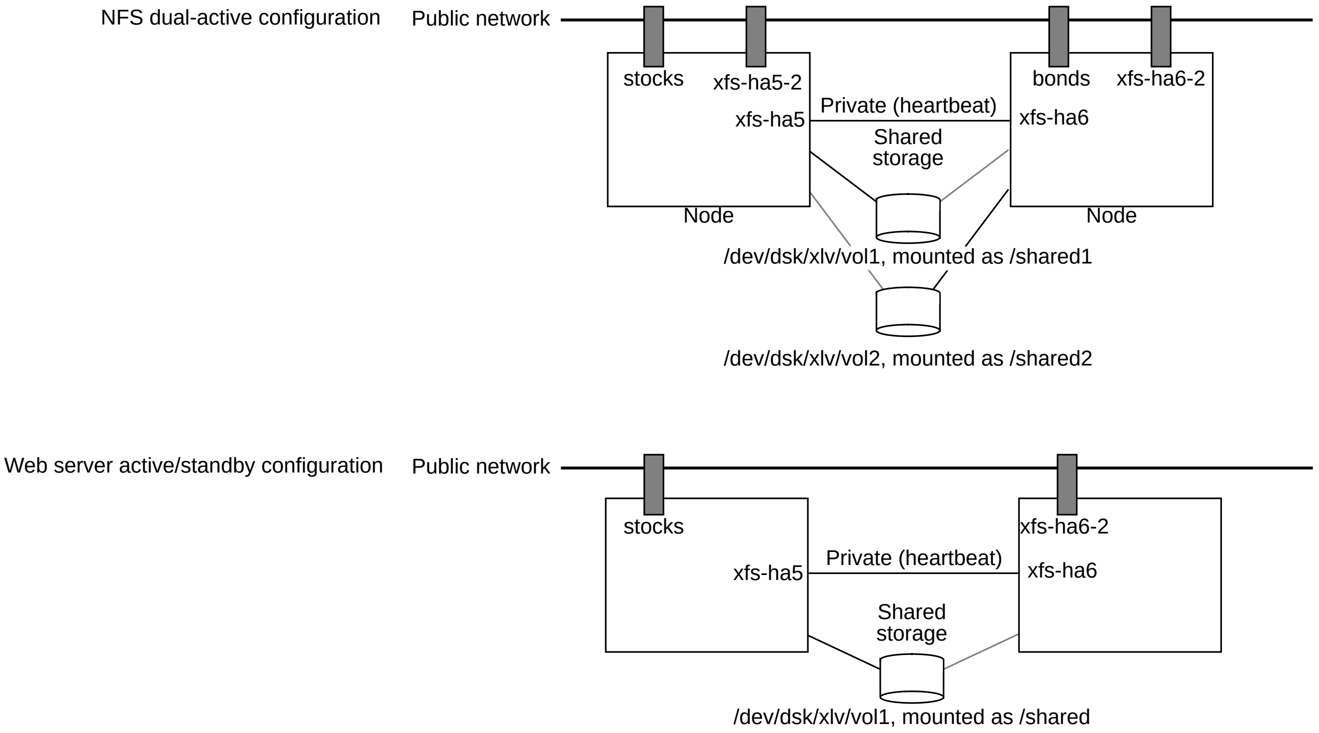

NFS dual-active configuration
Public network
stocks
xfs-ha5-2
bonds
xfs-ha6-2
Private (heartbeat)
xfs-ha5
xfs-ha6
Shared
storage
Node
Node
/dev/dsk/xlv/vol1, mounted as /shared1
/dev/dsk/xlv/vol2, mounted as /shared2
Web server active/standby configuration
Public network
stocks
xfs-ha6-2
Private (heartbeat)
xfs-ha5
xfs-ha6
Shared
storage
/dev/dsk/xlv/vol1, mounted as /shared

Figure 4-1 Configuration File Naming Conventions

In the dual-active NFS configuration, the cluster as a whole exports two filesystems, *stocks:/shared1* and *bonds:/shared2*. Clients access the mount points under the advertised names of *stocks* and *bonds*. In normal operation, both servers are active and service requests to different filesystems. The node xfs-ha5 services all requests to *stocks*; xfs-ha6 services requests to *bonds*. If xfs-ha5 fails, xfs-ha6 takes over the IP address for *stocks* and exports */shared1*, so that both filesystems are still available. Likewise, xfs-ha5 takes over the IP address of *bonds* and exports */shared2*, if xfs-ha6 should fail.

In the active / standby Web server configuration, the cluster has the external name of stocks. All documents are stored in a subdirectory of */shared*. the two nodes in the cluster are xfs-ha5 and xfs-ha6. Normally, the IP address *stocks* is exported by xfs-ha5. If this node fails, xfs-ha6 takes over the IP address of *stocks*, mounts the shared filesystem */shared*, and continues to service Web requests.

Editing the Blocks of the Configuration File

This section uses the example configuration files *ha.conf.nfs_dual_active* and *ha.conf.web_active_standby* to demonstrate the blocks or sections of an IRIS FailSafe configuration file.

Note: For convenience, the complete code for these samples is printed in Appendix D, "Sample Configuration Files."

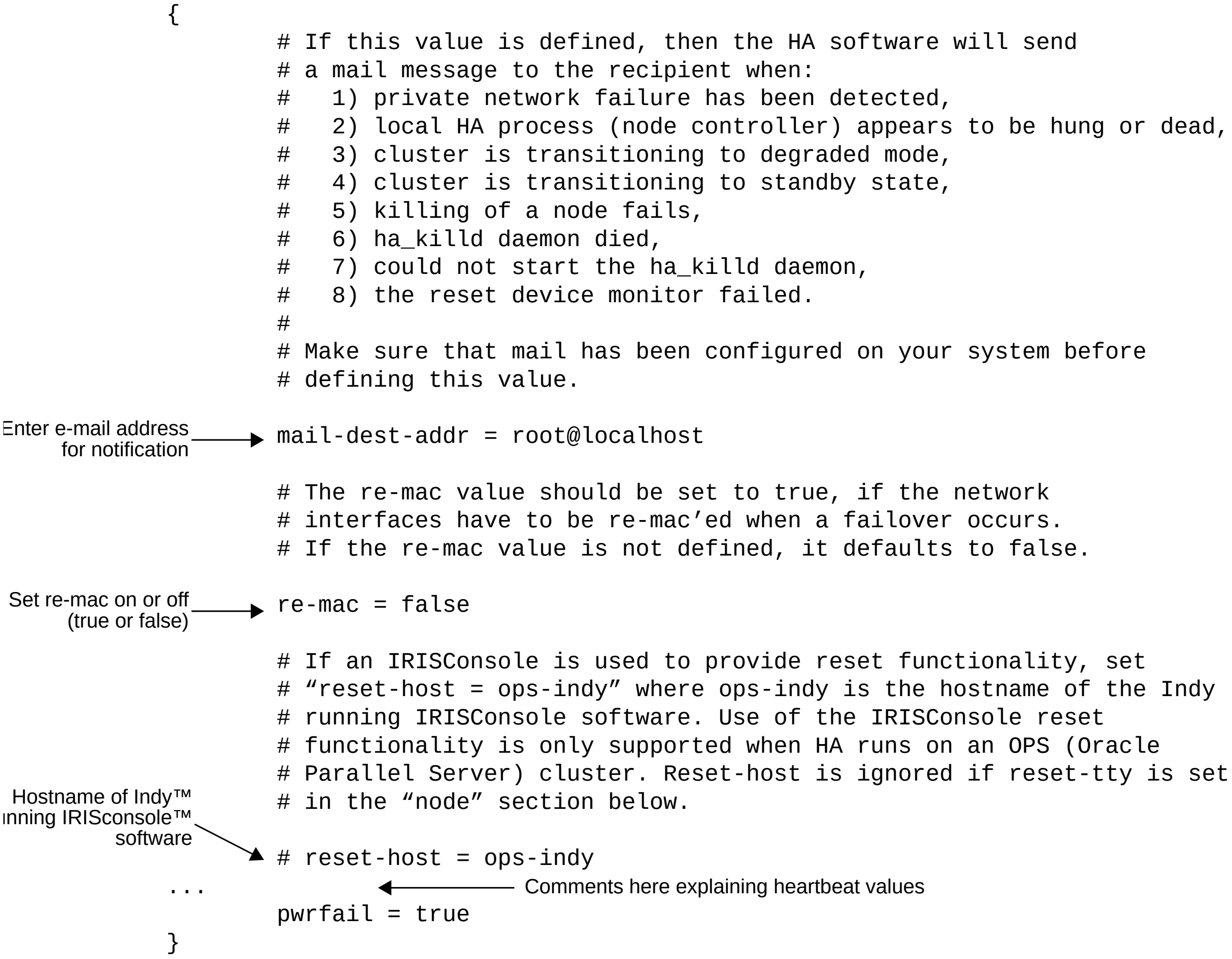
A configuration file includes the following:

- system configuration block
- node block: setup for all nodes
- filesystem block: sets the filesystems to be included in the IRIS FailSafe system, including filesystems on each node and, if present, NFS and Web server filesystems on each node
- application classes block: the application class *main* controls all system resources, such as interfaces and filesystems
- optional NFS blocks
- optional Web server blocks

Note: When you are naming entities in the IRIS FailSafe system, note the keywords in the configuration file; see Appendix C, "Keywords," for an alphabetical list.

System Configuration Block

Figure 4-2 shows the system configuration block for all sample configuration files.



```
{
        # If this value is defined, then the HA software will send
        # a mail message to the recipient when:
        #   1) private network failure has been detected,
        #   2) local HA process (node controller) appears to be hung or dead,
        #   3) cluster is transitioning to degraded mode,
        #   4) cluster is transitioning to standby state,
        #   5) killing of a node fails,
        #   6) ha_killd daemon died,
        #   7) could not start the ha_killd daemon,
        #   8) the reset device monitor failed.
        #
        # Make sure that mail has been configured on your system before
        # defining this value.

        mail-dest-addr = root@localhost

        # The re-mac value should be set to true, if the network
        # interfaces have to be re-mac'ed when a failover occurs.
        # If the re-mac value is not defined, it defaults to false.

        re-mac = false

        # If an IRISConsole is used to provide reset functionality, set
        # "reset-host = ops-indy" where ops-indy is the hostname of the Indy
        # running IRISConsole software. Use of the IRISConsole reset
        # functionality is only supported when HA runs on an OPS (Oracle
        # Parallel Server) cluster. Reset-host is ignored if reset-tty is set
        # in the "node" section below.

        # reset-host = ops-indy
    ...
        pwrfail = true
}
```

Figure 4-2 System Configuration Block

Node Block

The node section of the configuration file describes network interfaces to the servers in the cluster. Node settings vary, depending on whether you are using dual-active or active/standby configuration.

Dual-Active Configuration

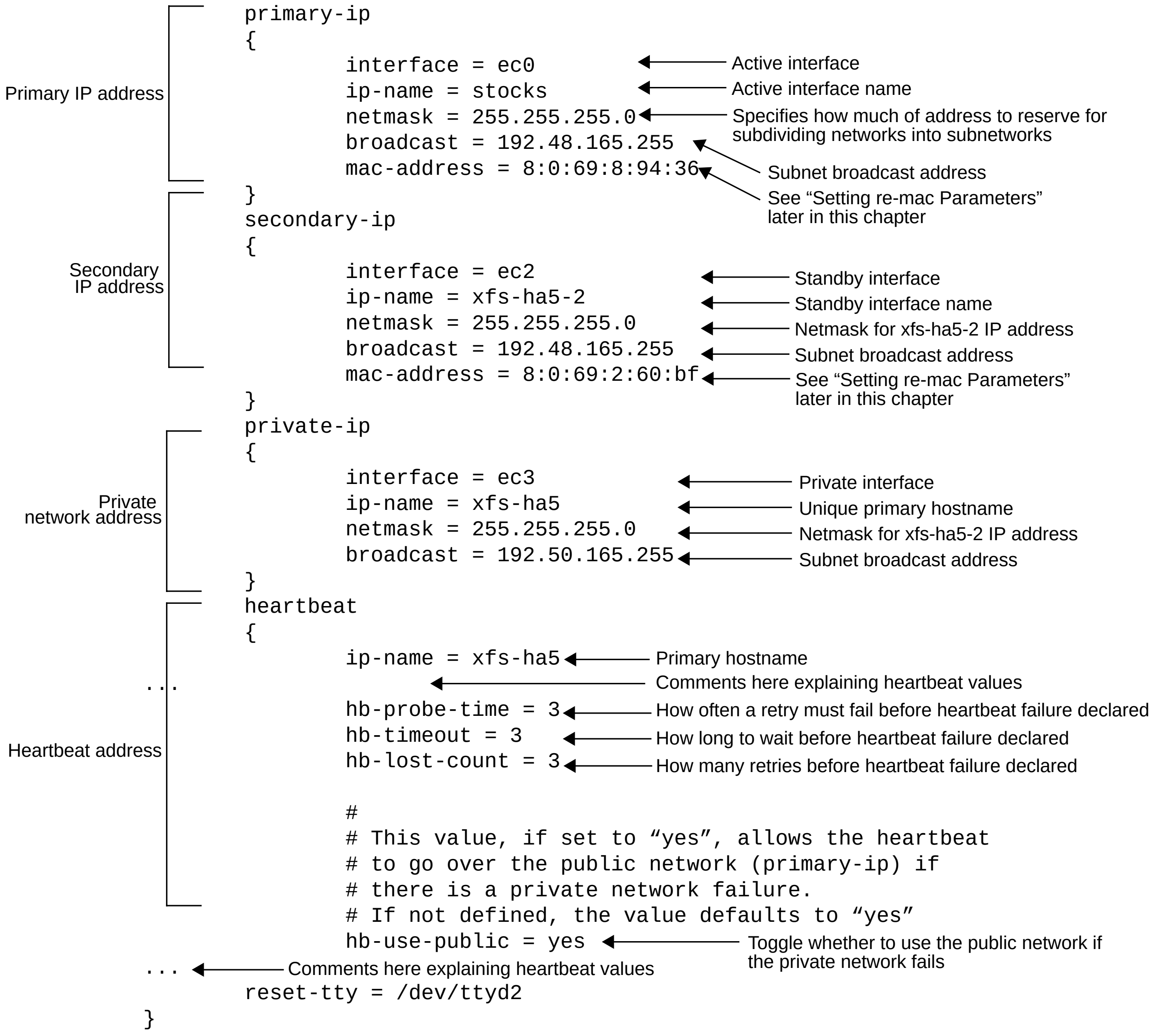
Each server in this configuration has three network interfaces. For example, for a CHALLENGE S server,

- **ec0** is the interface over which clients request services. It is associated with the IP address of the service, in this case, *stocks* or *bonds*. By default, this interface is not configured; IRIS FailSafe configures it based on the state of the cluster.
- **ec2** is a spare interface. A node uses this to service requests destined for the other node if it needs to take over the services offered by the other node. Normally, it is configured as $HOSTNAME-2. When it is used as a backup, it is configured as one of the service interfaces: *stocks* or *xfs-ha6*. The spare interface is present only in dual-active configurations.
- **ec3** is the network interface to a private network between the servers, which is used for keep-alive and other control messages.

For example, when both servers are up and exporting their own services, **ec0** on xfs-ha5 uses the IP address *stocks* and **ec0** on xfs-ha6 uses the IP address *bonds*. If xfs-ha5 fails and xfs-ha6 takes over its services, then **ec0** on xfs-ha6 still has its own IP address (*bonds*) and **ec2** on xfs-ha6 has the IP address *stocks*.

Figure 4-3 shows the node block for the first server in a dual-active cluster.

Note: In the dual-active configuration, referring to the servers and nodes as first and second does not imply primacy of one over the other.



```
        primary-ip
        {
                interface = ec0
                ip-name = stocks
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
                mac-address = 8:0:69:8:94:36
        }
        secondary-ip
        {
                interface = ec2
                ip-name = xfs-ha5-2
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
                mac-address = 8:0:69:2:60:bf
        }
        private-ip
        {
                interface = ec3
                ip-name = xfs-ha5
                netmask = 255.255.255.0
                broadcast = 192.50.165.255
        }
        heartbeat
        {
                ip-name = xfs-ha5
...
                hb-probe-time = 3
                hb-timeout = 3
                hb-lost-count = 3

                #
                # This value, if set to “yes”, allows the heartbeat
                # to go over the public network (primary-ip) if
                # there is a private network failure.
                # If not defined, the value defaults to “yes”
                hb-use-public = yes
...
        reset-tty = /dev/ttyd2
}
```

Active interface

Active interface name

Specifies how much of address to reserve for subdividing networks into subnetworks

Subnet broadcast address

See “Setting re-mac Parameters” later in this chapter

Standby interface

Standby interface name

Netmask for xfs-ha5-2 IP address

Subnet broadcast address

See “Setting re-mac Parameters” later in this chapter

Private interface

Unique primary hostname

Netmask for xfs-ha5-2 IP address

Subnet broadcast address

Primary hostname

Comments here explaining heartbeat values

How often a retry must fail before heartbeat failure declared

How long to wait before heartbeat failure declared

How many retries before heartbeat failure declared

Toggle whether to use the public network if the private network fails

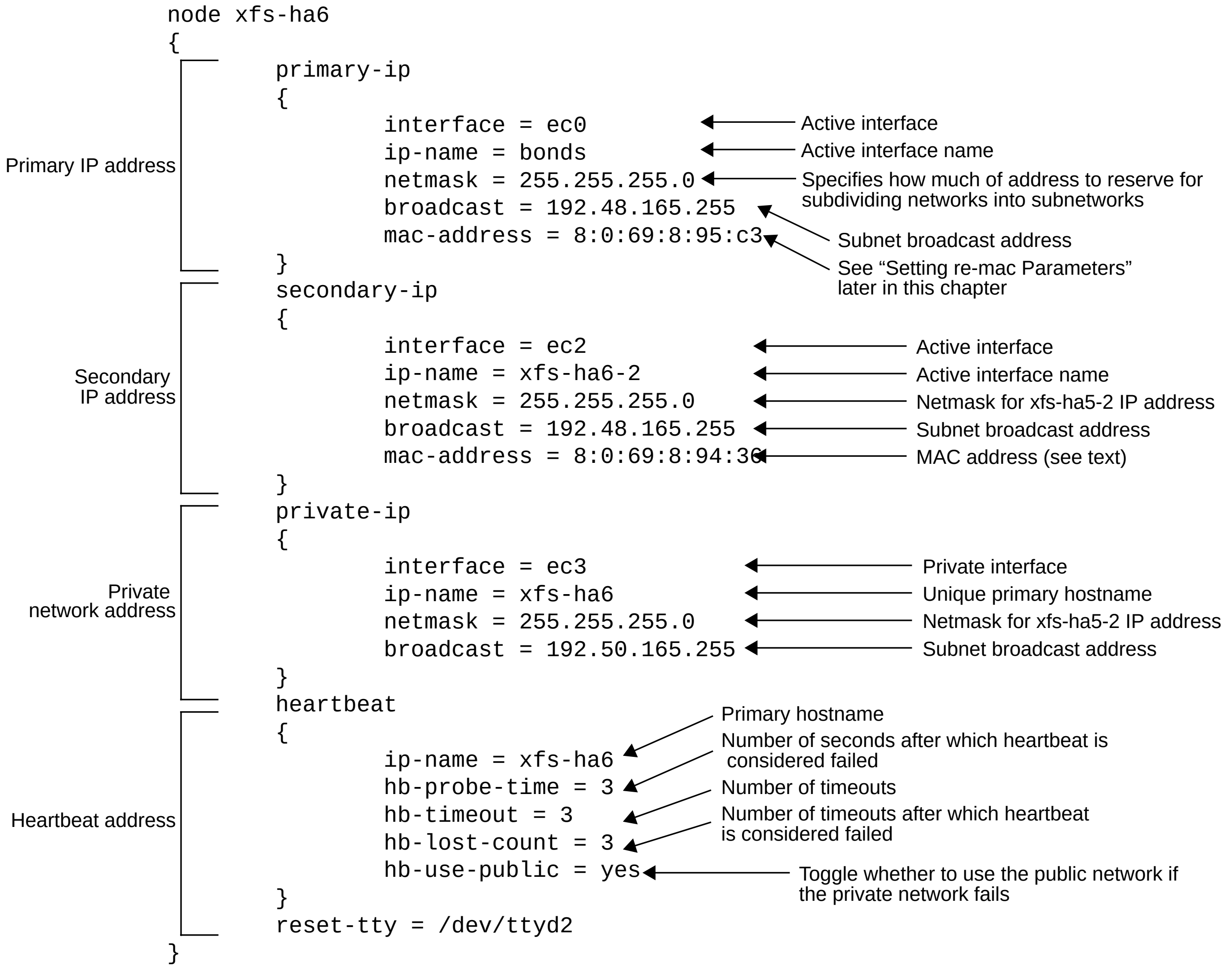
Comments here explaining heartbeat values

Figure 4-3 Dual-Active Configuration File Block: First Node

Note these values:

- The values specified in the *primary-ip*, *secondary-ip*, and *private-ip* are those passed to the *ifconfig* command. See this command's reference page for more details.
- The node label (xfs-ha5) must match the return value of the hostname on the server. It should match the private IP address *ip-name*.
- The "ip-name" value must be a name and not an address (such as 197.50.50.11). For the IP names (*ip-name*) for the primary and secondary IPs, use the network interface name. Reserve the server's name (hostname) for the private IP's *ip-name* field (see Figure 4-3).
- The reset-tty is the device name of the serial link connected to the remote power control unit system controller

Figure 4-4 shows the node block for the second server in a dual-active cluster.



```
node xfs-ha6
{
        primary-ip
        {
                interface = ec0
                ip-name = bonds
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
                mac-address = 8:0:69:8:95:c3
        }
        secondary-ip
        {
                interface = ec2
                ip-name = xfs-ha6-2
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
                mac-address = 8:0:69:8:94:36
        }
        private-ip
        {
                interface = ec3
                ip-name = xfs-ha6
                netmask = 255.255.255.0
                broadcast = 192.50.165.255
        }
        heartbeat
        {
                ip-name = xfs-ha6
                hb-probe-time = 3
                hb-timeout = 3
                hb-lost-count = 3
                hb-use-public = yes
        }
        reset-tty = /dev/ttyd2
}
```

Primary IP address
Active interface
Active interface name
Specifies how much of address to reserve for subdividing networks into subnetworks
Subnet broadcast address
See “Setting re-mac Parameters” later in this chapter
Secondary IP address
Active interface
Active interface name
Netmask for xfs-ha5-2 IP address
Subnet broadcast address
MAC address (see text)
Private network address
Private interface
Unique primary hostname
Netmask for xfs-ha5-2 IP address
Subnet broadcast address
Heartbeat address
Primary hostname
Number of seconds after which heartbeat is considered failed
Number of timeouts
Number of timeouts after which heartbeat is considered failed
Toggle whether to use the public network if the private network fails

Figure 4-4 Dual-Active Configuration File Block: Second Node

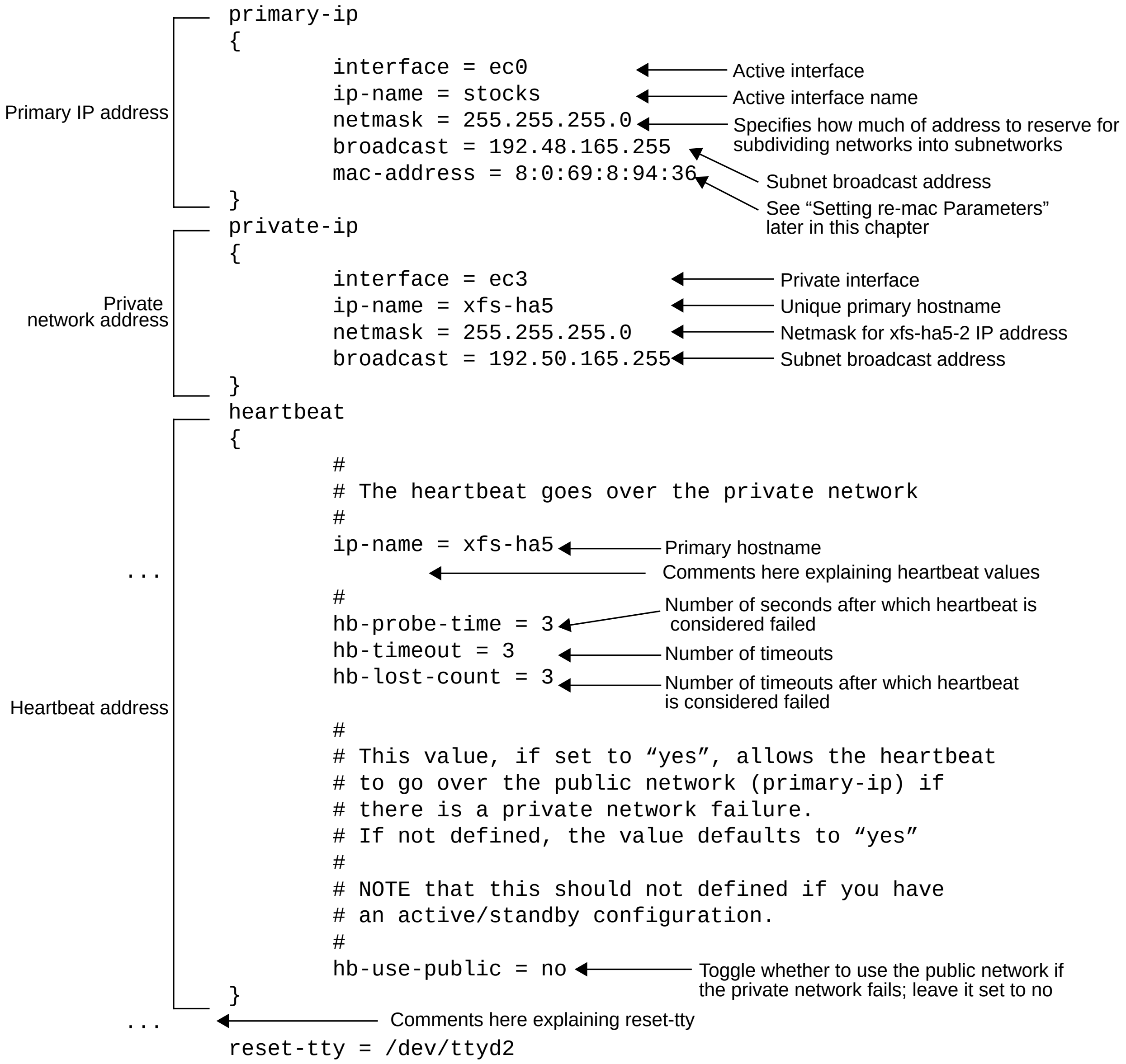
Active/Standby Configuration

Each server in this configuration has only two network interfaces, unlike the dual-active configuration, where each server has three network interfaces. For example, for a CHALLENGE S server,

- **ec0** is the interface over which clients request services. It is associated with the IP address of the service. By default, this interface is not configured; IRIS FailSafe configures it based on the state of the servers in the cluster.
- **ec3** is the network interface to a private network between the servers, which is used for keep-alive and other control messages.

Figure 4-5 shows the node block for the primary server in an active / standby configuration.

Note: For this configuration, the hb-use-public field must be set to “no.” the IRIS FailSafe software does not support the use of the public network a the heartbeat network in an active / standby configuration.

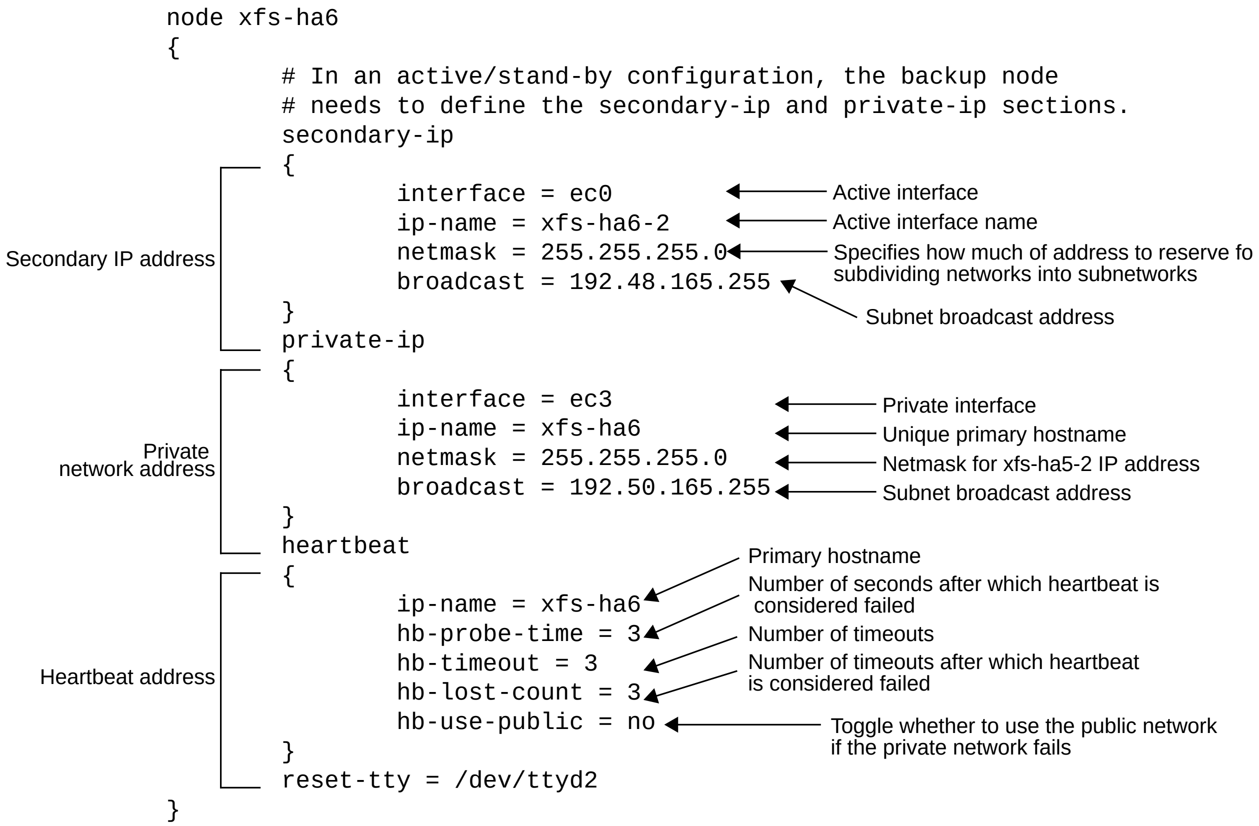


```
primary-ip
{
        interface = ec0
        ip-name = stocks
        netmask = 255.255.255.0
        broadcast = 192.48.165.255
        mac-address = 8:0:69:8:94:36
}
private-ip
{
        interface = ec3
        ip-name = xfs-ha5
        netmask = 255.255.255.0
        broadcast = 192.50.165.255
}
heartbeat
{
        #
        # The heartbeat goes over the private network
        #
        ip-name = xfs-ha5

        #
        hb-probe-time = 3
        hb-timeout = 3
        hb-lost-count = 3

        #
        # This value, if set to “yes”, allows the heartbeat
        # to go over the public network (primary-ip) if
        # there is a private network failure.
        # If not defined, the value defaults to “yes”
        #
        # NOTE that this should not defined if you have
        # an active/standby configuration.
        #
        hb-use-public = no
}
reset-tty = /dev/ttyd2
```

Primary IP address
Active interface
Active interface name
Specifies how much of address to reserve for subdividing networks into subnetworks
Subnet broadcast address
See “Setting re-mac Parameters” later in this chapter
Private network address
Private interface
Unique primary hostname
Netmask for xfs-ha5-2 IP address
Subnet broadcast address
...
Primary hostname
Comments here explaining heartbeat values
Number of seconds after which heartbeat is considered failed
Number of timeouts
Number of timeouts after which heartbeat is considered failed
Heartbeat address
Toggle whether to use the public network if the private network fails; leave it set to no
...
Comments here explaining reset-tty

Figure 4-5 Active/Standby Configuration File Block: Primary Node

Figure 4-6 shows the node block for the secondary server in an active/standby configuration.



```
node xfs-ha6
{
        # In an active/stand-by configuration, the backup node
        # needs to define the secondary-ip and private-ip sections.
        secondary-ip
        {
                interface = ec0
                ip-name = xfs-ha6-2
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
        }
        private-ip
        {
                interface = ec3
                ip-name = xfs-ha6
                netmask = 255.255.255.0
                broadcast = 192.50.165.255
        }
        heartbeat
        {
                ip-name = xfs-ha6
                hb-probe-time = 3
                hb-timeout = 3
                hb-lost-count = 3
                hb-use-public = no
        }
        reset-tty = /dev/ttyd2
}
```

Secondary IP address
Private network address
Heartbeat address
Active interface
Active interface name
Specifies how much of address to reserve fo subdividing networks into subnetworks
Subnet broadcast address
Private interface
Unique primary hostname
Netmask for xfs-ha5-2 IP address
Subnet broadcast address
Primary hostname
Number of seconds after which heartbeat is considered failed
Number of timeouts
Number of timeouts after which heartbeat is considered failed
Toggle whether to use the public network if the private network fails

Figure 4-6 Active/Standby Configuration File Block: Secondary Node

Notice that this block defines only the secondary IP and private IP addresses, in contrast to the block for the primary node, which defines the primary IP address as well.

Setting *re-mac* Parameters

The IRIS FailSafe software normally uses the *re-arp* protocol (re-arping) to change a server's IP address. The vast majority of TCP / IP implementations correctly support this protocol.

To support client systems that do not support the full IP *re-arp* protocol, the IRIS FailSafe software can also use *re-mac*, which changes the hardware Ethernet address of the surviving network adapter to those of the failed network adapter. MAC address failover is necessary for supporting NFS for PC clients.

Note: MAC address failover is supported on Ethernet networks only, not FDDI networks.

To support such a system, follow these steps:

1. Run *macconfig* to display the MAC addresses of all configured network interfaces in the node (including ec0). In the following *macconfig* example output, the Physical Address column lists the interface's MAC address.

   ```
   # macconfig
   Name      Physical Address    IP Address
   xpi0      10:0:96:20:98:12    195.50.165.11
   ec0       8:0:69:9:31:f8      192.48.165.89
   ```

   The command *macconfig* displays MAC addresses of network interfaces configured up (enabled or re-enabled) using *ifconfig*. If the interface is not configured, configure it with *ifconfig*. For example, **ec0** is the primary IP interface for the node in the following line:

   ```
   # ifconfig ec0 inet <ip-address> netmask <netmask> up
   ```

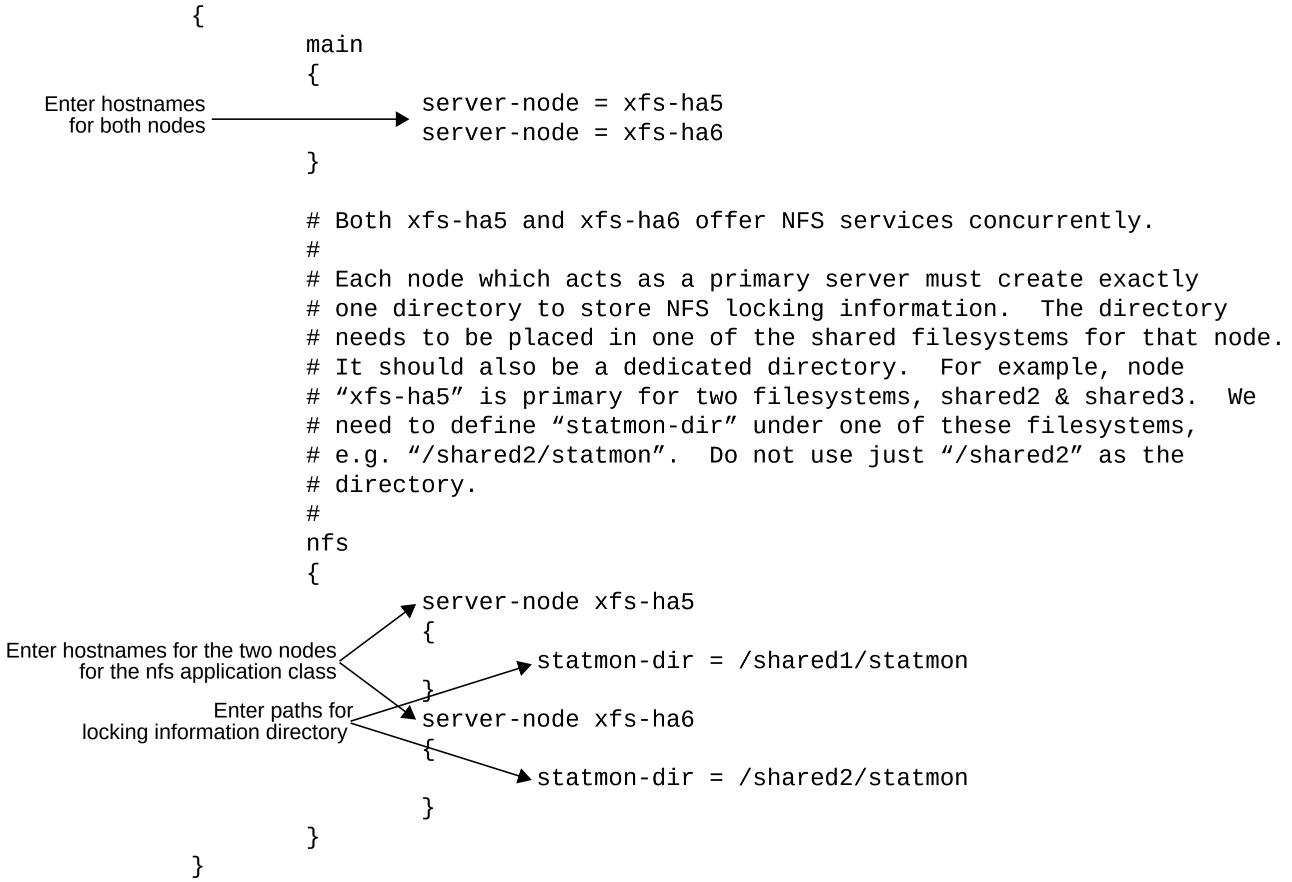
2. Enter the MAC addresses in the node block of the file, as explained in Figure 4-4 and Figure 4-5.

3. Set *re-mac* to TRUE in the system configuration block, as shown in Figure 4-2. This setting causes the IRIS FailSafe software to *re-mac* the interfaces in addition to *re-arp*ing them.

Application Classes Block

The application classes are IRIS FailSafe services provided by a node: main, NFS, and Web server. Each service is provided by at least one node (server-node).

Figure 4-7 shows the application classes block in a dual-active cluster.



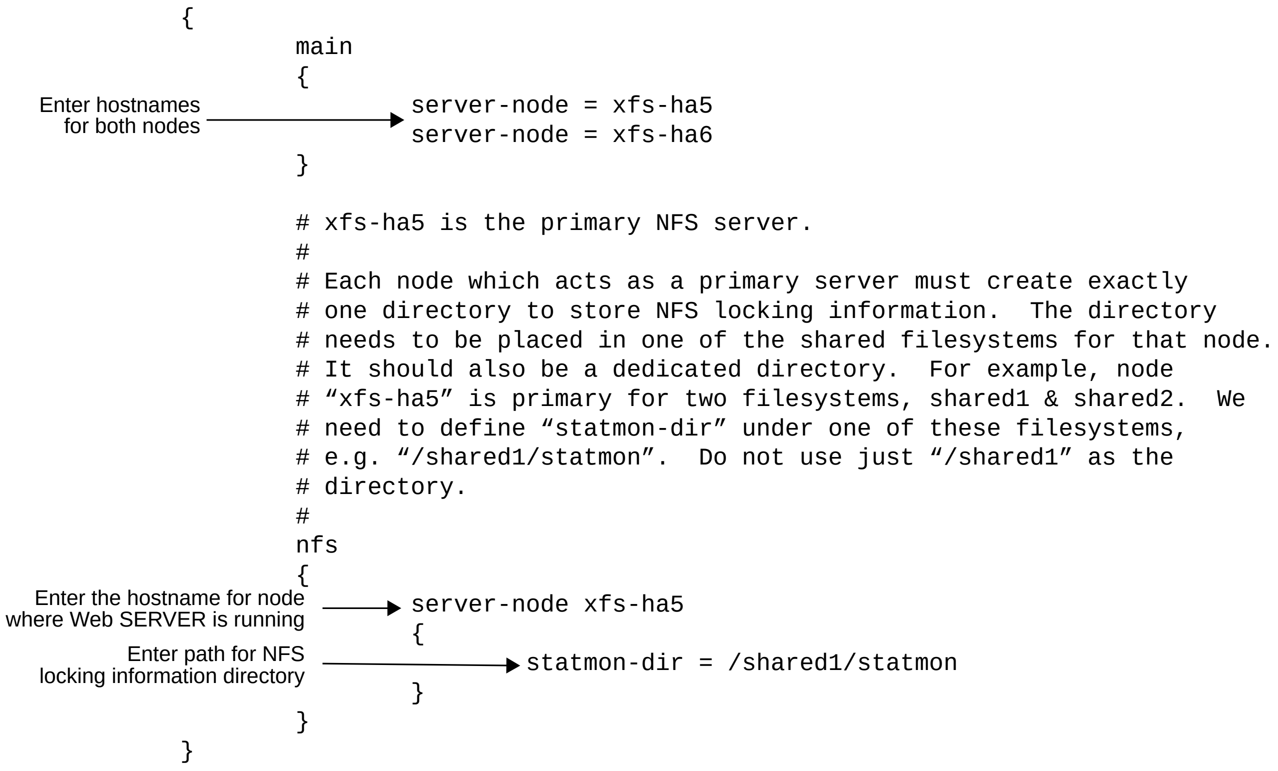
```
{
        main
        {
                server-node = xfs-ha5
                server-node = xfs-ha6
        }

        # Both xfs-ha5 and xfs-ha6 offer NFS services concurrently.
        #
        # Each node which acts as a primary server must create exactly
        # one directory to store NFS locking information.  The directory
        # needs to be placed in one of the shared filesystems for that node.
        # It should also be a dedicated directory.  For example, node
        # “xfs-ha5” is primary for two filesystems, shared2 & shared3.  We
        # need to define “statmon-dir” under one of these filesystems,
        # e.g. “/shared2/statmon”.  Do not use just “/shared2” as the
        # directory.
        #
        nfs
        {
                server-node xfs-ha5
                {
                        statmon-dir = /shared1/statmon
                }
                server-node xfs-ha6
                {
                        statmon-dir = /shared2/statmon
                }
        }
}
```

Enter hostnames for both nodes
Enter hostnames for the two nodes for the nfs application class
Enter paths for locking information directory

Figure 4-7 Dual-Active Configuration File Block: Application Classes

The directories */shared1* and */shared2* are shared filesystems that are defined later in the configuration file.

Figure 4-8 shows the application classes block in an active / standby NFS configuration.

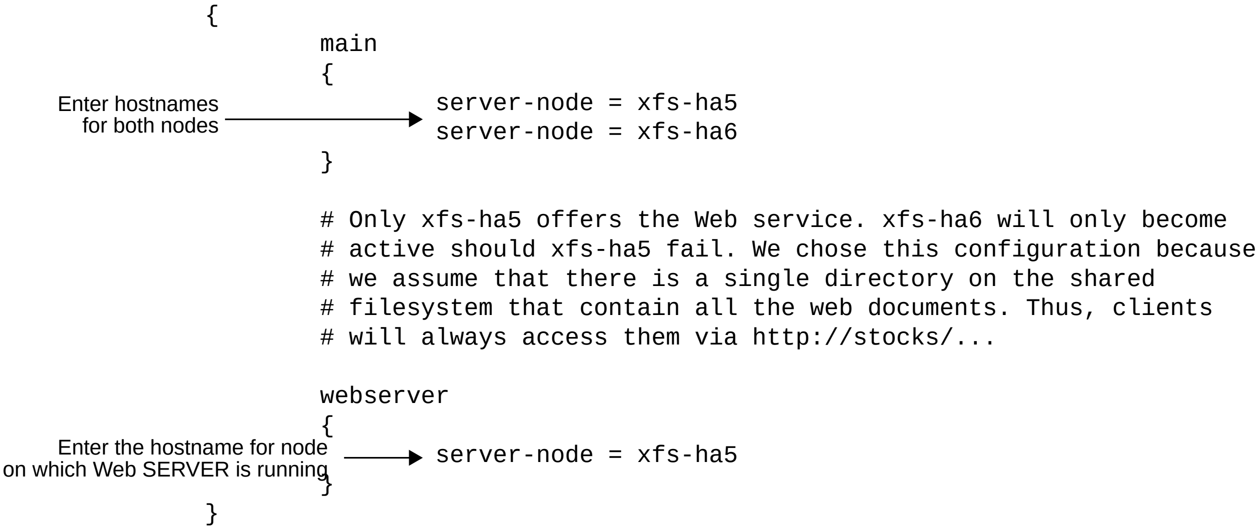


```
{
        main
        {
                server-node = xfs-ha5
                server-node = xfs-ha6
        }

        # xfs-ha5 is the primary NFS server.
        #
        # Each node which acts as a primary server must create exactly
        # one directory to store NFS locking information.  The directory
        # needs to be placed in one of the shared filesystems for that node.
        # It should also be a dedicated directory.  For example, node
        # “xfs-ha5” is primary for two filesystems, shared1 & shared2.  We
        # need to define “statmon-dir” under one of these filesystems,
        # e.g. “/shared1/statmon”.  Do not use just “/shared1” as the
        # directory.
        #
        nfs
        {
                server-node xfs-ha5
                {
                        statmon-dir = /shared1/statmon
                }
        }
}
```

Enter hostnames for both nodes
Enter the hostname for node where Web SERVER is running
Enter path for NFS locking information directory

Figure 4-8 NFS Active / Standby Configuration File Block: Application Classes

Figure 4-9 shows the application classes block in an active / standby Web server configuration.



```
{
        main
        {
                server-node = xfs-ha5
                server-node = xfs-ha6
        }

        # Only xfs-ha5 offers the Web service. xfs-ha6 will only become
        # active should xfs-ha5 fail. We chose this configuration because
        # we assume that there is a single directory on the shared
        # filesystem that contain all the web documents. Thus, clients
        # will always access them via http://stocks/...

        webserver
        {
                server-node = xfs-ha5
        }
}
```

Enter hostnames
for both nodes
Enter the hostname for node
on which Web SERVER is running

Figure 4-9 Web Server Active / Standby Configuration File Block: Application Classes

Disks/Filesystem Block

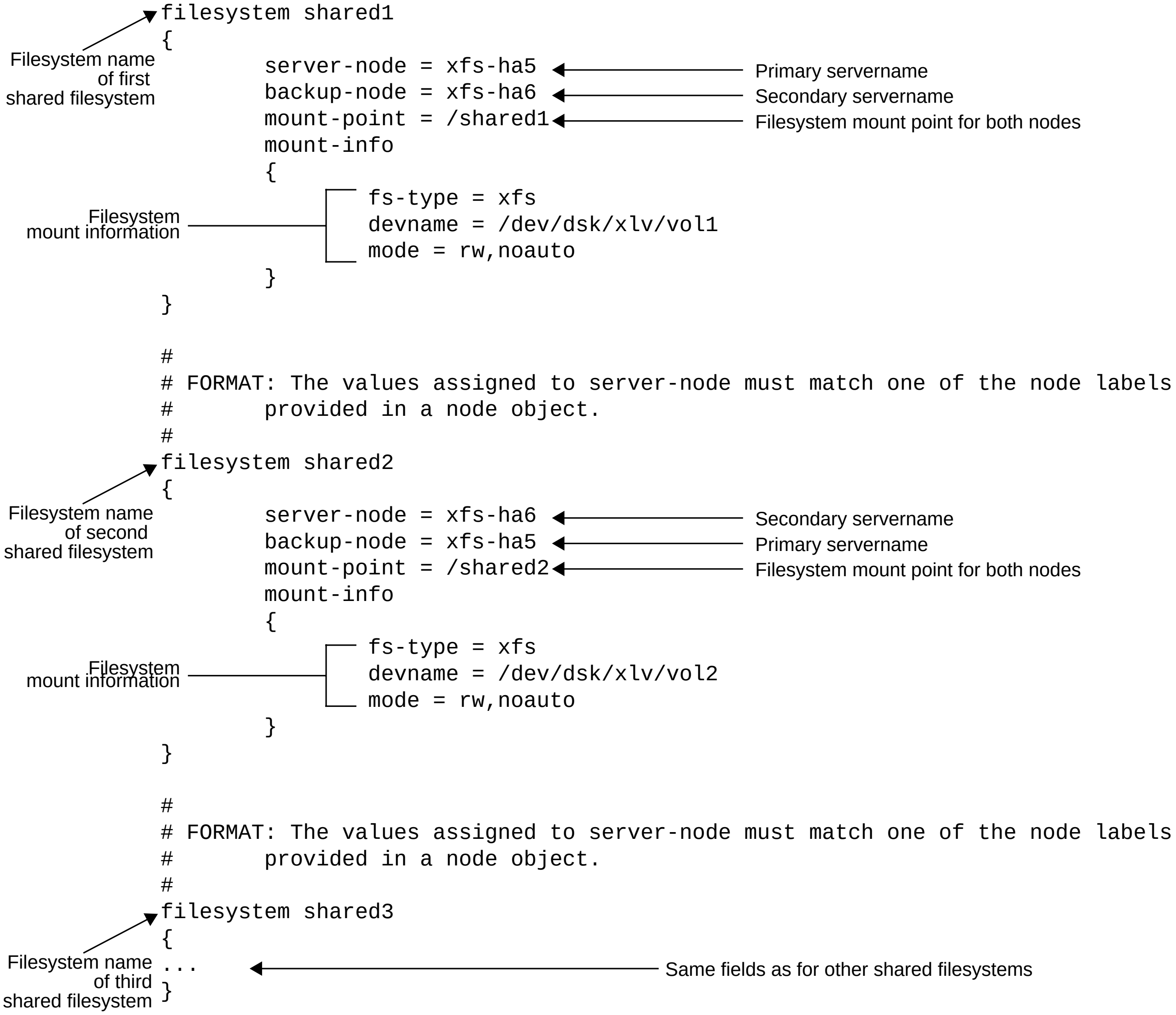
The IRIS FailSafe software supports failover for filesystems on shared disks, allowing the backup server to take over filesystems if the primary server fails. The filesystem block describes the values and flags passed to *mount*(1M).

Shared filesystems vary, depending on the IRIS FailSafe configuration:

- NFS dual-active
  - */shared1* is normally mounted on xfs-ha5
  - */shared2* is normally mounted on xfs-ha6

    These filesystems are created on separate XLV volumes, either on mirrored disks or on CHALLENGE RAID disks
- Web server active / standby: */shared* is normally mounted on xfs-ha5; created on an XLV volume

Figure 4-10 shows the disks / filesystem block for a dual-active configuration.



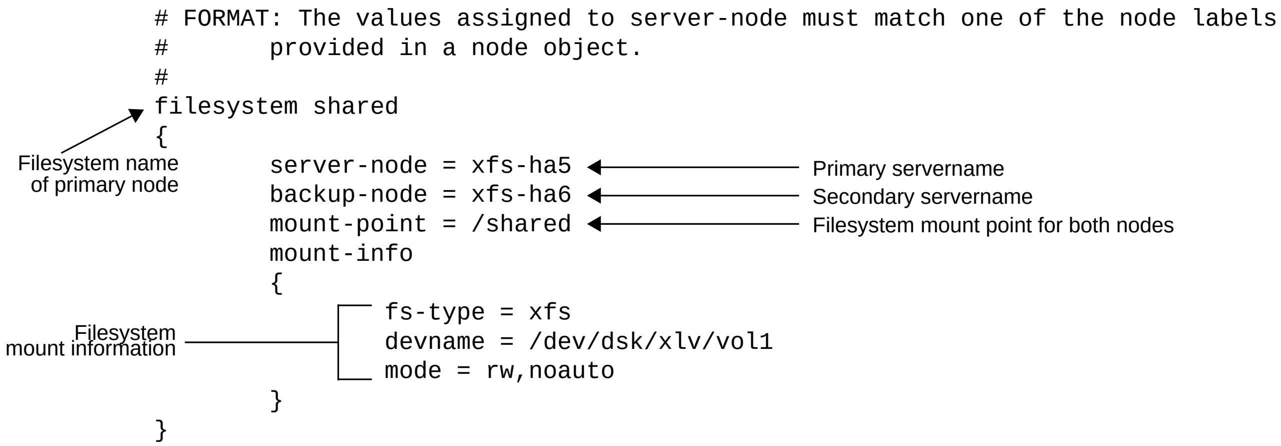
```
filesystem shared1
{
        server-node = xfs-ha5
        backup-node = xfs-ha6
        mount-point = /shared1
        mount-info
        {
                fs-type = xfs
                devname = /dev/dsk/xlv/vol1
                mode = rw,noauto
        }
}

#
# FORMAT: The values assigned to server-node must match one of the node labels
#         provided in a node object.
#
filesystem shared2
{
        server-node = xfs-ha6
        backup-node = xfs-ha5
        mount-point = /shared2
        mount-info
        {
                fs-type = xfs
                devname = /dev/dsk/xlv/vol2
                mode = rw,noauto
        }
}

#
# FORMAT: The values assigned to server-node must match one of the node labels
#         provided in a node object.
#
filesystem shared3
{
...
}
```

Filesystem name of first shared filesystem
Primary servername
Secondary servername
Filesystem mount point for both nodes
Filesystem mount information
Filesystem name of second shared filesystem
Secondary servername
Primary servername
Filesystem mount point for both nodes
Filesystem mount information
Filesystem name of third shared filesystem
Same fields as for other shared filesystems

Figure 4-10 Dual-Active NFS Configuration File Block: Disk / Filesystem

In this block, notice that the server-node field specifies the server that normally mounts the filesystem.

Figure 4-11 shows the disks / filesystem block, including introduction, for an active / standby configuration.



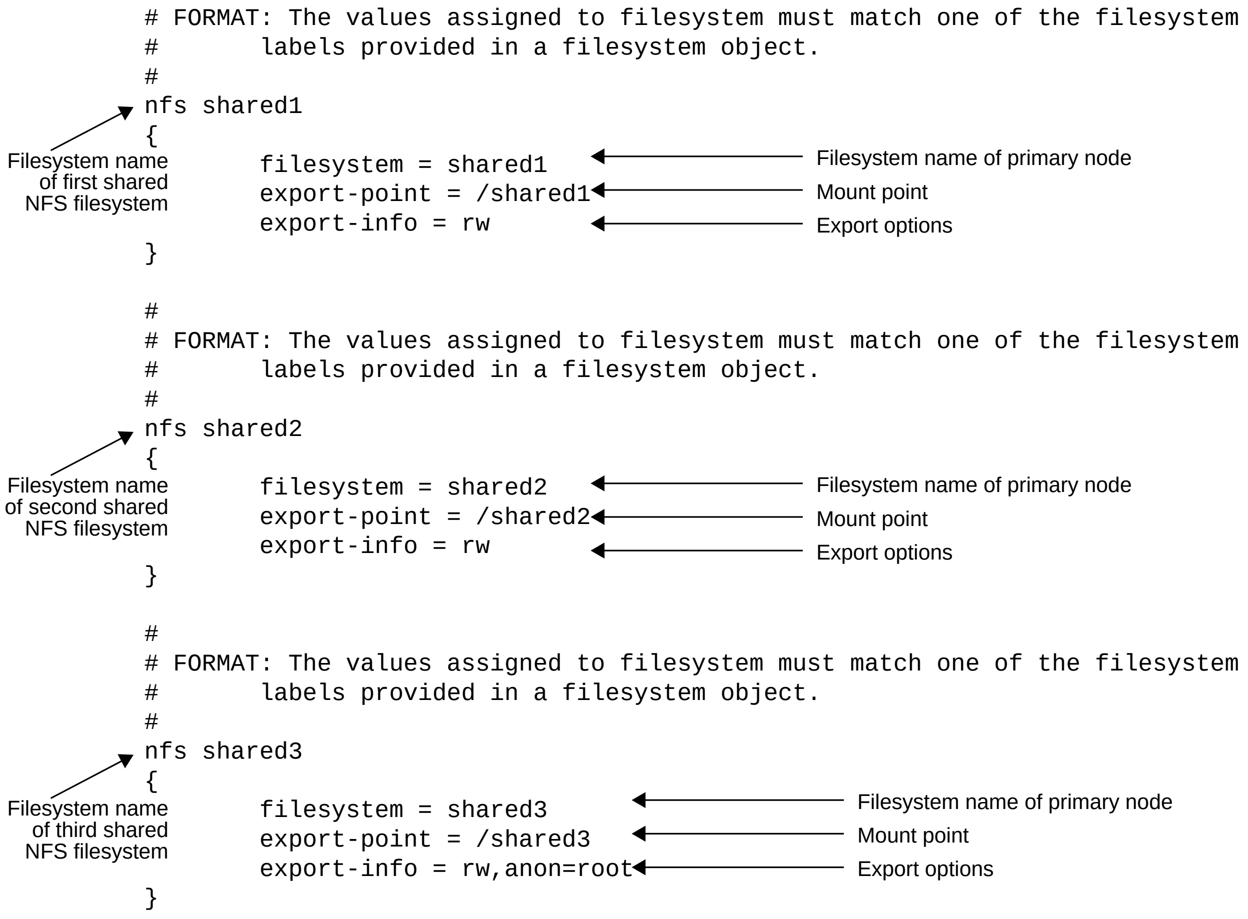
```
# FORMAT: The values assigned to server-node must match one of the node labels
#       provided in a node object.
#
filesystem shared
{
        server-node = xfs-ha5
        backup-node = xfs-ha6
        mount-point = /shared
        mount-info
        {
                fs-type = xfs
                devname = /dev/dsk/xlv/vol1
                mode = rw,noauto
        }
}
```

Filesystem name of primary node
Primary servername
Secondary servername
Filesystem mount point for both nodes
Filesystem mount information

Figure 4-11 Active / Standby Web Configuration File Block: Disk / Filesystem Configuration

Optional NFS Blocks

The exported NFS filesystems block in an NFS configuration file specifies the values and flags specified in the *exportfs*(1M) command. Figure 4-12 shows the exported NFS filesystems block in a dual-active NFS system.



```
# FORMAT: The values assigned to filesystem must match one of the filesystem
#       labels provided in a filesystem object.
#
nfs shared1
{
        filesystem = shared1
        export-point = /shared1
        export-info = rw
}

#
# FORMAT: The values assigned to filesystem must match one of the filesystem
#       labels provided in a filesystem object.
#
nfs shared2
{
        filesystem = shared2
        export-point = /shared2
        export-info = rw
}

#
# FORMAT: The values assigned to filesystem must match one of the filesystem
#       labels provided in a filesystem object.
#
nfs shared3
{
        filesystem = shared3
        export-point = /shared3
        export-info = rw,anon=root
}
```

Filesystem name of first shared NFS filesystem
Filesystem name of primary node
Mount point
Export options
Filesystem name of second shared NFS filesystem
Filesystem name of primary node
Mount point
Export options
Filesystem name of third shared NFS filesystem
Filesystem name of primary node
Mount point
Export options

Figure 4-12 NFS Dual-Active NFS Configuration File Block: Exported Filesystems

The filesystem fields in this block reference the physical filesystems that are exported. You must create at least one NFS section for every filesystem to be exported. Include only filesystems that are to be failed over. Figure 4-13 shows action and action timer blocks for NFS.

Local and remoter monitor script pathnames

```
action nfs
{
        local-monitor = /var/ha/actions/ha_nfs_lmon
        remote-monitor = /var/ha/actions/ha_nfs_rmon
}
```

Timers that affect monitoring of application classes

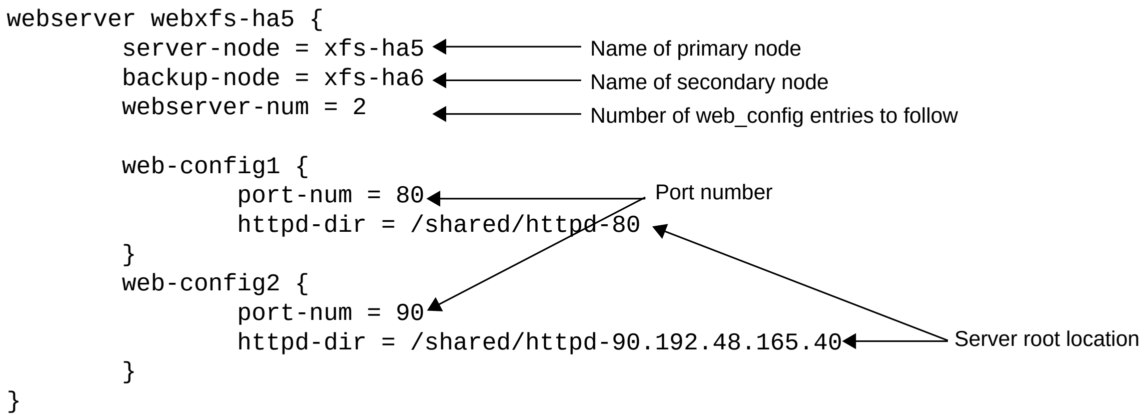
```
action-timer nfs
{
        #
        # once a node releases an application class (giveaway), it will
        # start the remote monitoring of it (application class) in
        # start-monitor-time seconds.
        #
        start-monitor-time = 30

        #
        # local monitoring will be done every lmon-probe-time seconds and
        # will timeout in lmon-timeout seconds.
        #
        lmon-probe-time = 20
        lmon-timeout = 30

        #
        # remote monitoring will be done every rmon-probe-time seconds and
        # will timeout in rmon-timeout seconds.
        #
        rmon-probe-time = 60
        rmon-timeout = 45

        #
        # some of the monitoring scripts do internal retries (i.e. the
        # application monitor sees it as a single ’probe’ but the actual
        # script can do retry-count ’probes’
        #
        retry-count = 1
}
```

Figure 4-13 NFS Action and Action Timer Blocks

The information in the action and timer blocks rarely needs to be changed.

Optional Web Server Blocks

The Web server block in a Web Server configuration file specifies the location and ports for the Web servers. Figure 4-14 shows this section for an active/standby configuration.



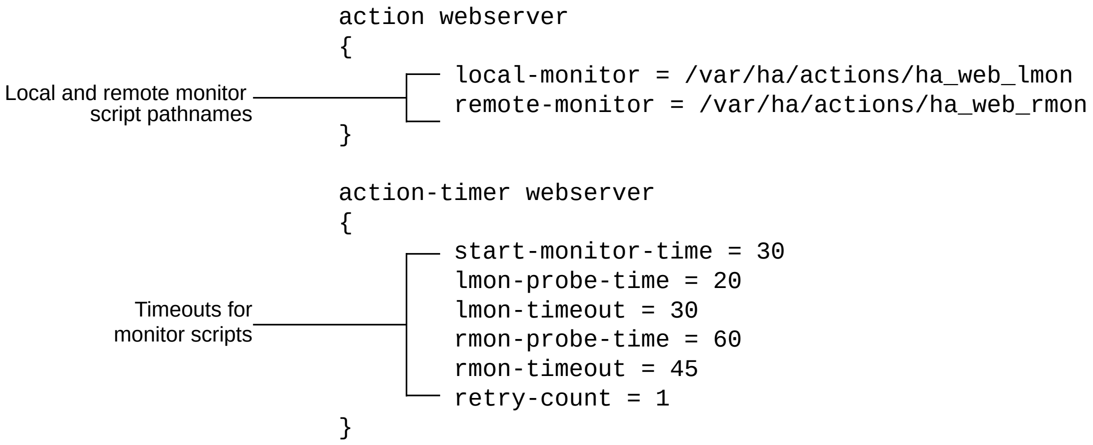
```
webserver webxfs-ha5 {
        server-node = xfs-ha5
        backup-node = xfs-ha6
        webserver-num = 2

        web-config1 {
                port-num = 80
                httpd-dir = /shared/httpd-80
        }
        web-config2 {
                port-num = 90
                httpd-dir = /shared/httpd-90.192.48.165.40
        }
}
```

Figure 4-14 Web Server Active/Standby Configuration: Web Servers Block

Notice that the primary server is configured with two Web servers: one listens to the default port, 80, and the other one listens to port 90. For more information on configuring for the Web server option, see “Configuring the Netscape Server Option” in Chapter 3.

Figure 4-15 shows the action and action timer blocks for Web Server in an active/standby configuration:

- the action field gives the pathnames of the respective monitoring scripts
- the action timer field lists the timers that affect the monitoring of the respective application classes



```
action webserver
{
        local-monitor = /var/ha/actions/ha_web_lmon
        remote-monitor = /var/ha/actions/ha_web_rmon
}

action-timer webserver
{
        start-monitor-time = 30
        lmon-probe-time = 20
        lmon-timeout = 30
        rmon-probe-time = 60
        rmon-timeout = 45
        retry-count = 1
}
```

Local and remote monitor script pathnames
Timeouts for monitor scripts

Figure 4-15 Web Server Active / Standby Configuration File Blocks: Action and Action Timer

The information in these blocks rarely needs to be changed.

Saving the Configuration File

When you are finished editing the file, save it as */var/ha/ha.conf*.

Chapter 5

Testing the Configuration

This chapter explains how to test the IRIS FailSafe system configuration. You test

- the configuration file
- filesystems
- public network interfaces
- NFS configuration
- Web server configuration
- system behavior

Testing the Configuration File

Edit one of the sample configuration files, following instructions in Chapter 4, “Creating the Configuration File.” After you have saved it as *ha.conf*, test it by running *ha_cfgverify*. Messages for this command are in Appendix B, “ha_cfgverify Error Messages.”

Testing Filesystems

For each filesystem on the primary node, execute the mount directory that the IRIS FailSafe software would execute. The parameters in the filesystem section of the configuration file *ha.conf* look like the following:

```
filesystem shared
{
   server-node = xfs-ha5
        backup-node = xfs-ha6
        mount-point = /shared
        mount-info
        {
                fs-type = xfs
                devname = /dev/dsk/xlv/vol1
                mode = rw,noauto
        }
}
```

To test the filesystems, enter

```
# mount -txfs -rw,noauto /dev/dsk/xlv/vol1 /shared
```

Repeat this process for each filesystem and on the secondary node after you umount them on the primary node. Make sure you do not mount the filesystem simultaneously from both nodes.

Testing Public Network Interfaces

To test the public network interface, execute the *ifconfig* command that the IRIS FailSafe system would execute; verify that it works. Repeat the process for each public network interface on each node.

The following example shows how the IRIS FailSafe software uses the parameters in the node block of the configuration file *ha.conf* to configure an interface up.

```
secondary-ip
{
        interface = ec2
        ip-name = xfs-ha5-2
        netmask = 255.255.255.0
        broadcast = 192.48.165.255
        mac-address = 8:0:69:2:60:bf
}
```

For more information on this block, see “Node Block” in Chapter 4.

Follow these steps:

1. Enter

   ```
   xfs-ha5# ifconfig ec2 inet xfs-ha5-2 up netmask
      255.255.255.0 broadcast 192.50.165.255
   ```

2. To test the public network connection from the first server (xfs-ha5), enter

   ```
   /usr/etc/ping stocks
   ```

3. If you have a dual-active configuration, you can also test the backup interface to the public network:

   ```
   /usr/etc/ping xfs-ha5-2
   ```

4. Repeat the ping process for the interface of the other server.

Testing the NFS Configuration

After the filesystem and network have been configured up, export the filesystems manually; determine if a client can access them. Execute this process from both nodes.

The following procedure presumes an NFS entry in *ha.conf* like the following:

```
nfs shared1
{
        filesystem = shared1
        export-point = /shared1
        export-info = rw
}
```

Follow these steps:

1. Make sure */shared1* is mounted. If it is not, verify the mount as described in the preceding section.
2. Export the filesystem from the primary node; for example:

   ```
   xfs-ha5# exportfs -i -o rw /shared1
   ```

3. Make sure the filesystem was exported:

   ```
   xfs-ha5# exportfs
   /shared1 -rw
   ```

4. Unexport it and umount it, so that you can run this test from the other server:

   ```
   xfs-ha5# exportfs -u /shared1
   xfs-ha5# umount /shared1
   ```

5. Repeat these steps on the other server. Make sure you do not mount the filesystem simultaneously from both nodes.

Testing Web Server Configuration

To make sure that the Web server addresses you are using are configured up, follow these steps:

1. Make sure that the network interfaces are *ifconfig*'d up.
2. Start the Web servers:

   ```
   chkconfig ns_httpd on
   ```

   (They are normally *chkconfig*'d on by default.)
3. Run

   ```
   /etc/init.d/ns_httpd start
   ```

   If you have multiple Web servers, you might need to create a */etc/config/ns_httpd.options* file.
4. Run a Web browser, such as Netscape, and try to access some Web pages exported by the server.

Testing System Behavior

After you have fully configured your IRIS FailSafe cluster, check to make sure that the services are available from the cluster when both nodes are up. For NFS, check the contents of any exported filesystems. For a Web server, run a browser and examine any Web pages that are installed.

After you have confirmed that the cluster operates correctly when both nodes are present, confirm that the cluster functions correctly in the face of failures by performing the following tests:

1. Power off one node in the cluster. The other node in the cluster should detect the failure and take over the services. If you have an active / standby configuration, power off the active node.
2. Disconnect the private network. If you have enabled heartbeat messages to be sent over the public network, the cluster should continue to function as before. Otherwise, one node takes over the services of the other node. The other node should enter the standby state.
3. Disconnect the public network from one of the active nodes in the cluster. The other node should take over the services.
4. Forcibly unmount a shared filesystem on an active node to make it unavailable. The other node should take over the services. Depending upon the node that detects this condition first, the failed node either enters standby mode or is restarted.
5. Kill the application daemons (for example, *ns_httpd*) on a node to make the service unavailable. The other node should take over the service.
6. Disconnect the serial line to the system controller port (if you are using CHALLENGE XL / L / DM) or the remote power control unit (if you are using CHALLENGE S). If you have configured the IRIS FailSafe software to send mail, it notifies the administrator of the failure and would otherwise continues to function.

   When the cluster is in this state, neither node can take over if another failure occurs. After you have reconnected the serial line, you can resume monitoring of the serial line by executing *ha_admin -m start <node_name>*.

Appendix A

Setting Up an IRIS FailSafe System With CHALLENGE S Servers

This appendix explains how to set up an IRIS FailSafe system when at least one of the nodes is a CHALLENGE S server. The process consists of:

1. setting up the component systems
2. cabling the private and public networks
3. setting up the IRIS FailSafe serial connection
4. testing the installed IRIS FailSafe hardware
5. testing the serial connection
6. installing the IRIS FailSafe software

The following equipment is required for installation:

- installation guides for the component systems
- laptop or ASCII terminal
- Phillips-head and small flat-blade screwdrivers

Setting Up the Component Systems

For the IRIS FailSafe system, you are setting up

- servers
- shared storage system
- public network interfaces (dual-active: two per node; active/standby: one per node):The following equipment is required for installation:
- installation guides for the component systems
- laptop or ASCII terminal

- Phillips-head and small flat-blade screwdrivers
  - primary public network (Ethernet AUI port)
  - optional secondary public network (optional Ethernet port) for dual-active configurations

    Note: If FDDI is used, only one public network is possible.
- private (heartbeat) network interface (Ethernet 10-base port): one per node
- shared SCSI bus to shared storage: one per node
- serial connection through the remote power control unit: one per node

Figure A-1 diagrams basic IRIS FailSafe cabling.

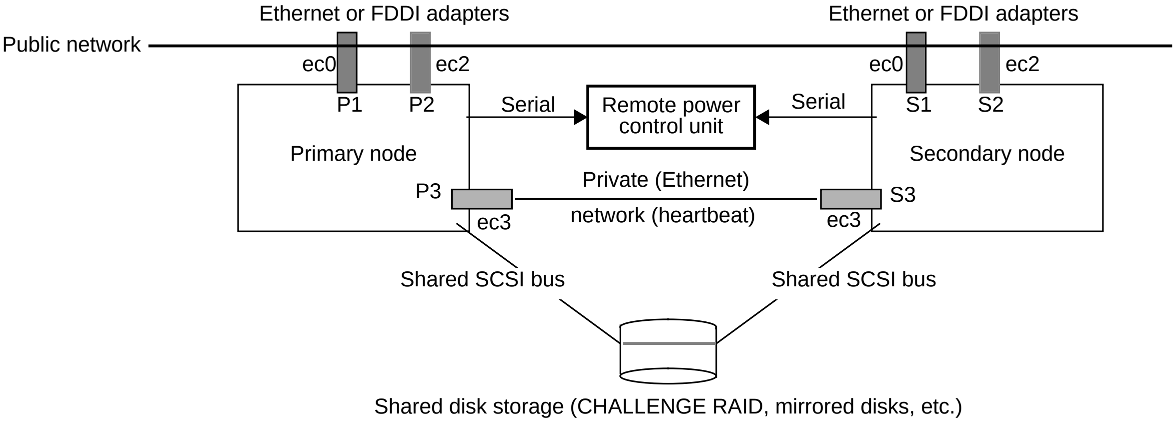

Ethernet or FDDI adapters
Ethernet or FDDI adapters
Public network
ec0
ec2
ec0
ec2
P1
P2
S1
S2
Serial
Remote power control unit
Serial
Primary node
Secondary node
P3
Private (Ethernet) network (heartbeat)
S3
ec3
ec3
Shared SCSI bus
Shared SCSI bus
Shared disk storage (CHALLENGE RAID, mirrored disks, etc.)

Figure A-1 Basic IRIS FailSafe Cabling Scheme

The dual-active configuration can use one public network or two. Figure A-2 diagrams IRIS FailSafe cabling with two public networks (dual-active configuration only).

Note: For information on IRIS FailSafe configurations, see “High Availability, Dual-Active Operation, and Active/Standby Operation” in Chapter 1.

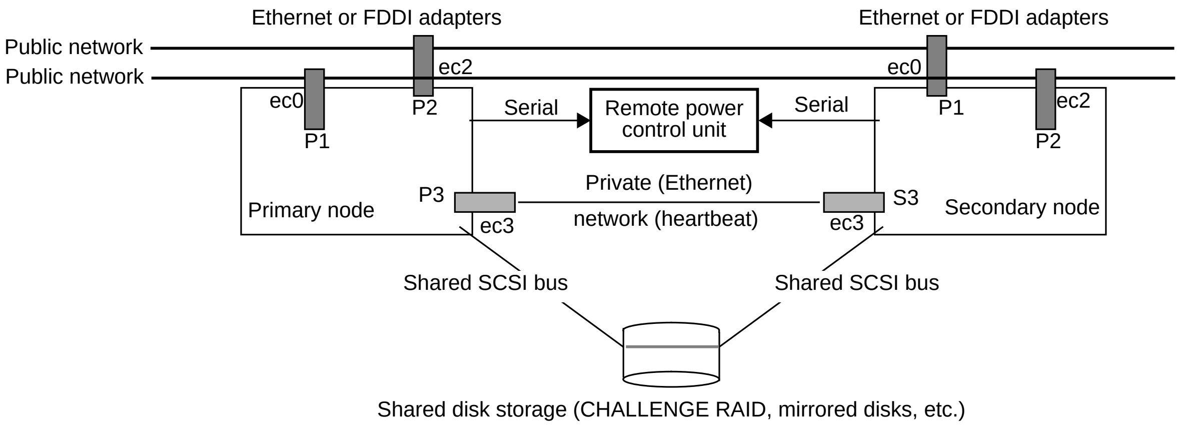

Ethernet or FDDI adapters
Ethernet or FDDI adapters
Public network
Public network
ec2
ec0
ec0
P2
Serial
Remote power control unit
Serial
P1
ec2
P1
P2
P3
Private (Ethernet) network (heartbeat)
S3
Primary node
ec3
ec3
Secondary node
Shared SCSI bus
Shared SCSI bus
Shared disk storage (CHALLENGE RAID, mirrored disks, etc.)

Figure A-2 IRIS FailSafe Cabling With Two Public Networks (Dual-Active Configuration)

To set up the component systems, follow these guidelines:

1. Make sure that the installation site meets the operating limits and AC power requirements of the equipment.
2. Prepare the physical location to allow for space, air flow, and floor-loading requirements for all component systems.

   Plan to situate the servers and vaults fairly close together. Differential SCSI cables, including cabling inside the chassis, can be no longer than 25 meters (80 feet). Use only the cables included in the shipment.
3. Make sure the site meets safety and operating considerations.
4. Prepare the site for the systems' electrical requirements.
5. For a dual-active configuration using two public networks, install the second network adapter (**ec2**) in the CHALLENGE S server if necessary. Follow instructions in the *CHALLENGE S Server Owner's Guide* (document number 007-2314-003).
6. Set up the shared storage system.
7. For testing, have ready a laptop or ASCII terminal and keyboard.

8. Make provisions for the network connections, such as obtaining Ethernet drop cables.

Figure A-3 shows the ports on the CHALLENGE S server for the private, primary, and secondary network interfaces and for the fast and wide SCSI interface (for the CHALLENGE RAID or CHALLENGE Vault storage system).

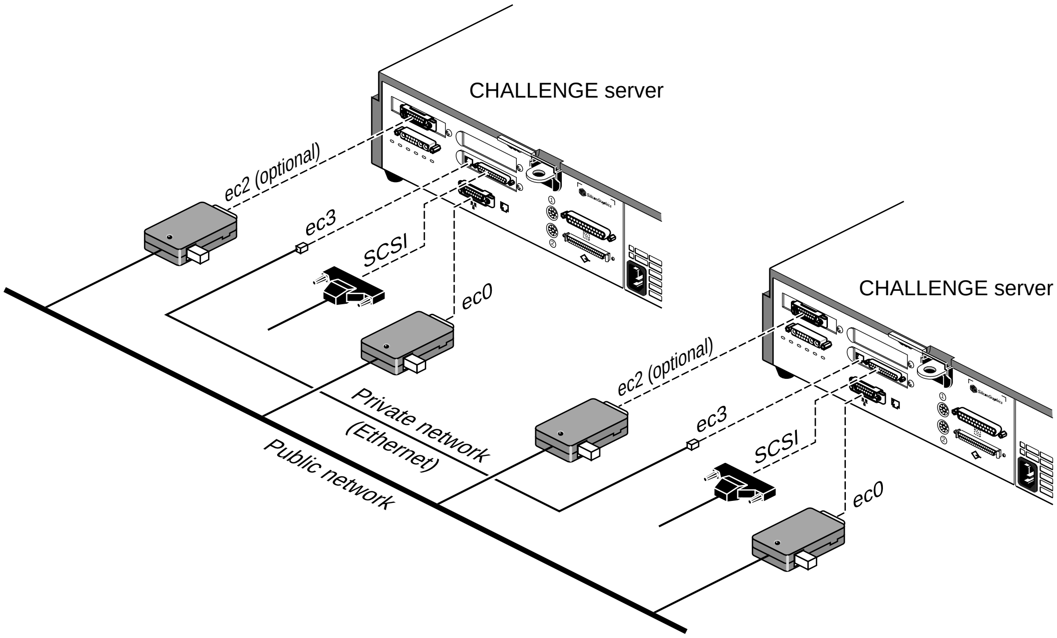


Figure A-3 CHALLENGE S Server Ports: Private and Public Network Cabling

Figure A-4 shows an example of serial connection for sites using two CHALLENGE S servers.

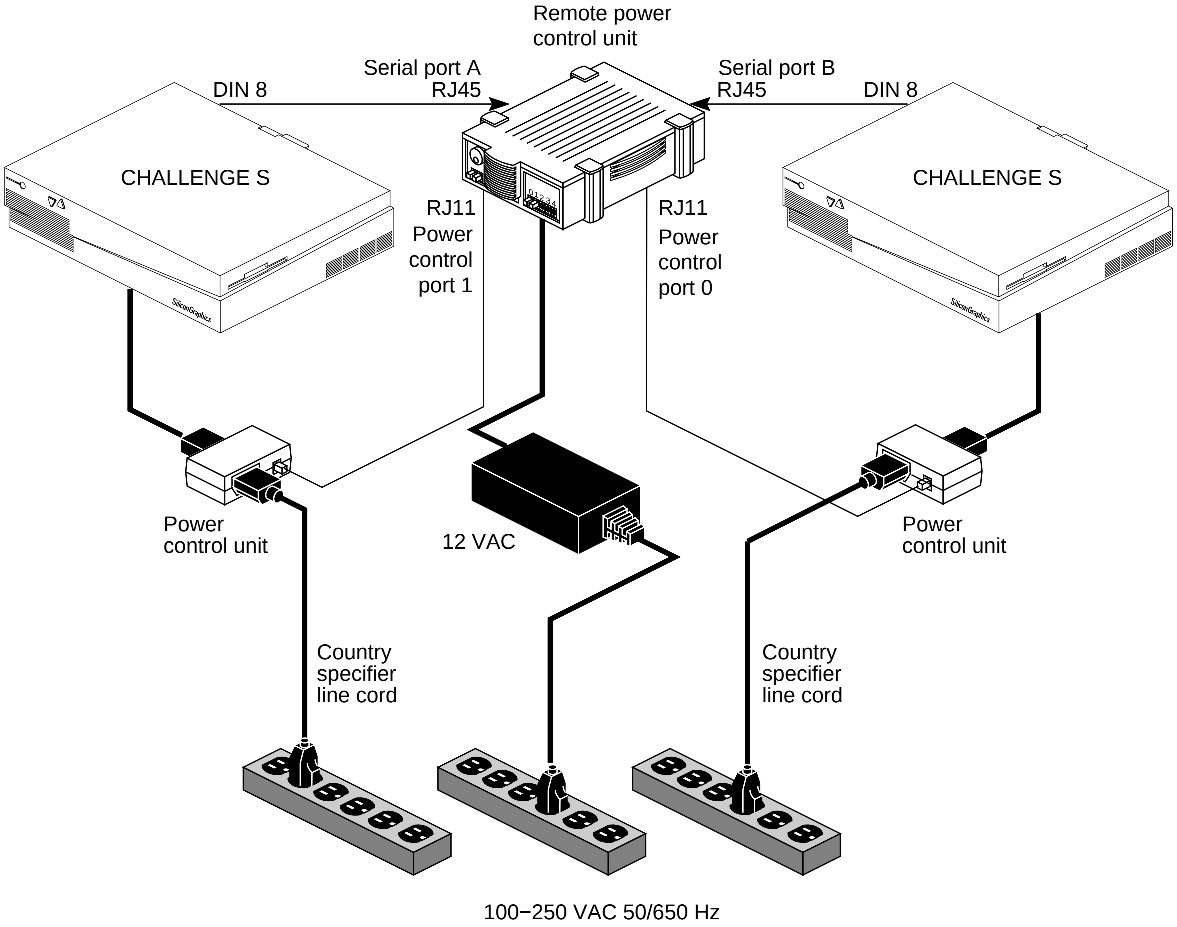


Figure A-4 Serial and Power Connections: Two CHALLENGE S Servers

Cabling the Private and Public Networks

This section explains

- cabling the private (heartbeat) connection between the two nodes
- cabling to the public network

Cabling the Private (Heartbeat) Connection Between the Two Nodes

The private network between the CHALLENGE servers supplies the heartbeat of one of the servers to the other. To cable the private network, connect the Ethernet 10-BASE-T (**ec3**) ports on the CHALLENGE S servers to each other with the blue Ethernet loopback cable supplied with the IRIS FailSafe option.

This port is between the two SCSI connectors on the back of the CHALLENGE S server, as shown in Figure A-5. Note that the Ethernet 10-BASE T and the ISDN port below it look similar. The Ethernet 10-BASE T port is the one on top, immediately next to the SCSI connector, as shown in Figure A-5.

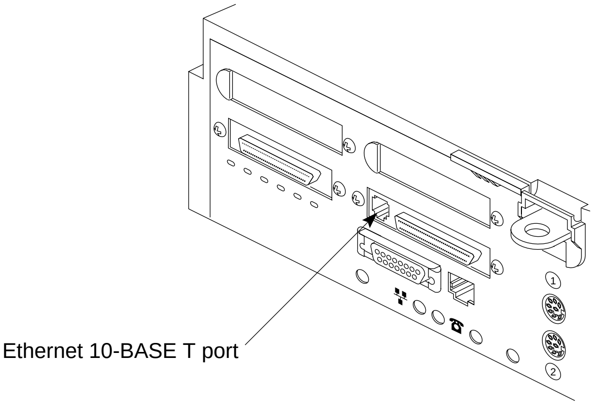


Figure A-5 Connecting an Ethernet 10-BASE T Cable to the CHALLENGE S Server (S-100 and S-150 Models)

Cabling to the Public Network

In an IRIS FailSafe dual-active configuration, the IRIS FailSafe system uses two connections to a public network: primary and secondary. The IRIS FailSafe active/standby configuration has only the primary network connection; the second node is a hot standby only and operates only if the first node fails.

To connect the Ethernet port of each CHALLENGE S server to the public network, follow these steps:

1. For each IRIS FailSafe server, obtain a drop cable that reaches from the wall to the back of the server.
2. Connect the Ethernet AUI cable to the Ethernet AUI port (**ec0**) on the back of the CHALLENGE S server, as shown in Figure A-6.

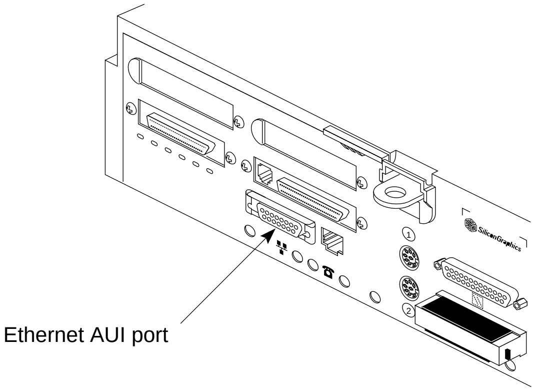


Figure A-6 Connecting an Ethernet AUI Cable to the CHALLENGE S Server

3. Make sure the sliding bracket on the Ethernet port on the system is pushed all the way left, as shown in Figure A-7.

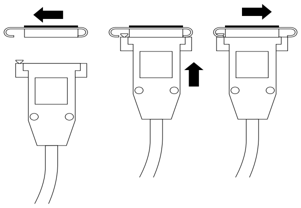


Figure A-7 Ethernet AUI cable

4. Plug the cable into the port. Slide the bracket right to hold it in place.
5. Cable the Ethernet connection of the second server in the system.
6. For the dual-active configuration, repeat this process to cable the optional Ethernet (**ec2**) board installed in the server. This connection can use the same public network as the primary Ethernet AUI ports (**ec0**) or can use a second public network.

   Caution: For a dual-active configuration using a second public network, make sure that the primary Ethernet connection (**ec0**) of the primary node and the secondary Ethernet connection (**ec2**) of the secondary node are on one network and the secondary Ethernet connection (**ec2**) of the primary node and the primary Ethernet connection (**ec0**) of the secondary node are on the secondary network. See Figure A-2.

Note: In the CHALLENGE S server, you can configure a single FDDI adapter, which takes up both GIO slots. Thus, if you use FDDI you can have only one connection to the public network. This configuration is the active-standby configuration; two same-type (both Ethernet or both FDDI) connections to the public network are required for dual-active configuration.

Setting Up the IRIS FailSafe Serial Connection

The serial connection between the two CHALLENGE servers in the IRIS FailSafe system makes it possible for one server to reboot or shut down the other in case of failure. In the case of two larger servers, the Remote System Control port of one server is cabled to a serial port of the other, and vice versa. Because the CHALLENGE S has no Remote System Control port, the Silicon Graphics remote power control unit takes its place for purposes of the serial connection.

This section explains how to cable the servers to the remote power control unit to the servers. It gives instructions for

- cabling two CHALLENGE S servers
- cabling one CHALLENGE S server and a larger CHALLENGE server

Cabling Two CHALLENGE S Servers

Figure A-8 shows an example of serial connection for sites using two CHALLENGE S servers.

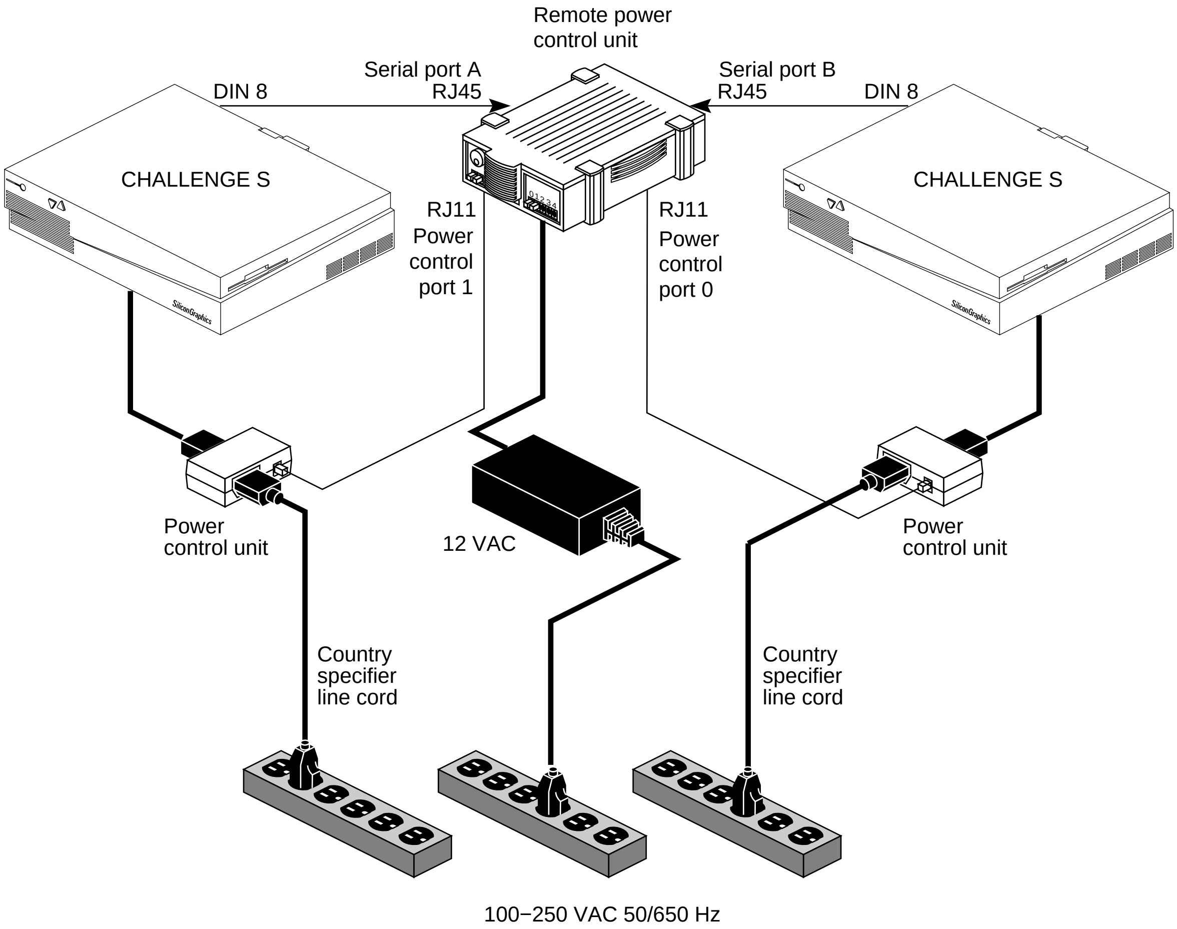


Figure A-8 Serial and Power Connections: Two CHALLENGE S Servers

Note: On the remote power control unit, the leftmost Serial Port (A) controls the leftmost Power Control Port (0), and the rightmost Serial Port (B) controls the next Power Control Port (1). Note that the jacks are different for the two types of ports.

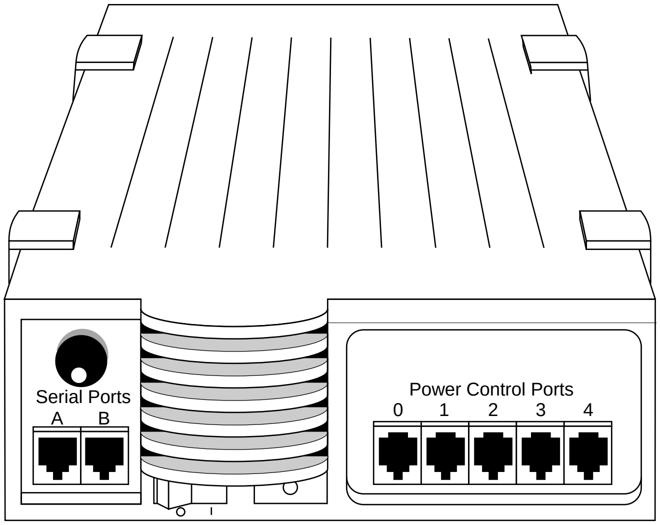


Figure A-9 Remote Power Control Unit: Connector Panel

Note that Power Control Ports 2 through 4 are not used; they plugged in to reduced the chance for error.

If both nodes in your IRIS FailSafe configuration are CHALLENGE S servers, follow these steps:

1. Plug one of the two serial cables included with the remote power control unit that has an RJ45 connector into Serial Port A on the remote power control unit, as shown in Figure A-10. This cable is labeled **CHALLENGE S TO RPCU CABLE**.

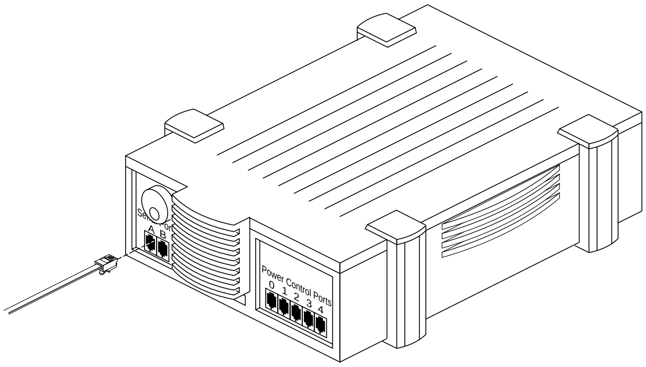


Figure A-10 Cabling the Serial Port on the Remote Power Control Unit

Note: For all RJ45 and RJ11 connections, be sure the connector is properly seated in the jack so that it is making good contact.

2. Plug the other end of this serial cable into the serial connector (**tty_2**) on the back of the CHALLENGE S server, as shown in Figure A-11.

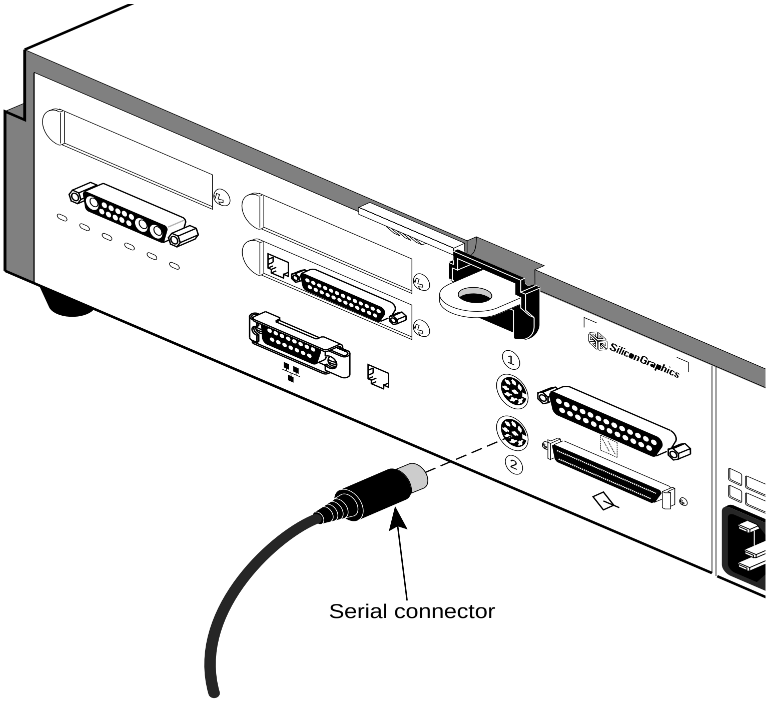


Figure A-11 Connecting the Serial Cable to the CHALLENGE S Server

3. Plug a serial cable with an RJ11 connector into the Power Control Port 0 on the remote power control unit, as shown in Figure A-12.

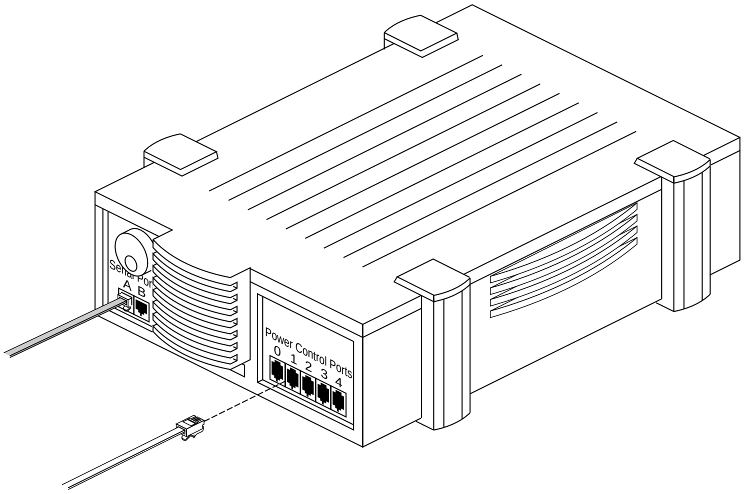


Figure A-12 Cabling the Power Control Port

4. Attach the RJ11 connector of the serial cable to the RJ11 connector in a power control unit, as shown in Figure A-13.

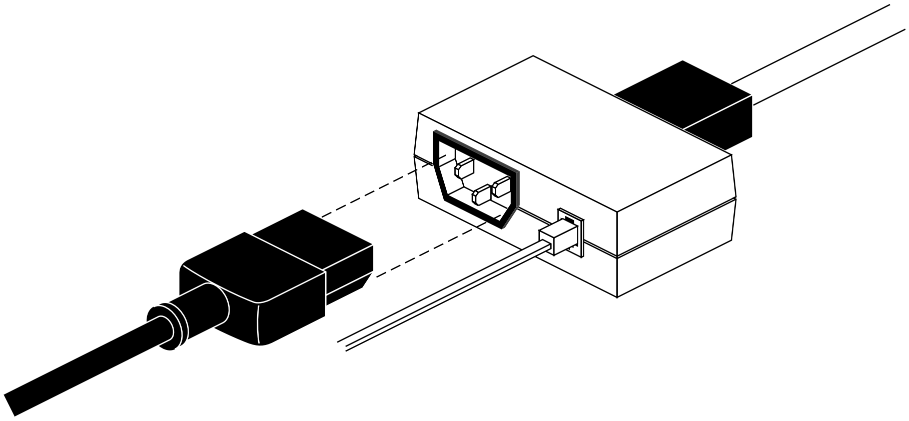


Figure A-13 Cabling the Power Control Unit

5. Attach the CHALLENGE S power cable to the server; plug the other end of the power cable into the power control unit.
6. Repeat these steps for the second CHALLENGE S server, using Serial Port B and Power Control Port 0.
7. Connect the power for the remote power control unit. Power it on; power on the servers and the ASCII terminal or laptop attached to each.
8. Type **hinv** on each IRIS FailSafe server; the screen should display output like the following:

   ```
   xfs-ha6 14# hinv
   1 150 MHZ IP22 Processor
   FPU: MIPS R4010 Floating Point Chip Revision: 0.0
   CPU: MIPS R4400 Processor Chip Revision: 5.0
   On-board serial ports: 2
   On-board bi-directional parallel port
   Data cache size: 16 Kbytes
   Instruction cache size: 16 Kbytes
   Secondary unified instruction/data cache size: 1 Mbyte
   Main memory size: 64 Mbytes
   Integral ISDN: Basic Rate Interface unit 0, revision 1.0
   E++ controller: ec2, version 1
   Integral Ethernet: ec3, version 1
   Integral Ethernet: ec0, version 1
   Integral SCSI controller 5: Version WD33C95A,
   differential, revision 0
   Disk drive: unit 5 on SCSI controller 5
   Integral SCSI controller 4: Version WD33C95A,
   differential, revision 0
   Integral SCSI controller 0: Version WD33C93B, revision D
   Disk drive: unit 1 on SCSI controller 0
   ```

   In this output, check for

   - network interfaces; for example:

     ```
     E++ controller: ec2, version 1
     Integral Ethernet: ec3, version 1
     Integral Ethernet: ec0, version 1
     ```

     This example shows the three network interfaces of a dual-active configuration; an active/standby configuration would lack the first line.

   - shared disks; for example:

```
Disk drive: unit 5 on SCSI controller 5
Integral SCSI controller 4: Version WD33C95A,
differential, revision 0
Integral SCSI controller 0: Version WD33C93B, revision
D
Disk drive: unit 1 on SCSI controller 0
```

Cabling a CHALLENGE S and a Larger Server

If you are using a CHALLENGE S and a CHALLENGE DM, L, or XL server, follow steps in this section. Figure A-14 diagrams the serial interface for this configuration.

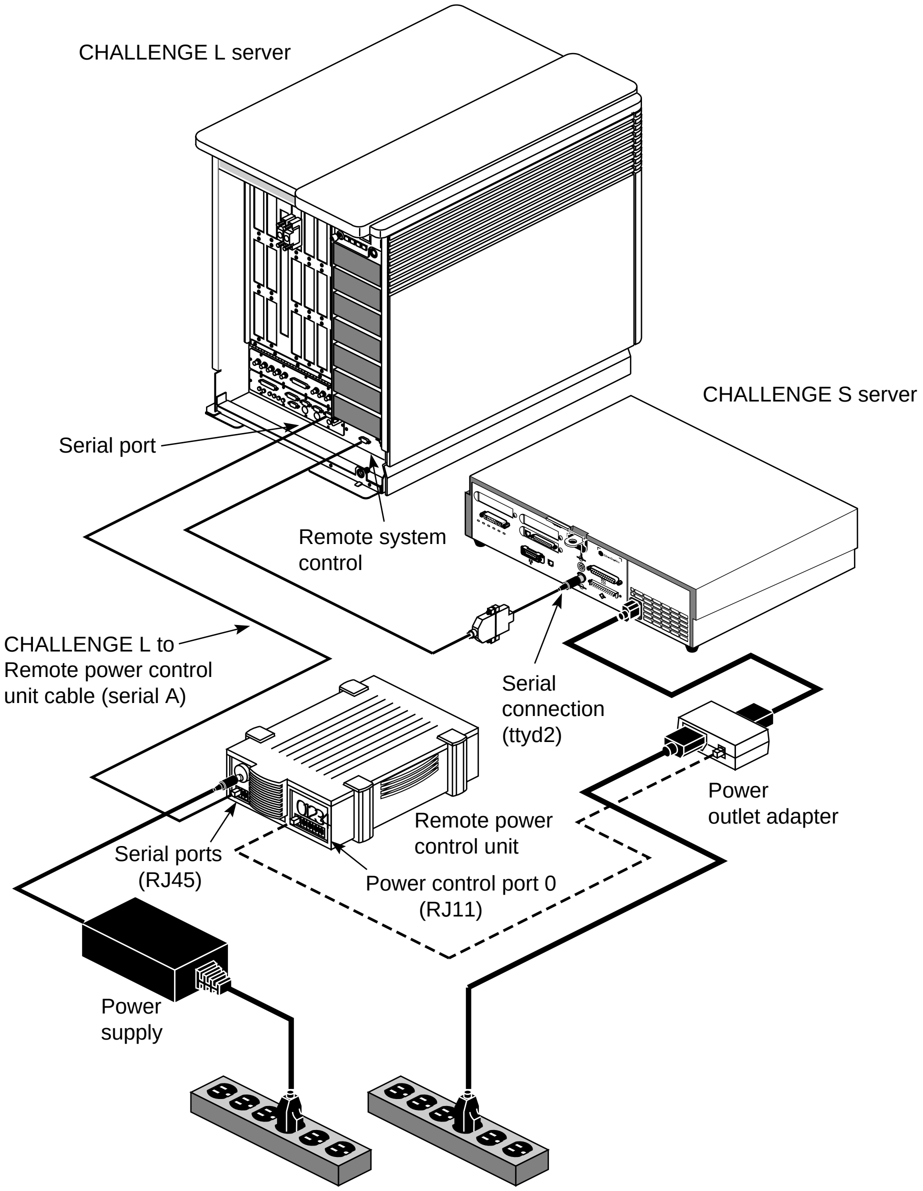

CHALLENGE L server
CHALLENGE S server
Serial port
Remote system control
CHALLENGE L to Remote power control unit cable (serial A)
Serial connection (ttyd2)
Power outlet adapter
Remote power control unit
Serial ports (RJ45)
Power control port 0 (RJ11)
Power supply

Figure A-14 Serial Connection: CHALLENGE S Server and Larger CHALLENGE Server

To set up the serial connection for sites using one CHALLENGE S server and another larger CHALLENGE server, follow these steps:

1. Connect the remote power control unit and the larger CHALLENGE server:
   - Attach a serial cable included with the remote power control unit to an RF45 Serial Port A on the remote power control unit, as shown in Figure A-10. This cable is labeled **CHALLENGE XL/L TO RPCU CABLE**.
   - Attach the other end of this cable to a serial port on the larger CHALLENGE server, as shown in Figure A-14.
2. Attach one end of a serial cable to a Power Control Port on the remote power control unit and the other end of the cable to the RJ45 connector port 1 in a power control unit. For details, see "Cabling Two CHALLENGE S Servers," earlier in this section.
3. Attach the CHALLENGE S power cable to the server; plug the other end of the power cable into the power outlet adapter.
4. Connect a serial cable to the serial port on the CHALLENGE S. Attach the other end to the 9-pin Remote System Control port on the larger CHALLENGE server.

Testing the Installed IRIS FailSafe Hardware

To test the IRIS FailSafe hardware installation, attach an ASCII terminal to each server and run *hinv*. Following is a sample output:

```
4 150 MHZ IP19 Processors
CPU: MIPS R4400 Processor Chip Revision: 5.0
FPU: MIPS R4010 Floating Point Chip Revision: 0.0
Data cache size: 16 Kbytes
Instruction cache size: 16 Kbytes
Secondary unified instruction/data cache size: 1 Mbyte
Main memory size: 64 Mbytes, 1-way interleaved
I/O board, Ebus slot 5: IO4 revision 1
Integral EPC serial ports: 4
Integral Ethernet controller: et0, Ebus slot 5
EFast FXP controller: fxp1
EFast FXP controller: fxp0
EPC external interrupts
```

```
Integral SCSI controller 1: Version WD33C95A, differential,
revision 0
Disk drive: unit 1 on SCSI controller 1
Integral SCSI controller 0: Version WD33C95A, single ended,
revision 0
Integral SCSI controller 4: Version SCIP/WD33C95A
Integral SCSI controller 3: Version SCIP/WD33C95A
Integral SCSI controller 2: Version SCIP/WD33C95A
Disk drive: unit 1 on SCSI controller 2
CC synchronization join counter
Integral EPC parallel port: Ebus slot 5
VME bus: adapter 0 mapped to adapter 21
```

In this output, check for

- network interfaces; for example:

  ```
  E++ controller: ec2, version 1
  Integral Ethernet: ec3, version 1
  Integral Ethernet: ec0, version 1
  ```

  This example shows the three network interfaces of a dual-active configuration; an active/standby configuration would lack the first line.

- shared disks; for example:

  ```
  Disk drive: unit 5 on SCSI controller 5
  Integral SCSI controller 4: Version WD33C95A,
  differential, revision 0
  Integral SCSI controller 0: Version WD33C93B, revision D
  Disk drive: unit 1 on SCSI controller 0
  ```

Testing the Serial Connection

To test the serial connection between the IRIS FailSafe servers, follow these steps:

1. Make sure the IRIS FailSafe servers are powered on.
2. Make sure that the remote power control unit is powered on.
3. Enter

   ```
   /usr/etc/ha_spng -i 10 -f /dev/ttyd2
   ```

   No output appears; check the return value of the command. If the return value is 0, the connection is good.
4. If the return value is 1**,** check the following:
   - verify that the IRIS FailSafe server is powered on
   - verify the cable connections from one server's serial port or remote power control unit and the other server's System Console port
5. Repeat the process on the second node.

If results are unsatisfactory, make sure that the RJ connector is completely seated at both ends and is making good contact.

Installing the IRIS FailSafe Software

The software needed to run IRIS FailSafe is on several CDs:

- IRIX 5.3 with XFS: two CDs, including the base filesystem and operating system, plexing, and networking software
- IRIS FailSafe software:
  - base IRIS FailSafe software, including patches, for the CHALLENGE S server and base IRIS FailSafe software including patches, for other platforms: use one of these
  - IRIS FailSafe software for the Web server option
  - IRIS FailSafe software for the NFS option

- optional software for Netscape and NFS
  - WebFORCE version 1.1.1 Netscape Communications Server
  - NFS

This section explains how to install the software for an IRIS FailSafe system. The process consists of:

- installing system software
- installing IRIS FailSafe software
- installing the Web server option
- installing the NFS option
- modifying and testing the configuration file
- installing the disk plexing license
- cloning the disk

If both servers in the IRIS FailSafe system are the same platform (both CHALLENGE S, for example), you can install the software two ways:

- on each server separately
- on one server, install and configure the software, copy the root disk to the other server, and adjust the configuration on the second server

The second method is followed in these instructions.

If the servers are different platforms, the software must be installed on each server separately. Only qualified Silicon Graphics System Service Engineers can install the software on a CHALLENGE DM, L, or XL server.

Installing System Software

To install system software, follow these steps:

1. Check to see if IRIX 5.3 XFS is installed on the first IRIS FailSafe server. If it is not, load the IRIX 5.3 XFS operating system from the two CDs, including the base filesystem and operating system and networking software. Install the base XLV volume manager and the XLV disk plexing option:

   ```
   i eoe2.sw.xlv
   i eoe2.sw.xlvplex
   ```

   You must install plexing regardless of whether you are using RAID disks or mirrored disks.

   Note: For complete information on installing an XFS filesystem, see *Getting Started with xFS Filesystems,* which is included with IRIX 5.3 XFS.

2. To verify if your system has the plexing software, type *xlv_mgr* to change to the XLV manager and type:

   ```
   show config
   ```

   Output appears, such as

   ```
   Allocated subvol locks: 30      locks in use: 0
   Plexing license: present
   Plexing support: present
   Maximum subvol block number: 0x7fffffff
   ```

3. Quit the XLV manager.

Installing IRIS FailSafe Software

To install the IRIS FailSafe system software, follow these steps:

1. Insert the CD for the IRIS FailSafe base software: use either the CD for the CHALLENGE S server or the CD for other platforms. Each CD includes necessary patches.

2. Type **`inst`**. At the `Inst>` prompt, enter

   ```
   list *
   ```

3. Type at the `Inst>` prompt

   ```
   i ha.sw.base
   ```

4. Following instructions in the IRIS FailSafe release nodes, install the required patches.
5. To install the IRIS FailSafe reference pages, enter

   ```
   i ha_man.base
   ```

6. If you are using the Netscape option, enter

   ```
   i ha_www.sw.base
   ```

7. If you are using the NFS option, enter

   ```
   i ha_nfs.sw.base
   ```

8. Exit *inst*.

Setting *nvram* Parameters

The IRIS Failsafe software requires the servers to be automatically booted when they are reset or when the system is powered on. Use **nvram**(1M) to change the boot parameter AutoLoad on both nodes.

```
# nvram -v AutoLoad yes
```

All nodes on a shared SCSI bus must be unique. By default, the SCSI ID of the host Challenge server) is 0. Since both hosts will be on the same bus with shared disk storage, you must set the SCSI ID of one server to be different. Use the **nvram**(1M) command to do so; for example:

```
# nvram -v scsihostid 3
```

Note that a host uses its SCSI ID on all buses attached to it. Therefore, you must make sure that no device attached to a server has the same SCSI unit number as the server’s scsihostid.

Installing the Web Server Option

To install the optional IRIS FailSafe Netscape software, insert the appropriate CD. At the `Inst>` prompt, type

```
i ns_httpd.sw.server
```

Note: By default, the IRIS FailSafe software supports the Netscape Communications Server. For another Web server, you must modify the recovery scripts (*takeover*, *giveover*, *takeback*, and *giveback*) in the */var/ha/actions.d* directory.

For instructions on configuring the Netscape software, see “Configuring the Netscape Server Option” in Chapter 3.

Installing the NFS Option

To install the optional IRIS FailSafe NFS software, follow these steps:

1. Insert the appropriate CD. At the `Inst>` prompt, type

   ```
   i nfs.sw.nfs
   ```

2. Following instructions in the IRIS FailSafe release nodes, install the required NFS-related patches.
3. Configure NFS following instructions in its documentation.

Modifying and Testing the Configuration File

The IRIS FailSafe system uses the *ha.conf* file to determine system resources, such as the primary and secondary (active and standby) server names, network interface names and addresses, and shared storage and filesystem parameters.

The IRIS FailSafe system software includes several versions of the configuration file for various configurations: dual-active Web server, active/standby Web server, dual-active NFS, active/standby NFS, and a combination of Web server and NFS. Choose one of these example configuration files, adapt it for your IRIS FailSafe system, and save it as *ha.conf*. Follow instructions in Chapter 3, “Configuring the IRIS FailSafe System.”

To test the validity of the *ha/conf* file, run *ha_cfgverify*. This command outputs error messages; see Appendix B, “ha_cfgverify Error Messages.” This command outputs warnings; all warnings must be verified manually.

Installing the Disk Plexing License

Install the disk plexing license in */etc/nodelock*.

Cloning the Disk

If your IRIS FailSafe system nodes are both CHALLENGE S servers, you can clone the disk of the first server. After you have finished installing and testing the software on the first server, follow these steps:

1. In the miniroot, make sure that IRIS FailSafe is *chkconfig*’d off.
2. Remove the system disk from the first server and install it in the secondary server.
3. Clone the disk.
4. Put the system disk from the first server back into the first server.
5. On the clone on the second server, change the following

   ```
   /etc/nodelock and /var/netls/nodelock (for any licenses)
   2ndservername 7#
   2ndservername 7# hostname -s 2ndservername
   2ndservername 8# echo 2ndservername > /etc/sys_id
   ```

6. Edit the interface names and other variables in */etc/config/netif.options* for the secondary server. For example, set the following for the second node in an active/standby configuration:

   ```
   if1name=ec3
   if1addr=$HOSTNAME
   if2name=ec0
   if2addr=$HOSTNAME-2
   ```

7. Update */etc/hosts* so that the name of the secondary server is the local server.

8. Use *chkconfig failsafe* on both servers.

   Caution: Do not use this command on a Web server configuration with ns_httpd on; the IRIS FailSafe software starts Netscape automatically, after it starts the necessary network interfaces.

9. Reboot both servers.

Appendix B

ha_cfgverify Error Messages

This appendix lists error messages and warnings that can appear as output of the *ha_cfgverify* command. This command aids in verifying the validity of the configuration file *ha.conf*. The error messages are given in alphabetical order.

```
ha_cfgverify: all nodes must be present in the server nodes
entry for main application class
```

Both nodes need server node entries for main application class.

```
ha_cfgverify: <name> application class does not have a server
node
```

All application class entries must have at least one server node entry.

```
ha_cfgverify: backup-node entry in filesystem <name> is missing
```

The filesystem section must have a server node and a backup node.

```
ha_cfgverify: backup-node entry <node name> in filesystem
<name> is not a valid node
```

The backup node does not have a node section in the configuration file.

```
ha_cfgverify: broadcast address for primary-ip not present for
node <node name>
ha_cfgverify: broadcast address for private-ip not present for
node <node name>
ha_cfgverify: broadcast address for secondary-ip not present
for node <node
name>
```

The broadcast address for primary-ip, secondary-ip, and private-ip must be specified in the *ha.conf* file.

```
ha_cfgverify: either remote or local monitor must be present
for <name> application class
```

All applications must have either local monitor or remote monitor.

```
ha_cfgverify: Failed to find hostname
```

The file */etc/sys_id* must have the hostname. The hostname must be configured using hostname *command hostname -s <host name>*.

```
ha_cfgverify: filesystem type entry in filesystem <name> is
missing
```

All filesystem sections must have mount point, filesystem type, device name, and mount mode entries.

```
ha_cfgverify: <type> filesystem type is not valid in filesystem
<name>
```

The filesystem can be either XFS or EFS.

```
ha_cfgverify: giveaway function <file name> is not valid in
<name> application class
ha_cfgverify: giveback function <file name> is not valid in
<name> application class
```

Either the filename is not present in the node or it does not have execute permission.

```
ha_cfgverify: heartbeat ip address not present for node <node
name>
```

The heartbeat IP address for the node must be specified. It is usually the private IP address.

```
ha_cfgverify: heartbeat_ip address <ip address> is not valid
for the node <node name>
```

The heartbeat IP address must be the private IP address. It must be same for all the nodes, that is, heartbeat IP address for all nodes must be the respective private IP addresses.

```
ha_cfgverify: heartbeat lost count for the node <node name> must
be specified
```

The heartbeat lost count must be specified for each node.

```
ha_cfgverify: heartbeat probe time for the node <node name> must
be specified
```

The heartbeat probe time in seconds. must be specified for each node.

```
ha_cfgverify: heartbeat timeout for the node <node name> must
be specified
```

The heartbeat timeout in seconds must be specified for each node.

```
ha_cfgverify: hostname <hostname> must be one of the node labels
```

The node label of the node section must be the node's hostname. The *ha_cfgverify* command must be run in one of the nodes of the cluster.

```
ha_cfgverify: interface name for primary-ip not present for
node <node name>
ha_cfgverify: interface name for private-ip not present for
node <node name>
ha_cfgverify: interface name for secondary-ip not present for
node <node name>
```

The interface name for primary-ip, secondary-ip, and private-ip must be specified.

```
ha_cfgverify: internal section: version-major value is invalid
(must be <num>)
```

The value of version-major field in the internal section must be <num>.

```
ha_cfgverify: internal section: version-minor value is invalid
(must be <num>)
```

The value of version-minor field in the internal section must be <num>.

```
ha_cfgverify: invalid checksum for /var/ha/ha.conf
```

The file */var/ha/ha.conf* cannot be opened for reading. There is a syntax error in the *ha.conf* file. There will be another message indicating the type of error.

```
ha_cfgverify: ip address for primary-ip not present for node
<node name>
ha_cfgverify: ip address for private-ip not present for node
<node name>
ha_cfgverify: ip address for secondary-ip not present for node
<node name>
```

The IP address for the primary-ip, secondary-ip, and private-ip must be specified in the *ha.conf* file.

```
ha_cfgverify: kill function <file name> is not valid in <name>
application class
```

Either the filename is not present in the node or it does not have execute permission.

```
ha_cfgverify: local monitor <file name> is not valid in <name>
application class
```

Either the filename is not present in the node or it does not have execute permission.

```
ha_cfgverify: local monitor probe time not specified for <name>
application class
ha_cfgverify: local monitor timeout not specified for <name>
application class
```

If the application has local monitor, local monitor timeout and probe time in seconds must be specified. If the application has remote monitor, remote monitor timeout and probe time in seconds must be specified.

```
ha_cfgverify: long-timeout value missing
```

The long-timeout value must be specified in the internal section of the *ha.conf* file.

```
ha_cfgverify: mac address for primary-ip not present for node
<node_name>
ha_cfgverify: mac address for secondary-ip not present for node
<node_name>
```

If re-mac'ing of interfaces is needed (the re-mac variable in the system configuration section of the configuration file is set to true), the MAC address of primary-ip and secondary-ip for the node *node_name* must be specified.

```
ha_cfgverify: main application class does not have all server
nodes
```

Either the main application class does not have server-node entries, or there is no main application class entry in application class section.

```
ha_cfgverify: main application class must have giveaway
function
ha_cfgverify: main application class must have giveback
function
ha_cfgverify: main application class must have kill function
ha_cfgverify: main application class must have takeback
function
ha_cfgverify: main application class must have takeover
function
ha_cfgverify: main application class must have start monitor
time
```

The main application class must have giveback, takeback, takeover, giveaway, and kill scripts. The start monitor time entry for main application class must be specified in seconds.

```
ha_cfgverify: missing node section
```

The configuration file *ha.conf* must have a node section describing the network interfaces.

```
ha_cfgverify: mount mode entry in filesystem <name> is missing
```

```
ha_cfgverify: mount point entry in filesystem <name> is missing
```

All filesystem sections must have mount point, filesystem type, server device name, server raw device name, backup device name, backup raw device name, and mount mode entries.

```
ha_cfgverify: mount point entry <name> in filesystem <name> is
present in /etc/fstab file
```

The mount point must not be present in the */etc/fstab* file. If it is present in the */etc/fstab* file, the “noauto” mount option must be specified. The IRIS FailSafe software mounts the directory.

```
ha_cfgverify: netmask for primary-ip not present for node <node
name>
ha_cfgverify: netmask for private-ip not present for node <node
name>
ha_cfgverify: netmask for secondary-ip not present for node
<node name>
```

The netmask for the primary-ip, secondary-ip, and private-ip must be specified in the *ha.conf* file.

```
ha_cfgverify: nfs is not configured in the system
```

Either the NFS software has not been configured in the system, or the NFS configuration flag is not *chkconfig*'d on.

```
ha_cfgverify: nfs <name> section: export point entry is invalid
```

The export point entry must be a subdirectory of the mount-point entry of the filesystem.

```
ha_cfgverify: nfs <name> section: export point entry is present
in /etc/exports file
```

The export point entry must not be present in the */etc/exports* file. It will be exported by the IRIS FailSafe software.

```
ha_cfgverify: nfs <name> section must have the export info entry
ha_cfgverify: nfs <name> section must have the export point
entry
ha_cfgverify: nfs <name> section must have the filesystem entry
```

All NFS sections must have filesystem, export-point, and export-info entries.

```
ha_cfgverify: nfs <name>: the filesystem entry <filesystem
name> is invalid.
```

The filesystem entry does not match the filesystem section labels.

```
ha_cfgverify: no application class section
```

The *ha.conf* file does not have application class section. The application class will have an entry for main application and optional entries for NFS and Web server applications.

```
ha_cfgverify: No nfs section present in the file
```

There is an NFS application class entry but there is no NFS section.

```
ha_cfgverify: No webserver section present in the file
```

The configuration file has a Web server application class entry, but there is no Web server section.

```
ha_cfgverify: node name <node_name> is not present in the hosts
database
```

The *node_name* must be present in the */etc/hosts* file, or its IP address can be obtained from the name server.

```
ha_cfgverify: <node_name> section: hb_use_public entry must be
set to “no” in hot standby configuration.
```

In an active / standby configuration (each node in the cluster has only two network interfaces), the hb_use_public entry must be set to “no”.

```
ha_cfgverify: primary-ip and secondary-ip information must be
present for the node <node_name>
```

The primary-ip and secondary-ip information is needed for all nodes in a dual-active configuration. In an active / standby configuration, primary-ip information is needed for one node and secondary-ip information is needed for the other node.

```
ha_cfgverify: primary-ip broadcast address <address> is not
valid for node <node name>
```

The broadcast address is not a valid internet address.

```
ha_cfgverify: primary-ip information must be present for the
node <node_name>
```

The primary-ip and secondary-ip information is needed for all nodes in a dual-active configuration. In an active / standby configuration, primary-ip information is needed for one node and secondary-ip information is needed for the other node.

```
ha_cfgverify: primary-ip interface name <interf. name> for node
<node name> is present in /etc/config/netif.options file
```

The primary-ip interface must not be configured using netif.options file. The IRIS FailSafe software configures the interface.

```
ha_cfgverify: primary-ip interface name <interface name> is not
valid for node <node name>
```

The interface name is not valid in the node. Check the interface name using the *netstat*(1) command.

```
ha_cfgverify: primary-ip mac address <mac_address> is not valid
for node <node_name>
```

The MAC address / physical address specified for primary-ip and secondary-ip is incorrect. Use *macconfig*(1M) command to find the correct physical address.

```
ha_cfgverify: primary-ip netmask <net mask> is not valid for
node <node name>
```

The netmask for the primary-ip, secondary-ip, or private-ip is not a valid internet netmask.

```
ha_cfgverify: private-ip broadcast address <address> is not
valid for node <node name>
```

The broadcast address is not a valid internet address.

```
ha_cfgverify: private-ip interface name <interface name> for
node <node name> must be present in /etc/config/netif.options
file
```

The private-ip must be specified in *netif.options* file.

```
ha_cfgverify: private-ip interface name <interface name> is not
valid for node <node name>
```

The interface name is not valid in the node. Check the interface name using the *netstat*(1) command.

```
ha_cfgverify: private-ip name <ip_name> for node <node_name>
must match the hostname
```

The private-ip name in the node section of the configuration file must be same as the hostname of that node.

```
ha_cfgverify: private-ip netmask <net mask> is not valid for
node <node name>
```

The netmask for the primary-ip, secondary-ip, or private-ip is not a valid internet netmask.

```
ha_cfgverify: remote monitor <file name> is not valid in <name>
application class
```

Either the filename is not present in the node or it does not have execute permission.

```
ha_cfgverify: remote monitor probe time not specified for
<name> application class
ha_cfgverify: remote monitor timeout not specified for <name>
application class
```

If the application has remote monitor, remote monitor timeout and probe time in seconds must be specified.

```
ha_cfgverify: reset-tty for the node <node name> must be
specified
```

The tty device which will used for resetting the other node must be specified.

```
ha_cfgverify: secondary-ip broadcast address <address> is not
valid for node <node name>
```

The broadcast address is not a valid internet address.

```
ha_cfgverify: secondary-ip information must be present for the
node <node_name>
```

The primary-ip and secondary-ip information is needed for all nodes in a dual-active configuration. In an active / standby configuration, primary-ip information is needed for one node and secondary-ip information is needed for the other node.

```
ha_cfgverify: secondary-ip interface name <interface name> for
node <node name> must be present in /etc/config/netif.options
file
```

The secondary-ip must be specified in *netif.options* file.

```
ha_cfgverify: secondary-ip interface name <interface name> is
not valid for node <node name>
```

The interface name is not valid in the node. Check the interface name using the *netstat*(1) command.

```
ha_cfgverify: secondary-ip mac address <mac_address> is not
valid for node <node_name>
```

The MAC address / physical address specified for primary-ip and secondary-ip is incorrect. Use *macconfig*(1M) command to find the correct physical address.

```
ha_cfgverify: secondary-ip netmask <net mask> is not valid for
node <node name>
```

The netmask for the primary-ip, secondary-ip, or private-ip is not a valid internet netmask.

```
ha_cfgverify: server device name entry in filesystem <name> is
missing
```

All filesystem sections must have mount point, filesystem type, device name, and mount mode entries.

```
ha_cfgverify: server-node entry in filesystem <name> is missing
```

The filesystem section must have a server node.

```
ha_cfgverify: server-node entry <node name> in filesystem
<name> is not a valid node;
```

The server node does not have node section.

```
ha_cfgverify: server node <node name> is not a valid node in
<name> application class
```

The server node in the application class does not have a node section.

```
ha_cfgverify: short-timeout value missing
```

The short-timeout value must be specified in the internal section of the *ha.conf* file.

```
ha_cfgverify: short-timeout value must be smaller than
long-timeout value
```

The long-timeout value in the internal section must be larger than the short-timeout value.

```
ha_cfgverify: Some xlv patches have not been installed in the
system
```

All XLV patches must be installed in the system. Check the release notes for the latest XLV patch numbers.

```
ha_cfgverify: statmon-dir must be present for nfs server node
<node_name>
```

The statmon-dir field in the application classes block of the configuration file must be specified for the NFS server node *node_name*.

```
ha_cfgverify: takeback function <file name> is not valid in
<name> application class
ha_cfgverify: takeover function <file name> is not valid in
<name> application class
```

Either the filename is not present in the node or it does not have execute permission.

```
ha_cfgverify: The /etc/config/routed.options must have -q
option
```

The routed daemon must be started with -q option. The routed options file */etc/config/routed.options* must have the *-q* option.

```
ha_cfgverify: Too many application classes: <num>
```

The IRIS FailSafe software supports up to 16 application class entries.

```
ha_cfgverify: Too many node entries: <num>. Only 2 two nodes
are permitted
```

The configuration file *ha.conf* can have only two node sections. The IRIS FailSafe software permits only clusters of two nodes.

```
ha_cfgverify: version-major value missing in internal section
ha_cfgverify: version-minor value missing in internal section
```

The internal section must have version-major and version-minor fields.

```
ha_cfgverify: Warning: check if xlv license has been installed
```

If XLV plexing is being used, a plexing license must be installed in the */etc/nodelock* file.

```
ha_cfgverify: Warning: device name entry in filesystem <name>
is invalid
```

Either the filename is not present in the node or it does not have execute permission.

```
ha_cfgverify: Warning: mount point entry <name> in filesystem
<name> is not a valid directory
```

The mount point must be a valid directory.

```
ha_cfgverify: Warning: netscape is not configured in the system
```

Either the Web server software has not been configured in the system, or the ns_httpd configuration flag is not *chkconfig*'d on.

```
ha_cfgverify : Warning NFS lockd/statd patch for Failsafe
(1032) is not installed in the system
```

The NFS *lockd/statd* patch (see the release notes for the patch number) is necessary for failing over NFS locks for the NFS service.

```
ha_cfgverify: Warning: no filesystems found in the
configuration file
```

The *ha.conf* file does not have filesystem section. Most IRIS FailSafe system configurations require a filesystem section. Check if your system configuration requires a filesystem section in the *ha.conf* file.

```
ha_cfgverify: webserver <section_name> section must have the
backup-node entry
ha_cfgverify: webserver <section_name> section must have the
server-node entry
```

The webserver <section_name> block of the configuration file must have server-node and backup-node entries.

```
ha_cfgverify: webserver <section_name> : the backup-node entry
<node_name> is invalid
ha_cfgverify: webserver <section_name> : the server-node entry
<node_name> is invalid
```

The server-node and backup-node entries must have a node section in the configuration file.

Appendix C

Keywords

This appendix is an alphabetical list of keywords used in the configuration file *ha.conf*.

- action
- action-timer
- application-class
- backup-node
- broadcast
- devname
- export-info
- export-point
- filesystem
- giveaway
- giveback
- hb-lost-count
- hb-probe-time
- hb-timeout
- hb-use-public
- heartbeat
- http-timeout
- httpd-dir
- interface
- internal
- ip-name

- kill
- lmon-probe-time
- lmon-timeout
- local-monitor
- long-timeout
- mail-dist-addr
- main
- mode
- mount-info
- mount-point
- netmask
- nfs
- node
- port-num
- primary-ip
- private-ip
- pwrfail
- re-mac
- remote-monitor
- reset-host
- reset-tty
- retry-count
- rmon-probe-time
- rmon-timeout
- secondary-ip
- server-node
- short-timeout

- start-monitor-time
- statmon-dir
- system-configuration
- takeback
- takeover
- web-config<number>
- webserver
- webserver-num

Appendix D

Sample Configuration Files

This appendix gives the full code for *ha.conf.nfs_dual_active* and *ha.conf.web_active_standby*, which are used as examples in Chapter 4. These files are among those available in */var/ha/samples*.

See Chapter 4, “Creating the Configuration File,” for block-by-block information on the configuration file.When you are finished editing the file, save it as */var/ha.conf*.

ha.conf.nfs_dual_active

This section presents the configuration file *ha.conf.nfs_dual_active*.

```
#       Sample configuration file
#
#       Dual-active NFS services.
#
# This file describes the configuration for a pair of
highly-available
# servers configured for NFS.
#
# The two nodes have the hostnames: xfs-ha5 and xfs-ha6.
#
# The cluster as a whole exports two filesystems:
#
#     stocks:/shared1 and bonds:/shared2.
#
# Note that client machines access these mount points under
the
# advertised names of stocks & bonds. (The hostnames xfs-ha5
and
# xfs-ha6 are not advertised over the public network.)
#
```

```
# In normal operation, both servers are active and service
requests
# to different filesystems. xfs-ha5 will service all requests
# directed to stocks and xfs-ha6 will service all requests
directed
# to bonds. When xfs-ha5 fails, xfs-ha6 will take over the IP
# address for stocks and export /shared1.
# Thus, the cluster as a whole would still offer these
filesystems
# even should a machine fail.
# Machine xfs-ha5 will likewise takeover the IP address of
bonds
# and export /shared2 should xfs-ha6 fail.


system-configuration
{
        # If this value is defined, then the HA software
will send
        # a mail message to the recipient when:
        #   1) private network failure has been detected,
        #   2) local HA process (node controller) appears to
be hung or dead,
        #   3) cluster is transitioning to degraded mode,
        #   4) cluster is transitioning to standby state,
        #   5) killing of a node fails,
        #   6) ha_killd daemon died,
        #   7) could not start the ha_killd daemon,
        #   8) the reset device monitor failed.
        #
        # Make sure that mail has been configured on your
system before
        # defining this value.

        mail-dest-addr = root@localhost

        # The re-mac value should be set to true, if the
network
        # interfaces have to be re-mac’ed when a failover
occurs.
        # If the re-mac value is not defined, it defaults to
false.

        re-mac = false
        # If an IRISConsole is used to provide reset
functionality, set
        # “reset-host = ops-indy” where ops-indy is the
hostname of the Indy
        # running IRISConsole software. Use of the
IRISConsole reset
        # functionality is only supported when HA runs on an
OPS (Oracle
        # Parallel Server) cluster. Reset-host is ignored if
reset-tty is set
        # in the “node” section below.

        # reset-host = ops-indy

        # When a Challenge DM/L/XL system loses power, the
heartbeat and the
        # failsafe mechanism both fail at the same time.
i.e. there is no
        # means to know for sure that applications on the
Challenge server have
        # terminated. Setting “pwrfail = true” allows the
FailSafe software to
        # takeover the failed node when both the heartbeat
and the failsafe
        # mechanism fail within the same ’small’ time
interval. On the
        # Challenge S, this option is ignored, since the
failsafe mechanism and
        # the heartbeat are independently powered.

        pwrfail = true
}


# The “node” section of the configuration file describes the
# network interfaces to the machines in the cluster.
#
#  Each of the two machines (xfs-ha5 and xfs-ha6) has 3
network
#  interfaces:
#
#    ec0 is the interface over which clients request
services. It is
#        therefore associated with the IP address of the
service -
```

```
#        in this case: stocks or bonds.
#        This interface is not configured by default. The HA
#        software will configure it based upon the state of
the
#        machines in the cluster.
#
#    ec2 is a spare interface. A node will use this to
service
#        requests destined for the other node if it needs to
#        take over the services offered by the other node.
#        Normally, it is configured as $HOSTNAME-2. When it
#        is used as a backup, it will be configured as one of
#        the service interfaces: stocks or bonds.
#        The spare interface is only present in dual-active
configurations.
#
#    ec3 is the network interface to a private network
between
#        machines A and B. This is used for keep-alive and
other
#        control messages. This is configured to be the
hostname
#        (xfs-ha5 or xfs-ha6).
#
#  For example,
#
#    When both machines are up and exporting their own
services,
#    then ec0 on xfs-ha5 would have the IP address stocks
while
#    ec0 on xfs-ha6 would have the IP address bonds.
#
#    If xfs-ha5 died and xfs-ha6 takes over its services,
then
#    ec0 on xfs-ha6 would still have its own IP address
(bonds),
#    ec2 on xfs-ha6 would now have the IP address of stocks.

# FORMAT:
#       1. The node label (xfs-ha5) must match the return
value of
#          ’hostname’ on the relevant machine.
#       2. The “ip-name” value must be a name and not an
#          address (X.X.X.X)
#
```

```
node xfs-ha5
{
        # The values specified in the “primary-ip”,
“secondary-ip”,
        # and “private-ip” are those passed to the
ifconfig(1M)
        # command.
        #
        # The mac-addresses for primary-ip and secondary-ip
are
        # required only if the network interfaces have to be
re-mac’ed.
        # This value can be obtained using macconfig(1M)
command.
        primary-ip
        {
                interface = ec0
                ip-name = stocks
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
                mac-address = 8:0:69:8:94:36
        }
        secondary-ip
        {
                interface = ec2
                ip-name = xfs-ha5-2
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
                mac-address = 8:0:69:2:60:bf
        }
        private-ip
        {
                interface = ec3
                ip-name = xfs-ha5
                netmask = 255.255.255.0
                broadcast = 192.50.165.255
        }
        heartbeat
        {
                #
                # The heartbeat goes over the private network
                #
                ip-name = xfs-ha5

                #
                # The following three values determine how
often
                # (hb-probe-time), how long to wait
(hb-timeout),
                # and now many retries (hb-lost-count) must
fail
                # before a heartbeat failure is declared.
                # In the worst case, the heartbeat failure
will
                # be declared after:
                #    hb-lost-count * hb-probe-time *
hb-timeout
                # or twice the above value if hb-use-public
is ’yes’
                #
                hb-probe-time = 3
                hb-timeout = 3
                hb-lost-count = 3

                #
                # This value, if set to “yes”, allows the
heartbeat
                # to go over the public network (primary-ip)
if
                # there is a private network failure.
                # If not defined, the value defaults to “yes”
                #
                hb-use-public = yes
        }
        #
        # The reset-tty is the device file name of the
serial connection.
        # In Challenge S, the serial link is connected to
the remote power
        # control unit. In Challenge DM/L/XL, the serial
link is connected
        # to the system controller of the other machine in
the cluster.
        #
        reset-tty = /dev/ttyd2
}

node xfs-ha6
{
        primary-ip
        {
                interface = ec0
                ip-name = bonds
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
                mac-address = 8:0:69:8:95:c3
        }
        secondary-ip
        {
                interface = ec2
                ip-name = xfs-ha6-2
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
                mac-address = 8:0:69:8:94:36
        }
        private-ip
        {
                interface = ec3
                ip-name = xfs-ha6
                netmask = 255.255.255.0
                broadcast = 192.50.165.255
        }
        heartbeat
        {
                ip-name = xfs-ha6
                hb-probe-time = 3
                hb-timeout = 3
                hb-lost-count = 3
                hb-use-public = yes
        }
        reset-tty = /dev/ttyd2
}


#
# The application classes are the different HA services
provided by a node.
# Each one of these HA services is provided by at least one
node (server-node).
# For the first release we will only fail-over the node,
that is why we
# define the ’main’ application class (we don’t fail-over
individual
# applications or application classes).
#
```

```
# FORMAT: The values assigned to server-node must match one
of the node labels
#       provided in a node object.
#
application-class
{
        main
        {
                server-node = xfs-ha5
                server-node = xfs-ha6
        }

        # Both xfs-ha5 and xfs-ha6 offer NFS services
concurrently.
        #
        # Each node which acts as a primary server must
create exactly
        # one directory to store NFS locking information.
The directory
        # needs to be placed in one of the shared
filesystems for that node.
        # It should also be a dedicated directory.  For
example, node
        # “xfs-ha5” is primary for two filesystems, shared2
& shared3.  We
        # need to define “statmon-dir” under one of these
filesystems,
        # e.g. “/shared2/statmon”.  Do not use just
“/shared2” as the
        # directory.
        #
        nfs
        {
                server-node xfs-ha5
                {
                        statmon-dir = /shared1/statmon
                }
                server-node xfs-ha6
                {
                        statmon-dir = /shared2/statmon
                }
        }
}
```

```
# DISK/FILESYSTEM CONFIGURATION
#
#   The HA software will failover filesystems on shared
disks.
#   This allows a backup machine to takeover filesystems
should
#   the primary machine fail.
#
#   The shared filesystems in this sample configuration are:
#
#     /shared1 is normally mounted on xfs-ha5. This
filesystem is
#     created on an XLV volume.
#
#     /shared2 is normally mounted on xfs-ha6. This
filesystem is
#     created on a different XLV volume.
#
#     /shared3 is normally mounted on xfs-ha6. This
filesystem is
#     created on a different XLV volume.
#

# The “filesystem” section of the configuration file
describes the
# values/flags that are passed to mount(1M).
#
# FORMAT: The values assigned to server-node must match one
of the node labels
#       provided in a node object.
#
filesystem shared1
{
        server-node = xfs-ha5
        backup-node = xfs-ha6
        mount-point = /shared1
        mount-info
        {
                fs-type = xfs
                devname = /dev/dsk/xlv/vol1
                mode = rw,noauto
        }
}
```

```
#
# FORMAT: The values assigned to server-node must match one
of the node labels
#       provided in a node object.
#
filesystem shared2
{
        server-node = xfs-ha6
        backup-node = xfs-ha5
        mount-point = /shared2
        mount-info
        {
                fs-type = xfs
                devname = /dev/dsk/xlv/vol2
                mode = rw,noauto
        }
}

#
# FORMAT: The values assigned to server-node must match one
of the node labels
#       provided in a node object.
#
filesystem shared3
{
        server-node = xfs-ha6
        backup-node = xfs-ha5
        mount-point = /shared3
        mount-info
        {
                fs-type = xfs
                devname = /dev/dsk/xlv/vol3
                mode = rw,noauto
        }
}

# EXPORTED NFS FILESYSTEMS
#
#   The NFS filesystems that are exported in our sample
configuration
#   are:
#
#     /shared1 from stocks (normally xfs-ha5) and /shared2 &
/shared3 from
```

```
#     bonds (normally xfs-ha6).
#
#   The “nfs” section of the configuration file specify the
values/flags
#   that are specified in the exportfs(1M) command. One
“nfs” section
#   must be present for every filesystem to be exported.
Note that the
#   “filesystem” field will reference the physical
filesystem that
#   is exported.

#
# FORMAT: The values assigned to filesystem must match one
of the filesystem
#       labels provided in a filesystem object.
#
nfs shared1
{
        filesystem = shared1
        export-point = /shared1
        export-info = rw
}

#
# FORMAT: The values assigned to filesystem must match one
of the filesystem
#       labels provided in a filesystem object.
#
nfs shared2
{
        filesystem = shared2
        export-point = /shared2
        export-info = rw
}

#
# FORMAT: The values assigned to filesystem must match one
of the filesystem
#       labels provided in a filesystem object.
#

nfs shared3
{
        filesystem = shared3
        export-point = /shared3
        export-info = rw,anon=root
}

#
# For each application class to be monitored, the following
parameters
# have to be defined.
#
#       action:
#               pathnames of the respective monitoring
scripts.
#
#       action-timer:
#               timers that affect the monitoring of the
respective
#               application classes
#
action nfs
{
        local-monitor = /var/ha/actions/ha_nfs_lmon
        remote-monitor = /var/ha/actions/ha_nfs_rmon
}

action-timer nfs
{
        #
        # once a node releases an application class
(giveaway), it will
        # start the remote monitoring of it (application
class) in
        # start-monitor-time seconds.
        #
        start-monitor-time = 30

        #
        # local monitoring will be done every
lmon-probe-time seconds and
        # will timeout in lmon-timeout seconds.
        #
        lmon-probe-time = 20
        lmon-timeout = 30

        #
        # remote monitoring will be done every
rmon-probe-time seconds and
        # will timeout in rmon-timeout seconds.
        #
        rmon-probe-time = 60
        rmon-timeout = 45

        #
        # some of the monitoring scripts do internal retries
(i.e. the
        # application monitor sees it as a single ’probe’
but the actual
        # script can do retry-count ’probes’
        #
        retry-count = 1
}


#------------------------------------------------------------
-----------------
#
# Do not change anything below the line
#
internal
{
        short-timeout = 5
        long-timeout = 60
        version-major = 1
        version-minor = 0
}

action main
{
        giveaway = /var/ha/actions/giveaway
        giveback = /var/ha/actions/giveback
        takeback = /var/ha/actions/takeback
        takeover = /var/ha/actions/takeover
        kill = /usr/etc/ha_kill
}

action-timer main
{
        start-monitor-time = 30
}
```

ha.conf.web_active_standby

This section presents the configuration file *ha.conf.web_active_standby*.

```
#
#       Sample configuration file
#
#
# This file describes the configuration for an active/standby
# Web server configuration.
#
# The cluster, with the external name of stocks, is
configured such
# that all the documents are stored in a subdirectory of
/shared.
# The 2 machines that make up the cluster are xfs-ha5 and
xfs-ha6.
# Normally, the IP address stocks is exported by xfs-ha5.
Should
# xfs-ha5 fail, xfs-ha6 will takeover the IP address of
stocks, mount
# the shared filesystem /shared, and continue to service web
requests.


system-configuration
{
        # If this value is defined, then the HA software
will send
        # a mail message to the recipient when:
        #   1) private network failure has been detected,
        #   2) local HA process (node controller) appears to
be hung or gone,
        #   3) cluster is transitioning to degraded mode,
        #   4) cluster is transitioning to standby state,
        #   5) killing of a node fails,
        #   6) ha_killd daemon died,
        #   7) could not start the ha_killd daemon,
        #   8) the reset device monitor failed.
        #
        # Make sure that mail has been configured on your
system before
        # defining this value.

        mail-dest-addr = root@localhost
        # The re-mac value should be set to true, if the
network
        # interfaces have to be re-mac’ed when a failover
occurs.
        # If the re-mac value is not defined, it defaults to
false.

        re-mac = false

        # If an IRISConsole is used to provide reset
functionality, set
        # “reset-host = ops-indy” where ops-indy is the
hostname of the Indy
        # running IRISConsole software. Use of the
IRISConsole reset
        # functionality is only supported when HA runs on an
OPS (Oracle
        # Parallel Server) cluster. Reset-host is ignored if
reset-tty is set
        # in the “node” section below.

        # reset-host = ops-indy

        # When a Challenge DM/L/XL system loses power, the
heartbeat and the
        # failsafe mechanism both fail at the same time.
i.e. there is no
        # means to know for sure that applications on the
Challenge server have
        # terminated. Setting “pwrfail = true” allows the
FailSafe software to
        # takeover the failed node when both the heartbeat
and the failsafe
        # mechanism fail within the same ’small’ time
interval. On the
        # Challenge S, this option is ignored, since the
failsafe mechanism and
        # the heartbeat are independently powered.

        pwrfail = true
}


# The “node” section of the configuration file describes the
# network interfaces to the machines in the cluster.
#
#  Since this is an active/standby configuration, each of
the two
#  machines has 2 network interfaces:
#
#    ec0 is the interface over which clients request
services. It is
#        associated with the IP address of the service.
#        This interface is not configured by default. The HA
#        software will configure it based upon the state of
the
#        machines in the cluster.
#
#    ec3 is the network interface to a private network
between
#        machines A and B. This is used for keep-alive and
other
#        control messages.
#
#  Note:
#       1. The node label (xfs-ha5) must match the return
value of
#          ’hostname’ on the relevant machine.
#       2. The “ip-name” value must be a name and not an
#          address (X.X.X.X)
#
node xfs-ha5
{
        # In an active/standby configuration, the primary
node
        # needs to define the primary-ip and private-ip
sections.
        # It should not define a secondary-ip section since
it
        # does not need to backup the standby system.
        #
        # The mac-addresses for primary-ip and secondary-ip
are
        # required only if the network interfaces have to be
re-mac’ed.
        # This value can be obtained using macconfig(1M)
command.
        primary-ip
        {
                interface = ec0
                ip-name = stocks
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
                mac-address = 8:0:69:8:94:36
        }
        private-ip
        {
                interface = ec3
                ip-name = xfs-ha5
                netmask = 255.255.255.0
                broadcast = 192.50.165.255
        }
        heartbeat
        {
                #
                # The heartbeat goes over the private network
                #
                ip-name = xfs-ha5

                #
                # The following three values determine how
often
                # (hb-probe-time), how long to wait
(hb-timeout),
                # and now many retries (hb-lost-count) must
fail
                # before a heartbeat failure is declared.
                # In the worst case, the heartbeat failure
will
                # be declared after
                #       hb-lost-count * hb-probe-time *
hb-timeout
                #
                hb-probe-time = 3
                hb-timeout = 3
                hb-lost-count = 3

                #
                # This value, if set to “yes”, allows the
heartbeat
                # to go over the public network (primary-ip)
if
                # there is a private network failure.
                # If not defined, the value defaults to “yes”
                #
                # NOTE that this should not defined if you
have
                # an active/standby configuration.
                #
                hb-use-public = no
        }
        #
        # The reset-tty is the device file name of the
serial connection.
        # In Challenge S, the serial link is connected to
the remote power
        # control unit. In Challenge DM/L/XL, the serial
link is connected
        # to the system controller of the other machine in
the cluster.
        #
        reset-tty = /dev/ttyd2
}

node xfs-ha6
{
        # In an active/standby configuration, the backup node
        # needs to define the secondary-ip and private-ip
sections.
        secondary-ip
        {
                interface = ec0
                ip-name = xfs-ha6-2
                netmask = 255.255.255.0
                broadcast = 192.48.165.255
        }
        private-ip
        {
                interface = ec3
                ip-name = xfs-ha6
                netmask = 255.255.255.0
                broadcast = 192.50.165.255
        }
        heartbeat
        {
                ip-name = xfs-ha6
                hb-probe-time = 3
                hb-timeout = 3
                hb-lost-count = 3
                hb-use-public = no
        }
        reset-tty = /dev/ttyd2
}


#
# The application classes are the different HA services
provided by a node.
# Each one of these HA services is provided by at least one
node (server-node).
# For the first release we will only fail-over the node,
that is why we
# define the ’main’ application class (we don’t fail-over
individual
# applications or application classes).
#
# FORMAT: The values assigned to server-node must match one
of the node labels
#       provided in a node object.
#
application-class
{
        main
        {
                server-node = xfs-ha5
                server-node = xfs-ha6
        }

        # Only xfs-ha5 offers the Web service. xfs-ha6 will
only become
        # active should xfs-ha5 fail. We chose this
configuration because
        # we assume that there is a single directory on the
shared
        # filesystem that contain all the web documents.
Thus, clients
        # will always access them via http://stocks/...

        webserver
        {
                server-node = xfs-ha5
        }
}
```

```
# DISK/FILESYSTEM CONFIGURATION
#
#   The HA software will failover filesystems on shared
disks.
#   This allows a backup machine to takeover filesystems
should
#   the primary machine fail.
#
#   The shared filesystems in this sample configuration are:
#
#     /shared is normally mounted on xfs-ha5. This
filesystem is
#     created on an XLV volume.
#

# The “filesystem” section of the configuration file
describes the
# values/flags that are passed to mount(1M).
#
# FORMAT: The values assigned to server-node must match one
of the node labels
#       provided in a node object.
#
filesystem shared
{
        server-node = xfs-ha5
        backup-node = xfs-ha6
        mount-point = /shared
        mount-info
        {
                fs-type = xfs
                devname = /dev/dsk/xlv/vol1
                mode = rw,noauto
        }
}


#
#  WEBSERVERS
#
#    This section describes the configuration of the highly
#    available webservers.
#    Each web server configuration, web-config<number> has
the server
```

```
#    port number, port-num and the server root location,
httpd-dir.
#    The server root location is usually a directory in the
shared
#    filesystem.
#
#    There are two web servers configured. The web server
#    configuration, web-config1 has the server port number
as 80,
#    and the server root location /shared/httpd-80. The
#    web-config2 configuration has the port number 90 and the
#    server root location /shared/httpd-90.192.48.165.40
#
#    The web servers will be served by the node xfs-ha5. The
#    node xfs-ha6 will provide web service when the node
xfs-ha5
#    fails to provide the service.
#    The clients can access the two web servers as
#              http://stocks:80/... and
#              http://stocks:90/...
#
#    The values provided for server-node and backup-node
must match
#    one of the node labels. The value for the webserver-num
is the
#    count of webservers configured.
#    Note: “web-config<number>” is a keyword.
#

webserver webxfs-ha5 {
        server-node = xfs-ha5
        backup-node = xfs-ha6
        webserver-num = 2

        web-config1 {
                port-num = 80
                httpd-dir = /shared/httpd-80
        }
        web-config2 {
                port-num = 90
                httpd-dir = /shared/httpd-90.192.48.165.40
        }
}

action webserver
```

```
{
        local-monitor = /var/ha/actions/ha_web_lmon
        remote-monitor = /var/ha/actions/ha_web_rmon
}

action-timer webserver
{
        start-monitor-time = 30
        lmon-probe-time = 20
        lmon-timeout = 30
        rmon-probe-time = 60
        rmon-timeout = 45
        retry-count = 1
}


# Contents of /etc/config/netif.options:
#
# For xfs-ha5
#
# Note that we explicitly put in an entry for
if2addr/if2name because
# we do not want it to be configured by default as
gate-$HOSTNAME.
#
# if1name=ec3
# if1addr=$HOSTNAME
# if2addr=
# if2name=
#
# For xfs-ha6 (the backup machine)
#
# if1name=ec3
# if1addr=$HOSTNAME
# if2name=ec0
# if2addr=$HOSTNAME-2


#-------------------------------------------------------------
-----------------
#
# Do not change anything below the line
#
internal
{
        short-timeout = 5
        long-timeout = 60
        version-major = 1
        version-minor = 0
}

action main
{
        giveaway = /var/ha/actions/giveaway
        giveback = /var/ha/actions/giveback
        takeback = /var/ha/actions/takeback
        takeover = /var/ha/actions/takeover
        kill = /usr/etc/ha_kill
}

action-timer main
{
        start-monitor-time = 30
}
```

Appendix E

System Maintenance and Troubleshooting

This appendix explains

- monitoring the serial connection
- troubleshooting system problems
- replacing batteries in the remote power control unit

Monitoring the Serial Connection

To stop monitoring the serial connection to a server, run

ha_admin -m stop *servername*

This command returns

```
ha_admin: Stopped monitoring the serial connection to xfs-ha5
```

Run

ha_spng -i10 -f /dev/ttjd2

and check the return code. If you are running *csh*,

```
xfs-ha6 14# ha_spng -i 10 -f /dev/ttyd2
xfs-ha6 15# echo $status
0
```

The zero at the end of this output indicates normal operation. If you are running *sh*, test the return code of *ha_spng* by checking $?:

```
# ha_spng -i 10 -f /dev/ttyd2
# echo $?
0
```

The zero at the end of this output indicates normal operation.

To start monitoring the serial connection to a server, run

`ha_admin -m start` *servername*

This command returns

```
ha_admin: Started monitoring the serial connection to
<servername>serial
```

Troubleshooting System Problems

When you encounter a failure, the first thing to do is to secure the integrity of your data. Use **df**(1M) to make sure that all of your shared filesystems are only mounted on one node; no filesystem should be simultaneously mounted by two nodes. After you use **df**(1M), look in */var/adm/SYSLOG* for causes of failure.

Other problems that might arise are

- duplicate SCSI IDs
- volumes that have not been mounted
- Netscape server warning messages at startup
- inability to access a node via the network
- trouble mounting shared filesystems
- trouble accessing the network interfaces on a cluster node
- inability to access a shared filesystem over NFS
- Web server not responding
- IRIS FailSafe does not start
- local monitor failures
- recovery script failures
- errors logged to */var/adm/SYSLOG*

Duplicate SCSI IDs

If you see SCSI bus-related errors after configuring the cluster, or nonexistent devices show up in **hinv**(1M), follow these steps:

1. Verify that the SCSI host IDs of the two nodes are different by running **nvram**(1M).
2. Verify that the SCSI IDs of all disks and other peripherals on the same SCSI bus have distinct SCSI unit numbers and that they are different from the SCSI host IDs of the two nodes in the cluster.

Volumes That Have Not Been Mounted

To fix this problem, follow these steps:

1. Verify that the system is licensed for plexing and that the plexing software is installed:

   ```
   xfs-ha5 1# xlv_mgr
   xlv_mgr> show config
   Allocated subvol locks: 30 locks in use: 7
   Plexing license: present
   Plexing support: present
   Maximum subvol block number: 0x7fffffff
   ```

   If you have just installed the plexing software, you must reboot he system for the plexing support to be included in the kernel.

2. Verify that the system sees the volume. In an IRIS FailSafe environment, a node accesses (and mounts filesystems on) only those XLV volumes that it owns. To see all the volumes in the cluster, run

   ```
   xlv_mgr
   xlv_mgr> show all
   Volume: vol1 (complete)
   Volume: vol2 (complete)
   Volume: shared_vol (complete)
   ```

   This command shows all the XLV volumes in the cluster.

3. To see volumes owned by a cluster node, type on that node

   ```
   xlv_assemble -ln

   VOL vol2        flags=0x1, [complete]
   DATA    flags=0x0()     open_flag=0x0() device=(192, 4)
   PLEX 0  flags=0x0
   VE 0    [active]
           start=0, end=687999, (cat)grp_size=1
           /dev/dsk/dks5d9s0 (688000 blks)
   PLEX 1  flags=0x0
   VE 0    [active]
           start=0, end=687999, (cat)grp_size=1
           /dev/dsk/dks5d9s1 (688000 blks)
   ```

   This command displays only the volumes owned by this node.

4. If a volume is owned by the wrong node, change the ownership of a volume (for example, to make vol2 owned by xfs-ha6) by first dismounting the filesystem mounted on that volume (vol2):

   ```
   xfs-ha5 2# umount /vol2
   xfs-ha5 3# xlv_mgr
   xlv_mgr> change nodename xfs-ha6 vol2
   set node name “xfs-ha6” for object “vol2” done
   ```

5. Run *xlv_assemble -l* on both xfs-ha6 and xfs-ha5.

Netscape Server Warning Messages at Startup

The error messages

```
error: could not bind to 192.48.165.92 port 80 (Cannot
assign requested address)
error: could not bind to 192.48.165.94 port 80 (Cannot
assign requested address)
```

are normal. They appear because the Netscape Communications Server (*ns_httpd*) is started before the IRIS FailSafe configures the network interfaces up. (In an IRIS FailSafe system, the high-availability software configures the network interfaces.) The IRIS FailSafe software automatically starts the Netscape Communications Server.

Inability to Access a Node via the Network

If you cannot access a node using the network, run *netstat -i* to see if the IP address to which you are tying to connect is configured on one of the node's public interfaces.

Because the primary interface is configured by IRIS FailSafe, it might not be configured if IRIS FailSafe is not started. Also, the IP address might have been taken over by the other node.

The following shows typical output of *netstat -i* t for the IRIS FailSafe system.

```
xfs-ha6 4# netstat -i
Name Mtu   Network     Address          Ipkts Ierrs    Opkts Oerrs  Coll
ec3  1500  197.50.50   xfs-ha6.engr.sg   1781     0     1752     0     1
ec0  1500  b7l-oslab   stocks           37844     1     2246     0  1692
ec2* 1500  none        none                 0     0        0     0     0
lo0  8304  loopback    localhost         8241     0     8241     0     0
```

Note: You cannot access an IP address associated with a private interface from a node on the public network.

Trouble Mounting Shared Filesystems

If you are having trouble mounting shared filesystems, execute the mount directive that would be executed by the IRIS FailSafe software. If your */var/ha/ha.conf* file contains directives like the following:

```
...

        server-node = xfs-ha5
        backup-node = xfs-ha6
        mount-point = /shared
        mount-info
        {
                fs-type = xfs
                devname = /dev/dsk/xlv/vol1
                mode = rw,noauto
        }
```

then run

```
# mount -txfs -rw,noauto /dev/dsk/xlv/vol1 /shared
```

Run this command for every filesystem. Run it on the other node after you umount them here. Make sure you do not mount the same filesystem simultaneously from both nodes, which would cause data corruption.

Trouble Accessing the Network Interfaces on a Cluster Node

If you cannot access a node's network interfaces, execute the *ifconfig* command that the IRIS FailSafe software would execute, and verify that it works. Run this command on both nodes for the private and primary interfaces; for a dual-active configuration, run it also for the secondary interfaces.

The following example shows how IRIS FailSafe uses the parameters in the node block of the configuration file *ha.conf* to configure an interface up.

```
xfs-ha5# ifconfig ec2 inet xfs-ha5-2 up netmask
255.255.255.0 broadcast 192.50.165.255
```

Then ping the interface.

Inability to Access a Shared Filesystem Over NFS

In this case, make sure that the network interface is *ifconfig*'d up and the filesystem is mounted, as explained in "Trouble Accessing the Network Interfaces on a Cluster Node" earlier in this section. Export the filesystems manually and see if a client can access it. Repeat the process on the other node.

If *ha.conf* has an entry with the following form:

```
nfs shared1
{
        filesystem = shared1
        export-point = /shared1
        export-info = rw
}
```

then follow these steps:

1. Verify that */shared1* is mounted. If it is not, follow instructions in "Volumes That Have Not Been Mounted" earlier in this section.
2. Run

   ```
   xfs-ha5# exportfs -i -o rw /shared1
   ```

3. Verify that the filesystem is exported:

   ```
   xfs-ha5 90# exportfs
   /shared1 -rw
   ```

4. Unexport it and unmount it so that you can redo this test from the other node and avoid simultaneously mounting this filesystem from it.

   ```
   exportfs -u /shared1
   umount /shared1
   ```

Web Server Not Responding

If a Web server is not responding after the network and the Web server have been installed and configured, make sure that the addresses you are using are configured up. Follow these steps:

1. Start up the Web servers

   ```
   chkconfig ns_httpd on
   ```

   By default, *ns_httpd* is normally *chkconfig*'d off.
2. Run

   ```
   /etc/init.d/ns_httpd start
   ```

   If you have multiple Web servers, you might need to create a */etc/config/ns_httpd.options* file.
3. run a Web browser, such as Netscape, and try to access some Web pages exported by the server.

IRIS Failsafe System Does Not Start

If the IRIS FailSafe system does not start, follow these steps:

1. Make sure that IRIS FailSafe is *chkconfig*'d on.
2. Make sure that */var/ha/ha.conf* is identical on both nodes in the cluster.
3. Run *ha_cfgverify*.
4. Verify the network interfaces, serial connections, and filesystems.
5. Look at */var/adm/SYSLOG* to see what errors are printed out by the IRIS FailSafe daemons. When a node in the IRIS FailSafe cluster starts up, the following *syslog* messages appear;

   ```
   ha_appmon[6141]: Received XRELEASE_PEER
   ha_nc[6135]: Received JOINING
   ha_nc[6135]: Received JOINING
   ha_nc[6135]: New state: NC_JOINING
   ha_nc[6135]: Received REJOIN
   ha_appmon[6141]: Received XACQUIRE
   ha_appmon[6141]: Received START_REMMON
   ha_nc[6135]: New state: NC_NORMAL
   ```

   If the node is part of an active / standby cluster, this message appears on the standby node:

   ```
   root: /var/ha/actions.d/takeback/S800interfaces: xfs-ha6
   has no primary interface, nothing to takeback
   ```

Local Monitor Failures

When the IRIS FailSafe software detects a local application failure, the event is written to */var/adm/SYSLOG*. The following shows the entries that are written when it detects that the http daemons are no longer responding:

```
Nov 29 11:25:36 6D:xfs-ha6 syslog[775]: /usr/etc/ha_exec:
command /usr/etc/http_ping failed error=1. retrying
Nov 29 11:25:36 5B:xfs-ha6 root: Failed to ping local
webserver [port: 80]
Nov 29 11:25:36 6D:xfs-ha6 ha_appmon[241]: webserver_xfs-ha5
local monitoring failed: status = 3
Nov 29 11:25:36 6D:xfs-ha6 ha_nc[235]: Received LOCMONFAIL
```

Note that when a node detects a local failure, it hands off its services to the other node. If this process was successful, the failed node would not be power-cycled. Thus you can diagnose the failure and reintegrate the node after it has been fixed.

Recovery Script Failures

IRIS FailSafe uses application-specific recovery scripts,. If a script fails, the error is logged to */var/adm/SYSLOG*. A sample entry might look like the following:

```
Nov 29 11:07:32 5B:xfs-ha5 root: ERROR: /sbin/xlv_mgr
/tmp/.ha_a.tak
Nov 29 11:07:32 6D:xfs-ha5 ha_appmon[238]: takeback script
exited with status 3
Nov 29 11:07:32 6D:xfs-ha5 ha_nc[232]: process appmon died
with status 1
```

Normally, the other node in the cluster restart's this node and takes over its services. To diagnose what caused the original script failure, follow these steps:

1. Shut the cluster down.
2. Rerun the command that failed. In the example above, the command would be the call to */sbin/xlv_mgr*.
3. Rerun the appropriate script from */var/ha/actions*. In this case, it would be */var/ha/actions/takeback*. For example

   ```
   /var/ha/actions/takeback ‘ha_cfgchksum‘
   ```
4. Make the appropriate fixes and bring the cluster back up.

Note: The likeliest cause of errors is misconfiguration.

Replacing the Private Network

If the private network between the nodes is disconnected, IRIS FailSafe switches to using the public network if the hb-use-public field in the node block is set to yes in *ha.conf*.

To replace the private network connection, follow these steps:

1. Put the non-active node into standby state by running *ha_admin -s*. In a dual-active configuration, select either node.
2. Replace the private network cable.
3. Reintegrate the node that is in standby state into the cluster using *ha_admin* -r.

If the problem lies with the controller on one of the nodes, you must power off the failed node, replace the hardware, and then reboot the node.

Errors Logged to */var/adm/SYSLOG*

This section lists errors that might be logged to */var/adm/SYSLOG*.

```
wd95_5:  WD95 saw SCSI reset
```

This message is benign. A CHALLENGE server resets the SCSI bus in the process of starting up. For a dual-hosted system like IRIS FailSafe, the other server on the shared scsi bus also sees the resets.

```
checksum mismatch error
```

Either the configuration files on the two nodes do not match or the configuration file has been changed after the IRIS FailSafe daemons were started.

Make sure that */var/ha/ha.conf* are the same on both nodes and then reboot both nodes in the cluster.

```
read_conf error
error return from read_config
nc_readconfig: Bad config file ...
```

The IRIS FailSafe daemons could not start up because the */var/ha/ha.conf* file on the node is invalid. In some cases, the error message will also indicate the missing or invalid fields. Run *ha_cfgverify* and fix any problems identified.

```
Kill failed (%d) after remote monitor failed or no heartbeat
```

The local node tried to take over the other node's services but failed because it could not reset the other node. The local node went into the standby state.

Check the public and private network interfaces. After the problem has been fixed, reintegrate this node into the cluster by running *ha_admin -r*.

```
connect to opsnc failed
```

In a mixed OPS / IRIS FailSafe configuration, the *ha_killd* daemon could not connect to the OPS node controller on the Indy that is running the IRISconsole software.

Verify that the cables from the Indy running the IRISconsole are properly attached to the nodes in the IRIS FailSafe cluster, that the Indy is running, and that the *opsnc* daemon is running on the Indy.

```
open_tty failed -- cannot monitor node
```

The IRIS FailSafe system cannot open the serial connection to the remote power control unit or the system controller port of the other node in the cluster.

Verify that the serial connection is hooked up and test it using *ha_spng*. After the problem has been resolved, restart monitoring the serial line by running *ha_admin -m start*.

```
assumed power failed
```

This node detected that the heartbeat and the failsafe mechanism have both failed on the remote node and assumed that the remote node experienced a power failure. This node then took over the other node's services.

```
Monitoring of reset-tty on %s failed
```

The IRIS FailSafe system could not communicate with the either the system controller port (CHALLENGE L) or the remote power control unit (CHALLENGE S). Although the system continues to function, this situation prevents a node from taking over from another node if another failure occurs.

Check the physical connections. Run *ha_spng* to verify the serial connection. After the problem has been resolved, restart monitoring the serial line by running *ha_admin -m start*.

```
mail script failed
```

The application monitor detected a change in the state of the cluster, tried to send mail to the recipient specified in the */var/ha/ha.conf* file, and failed.

Verify the mail configuration on your nodes.

```
xlv_plexd[31]: DIOCXLVPLEXCOPY on xxx (dev 192.4) failed: No
such device or address
```

The plex revive operation was interrupted because the underlying device went away. This situation is usually because the shared volume has been given away to the other node in the cluster.

This message is benign.

```
New state: NC_ERROR
```

The IRIS FailSafe software has detected an internal inconsistency and has suspended operation. The nodes of the cluster are still running.

Verify the software and hardware configuration; reboot this node. Report this problem to Silicon Graphics Technical Support.

```
lost_heartbeat: heartabeat_<nodenumber> failed
Retrying heartbeat monitor over <public interface>
```

The cluster is in normal state,but a failure was detected in the private network.

Repair the private network. On each node, run *ha_admin -x* to switch the heartbeat back to the private network. .

Replacing Batteries in the Remote Power Control Unit

The remote power control unit accepts inlet power from a standard 12 VDC-AC wall adapter. The remote power control unit retains all configuration settings in the absence of power for ten years.

The remote power control unit uses eight AA alkaline batteries as an on-board uninterruptable power supply to power the unit and sustain it for up to five hours during a power outage. The batteries are in a removable battery tray, which is accessed from the front of the unit.

An LED on the battery tray alerts you that the batteries have lost sufficient power to drive the remote power control unit. This LED also lights up immediately after you power on the remote power control unit.

You can replace batteries in the remote power control unit with the unit powered up and the IRIS FailSafe cluster running (the battery tray is hot-pluggable).

To install fresh batteries, follow these steps:

1. Have ready eight fresh AA alkaline batteries. Do not mix fresh and used batteries.
2. Unlock the battery tray by turning the lock screw from the **LOCK** to the **UNLOCK** position. Pull out the battery tray using its handle.
3. Remove the three screws on the outer part of the battery holder cage; slide the holder cage off. The two inner battery holders are exposed; each holds four batteries.
4. With your fingers, carefully pry the used batteries out of one holder. Replace the batteries, following the correct orientation. (Use the other holder as a guide if necessary.) Repeat the process for the second holder.

5. When you have replaced the batteries, slide the outer battery holder cage back onto the battery tray, and replace the screws.
6. Replace the battery tray in the remote power control unit. and turn the lock screw back to the **LOCK** position

Index

A

actions.d directory, 20

actions directory, 19

active / standby configuration
- configuring, 60-62, 65-66, 68
  - networks, 32-36
- defined, 2
- NFS, 47-48

application monitor process, 11

C

chkconfig, 22, 25, 31, 36, 103, 104, 153, 154
- and controlled failback, 21

cluster state, 10

configuration file, 51-72
- application classes block, 64-66
- blocks, editing, 54-72
- disks / filesystems block, 66-68
- naming conventions, 53
- NFS blocks, 69-71
- node block, 56-63
- prerequisites, 52
- samples, 123-145
  - location, 51, 123
- saving, 72
- system configuration block, 55
- testing, 73
- Web server blocks, 71-72

configuring. *See at* IRIS FailSafe system, network, and so on

D

disk controller, failover, 15

documentation, related, xv

dual-active configuration
- configuring, 56-59, 64, 66-67
  - networks, 27-32
- defined, 2
- NFS, 45-47

E

error messages, 105-118

/etc/config/netif.options, 30, 34

F

failback, controlled, 21-22

failover, 3-4
- defined, 1
- disk controller, 15
- network adapter, 12-14
- requirements, 3
- server, 12
- service, 16
- storage, 15

filesystem
  block, 66-68
  configuring
    shared, 48-49
  See also NFS

H

ha_admin, 18
  -*i*, 21
  parameters, 19
ha_appmon, 11, 18
ha_cfg_info, 18
ha_cfgcksum, 18
ha_cfgverify, 18, 73
  messages, 105-118
ha_exec, 18
ha_killd, 18
ha_nc, 10, 18
ha_spng, 18
ha.conf, 51-72, 73
  keywords, 119-121
  location, 20
  See also configuration file
ha.conf.nfs_dual_active, 123-135
ha.conf.web_active_standby, 136-145
high-availability system, defined, 2

I

ifconfig, 37, 40, 43, 58, 63, 152
IP address
  See also at network, 14
  takeover, 14
IRIS FailSafe system
  and OPS, xiv
  commands, 17-19
  components, 5-7
  configuring, 27-49, 73-77
  files and directories, 19-20
  hardware
    components, xiii
    setting up, 79-98
    testing installation, 96-98
  installing, 79-104
  options, xiii
  processes and scripts, 8
  software, 7-11
    architecture, 9
    components, xiii
    setting up, 98-104
  starting, 25
  stopping, 26
  testing configuration, 73-77

K

keywords in *ha.conf*, 119-121

M

MAC address, 29, 33, 63
macconfig, 18
messages, location, 17

N

Netscape
  configuring, 37-45
    for active/standby configuration, 43-44
    for dual-active configuration, 38-41
        serving same document root, 42
  See also Web server

network
  adapter
    and configurations, 2
    and failover, 12-14
  configuring for active/standby system, 32-36
  configuring for dual-active system, 27-32
  testing configuration, 36-37
NFS
  configuring, 45-48, 69-71
  installing IRIS FailSafe option, 102
  locking, 16
node
  controller process, 10
  defined, 6
  state, 10
    displaying, 21
nvram, 101

O

OPS, xiv
Oracle Parallel Server. *See* OPS

P

point of failure, handling, 11-16
primary adapter, 12-14
private network, 6, 28
  configuring
    active/standby configuration, 32-36
    dual-active configuration, 29-32
public network, 6, 28
  adapter, 6
  configuring
    active/standby configuration, 32-36
    dual-active configuration, 29-32

R

re-arp, 13
re-mac, 13
  parameters, setting, 63
remote power control unit, replacing batteries, 159-160
remote procedure calls, 17
Remote System Control port, 7
reset-tty, 58
resources directory, 20

S

secondary adapter, 12-14
serial connection, 7
  monitoring, 147-148
service, failover, 16
storage, shared, 6
  failover, 15

T

takeback, 4
  defined, 1
  requirements, 3
troubleshooting, 148-159

V

/var/ha directory, 19-20
/var/ha/samples, 51

W

Web server
  accounting, 45
  configuring, 71-72
  installing IRIS FailSafe option, 102
  other than Netscape, configuring for, 37
  See also Netscape

We'd Like to Hear From You

As a user of Silicon Graphics documentation, your comments are important to us. They help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics to comment on:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please include the title and part number of the document you are commenting on. The part number for this document is
007-3109-001.

Thank you!

Three Ways to Reach Us

The **postcard** opposite this page has space for your comments. Write your comments on the postage-paid card for your country, then detach and mail it. If your country is not listed, either use the international card and apply the necessary postage or use electronic mail or FAX for your reply.

If **electronic mail** is available to you, write your comments in an e-mail message and mail it to either of these addresses:

- If you are on the Internet, use this address: techpubs@sgi.com
- For UUCP mail, use this address through any backbone site: *[your_site]*!sgi!techpubs

You can forward your comments (or annotated copies of manual pages) to Technical Publications at this **FAX** number:

415 965-0964