IRIS FailSafe™ Version 2 Administrator's Guide

007–3901–004

CONTRIBUTORS

Written by Jenn Byrnes, Steven Levine, Susan Ellis

Edited by Rick Thompson

Illustrated by Dany Galgani

Production by Glen Traefald

Engineering contributions by Vidula Iyer, Ashwinee Khaladkar, Tony Kavadias, Linda Lait, Michael Nishimoto, Wesley Smith, Bill Sparks, Paddy Sreenivasan, Dan Stekloff, Rebecca Underwood, Manish Verma

COPYRIGHT

© 1999, 2000, 2001, Silicon Graphics, Inc. All Rights Reserved. The contents of this document may not be copied or duplicated in any manner, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

LIMITED RIGHTS LEGEND

The electronic (software) version of this document was developed at private expense; if acquired under an agreement with the USA government or any contractor thereto, it is acquired as "commercial computer software" subject to the provisions of its applicable license agreement, as specified in (a) 48 CFR 12.212 of the FAR; or, if acquired for Department of Defense units, (b) 48 CFR 227-7202 of the DoD FAR Supplement; or sections succeeding thereto. Contractor/manufacturer is Silicon Graphics, Inc., 1600 Amphitheatre Pkwy 2E, Mountain View, CA 94043-1351.

TRADEMARKS AND ATTRIBUTIONS

Silicon Graphics, CHALLENGE, IRIS, IRIX, Performance Co-Pilot, and WebFORCE are registered trademarks and SGI, CXFS, IRISconsole, IRIS FailSafe, Origin, Origin2000, POWER CHALLENGE, the Silicon Graphics logo, and XFS are trademarks of Silicon Graphics, Inc.

Macintosh is a registered trademark of Apple Computer, Inc. INFORMIX is a registered trademark of Informix Software, Inc. Windows is a registered trademark of Microsoft Corporation. Netscape, Netscape Enterprise Server, and Netscape FastTrack Server are trademarks of Netscape Communications Corporation. Oracle is a registered trademark of Oracle Corporation. NFS (Network File System) and Java are trademarks of Sun Microsystems, Inc. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Cover design by Sarah Bolles, Sarah Bolles Design, and Dany Galgani, SGI Technical Publications.

New Features in This Guide

This revision contains the following new information:

- The new `cluster_status` command; see "System Status", page 177.
- Partition ID information; see "Defining Nodes", page 97.
- Configuration of nodes supported in an IRIS FailSafe and CXFS coexecution environment; see "Coexecution with CXFS", page 49.
- Number of nodes supported: 16 CSFS nodes, and up to 8 IRIS FailSafe nodes with coexecution; see "Hardware Components of an IRIS FailSafe Cluster", page 11.
- Node_Failures_Only attribute; see "Failover Attributes", page 145.
- Added TP9100 and TP9400 as supported disk storage; see "Hardware Components of an IRIS FailSafe Cluster", page 11.
- Added section on Log File Management; see "Log File Management", page 202.
- Numerous clarifications, notes, and updates throughout manual.

Record of Revision

Version	Description
002	December 1999 Published in conjunction with FailSafe 2.0 rollup patch. Supports IRIX 6.5.2 and later.
003	November 2000 Supports the IRIS FailSafe 2.1 release
004	May 2001 Supports the IRIS FailSafe 2.1.1 release

Contents

Figures

Tables

About This Guide

This guide describes the configuration and administration of an IRIS FailSafe™ highly available system.

This guide was prepared in conjunction with Release 2.1.1 of the IRIS FailSafe product. It supports IRIX 6.5.10 and later.

Audience

The *IRIS FailSafe Version 2 Administrator's Guide* is written for the person who administers the IRIS FailSafe system. The IRIS FailSafe administrator must be familiar with the operation of Origin™ servers, as well as optional Origin Vault, Fibre Channel RAID, JBOD, TP9100, or TP9400 storage systems, whichever is used in the IRIS FailSafe configuration. Good knowledge of XLV and XFS™ is also required.

Assumptions

To use Performance Co-Pilot (PCP) for FailSafe, you must have the following licenses:

- Two or more PCP Collector licenses (`PCPCOL`), one for each node in the FailSafe cluster from which you want to collect performance metrics
- One PCP Monitor license (`PCPMON`) for the workstation that is to run the visualization tools

Structure of This Guide

IRIS FailSafe configuration and administration information is presented in the following chapters and appendices:

- Chapter 1, "Overview of the IRIS FailSafe System", introduces the components of the IRIS FailSafe system and explains its hardware and software architecture.
- Chapter 2, "Planning IRIS FailSafe Configuration", describes how to plan the configuration of an IRIS FailSafe cluster.

- Chapter 3, "Installing IRIS FailSafe Software and Preparing the System", describes several procedures that must be performed on nodes in an IRIS FailSafe cluster to prepare them for IRIS FailSafe.
- Chapter 4, "IRIS FailSafe Administration Tools", describes the cluster manager tools you can use to administer and IRIS FailSafe system.
- Chapter 5, "IRIS FailSafe Configuration", explains how to perform the administrative tasks to configure a FailSafe system.
- Chapter 6, "IRIS FailSafe Configuration Examples", shows an example FailSafe three-node configuration, and some variations on that configuration.
- Chapter 7, "IRIS FailSafe System Operation", explains how to perform the administrative tasks to operate and monitor a FailSafe system.
- Chapter 8, "Testing IRIS FailSafe Configuration", describes how to test the configured IRIS FailSafe system.
- Chapter 9, "IRIS FailSafe Recovery", describes the log files used by FailSafe and how to evaluate problems in a FailSafe system.
- Chapter 10, "Upgrading and Maintaining Active Clusters", describes some procedures you may need to perform without shutting down a FailSafe cluster.
- Chapter 11, "Performance Co-Pilot for FailSafe", tells you how to use PCP to monitor the availability of a FailSafe cluster.
- Appendix A, "Updating from IRIS FailSafe 1.2 to IRIS FailSafe 2.X", provides a description of the procedures you perform to upgrade a system from IRIS FailSafe 1.X to IRIS FailSafe 2.X.
- Appendix B, "IRIS FailSafe 2.1 Software", summarizes the systems to install on each component of a cluster or node.
- Appendix C, "Metrics Exported by PCP for FailSafe", lists the metrics implemented by `pmdafsafe`(1).

Related Documentation

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

- *IRIS FailSafe Version 2 Programmer's Guide*

- *Performance Co-Pilot User's and Administrator's Guide*
- *CXFS Software Installation and Administration Guide*
- *IRIS FailSafe 2.0 INFORMIX Administrator's Guide*
- *IRIS FailSafe 2.0 Netscape Server Administrator's Guide*
- *IRIS FailSafe Version 2 NFS Administrator's Guide*
- *IRIS FailSafe 2.0 Oracle Administrator's Guide*
- *IRIS FailSafe Version 2 Samba Administrator's Guide*

The IRIS FailSafe reference pages are as follows:

- `cdbBackup`(1M)
- `cdbRestore`(1M)
- `cluster_mgr`(1M)
- `crsd`(1M)
- `failsafe`(7M)
- `fs2d`(1M)
- `ha_cilog`(1M)
- `ha_cmsd`(1M)
- `ha_exec2`(1M)
- `ha_fsd`(1M)
- `ha_gcd`(1M)
- `ha_ifd`(1M)
- `ha_ifdadmin`(1M)
- `ha_macconfig2`(1M)
- `ha_srmd`(1M)
- `ha_statd2`(1M)
- `haStatus`(1M)

Release notes are included with each IRIS FailSafe product. The names of the release notes are as follows:

Release Note	Product
failsafe2	IRIS FailSafe 2.1
failsafe2_nfs	IRIS FailSafe NFS
failsafe2_web	IRIS FailSafe Netscape Web
failsafe2_informix	IRIS FailSafe INFORMIX
failsafe2_oracle	IRIS FailSafe Oracle
failsafe2_samba	IRIS FailSafe Samba
cluster_admin	Cluster administration services
cluster_control	Node control services
cluster_services	Cluster services

Conventions

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
`manpage`(*x*)	Man page section identifiers appear in parentheses after man page names. The following list describes the identifiers:

1	User commands
1B	User commands ported from BSD
2	System calls
3	Library routines, macros, and opdefs
4	Devices (special files)
4P	Protocols

	5	File formats
	7	Miscellaneous topics
	7D	DWB-related information
	8	Administrator commands
	Some internal routines (for example, the `_assign_asgcmd_info`() routine) do not have man pages associated with them.	
variable	Italic typeface denotes variable entries and words or concepts being defined.	
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.	
[ ]	Brackets enclose optional portions of a command or directive line.	
...	Ellipses indicate that a preceding element can be repeated.	

Reader Comments

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

You can contact us in any of the following ways:

- Send e-mail to the following address:

  `techpubs@sgi.com`

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

  `http://techpubs.sgi.com`

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

- Send mail to the following address:

  Technical Publications
  SGI
  1600 Amphitheatre Pkwy., M/S 535
  Mountain View, California 94043–1351

- Send a fax to the attention of “Technical Publications” at +1 650 932 0801.

We value your comments and will respond to them promptly.

Chapter 1

Overview of the IRIS FailSafe System

This chapter provides an overview of the components and operation of the IRIS FailSafe system. It contains these major sections:

High Availability and IRIS FailSafe

In the world of mission critical computing, the availability of information and computing resources is extremely important. The availability of a system is affected by how long it is unavailable after a failure in any of its components. Different degrees of availability are provided by different types of systems:

- Fault-tolerant systems (continuous availability). These systems use redundant components and specialized logic to ensure continuous operation and to provide complete data integrity. On these systems the degree of availability is extremely high. Some of these systems can also tolerate outages due to hardware or software upgrades (continuous availability). This solution is very expensive and requires specialized hardware and software.

- Highly available systems. These systems survive single points of failure by using redundant off-the-shelf components and specialized software. They provide a lower degree of availability than the fault-tolerant systems, but at much lower cost. Typically these systems provide high availability only for client/server applications, and base their redundancy on cluster architectures with shared resources.

The Silicon Graphics® IRIS FailSafe product provides a general facility for providing highly available services. IRIS FailSafe provides highly available services for a cluster that contains multiple nodes (N-node configuration). Using IRIS FailSafe, you can configure a highly available system in any of the following topologies:

- Basic two-node configuration
- Ring configuration
- Star configuration, in which multiple applications running on multiple nodes are backed up by one node
- Symmetric pool configuration

These configurations provide redundancy of processors and I/O controllers. Redundancy of storage is obtained through the use of multi-hosted RAID disk devices and plexed (mirrored) disks.

If one of the nodes in the cluster or one of the nodes' components fails, a different node in the cluster restarts the highly available services of the failed node. To clients, the services on the replacement node are indistinguishable from the original services before failure occurred. It appears as if the original node has crashed and rebooted quickly. The clients notice only a brief interruption in the highly available service.

In an IRIS FailSafe high availability environment, nodes can serve as backup for other nodes. Unlike the backup resources in a fault-tolerant system, which serve purely as redundant hardware for backup in case of failure, the resources of each node in a highly available system can be used during normal operation to run other applications that are not necessarily highly available services. All highly available services are owned by one node in the cluster at a time.

Highly available services are monitored by the IRIS FailSafe software. During normal operation, if a failure is detected on any of these components, a *failover* process is initiated. Using IRIS FailSafe, you can define a failover policy to establish which node will take over the services under what conditions. This process consists of resetting the failed node (to ensure data consistency), doing any recovery required by the failed over services, and quickly restarting the services on the node that will take them over.

IRIS FailSafe supports *selective failover* in which individual highly available applications can be failed over to a backup node independent of the other highly available applications on that node.

IRIS FailSafe highly available services fall into two groups: highly available resources and highly available applications. Highly available resources include network interfaces, XLV logical volumes, and XFS filesystems that have been configured for IRIS FailSafe. Silicon Graphics has developed IRIS FailSafe software options for some highly available applications. This optional software includes

- IRIS FailSafe NFS
- IRIS FailSafe Web (for Netscape servers)
- IRIS FailSafe INFORMIX
- IRIS FailSafe Oracle
- IRIS FailSafe Samba

IRIS FailSafe provides a framework for making additional applications into highly available services. If you want to add highly available applications on an IRIS FailSafe cluster, you must write scripts to handle monitoring and failover functions. Information on developing these scripts is described in the *IRIS FailSafe Version 2 Programmer's Guide*. If you need assistance in this regard, contact SGI Professional Services, which offers custom FailSafe agent development and HA integration services.

IRIS FailSafe System Components and Concepts

An IRIS FailSafe system is a *cluster* system, which consists of various components with defined interrelationships. In order to use IRIS FailSafe to configure and monitor highly available services, you should be familiar with the following concepts and definitions of the system's components. These are the entities and attributes that define an IRIS FailSafe system; all system administration tasks are based on these concepts.

Node

A *node* is a single IRIX kernel image. Usually, a node is an individual computer.

The nodes might be connected to a storage area network (SAN) consisting of a number of disks.

This use of the term *node* does not have the same meaning as a node in an Origin system.

Pool

The *pool* is the entire set of nodes that are coupled to each other by networks and are defined as nodes in the cluster database. The nodes are usually close together and should always serve a common purpose. A replicated cluster configuration database is stored on each node in the pool.

All nodes that can be added to a cluster are part of the pool, but not all nodes in the pool must be part of the cluster. There is only one pool. Other pools may exist, but each is disjoint from the other. They share no node or cluster definitions.

The cluster software uses the network to send the heartbeat and control messages necessary for the cluster database to function. SGI recommends that all nodes be on the same subnet.

If there are delays in receiving heartbeat messages, the cluster software may determine that a node is not responding and cause that node to be reset.

Caution: To avoid unnecessary resets, SGI strongly recommends a private network dedicated to cluster communication. (The rest of this document refers to the *private network*.)

Cluster

The *cluster* is the set of nodes in the pool that have been defined as a cluster. A cluster is identified by a simple name; this name must be unique within the pool.

All nodes in the cluster are also in the pool. However, all nodes in the pool are not necessarily in the cluster; that is, the cluster may consist of a subset of the nodes in the pool. There is only one cluster per pool.

Membership

There are the following types of membership:

- *FailSafe membership* is the list of FailSafe nodes in the cluster on which FailSafe can make resource groups online. It differs from CXFS Membership. For more information about CXFS, see *CXFS Software Installation and Administration Guide*.
- `fs2d` *membership* (also known as *user-space membership*) is the group of nodes in the pool that are accessible to `fs2d` and therefore can receive cluster configuration database updates; this may be a subset of the nodes defined in the pool.

Resource

A *resource* is a single physical or logical entity that provides a service to clients or other resources. For example, a resource can be a single disk volume, a particular network address, or an application such as a web server. A resource is generally available for use over time on two or more nodes in a cluster, although it can be allocated to only one node at any given time.

Resources are identified by a *resource name* and a *resource type*. One resource can be dependent on one or more other resources; if so, it will not be able to start (that is, be made available for use) unless the dependent resources are also started. Dependent resources must be part of the same *resource group* and are identified in a *resource dependency list*. Resource dependencies are verified when resources are added to a resource group, not when resources are defined.

Resource Type

A *resource type* is a particular class of resource. All of the resources in a particular resource type can be handled in the same way for the purposes of *failover*. Every resource is an instance of exactly one resource type.

A resource type is identified by a simple name; this name must be unique within the cluster. A resource type can be defined for a specific node, or it can be defined for an entire cluster. A resource type that is defined for a specific node overrides a clusterwide resource type definition with the same name; this allows an individual node to override global settings from a clusterwide resource type definition.

Like resources, a resource type can be dependent on one or more other resource types. If such a dependency exists, at least one instance of each of the dependent resource types must be defined. For example, a resource type named `Netscape_web` might have resource type dependencies on resource types named `IP_address` and `volume`. If a resource named `web1` is defined with the `Netscape_web` resource type,

then the resource group containing `web1` must also contain at least one resource of the type `IP_address` and one resource of the type `volume`.

The FailSafe software includes many predefined resource types. If these types fit the application you want to make highly available, you can reuse them. If none fit, you can create additional resource types by using the instructions in this guide.

Resource Name

A *resource name* identifies a specific instance of a *resource type*. A resource name must be unique for a given resource type.

Resource Group

A *resource group* is a collection of interdependent resources. A resource group is identified by a simple name; this name must be unique within a cluster. Table 1-1 shows an example of the resources and their corresponding resource types for a resource group named `WebGroup.`

Table 1-1 Example Resource Group

Resource	Resource Type
`10.10.48.22`	`IP_address`
`/fs1`	`filesystem`
`vol1`	`volume`
`web1`	`Netscape_web`

If any individual resource in a resource group becomes unavailable for its intended use, then the entire resource group is considered unavailable. Therefore, a resource group is the unit of failover.

Resource groups cannot overlap; that is, two resource groups cannot contain the same resource.

Resource Dependency List

A *resource dependency list* is a list of resources upon which a resource depends. Each resource instance must have resource dependencies that satisfy its resource type dependencies before it can be added to a resource group.

Resource Type Dependency List

A *resource type dependency list* is a list of resource types upon which a resource type depends. For example, the `filesystem` resource type depends upon the `volume` resource type, and the `Netscape_web` resource type depends upon the `filesystem` and `IP_address` resource types.

For example, suppose a file system instance `fs1` is mounted on volume `vol1`. Before `fs1` can be added to a resource group, `fs1` must be defined to depend on `vol1`. FailSafe only knows that a file system instance must have one volume instance in its dependency list. This requirement is inferred from the resource type dependency list.

Failover

A *failover* is the process of allocating a resource group (or application) to another node, according to a *failover policy*. A failover may be triggered by the failure of a resource, a change in the FailSafe membership (such as when a node fails or starts), or a manual request by the administrator.

Failover Policy

A *failover policy* is the method used by FailSafe to determine the destination node of a failover. A failover policy consists of the following:

- failover domain
- failover attributes
- failover script

FailSafe uses the failover domain output from a failover script along with failover attributes to determine on which node a resource group should reside.

The administrator must configure a failover policy for each resource group. A failover policy name must be unique within the *pool*.

Failover Domain

A *failover domain* is the ordered list of nodes on which a given resource group can be allocated. The nodes listed in the failover domain must be within the same cluster; however, the failover domain does not have to include every node in the cluster.

The administrator defines the *initial failover domain* when creating a failover policy. This list is transformed into a *run-time failover domain* by the *failover script*; FailSafe uses the run-time failover domain along with failover attributes and the FailSafe membership to determine the node on which a resource group should reside. FailSafe stores the run-time failover domain and uses it as input to the next failover script invocation. Depending on the run-time conditions and contents of the failover script, the initial and run-time failover domains may be identical.

In general, FailSafe allocates a given resource group to the first node listed in the run-time failover domain that is also in the FailSafe membership; the point at which this allocation takes place is affected by the failover attributes.

Failover Attribute

A *failover attribute* is a string that affects the allocation of a resource group in a cluster. The administrator must specify system attributes (such as `Auto_Failback` or `Controlled_Failback`), and can optionally supply site-specific attributes.

Failover Scripts

A *failover script* is a shell script that generates a *run-time failover domain* and returns it to the `ha_fsd` process. The (`ha_fsd`) process applies the failover attributes and then selects the first node in the returned failover domain that is also in the current FailSafe membership.

The following failover scripts are provided with the FailSafe release:

- `ordered`, which never changes the initial failover domain. When using this script, the initial and run-time failover domains are equivalent.
- `round-robin`, which selects the resource group owner in a round-robin (circular) fashion. This policy can be used for resource groups that can be run in any node in the cluster.

If these scripts do not meet your needs, you can create a new failover script using the information provided in *IRIS FailSafe Version 2 Programmer's Guide*.

Action Scripts

The *action scripts* are the set of scripts that determine how a resource is started, monitored, and stopped. There must be a set of action scripts specified for each resource type.

The following is the complete set of action scripts that can be specified for each resource type:

- `exclusive`, which verifies that a resource is not already running
- `start`, which starts a resource
- `stop`, which stops a resource
- `monitor`, which monitors a resource
- `restart`, which restarts a resource on the same server after a monitoring failure occurs

The release includes action scripts for predefined resource types. If these scripts fit the resource type that you want to make highly available, you can reuse them by copying them and modifying them as needed. If none fits, you can create additional action scripts by using the instructions provided in *IRIS FailSafe Version 2 Programmer's Guide*.

Additional IRIS FailSafe Features

IRIS FailSafe provides the following features to increase the flexibility and ease of operation of a highly available system:

- dynamic management
- fine grain failover
- local restarts

These features are summarized in the following sections.

Dynamic Management

FailSafe allows you to perform a variety of administrative tasks while the system is running:

- Dynamically managed application monitoring

  FailSafe allows you to turn FailSafe monitoring of an application on and off while FailSafe continues to run. This allows you to perform online application upgrades without bringing down the FailSafe system.

- Dynamically managed FailSafe resources

  FailSafe allows you to add resources while the FailSafe system is online.

- Dynamically managed FailSafe upgrades

  FailSafe allows you to upgrade FailSafe software on one node at a time without taking down the entire FailSafe cluster.

Fine-Grain Failover

In an IRIS FailSafe environment, you can specify *fine-grain failover*.

In FailSafe version 2, the unit of failover is a resource group, not the whole node. This limits the impact of a component failure to the resource group to which that component belongs, and does not affect other resource groups or services on the same node. The process in which a specific resource group is failed over from one node to another node while other resource groups continue to run on the first node is called *fine-grain failover*.

Local Restarts

FailSafe allows you to fail over a resource group onto the same node. This feature enables you to configure a single-node system, where backup for a particular application is provided on the same machine, if possible. It also enables you to indicate that a specified number of local restarts be attempted before the resource group fails over to a different node.

IRIS FailSafe Administration

You can perform all IRIS FailSafe administrative tasks by means of the IRIS FailSafe Cluster Manager Graphical User Interface (GUI). The FailSafe GUI provides a guided interface to configure, administer, and monitor a FailSafe-controlled highly available cluster. The FailSafe GUI also provides screen-by-screen help text.

If you wish, you can perform IRIS FailSafe administrative tasks directly by means of the IRIS FailSafe Cluster Manager CLI, which provides a command-line interface for the administration tasks.

For information on IRIS FailSafe Cluster manager tools, see Chapter 4, "IRIS FailSafe Administration Tools".

For information on IRIS FailSafe configuration and administration tasks, see Chapter 5, "IRIS FailSafe Configuration" and Chapter 7, "IRIS FailSafe System Operation".

Hardware Components of an IRIS FailSafe Cluster

Figure 1-1 shows an example of IRIS FailSafe hardware components, in this case for a two-node system.

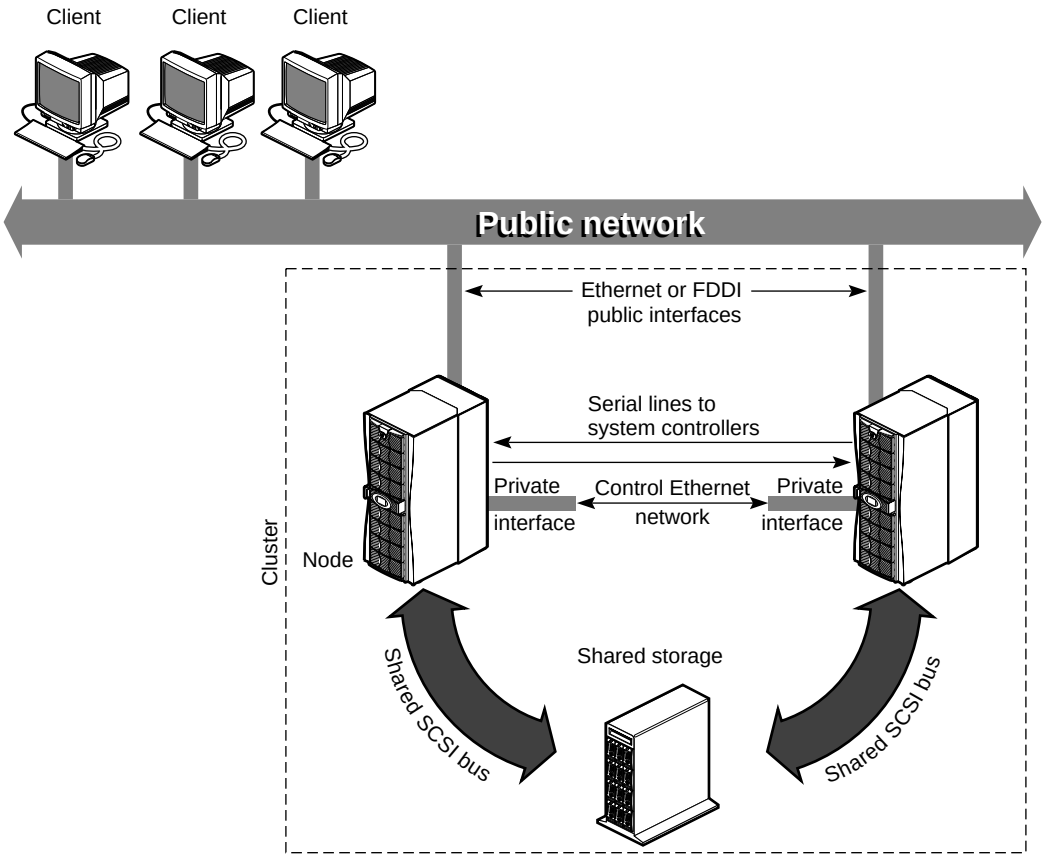

Figure 1-1 Sample IRIS FailSafe System Components

The hardware components of the IRIS FailSafe system are as follows:

- Up to eight Origin 2000, Origin 200, or Origin 3000 nodes
- More than two interfaces on each node for control networks

  At least two Ethernet or FDDI interfaces on each node are required for the control network *heartbeat* connection, by which each node monitors the state of other nodes. The IRIS FailSafe software also uses this connection to pass *control* messages between nodes. These interfaces have distinct IP addresses.

- A serial line from a serial port on each node to a Remote System Control port on another node

  A node that is taking over services on the failed node uses this line to reboot the failed node during takeover. This procedure ensures that the failed node is not using the shared disks when the replacement node takes them over.

- Disk storage and SCSI bus shared by the nodes in the cluster

  The nodes in the IRIS FailSafe system share dual-hosted 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 hardware required for the disk storage can be one of the following:

  – Origin JBOD peripheral enclosures with SCSI disks

  – Origin RAID deskside or rackmount storage systems; each chassis assembly has two storage-control processors (SPs) and at least five disk modules with caching enabled

  – TP9100

  – TP9400

- An EL-8+ (FAILSAFE-N_NODE) hardware component to reset machines in a cluster or, optionally, an ST16XX or EL-16 hardware component.

In addition, IRIS FailSafe supports ATM LAN emulation failover when FORE Systems ATM cards are used with a FORE Systems switch.

Note: 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 two or more failed components.

IRIS FailSafe Disk Connections

An IRIS FailSafe system supports the following disk connections:

- RAID support

  – single controller or dual controllers

  – single or dual hubs

  - single or dual pathing
- JBOD support
  - single or dual vaults
  - single or dual hubs

SCSI disks can be connected to two machines only. Fibre channel disks can be connected to multiple machines.

IRIS FailSafe Supported Configurations

IRIS FailSafe supports the following highly available configurations:

- Basic two-node configuration
- Star configuration of multiple primary and 1 backup node
- Ring configuration

You can use the following reset models when configuring an IRIS FailSafe system:

- Server-to-server. Each server is directly connected to another for reset. May be unidirectional.
- Network. Each server can reset any other by sending a signal over the control network to an EL-16 multiplexer.
- IRISconsole. Each server can request that the IRISconsole™ perform resets.

In a basic two-node configuration, the following arrangements are possible:

- All highly available services run on one node. The other node is the backup node. After failover, the services run on the backup node. In this case, the backup node is a hot standby for failover purposes only. The backup node can run other applications that are not highly available services.
- Highly available services run concurrently on both nodes. For each service, the other node serves as a backup node. For example, both nodes can be exporting different NFS filesystems. If a failover occurs, one node then exports all of the NFS filesystems.

High-Availability Resources

This section discusses the highly available resources that are provided on an IRIS FailSafe system.

Nodes

If a node crashes or hangs (for example, due to a parity error or bus error), the IRIS FailSafe software detects this. A different node, determined by the failover policy, takes over the failed node's services after resetting the failed node.

If a node fails, the interfaces, 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 Interfaces and IP Addresses

Clients access the highly available services provided by the IRIS FailSafe cluster using IP addresses. Each highly available service can use multiple IP addresses. The IP addresses are not tied to a particular highly available service; they can be shared by all the highly available services in the cluster.

IRIS FailSafe uses the IP aliasing mechanism to support multiple IP addresses on a single network interface. Clients can use a highly available service that uses multiple IP addresses even when there is only one network interface in the server node.

The IP aliasing mechanism allows an IRIS FailSafe configuration that has a node with multiple network interfaces to be backed up by a node with a single network interface. IP addresses configured on multiple network interfaces are moved to the single interface on the other node in case of a failure.

IRIS FailSafe requires that each network interface in a cluster have an IP address that does not failover. These IP addresses, called *fixed IP addresses*, are used to monitor network interfaces. Each fixed IP address must be configured to a network interface at system boot up time. All other IP addresses in the cluster are configured as *highly available IP addresses*.

Highly available IP addresses are configured on a network interface. During failover and recovery processes they moved to another network interface in the other node by IRIS FailSafe. Highly available IP addresses are specified when you configure the IRIS FailSafe system. IRIS FailSafe uses the `ifconfig` command to configure an IP address on a network interface and to move IP addresses from one interface to another.

In some networking implementations, IP addresses cannot be moved from one interface to another by using only the `ifconfig` command. IRIS FailSafe uses *re-MACing* (*MAC address impersonation*) to support these networking implementations. Re-MACing moves the physical (MAC) address of a network interface to another interface. It is done by using the `macconfig` command. Re-MACing is done in addition to the standard `ifconfig` process that IRIS FailSafe uses to move IP addresses. To do RE-MACing in FailSafe, a resource of type MAC_Address is used.

Note: Re-MACing can be used only on Ethernet networks. It cannot be used on FDDI networks.

Re-MACing is required when packets called gratuitous ARP packets are not passed through the network. These packets are generated automatically when an IP address is added to an interface (as in a failover process). They announce a new mapping of an IP address to MAC address. This tells clients on the local subnet that a particular interface now has a particular IP address. Clients then update their internal ARP caches with the new MAC address for the IP address. (The IP address just moved from interface to interface.) When gratuitous ARP packets are not passed through the network, the internal ARP caches of subnet clients cannot be updated. In these cases, re-MACing is used. This moves the MAC address of the original interface to the new interface. Thus, both the IP address and the MAC address are moved to the new interface and the internal ARP caches of clients do not need updating.

Re-MACing is not done by default; you must specify that it be done for each pair of primary and secondary interfaces that requires it. A procedure in the section "Planning Network Interface and IP Address Configuration" describes how you can determine whether re-MACing is required. In general, routers and PC/NFS clients may require re-MACing interfaces.

A side effect of re-MACing is that the original MAC address of an interface that has received a new MAC address is no longer available for use. Because of this, each network interface has to be backed up by a dedicated backup interface. This backup interface cannot be used by clients as a primary interface. (After a failover to this interface, packets sent to the original MAC address are ignored by every node on the network.) Each backup interface backs up only one network interface.

Disks

The IRIS FailSafe system can include shared SCSI-based storage in the form of one or more Origin FibreVault RAID storage systems. It can also include FibreVault

peripheral enclosures with SCSI disks with plexed disks. If highly available applications use filesystems, XFS filesystems or (as of the FailSafe 2.1 release) CXFS filesystems must be used.All data for highly available applications must be stored in XLV logical volumes on shared disks or, if CXFS filesystems are used, in XVM logical volumes.

For Fibre Channel RAID storage systems, if a disk or disk controller fails, the RAID storage system is equipped to keep services available through its own capabilities.

With plexed XLV logical volumes on the disks in a FibreVault, the XLV system provides redundancy. No participation of the IRIS FailSafe system software is required for a disk failure. If a disk controller fails, the IRIS FailSafe system software initiates the failover process.

IRIS FailSafe assumes that CXFS filesystems are highly-available because they do not require a FailSafe failover in order to be made available on another node in the cluster. Therefore, FailSafe does not directly start, stop, or monitor CXFS filesystems and CXFS filesystems should not be added to the FailSafe resource groups. For information on IRIS FailSafe and CXFS coexecution, see "Coexecution with CXFS", page 49.

Figure 1-2 shows disk storage takeover on a two-node system. The surviving node takes over the shared disks and recovers the logical volumes and filesystems on the disks. This process is expedited by the XFS filesystem, which supports fast recovery because it uses journaling technology that does not require the use of the `fsck` command for filesystem consistency checking.

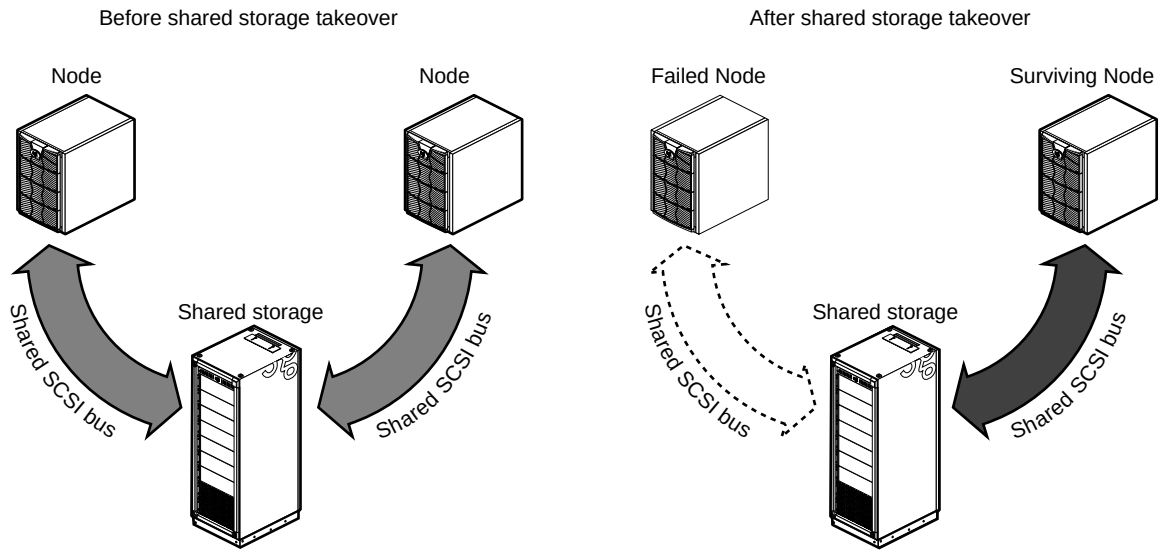

Figure 1-2 Disk Storage Failover on a Two-Node System

Highly Available Applications

Each application has a primary node and up to seven additional nodes that you can use as a backup node, according to the failover policy you define. The primary node is the node on which the application runs when FailSafe is in *normal state*. When a failure of any highly available resources or highly available application is detected by IRIS FailSafe software, all highly available resources in the affected resource group on the failed node are failed over to a different node and the highly available applications on the failed node are stopped. When these operations are complete, the highly available applications are started on the backup node.

All information about highly available applications, including the primary node, components of the resource group, and failover policy for the application and monitoring, is specified when you configure your IRIS FailSafe system with the Cluster Manager GUI or with the Cluster Manager CLI. Information on configuring the system is provided in Chapter 5, "IRIS FailSafe Configuration". Monitoring scripts detect the failure of a highly available application.

The IRIS FailSafe software provides a framework for making applications highly available services. By writing scripts and configuring the system in accordance with

those scripts, you can turn client/server applications into highly available applications. For information, see the *IRIS FailSafe Version 2 Programmer's Guide*.

Failover and Recovery Processes

When a failure is detected on one node (the node has crashed, hung, or been shut down, or a highly available service is no longer operating), a different node performs a failover of the highly available services that are being provided on the node with the failure (called the *failed node*). Failover allows all of the highly available services, including those provided by the failed node, to remain available within the cluster.

A failure in a highly available service can be detected by IRIS FailSafe processes running on another node. Depending on which node detects the failure, the sequence of actions following the failure is different.

If the failure is detected by the IRIS FailSafe software running on the same node, the failed node performs these operations:

- stops the highly available resource group running on the node
- moves the highly available resource group to a different node, according to the defined failover policy for the resource group
- sends a message to the node that will take over the services to start providing all resource group services previously provided by the failed node

When it receives the message, the node that is taking over the resource group performs these operations:

- transfers ownership of the resource group from the failed node to itself
- starts offering the resource group services that were running on the failed node

If the failure is detected by FailSafe software running on a different node, the node detecting the failure performs these operations:

- using the serial connection between the nodes, power cycles the failed node to prevent corruption of data
- transfers ownership of the resource group from the failed node to the other nodes in the cluster, based on the resource group failover policy.
- starts offering the resource group services that were running on the failed node

When a failed node comes back up, whether the node automatically starts to provide highly available services again depends on the failover policy you define. For information on defining failover policies, see "Defining a Failover Policy", page 144 in Chapter 5, "IRIS FailSafe Configuration".

Normally, a node that experiences a failure automatically reboots and resumes providing highly available services. This scenario works well for transient errors (as well as for planned outages for equipment and software upgrades).

For further information on FailSafe execution during startup and failover, see "Execution of FailSafe Action and Failover Scripts", page 26.

Overview of Configuring and Testing a New IRIS FailSafe Cluster

After the IRIS FailSafe cluster hardware has been installed, follow this general procedure to configure and test the IRIS FailSafe system:

1. Become familiar with IRIS FailSafe terms by reviewing this chapter.
2. Plan the configuration of highly available applications and services on the cluster using Chapter 2, "Planning IRIS FailSafe Configuration".
3. Perform various administrative tasks, including the installation of prerequisite software, that are required by IRIS FailSafe, as described in Chapter 3, "Installing IRIS FailSafe Software and Preparing the System".
4. Define the IRIS FailSafe configuration as explained in Chapter 5, "IRIS FailSafe Configuration".
5. Test the IRIS FailSafe system in three phases: test individual components prior to starting IRIS FailSafe software, test normal operation of the IRIS FailSafe system, and simulate failures to test the operation of the system after a failure occurs.

IRIS FailSafe Software Overview

This section describes the software layers, communication paths, and cluster configuration database of an IRIS FailSafe system.

Layers

A FailSafe system has the following software layers:

- Plug-ins, which create highly available services. If the application you want is not available, you can hire the SGI Professional Services group to develop the required software, or you can use the *IRIS FailSafe Version 2 Programmer's Guide* to write the software yourself.
- IRIS FailSafe base, which includes the ability to define resource groups and failover policies
- Cluster services, which lets you define clusters, resources, and resource types (this consists of the `cluster_services` installation package)
- Cluster software infrastructure, which lets you do the following:
  - Perform node logging
  - Administer the cluster
  - Define nodes

  The cluster software infrastructure consists of the `cluster_admin` and `cluster_control` subsystems).

Figure 1-3 shows a graphic representation of these layers. Table 1-2 describes the layers for FailSafe, which are located in the `/usr/cluster/bin` directory. The cluster services and cluster software infrastructure layers are shared with CXFS. For more information about CXFS, see *CXFS Software Installation and Administration Guide*.

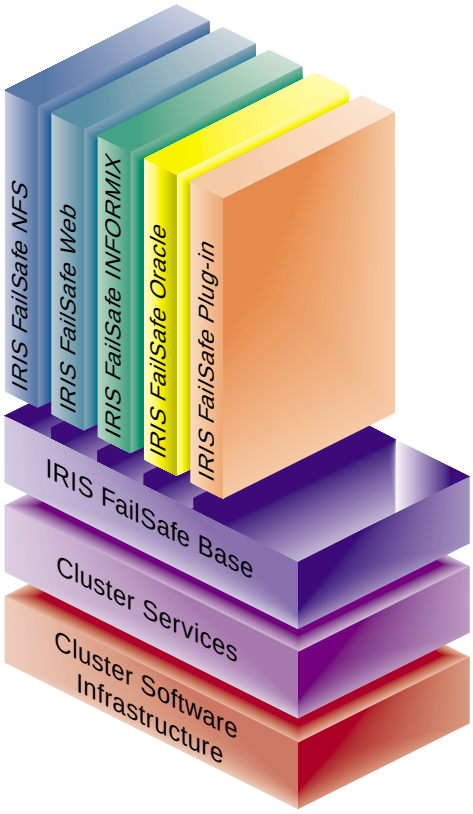

Figure 1-3 Software Layers

Table 1-2 Contents of `/usr/cluster/bin`

Layer	Subsystem	Process	Description
Plug-ins	`failsafe_informix` `failsafe2_oracle`	`ha_ifmx2`	IRIS FailSafe database agents. Each database agent monitors all instances of one type of database.
IRIS FailSafe Base	`failsafe2`	`ha_fsd`	IRIS FailSafe daemon. Provides basic component of the IRIS FailSafe software.
Cluster services (high-availability processes)	`cluster_services`	`ha_cmsd`	The FailSafe membership daemon. Provides the list of nodes, called *FailSafe membership*, available to the cluster.
		`ha_gcd`	Group membership daemon. Provides group membership and reliable communication services in the presence of failures to IRIS FailSafe processes.
		`ha_srmd`	System resource manager daemon. Manages resources, resource groups, and resource types. Executes action scripts for resources.
		`ha_ifd`	Interface agent daemon. Monitors the local node's network interfaces. This daemon is described in detail in "The Interface Agent Daemon (IFD)", page 24.
Cluster software infrastructure (cluster administrative processes)	`cluster_admin`	`cad`	Cluster administration daemon. Provides administration services.
	`cluster_control`	`crsd`	Node control daemon. Monitors the serial connection to other nodes. Has the ability to reset other nodes.

Layer	Subsystem	Process	Description
		`cmond`	Daemon that manages all other daemons. This process starts other processes in all nodes in the cluster and restarts them on failures.
		`fs2d`	Manages the configuration database and keeps each copy in sync on all nodes in the pool

The Interface Agent Daemon (IFD)

The IFD is an agent that monitors network interfaces and IP addresses. The IFD monitors all network interfaces and IP addresses configured in the node even when there are no HA IP addresses in the node.

The IFD checks the number of input packets for each interace. If the number of input packets does not increase for a ten second period, the IFD pings the broadcast address of the interface. If the input packet count does not increase in the next ten second period, the network interface and all IP addresses on the interface are marked as bad.

The IFD reads the configuration of IP addresses from the configuration database.

`IP_address` resource type action scripts use the `ha_ifdadmin` command to communicate with the IFD. Action scripts obtain status and configuration IP address from the IFD.

IFD logging can be controlled with the Cluster Manager GUI and the Cluster Manager CLI.

Communication Paths

The following figures show communication paths in FailSafe. Note that they do not represent the `cmond` cluster manager daemon.

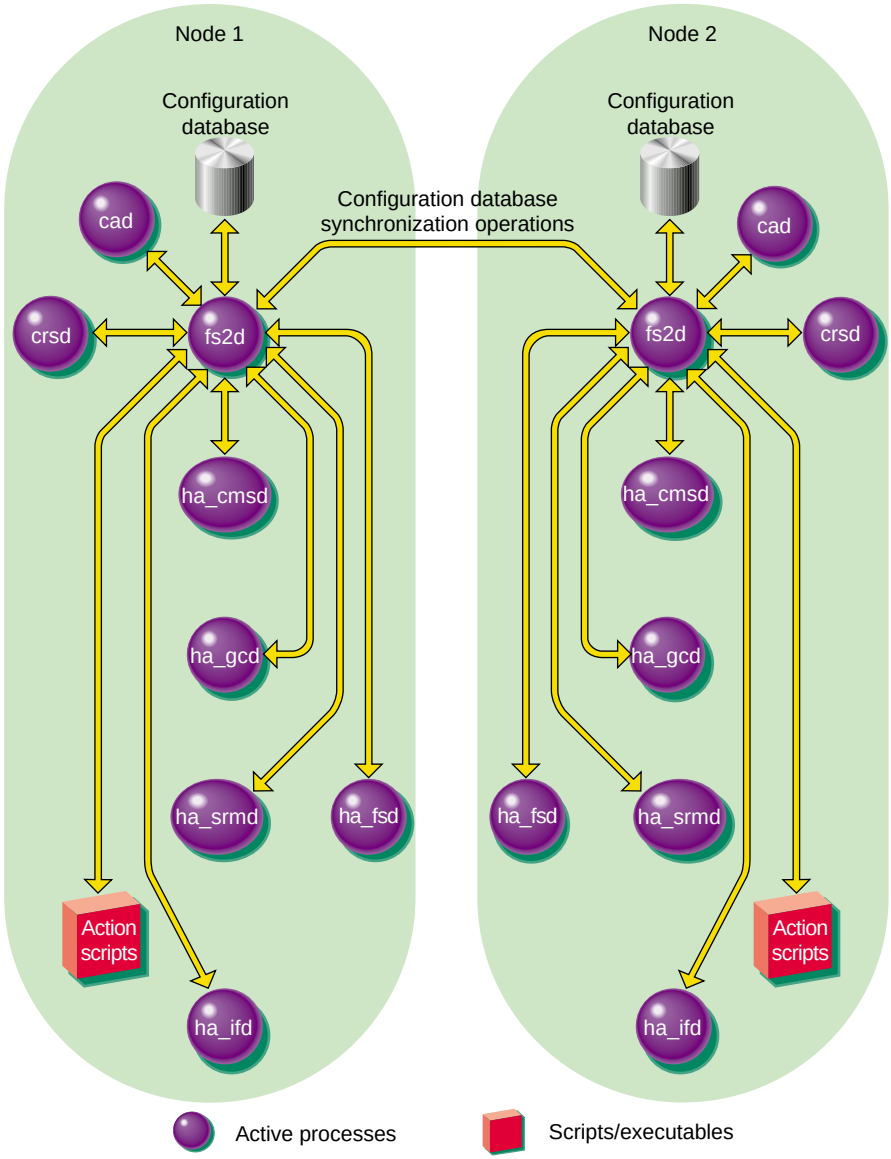

Figure 1-4 Read/Write Actions to the Cluster Configuration Database

Figure 1-5 shows the communication path for a node that is in the pool but not in a cluster.

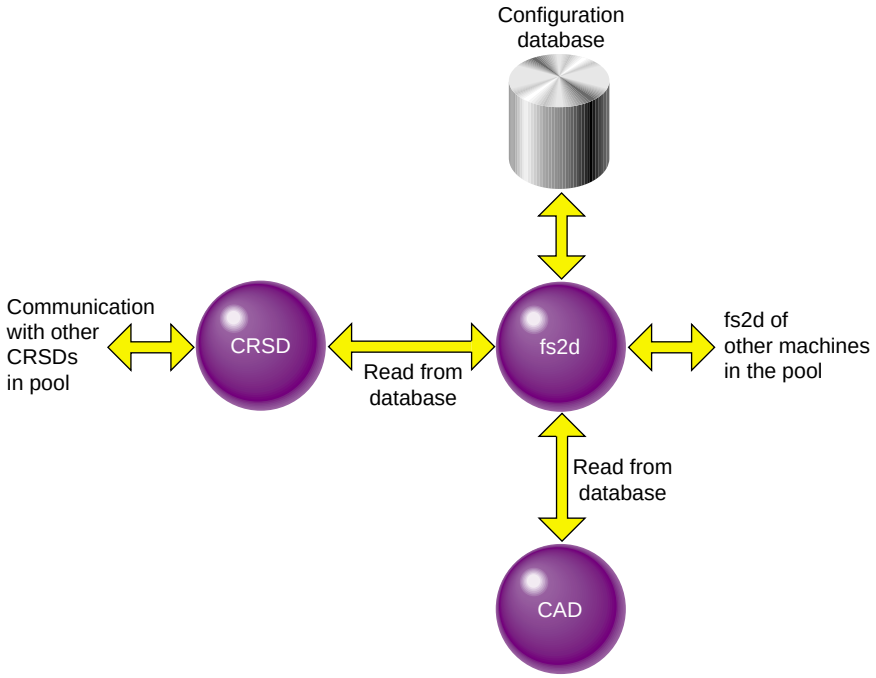

Figure 1-5 Communication Path for a Node that is Not in a Cluster

Execution of FailSafe Action and Failover Scripts

The order of execution is as follows:

1. FailSafe starts up by the `start ha_services` command in `cmgr` or as part of the node bootup procedure. It then reads the resource group information from the cluster configuration database.
2. FailSafe asks the system resource manager (SRM) to run `exclusive` scripts for the all resource groups that are in the `Online ready` state.
3. SRM returns one of the following states for each resource group:
   - `running`

- `partially running`
- `not running`

4. If a resource group has a state of `not running` in a node where HA services have been started, the following occurs:

   a. FailSafe runs the failover policy script associated with the resource group. The failover policy scripts takes the list of nodes that are capable of running the resource group (the *failover domain*) as a parameter.

   b. The failover policy script returns an ordered list of nodes in descending order of priority (the *run-time failover domain*) where the resource group can be placed.

   c. FailSafe sends a request to SRM to move the resource group to the first node in the run-time failover domain.

   d. SRM executes the `start` action script for all resources in the resource group:

      - If the `start` script fails, the resource group is marked `online` on that node with `srmd executable error` error.
      - If the `start` script is successful, SRM automatically starts monitoring those resources. After the specified start monitoring time passes, SRM executes the `monitor` action script for the resource in the resource group.

5. If state of the resource group is `running` or `partially running` on only one node in the cluster, FailSafe runs the associated failover policy script:

   - If the highest priority node is the same node where the resource group is partially running or running, the resource group is made online on the same node. In the `partially running` case, FailSafe asks SRM to execute `start` scripts for resources in the resource group that are not running.
   - If the highest priority node is a another node in the cluster, FailSafe asks SRM to execute `stop` action scripts for resources in the resource group. FailSafe makes the resource group online in the highest priority node in the cluster.

6. If the state of the resource group is `running` or `partially running` in multiple nodes in the cluster, the resource group is marked with an `error exclusivity` error. These resource groups will require operator intervention to become online in the cluster.

Figure 1-6 shows the message paths for action scripts and failover policy scripts.

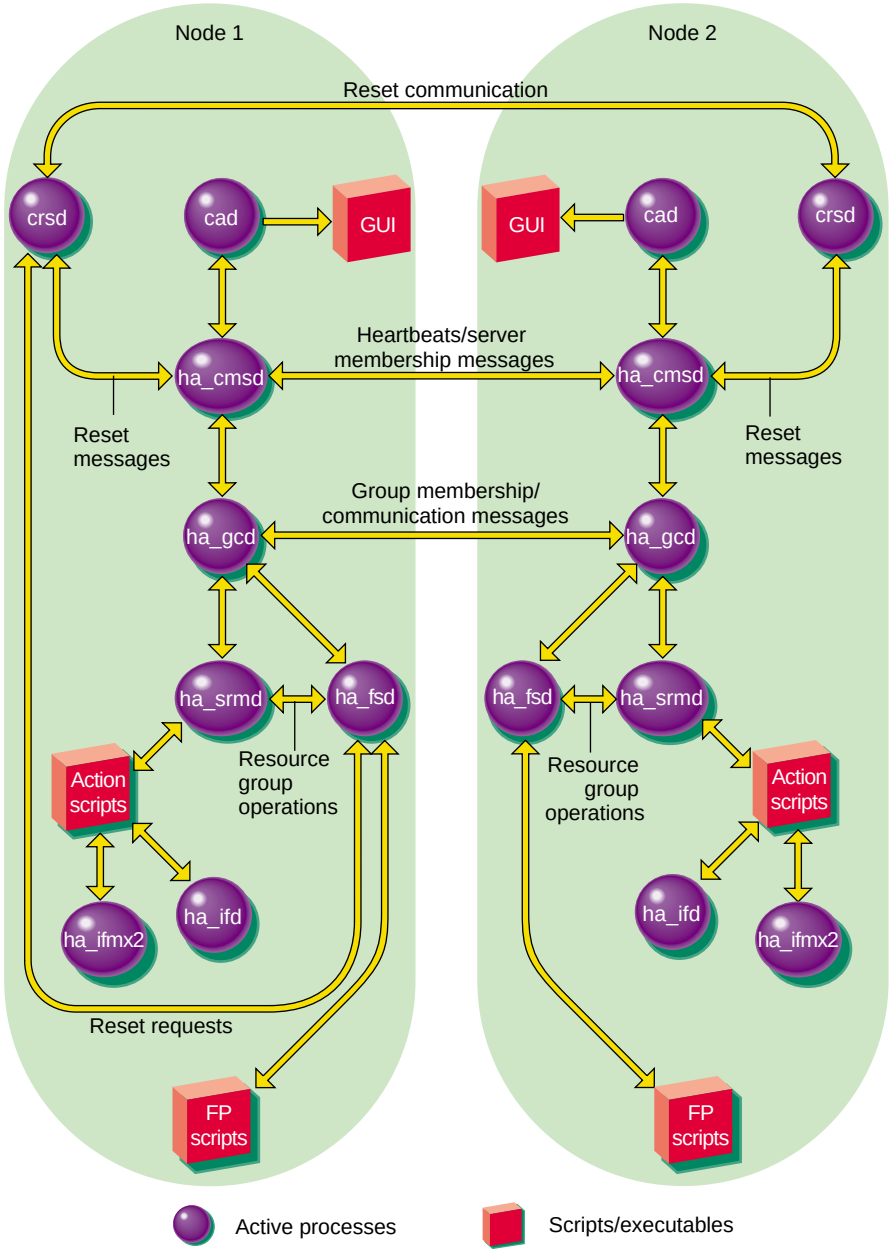

Figure 1-6 Message Paths for Action Scripts and Failover Policy Scripts

When a start Script Fails

When the `start` action script fails, the order of execution is as follows:

1. SRM notifies FailSafe of the `start` action script failure as a resource group failure.
2. FailSafe runs the failover policy script to determine the next node for the resource group.
3. FailSafe sends a request to SRM to release the resource group and allocate the resource group in the next node in the cluster.

When a stop Script Fails

When the `stop` action script fails, the order of execution is as follows:

1. SRM notifies FailSafe of the `stop` action script failure as a resource group failure.
2. FailSafe marks the resource group with an `srmd executable error` error.
3. The system administrator must use the `offline force` command to clear the error state after stopping the resource group in the node.

Components

The cluster configuration database is a key component of FailSafe software. It contains all information about the following:

- Resources
- Resource types
- Resource groups
- Failover policies
- Nodes
- Clusters

The cluster configuration database daemon (`fs2d`) maintains identical databases on each node in the cluster.

Table 1-3 shows the contents of the `/var/cluster/ha` directory.

Table 1-3 Contents of `/var/cluster/ha` directory

Directory or File	Purpose
`comm/`	Directory that contains files that communicate between various daemons. FailSafe processes create temporary files in this directory. FailSafe interprocess communication will fail if there is not sufficient disk space for this directory (approximately 2–3 MB) in the root filesystem on every node in a FailSafe cluster.
`common_scripts/`	Directory that contains the script library (the common functions that may be used in action scripts).
`log/`	Directory that contains the logs of all scripts and daemons executed by IRIS FailSafe. The outputs and errors from the commands within the scripts are logged in the `script_`*nodename* file.
`policies/`	Directory that contains the failover scripts used for resource groups.
`resource_types/template`	Directory that contains the template action scripts.
`resource_types/`*rt_name*	Directory that contains the action scripts for the *rt_name* resource type. For example, `/var/cluster/ha/resource_types/filesystem`.
`resource_types/`*rt_name*`/exclusive`	Script that verifies that a resource of this resource type is not already running.
`resource_types/`*rt_name*`/monitor`	Script that monitors a resource this resource type.
`resource_types/`*rt_name*`/restart`	Script that restarts a resource of this resource type on the same node after a monitoring failure.
`resource_types/`*rt_name*`/start`	Script that starts a resource of this resource type.
`resource_types/`*rt_name*`/stop`	Script that stops a resource of this resource type.

Planning IRIS FailSafe Configuration

This chapter explains how to plan the configuration of highly available services on your IRIS FailSafe cluster. The major sections of this chapter are as follows:

Introduction to Configuration Planning

Configuration planning involves making decisions about how you plan to use the IRIS FailSafe cluster, and based on that, how the disks and interfaces must be set up to meet the needs of the highly available services you want the cluster to provide. Questions you must answer during the planning process are:

- What do you plan to use the nodes for?

  Your answers might include uses such as offering home directories for users, running particular applications, supporting an Oracle database, providing Netscape World Wide Web service, and providing file service.

- Which of these uses will be provided as a highly available service?

  SGI has developed IRIS FailSafe software options for some highly-available applications:

  – The IRIS FailSafe NFS option enables you to provide exported NFS filesystems as highly available services.

  – The IRIS FailSafe Web option is used for the Netscape FastTrack and Enterprise Servers.

  – The IRIS FailSafe INFORMIX option is used for Informix databases.

  - The IRIS FailSafe Oracle option is used for Oracle databases.
  - The IRIS FailSafe Samba option is used for Samba for IRIX.

  To offer other applications as highly available services, a set of application monitoring shell scripts need to be developed that provide switch over and switch back functionality. Developing these scripts is described in the *IRIS FailSafe Version 2 Programmer's Guide*. If you need assistance in this regard, contact SGI Professional Services, which offers custom FailSafe agent development and HA integration services.

- Which node will be the primary node for each highly available service?

  The primary node is the node that provides the service (exports the filesystem, is a Netscape server, provides the database, and so on) when the node is in an UP state.

- For each highly available service, how will the software and data be distributed on shared and non-shared disks?

  Each application has requirements and choices for placing its software on disks that are failed over (shared) or not failed over (non-shared).

- Are the shared disks going to be part of a RAID storage system or are they going to be disks in SCSI/Fibre channel disk storage that have plexed XLV logical volumes on them?

  Shared disks must be part of a RAID storage system or in SCSI/Fibre channel disk storage with plexed XLV logical volumes on them.

- Will the shared disks be used as raw XLV logical volumes or XLV logical volumes with XFS filesystems on them?

  XLV logical volumes are required by IRIS FailSafe; filesystems must be XFS filesystems. The choice of volumes or filesystems depends on the application that is going to use the disk space.

- Will the shared disks contain CXFS filesystems, which use XVM logical volumes? For information on using FailSafe and CXFS, see "Coexecution with CXFS", page 49.

- Which IP addresses will be used by clients of highly available services?

  Multiple interfaces may be required on each node because a node could be connected to more than one network or because there could be more than one interface to a single network.

- Which resources will be part of a resource group?

  All resources that are dependent on each other have to be in the resource group.

- What will be the failover domain of the resource group?

  The failover domain determines the list of nodes in the cluster where the resource group can reside. For example, a volume resource that is part of a resource group can reside only in nodes from which the disks composing the volume can be accessed.

- How many highly available IP addresses on each network interface will be available to clients of the highly available services?

  At least one highly available IP address must be available for each interface on each node that is used by clients of highly available services.

- Which IP addresses on nodes in the failover domain are going to be available to clients of the highly available services?

- For each highly available IP address that is available on a node in the failover domain to clients of highly available services, which interface on the other nodes will be assigned that IP address after a failover?

  Every highly available IP address used by a highly available service must be mapped to at least one interface in each node that can take over the resource group service. The highly available IP addresses are failed over from the interface in the primary node of the resource group to the interface in the replacement node.

As an example of the configuration planning process, say that you have a two-node IRIS FailSafe cluster that is a departmental server. You want to make four XFS filesystems available for NFS mounting and have two Netscape FastTrack servers, each serving a different set of documents. These applications will be highly available services.

You decide to distribute the services across two nodes, so each node will be the primary node for two filesystems and one Netscape server. The filesystems and the document roots for the Netscape servers (on XFS filesystems) are each on their own plexed XLV logical volume. The logical volumes are created from disks in a Fibre Channel storage system connected to both nodes.

There are four resource groups: NFSgroup1 and NFSgroup2 are the NFS resource groups, and Webgroup1 and Webgroup2 are the Web resource groups. NFSgroup1 and Webgroup1 will have one node as the primary node. NFSgroup2 and Webgroup2 will have the other node as the primary node.

Two networks are available on each node, ef0 and ef1. The ef0 interfaces in each node are connected to each other to form a private network.

The following sections help you answer the configuration questions above, make additional configuration decisions required by IRIS FailSafe, and collect the information you need to perform the configuration tasks described in Chapter 3, "Installing IRIS FailSafe Software and Preparing the System", and Chapter 5, "IRIS FailSafe Configuration".

Disk Configuration

The first subsection below describes the disk configuration issues that must be considered when planning an IRIS FailSafe system. It explains the basic configurations of shared and non-shared disks and how they are reconfigured by IRIS FailSafe after a failover. The second subsection explains how disk configurations are specified when you configure the IRIS FailSafe system.

Planning Disk Configuration

For each disk in an IRIS FailSafe cluster, you must choose whether to make it a shared disk, which enables it to be failed over, or a non-shared disk. Non-shared disks are not failed over.

The nodes in an IRIS FailSafe cluster must follow these requirements:

- The system disk must be a non-shared disk.
- The IRIS FailSafe software, in particular the directory `/var/ha`, must be on a non-shared disk. In general, `/var` and its subdirectories should not typically be made highly available.

Choosing to make a disk shared or non-shared depends on the needs of the highly available services that use the disk. Each highly available service has requirements about the location of data associated with the service:

- Some data must be placed on non-shared disks.
- Some data must not be placed on shared disks.
- Some data can be on shared or non-shared disks.

The figures in the remainder of this section show the basic disk configurations on IRIS FailSafe clusters before failover. Each figure also shows the configuration after failover. The basic disk configurations are these:

- a non-shared disk on each node
- multiple shared disks contained Web server and NFS file server documents

In each of the before and after failover diagrams, just one or two disks are shown. In fact, many disks could be connected in the same way as each disk shown. Thus each disk shown can represent a set of disks.

An IRIS cluster can contain a combination of the basic disk configurations listed above.

Figure 2-1 shows two nodes in an IRIS FailSafe cluster, each of which has a non-shared disk with two resource groups. When non-shared disks are used by highly available applications, the data required by those applications must be duplicated on non-shared disks on both nodes. When a failover occurs, IP aliases fail over. The data that was originally available on the failed node is still available from the replacement node by using the IP alias to access it.

The configuration in Figure 2-1 contains two resource groups, Group1 and Group2. Group1 contains resource 192.26.50.1 of IP_address resource type. Group2 contains resource 192.26.50.2 of IP_address resource type.

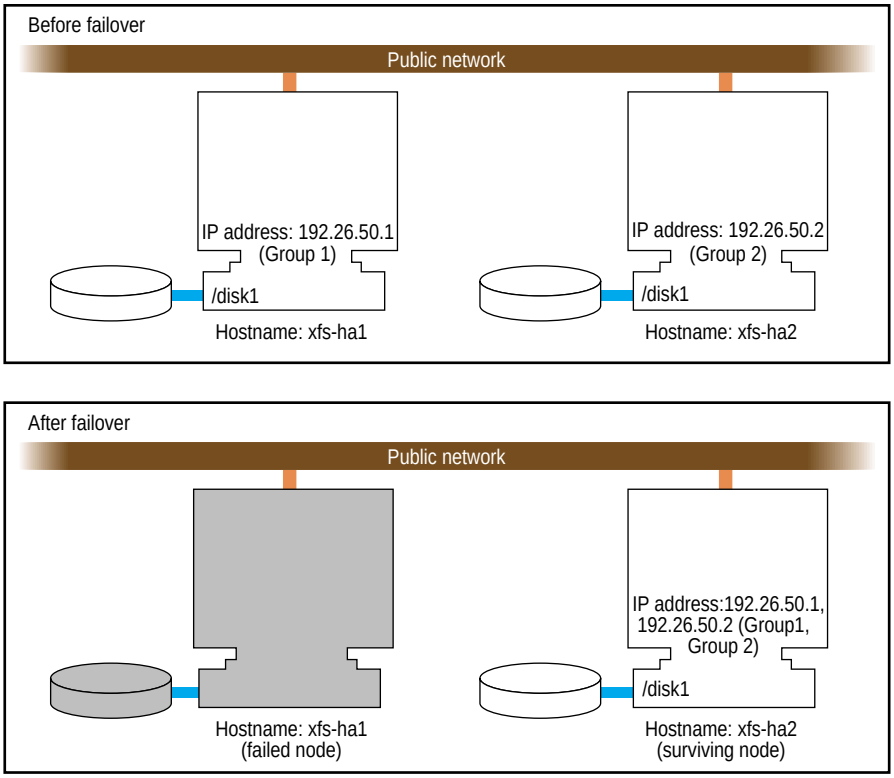

Figure 2-1 Non-Shared Disk Configuration and Failover

Figure 2-2 shows a two-node configuration with one resource group, Group1. Resource group Group1 has a failover domain of (xfs-ha1, xfs-ha2). Resource group Group1 contains three resources: resource 192.26.50.1 of resource type IP_address, resource /shared of resource type filesystem, and resource shared_vol of resource type volume.

In this configuration, the resource group Group1 has a *primary node*, which is the node that accesses the disk prior to a failover. It is shown by a solid line connection. The backup node, which accesses the disk after a failover, is shown by a dotted line. Thus, the disk is shared between the nodes. In an active/backup configuration, all resource groups have the same primary node. The backup node doesn't run any highly available resource groups until a failover occurs.

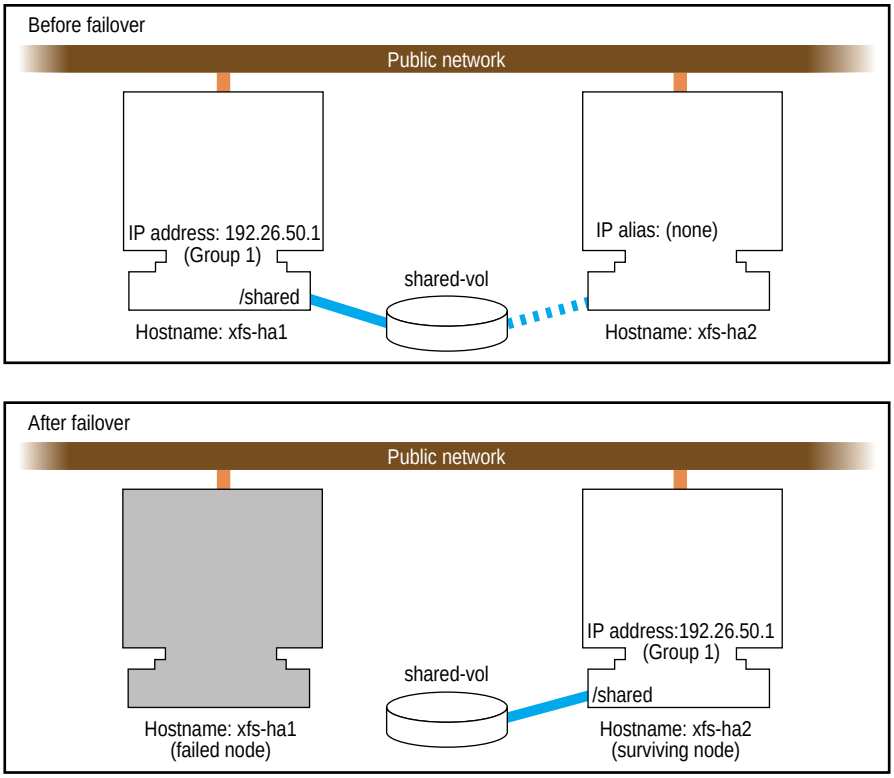

Figure 2-2 Shared Disk Configuration for Active/Backup Use

Figure 2-3 shows two shared disks in a two-node cluster with two resource groups, Group1 and Group2. Resource group Group1 contains the following resources:

- resource 192.26.50.1 of type IP_address
- resource shared1_vol of type volume
- resource /shared1 of type filesystem

Resource group Group1 has a failover domain of (xfs-ha1, xfs-ha2).

Resource group Group2 contains the following resources:

- resource 192.26.50.2 of type IP_address

- resource shared2_vol of type volume
- resource /shared2 of type filesystem

Resource group Group2 has a failover domain of (xfs-ha2, xfs-ha2).

In this configuration, each node serves as a primary node for one resource group. The solid line connections show the connection to the primary node prior to failover. The dotted lines show the connections to the backup nodes. After a failover, the surviving node has all the resource groups.

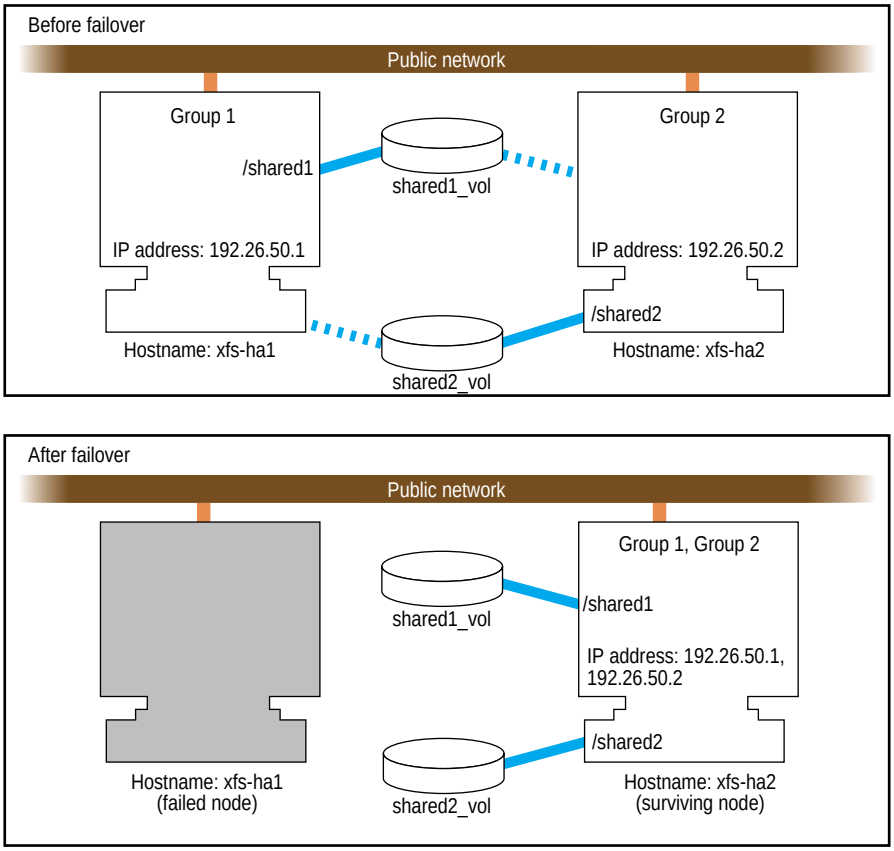

Figure 2-3 Shared Disk Configuration For Dual-Active Use

Other sections in this chapter and similar sections in the *IRIS FailSafe 2.0 Oracle Administrator's Guide*, and *IRIS FailSafe 2.0 INFORMIX Administrator's Guide* provide more specific information about choosing between shared and non-shared disks for various types of data associated with each highly available service.

Configuration Parameters for Disks

There are no configuration parameters associated with non-shared disks. They are not specified when you configure an IRIS FailSafe system. Only shared disks (actually, the XLV logical volumes on shared disks) are specified at configuration. See the section "Configuration Parameters for Logical Volumes", page 43 for details.

For information on using CXFS filesystems (which use XVM logical volumes) in a FailSafe configuration, see "Coexecution with CXFS", page 49.

Logical Volume Configuration

The first subsection below describes logical volume issues that must be considered when planning an IRIS FailSafe system. The second subsection gives an example of an XLV logical volume configuration on an IRIS FailSafe system. The third subsection explains the aspects of the configuration that must be specified for an IRIS FailSafe system.

Note: This section describes logical volume configuration using XLV logical volumes. For information on coexecution of FailSafe and CXFS filesystems (which use XVM logical volumes), see "Coexecution with CXFS", page 49.

Planning Logical Volumes

All shared disks must have XLV logical volumes on them. You can work with XLV logical volumes on shared disks as you would work with other disks. However, for correct operation of the IRIS FailSafe configuration, you must follow these rules:

- All data that is used by highly available applications on shared disks must be stored in XLV logical volumes.
- 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 part of the same resource group. (See the section "Creating XLV Logical Volumes and XFS Filesystems" for more information about XLV volume ownership.)

- Each disk in a Fibre Channel Vault or RAID LUN must be part of one resource group. Therefore, you must divide the Vault disks and RAID LUNs into one set for each resource group. If you create multiple volumes on a Vault disk or RAID LUN, all those volumes must be part of one resource group.
- Do not access a shared XLV volume from more than one node simultaneously. Doing so causes data corruption.

The IRIS FailSafe software relies on the XLV naming scheme to operate correctly. A fully qualified XLV volume name is *pathname*/*volname* or *pathname*/*nodename.volname*. The components are these:

- *pathname*, which is `/dev/xlv` or `/dev/rxlv`
- *nodename*, which by default is the same as the hostname of the node the volume was created on
- *volname*, a name specified when the volume was created; this component is commonly used when a volume is to be operated on by any of the XLV tools

For example, if volume *vol1* is created on node ha1 using disk partitions located on a shared disk, the raw character device name for the assembled volume is `/dev/rxlv/vol1` on IRIX 6.5. On the peer ha2, however, the same raw character volume appears as `/dev/rxlv/ha1.vol1` on IRIX 6.5, where *ha1* is the nodename component, and *vol1* is the volname component. As can be seen from this example, when the nodename component is the same as the local hostname, it does not appear as part of the device node name.

One *nodename* is stored in each disk or LUN volume header. This is why all volumes with volume elements on any single disk must have the same *nodename* component. If this rule is not followed, the IRIS FailSafe software does not operate correctly.

The IRIS FailSafe software modifies the *nodename* component of the volume header as volumes are transferred between nodes during failover and recovery operations. This is important because `xlv_assemble` assembles only those volumes whose *nodename* matches the local hostname. Some of the other XLV utilities allow you to see (and modify) all volumes, regardless of which node owns them.

The resource name for a resource of resource type “volume” is the XLV volume name.

If you use XLV logical volumes as raw volumes (no filesystem) for storing database data, the database system may require that the device names (in `/dev/rxlv` and `/dev/xlv` on IRIX 6.5) have specific owners, groups, and modes. See the documentation provided by the database vendor to determine if the XLV logical volume device names must have owners, groups, and modes that are different from the default values (the default owner, group, and mode for XLV logical volumes are root, sys, and 0600).

Example Logical Volume Configuration

As an example of XLV logical volume configuration, say that you have these logical volumes on four disks on an IRIX 6.5 system that we will call disk 1 through disk 5:

- A logical volume called `/dev/xlv/volA` (volume A) that contains disk 1 and a portion of disk 2.
- A logical volume called `/dev/xlv/volB` (volume B) that contains the remainder of disk 2 and disk 3.
- A logical volume called `/dev/xlv/volC` (volume C) that contains disks 4 and 5.

Volumes A and B must be part of the same resource group because they share a disk. Volume C could be part of any resource group.

Configuration Parameters for Logical Volumes

Configuration parameters for XLV logical volumes list

- owner of device filename (default value: root)
- group of device filename (default value: sys)
- mode of device filename (default value: 600)

Table 2-1 lists a label and parameters for individual logical volumes.

Table 2-1 XLV Logical Volume Configuration Parameters

Resource Attribute	volA	volB	volC	Comments
devname-owner	root	root	root	The owner of the device name.
devname-group	sys	sys	root	The group of the device name.
devname-mode	0600	0600	0600	The mode of the device name.

See the section "Creating XLV Logical Volumes and XFS Filesystems" for information about creating XLV logical volumes.

Filesystem Configuration

The first subsection below describes filesystem issues that must be considered when planning an IRIS FailSafe system. The second subsection gives an example of an XFS filesystem configuration on an IRIS FailSafe system. The third subsection explains the aspects of the configuration that must be specified for an IRIS FailSafe system.

Note: This section describes filesystem configuration for FailSafe using XFS filesystems. For information on coexecution of FailSafe and CXFS filesystems, see "Coexecution with CXFS", page 49.

Planning Filesystems

The IRIS FailSafe software supports the automatic failover of XFS filesystems on shared disks. Shared disks must be in Fibre Channel Vault or RAID storage systems that are shared between the nodes in the IRIS FailSafe cluster.

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

- All filesystems to be failed over must be XFS filesystems.
- All filesystems to be failed over must be created on XLV logical volumes on shared disks.
- For availability, filesystems to be failed over in an IRIS FailSafe cluster must be created on either mirrored disks (using the XLV plexing software) or on the Fibre Channel RAID storage system.

- Create the mount points for the filesystems on all nodes in the failover domain.
- When you set up the various IRIS FailSafe filesystems on each node, make sure that each filesystem uses a different mount point.
- Do not simultaneously mount filesystems on shared disks on more than one node. Doing so causes data corruption. Normally, IRIS FailSafe performs all mounts of filesystems on shared disks. If you manually mount a filesystem on a shared disk, make sure that it is not being used by another node.
- Do not place filesystems on shared disks in the `/etc/fstab` file. IRIS FailSafe mounts these filesystems only after making sure that another node does not have these filesystems mounted.

The resource name of a resource of the filesystem resource type is the mount point of the filesystem.

Note: When clients are actively writing to a FailSafe NFS filesystem during failover of filesystems, data corruption can occur unless filesystems are exported with the mode `wsync`. This mode requires that local mounts of the XFS filesystems use the `wsync` mount mode as well. Using `wsync` affects performance considerably.

Example Filesystem Configuration

Continuing with the example configuration from the section "Example Logical Volume Configuration" in this chapter, say that volumes A and B have XFS filesystems on them:

- The filesystem on volume A is mounted at `/sharedA` with modes rw and noauto. Call it filesystem A.
- The filesystem on volume B is mounted at `/sharedB` with modes rw and noauto."Example Logical Volume Configuration" Call it filesystem B.

Configuration Parameters for Filesystems

Table 2-2 lists a label and configuration parameters for each filesystem.

Table 2-2 Filesystem Configuration Parameters

Resource Attribute	/sharedA	/sharedB	Comments
monitoring-level	2	2	There are 2 types of monitoring 1 – checks /etc/mtab file 2 – checks if the filesystem is mounted using `stat`(1M) command
volume-name	volA	volB	The label of the logical volume on which the filesystem was created.
mode	rw,noauto	rw,noauto,wsync	The modes of the filesystem (identical to the modes specified in `/etc/fstab`).

See the section "Creating XLV Logical Volumes and XFS Filesystems" for information about creating XFS filesystems.

IP Address Configuration

The first subsection below describes network interface and IP address issues that must be considered when planning an IRIS FailSafe system. The second subsection gives an example of the configuration of network interfaces and IP addresses on an IRIS FailSafe system. The third subsection explains the aspects of the configuration that must be specified for an IRIS FailSafe configuration.

Planning Network Interface and IP Address Configuration

Follow these guidelines when planning the configuration of the interfaces to the private network between nodes in a cluster that can be used as a control network between nodes. This information is used when you define the nodes:

- Each interface has one IP address.
- The IP addresses used on each node for the interfaces to the private network are on a different subnet from the IP addresses used for public networks.
- An IP name can be specified for each IP address in `/etc/hosts`.

- Choosing a naming convention for these IP addresses that identifies them with the private network can be helpful. For example, precede the hostname with "priv-" (for private), as in priv-xfs-ha1 and priv-xfs-ha2.

Follow these guidelines when planning the configuration of the node interfaces in a cluster to one or more public networks:

- If re-MACing is required, each interface to be failed over requires a dedicated backup interface on the other node (an interface that does not have a highly available IP address). Thus, for each IP address on an interface that requires re-MACing, there should be one interface in each node in the failover domain dedicated for the interface.
- Each interface has a primary IP address also known as the fixed address. The primary IP address does not fail over.
- The hostname of a node cannot be a highly available IP address.
- All IP addresses used by clients to access highly available services must be part of the resource group to which the HA service belongs.
- If re-MACing is required, all of the highly available IP addresses must have the same backup interface.
- Making good choices for highly available IP addresses is important; these are the "hostnames" that will be used by users of the highly available services, not the true hostnames of the nodes.
- Make a plan for publicizing the highly available IP addresses to the user community, since users of highly available services must use highly available IP addresses instead of the output of the `hostname` command.
- Highly available IP addresses should not be configured in the `/etc/config/netif.options` file. Highly available IP addresses also should not be defined in the `/etc/config/ipaliases.options` file.

Follow the procedure below to determine whether re-MACing is required (see the section "Network Interfaces and IP Addresses" for information about re-MACing). It requires the use of three nodes: *node1*, *node2*, and *node3*. *node1* and *node2* can be nodes of an IRIS FailSafe cluster, but they need not be. They must be on the same subnet. *node3* is a third node. If you need to verify that a router accepts gratuitous ARP packets (which means that re-MACing is not required), *node3* must be on the other side of the router from *node1* and *node2*.

1. Configure an IP address on one of the interfaces of *node1*:

   # **/usr/etc/ifconfig** *interface* **inet** *ip_address* **netmask** *netmask* **up**

   interface is the interface to be used access the node. *ip_address* is an IP address for *node1*. This IP address is used throughout this procedure. *netmask* is the netmask of the IP address.

2. From *node3*, `ping` the IP address used in Step 1 :

   # **ping -c 2** *ip_address*
   ```
   PING 190.0.2.1 (190.0.2.1): 56 data bytes
   64 bytes from 190.0.2.1: icmp_seq=0 ttl=255 time=29 ms
   64 bytes from 190.0.2.1: icmp_seq=1 ttl=255 time=1 ms

   ----190.0.2.1 PING Statistics----
   2 packets transmitted, 2 packets received, 0% packet loss
   round-trip min/avg/max = 1/1/1 ms
   ```

3. Enter this command on *node1* to shut down the interface you configured in Step 1 :

   # **/usr/etc/ifconfig** *interface* **down**

4. On *node2*, enter this command to move the IP address to *node2*:

   # **/usr/etc/ifconfig** *interface* **inet** *ip_address* **netmask** *netmask* **up**

5. From *node3*, `ping` the IP address:

   # **ping -c 2** *ip_address*

   If the `ping` command fails, gratuitous ARP packets are not being accepted and re-MACing is needed to fail over the IP address.

Example IP Address Configuration

For this example, you are configuring an IP address of 192.26.50.1. This address has a network mask of 0xffffff00, a broadcast address of 192.26.50.255, and it is configured on interface ef0.

In this example, you are also configuring an IP address of 192.26.50.2. This address also has a network mask of 0xffffff00, a broadcast address of 192.26.50.255, and it is configured on interface ef0.

Table 2-3 shows the FailSafe configuration parameters you specify for these IP addresses.

Table 2-3 IP Address Configuration Parameters

Resource Attribute	Resource Name: 192.26.50.1	Resource Name: 192.26.50.1
network mask	0xffffff00	0xffffff00
broadcast address	192.26.50.255	192.26.50.255
interface	ef0	ef0

Local Failover of IP Addresses

You can configure your system so that an IP address will fail over to a second interface within the same host, for example from ef0 to ef1 on a single node. A configuration example that shows the steps you must follow for this configuration is provided in "Local Failover of IP Address", page 172.

Coexecution with CXFS

CXFS 6.5.10 and IRIS FailSafe 2.1 may be installed and run on the same system, which is known as *coexecution*. This allows you to have application-level high availability and a clustered filesystem.

Note: IRIS FailSafe assumes that CXFS filesystems are highly-available because they do not require a FailSafe failover in order to be made available on another node in the cluster. Therefore, FailSafe does not directly start, stop, or monitor CXFS filesystems and CXFS filesystems should not be added to the FailSafe resource groups.

If a highly available application uses a CXFS filesystem and IP address, add the application and the IP address to a resource group. The CXFS filesystem that the application depends on should be mounted on all nodes in the failover domain using the CXFS GUI or `cmgr`(1m) tool.

If the node acting as the metadata server for a filesystem dies, another node in the list of potential metadata servers will be chosen as the new metadata server.

Note the following:

- Even when you are running CXFS and FailSafe, there is still only one pool, one cluster, and one cluster configuration.
- The cluster can be one of three types:
  - `FailSafe`. In this case, all nodes will also be of type `FailSafe`.
  - `CXFS`. In this case, all nodes will be of type `CXFS`.
  - `CXFS and FailSafe` (coexecution). In this case, nodes will be a mix of type `CXFS` and type `CXFS and FailSafe`, using FailSafe for application-level high availability and CXFS.

  Note: Although it is possible to configure a coexecution cluster with type `FailSafe` nodes, SGI does not support this configuration.

- It is recommended that a production cluster be configured with a minimum of 3 weighted nodes and a maximum of 16 nodes. (A 2–weighted-node cluster with reset cables is supported, but there are inherent issues with this configuration.) All the nodes in the cluster must be running CXFS, and as many as 8 nodes can also run IRIS FailSafe.
- All potential metadata server nodes must be of type `CXFS` or `CXFS and FailSafe`
- There is one `cmgr`(1m) (`cluster_mgr`) command but separate grapical user interfaces (GUIs) for CXFS and for FailSafe. You must manage CXFS configuration with the CXFS GUI and FailSafe configuration with the FailSafe GUI.
- Using the CXFS GUI or the CLI, you can convert an existing `FailSafe` cluster and nodes to `CXFS` or `CXFS and FailSafe`. You can perform a parallel action using the FailSafe GUI. A converted node can be used by FailSafe to provide application-level high-availability and by CXFS to provide clustered filesystems.

  However:

  - You cannot change the type of a node if the respective high availability (HA) or CXFS services are active. You must first stop the services for the node.
  - The cluster must support all of the functionalities (FailSafe and/or CXFS) that are turned on for its nodes; that is, if your cluster is `CXFS`, then you **cannot** modify a node that is already part of the cluster so that it is `FailSafe`.

However, the nodes do not have to support all the functionalities of the cluster; that is, you can have a `CXFS` node in a `CXFS and FailSafe` cluster.

- For FailSafe, you must have at least two network interfaces. However, CXFS uses only one interface for both heartbeat and control messages. When using FailSafe and CXFS on the same node, only the priority 1 network will be used for CXFS and it must be set to allow both heartbeat and control messages.

Note: CXFS will not fail over to the second network. If the priority 1 network fails, CXFS will fail but FailSafe services may move to the second network if the node is `CXFS and FailSafe`. If CXFS resets the node due to the loss of the priority 1 network, it will cause FailSafe to remove the node from the FailSafe membership; this in turn will cause resource groups to fail over to other FailSafe nodes in the cluster.

- All relevant IRIX patches for CXFS and FailSafe must be installed.

For information on converting a CXFS cluster for FailSafe use, see "Converting a CXFS Cluster to FailSafe", page 112. For information on converting a CXFS node for FailSafe use, see "Converting a CXFS Node to FailSafe", page 102. For information on configuring a FailSafe system to export CXFS filesystems, see "Exporting CXFS Filesystems", page 173.

For more information on CXFS, see *CXFS Software Installation and Administration Guide*

Installing IRIS FailSafe Software and Preparing the System

This chapter describes several system administration procedures that must be performed on the nodes in a cluster to prepare and configure them for IRIS FailSafe. These procedures assume that you have done the planning described in Chapter 2, "Planning IRIS FailSafe Configuration".

The major sections in this chapter are as follows:

Overview of Configuring Nodes for IRIS FailSafe

Performing the system administration procedures required to prepare nodes for IRIS FailSafe involves these steps:

1. Install required software as described in the section "Installing Required Software".
2. Configure the system files on each node, as described in the section "Configuring System Files".
3. Check the setting of two important NVRAM variables on each node as described in the section "Setting NVRAM Variables".

4. Create the logical volumes and filesystems required by the highly available applications you plan to run on the cluster. See the section "Creating XLV Logical Volumes and XFS Filesystems".
5. Configure the network interfaces on the nodes using the procedure in the section "Configuring Network Interfaces".
6. Configure the serial ports used on each node for the serial connection to the other nodes by following the procedure in the section "Configuring the Serial Ports".
7. When you are ready configure the nodes so that IRIS FailSafe software starts up when they are rebooted.

To complete the configuration of nodes for IRIS FailSafe, you must configure the components of the IRIS FailSafe system, as described in Chapter 5, "IRIS FailSafe Configuration".

Installing Required Software

Note: The IRIS FailSafe base CD requires about 10 MB.

Note: Users must install base system administration (*sysadm_base*), cluster administration (*cluster_admin*), cluster control (*cluster_control*), cluster services (*cluster_services*), java (*java_eoe*), and Java Plug-in (*java_plugin*) from the IRIX CD set.

Note: For instructions on installing an IRIS FailSafe patch, see "Installing an IRIS FailSafe Patch", page 70.

To install the software, follow these steps:

1. Make sure all servers in the cluster are running a supported release of IRIX.
2. Depending on the servers and storage in the configuration and the IRIX revision level, install the latest recommended patches. For information on recommended patches for each platform, see
   `http://bits.csd.sgi.com/digest/patches/recommended/`

3. On each system in the pool, install the version of the EL-8+ multiplexer driver that is appropriate to the operating system. Use the CD that accompanies the EL-8+ multiplexer. Reboot the system after installation.
4. Install the software on pool nodes:

   On each node that is part of the pool, install the following software, in this order:

   - `sysadm_base.sw.dso`
   - `sysadm_base.sw.server`
   - `cluster_admin.sw.base`
   - `cluster_control.sw.base`
   - `cluster_services.sw.base`
   - `cluster_services.sw.cli`
   - `cluster_control.sw.cli`
   - `failsafe2.sw.cli`
   - `sysadm_failsafe2.sw.server`
   - `cluster_control.sw`

   Note: For 6.5 systems that do not have `sysadmdesktop` installed,`inst` reports missing prerequisites. Resolve this conflict by installing `sysadm_base.sw.priv`, which provides a subset of the functionality of `sysadmdesktop.sw.base` and is included in this distribution, or by installing `sysadmdesktop.sw.base` from the IRIX distribution.

   If you try to install `sysadm_base.sw.priv` on a system that already has `sysadmdesktop.sw.base`, `inst` reports incompatible subsystems. Resolve this conflict by not installing `sysadm_base.sw.priv`. Similar conflicts occur if you try to install `sysadmdesktop.sw.base` on a system that already has `sysadm_base.sw.priv`.

   If the pool nodes are to be administered by a Web-based version of the IRIS FailSafe Cluster Manager GUI, install these subsystems, in this order:

   – `java_eoe.sw`, version 3.1.1

- `sysadm_base.sw.client`
- `sysadm_failsafe2.sw.client`
- `sysadm_failsafe2.sw.web`

5. Install additional software on nodes:

   On each node that is part of the cluster, install the following software, in the order given. This software is required for nodes in addition to that listed in Step 4..

   - `cluster_services.sw`
   - `failsafe2.sw`
   - if necessary: `nfs.ws.nfs` (IRIX; might already be present)
   - `failsafe2_nfs.sw`
   - if necessary: `ns_admin.sw.server` (from Netscape; might already be present)
   - if necessary: `ns_fasttrack.sw.server` OR `ns_enterprise.sw.server` (from Netscape; might already be present)
   - `failsafe2_web.sw`

6. Install software on the administrative workstation (GUI client).

   If the workstation runs the GUI client from an IRIX desktop, install these subsystems:

   - `sysadm_failsafe2.sw.desktop`
   - `sysadm_failsafe2.sw.client`
   - `sysadm_base.sw.client`
   - `java_eoe.sw`, version 3.1.1
   - if the workstation launches the GUI client from a Web browser that supports Java™: `java_plugin` from the IRIS FailSafe CD

Note: If you try to install all subsystems in `java_plugin`, `inst` reports incompatible subsystems (`java_plugin.sw.swing101`, `java_plugin.sw.swing102`, and `java_plugin.sw.swing103`). Resolve this conflict by not installing these three subsystems; the IRIS FailSafe Cluster Manager GUI does not use them.

If the Java plug-in is not installed when the IRIS FailSafe Manager GUI is run from a browser, the browser is redirected to `http://java.sun.com/products/plugin/1.1/plugin-install.html`

After installing the Java plug-in, you must close all browser windows and restart the browser.

For a non-IRIX workstation, download the Java Plug-in from `http://java.sun.com/products/plugin/1.1/plugin-install.html`

If the Java plug-in is not installed when the IRIS FailSafe Manager GUI is run from a browser, the browser is redirected to this site.

7. On the appropriate servers, install other optional software the customer may have ordered, such as storage management or network board software.
8. If the customer is using plexed XLV logical volumes, do the following:
   - Install a disk plexing license on each server in the cluster in `/var/flex1m/license.dat`. For more information on XLV logical volumes and on XFS plexing and filesystems, see Chapter 2, "Planning IRIS FailSafe Configuration".
   - Verify that the license has been successfully installed on each node in the cluster:

     ```
     # xlv_mgr
     xlv_mgr> show config
     ```

     If the license is successfully installed, the following line appears:

     ```
     Plexing license: present
     ```

   - Quit `xlv_mgr`.
9. Install recommended patches for IRIS FailSafe.

For IRIS FailSafe, you must set the AutoLoad variable to Yes; this can be done when you set host SCSI IDs, as explained in "Setting NVRAM Variables", page 62.

Note: For reference, Appendix B, "IRIS FailSafe 2.1 Software", page 261, summarizes systems to install on each component of a cluster or node.

Configuring System Files

When you install the FailSafe Software, there are some considerations when configuring your system files and system parameters for every node in the pool. This section includes information on the following topics:

- "Configuring /etc/services for FailSafe", page 58
- "Configuring /etc/config/cad.options for FailSafe", page 59
- "Configuring /etc/config/fs2d.options for FailSafe", page 59
- "Configuring /etc/config/cmond.options for FailSafe", page 62
- "Setting the coreplusid System Parameter", page 62

Configuring /etc/services for FailSafe

The `/etc/services` file must contain entries for `sgi-cad` and `sgi-crsd` before installing the `cluster_admin` product on each node in the pool. The port numbers assigned for these processes must be the same in all nodes in the pool. Note that `sgi-cad` requires a tcp port.

The following shows an example of `/etc/services` entries for `sgi-cad` and `sgi-crsd`:

```
sgi-crsd        7500/udp          # Cluster Reset Services Daemon
sgi-cad         9000/tcp          # Cluster Admin daemon
```

The `/etc/services` file must contain entries for `sgi-cmsd` and `sgi-gcd` on each node before starting HA services in the node. The port numbers assigned for these processes must be the same in all nodes in the cluster.

The following shows an example of `/etc/services` entries for `sgi-cmsd` and `sgi-gcd`:

```
sgi-cmsd        7000/udp        # SGI FailSafe Membership Daemon
sgi-gcd         8000/udp        # SGI Group Communication Daemon
```

Configuring /etc/config/cad.options for FailSafe

The `/etc/config/cad.options` file contains the list of parameters that the cluster administration daemon (CAD) reads when the process is started. The CAD provides cluster information to the FailSafe Cluster Manager GUI.

The following options can be set in the `cad.options` file:

–append_log	Append CAD logging information to the CAD log file instead of overwriting it.
–log_file *filename*	CAD log file name. Alternately, this can be specified as -lf *filename.*
-vvvv	Verbosity level. The number of "v"s indicates the level of logging. Setting -v logs the fewest messages. Setting -vvvv logs the highest number of messages.

The following example shows an `/etc/config/cad.options` file:

```
-vv -lf /var/cluster/ha/log/cad_nodename -append_log
```

When you change the `cad.options` file, you must restart the CAD processes with the `/etc/init.d/cluster restart` command for those changes to take affect.

Configuring /etc/config/fs2d.options for FailSafe

The `/etc/config/fs2d.options` file contains the list of parameters that the fs2d daemon reads when the process is started. The fs2d daemon is the configuration database daemon that manages the distribution of cluster configuration database (CDB) across the nodes in the pool.

The following options can be set in the `fs2d.options` file:

-logevents *event name*	Log selected events. These event names may be used: `all`, `internal`, `args`, `attach`, `chandle`, `node`, `tree`, `lock`, `datacon`, `trap`, `notify`, `access`, `storage`. The default value for this option is `all`.

-logdest *log destination*	Set log destination. These log destinations may be used: `all`, `stdout`, `stderr`, `syslog`, `logfile`. If multiple destinations are specified, the log messages are written to all of them. If `logfile` is specified, it has no effect unless the -logfile option is also specified. The default is -logdest stderr, but logging is then disabled if fs2d runs as a daemon, since stdout and stderr are closed when fs2d is running as a daemon. The default value for this option is `logfile`.
-logfile *filename*	Set log file name. The default value is `/var/cluster/ha/log/fs2d_log`
-logfilemax *maximum size*	Set log file maximum size (in bytes). If the file exceeds the maximum size, any preexisting `filename.old` will be deleted, the current file will be renamed to `filename.old`, and a new file will be created. A single message will not be split across files. If -logfile is set, the default value for this option is 10000000.
-loglevel *log level*	Set log level. These log levels may be used: `always`, `critical`, `error`, `warning`, `info`, `moreinfo`, `freq`, `morefreq`, `trace`, `busy`. The default value for this option is `info`.
-trace *trace class*	Trace selected events. These trace classes may be used: `all`, `rpcs`, `updates`, `transactions`, `monitor`. No tracing is done, even if it is requested for one or more classes of events, unless either or both of -tracefile or -tracelog is specified. The default value for this option is `transactions`.
-tracefile *filename*	Set trace file name.
-tracefilemax *maximum size*	Set trace file maximum size (in bytes). If the file exceeds the maximum size, any preexisting `filename.old` will be deleted, the current file will be renamed to `filename.old`.

-[no]tracelog	[Do not] trace to log destination. When this option is set, tracing messages are directed to the log destination or destinations. If there is also a trace file, the tracing messages are written there as well.
-[no]parent_timer	[Do not] exit when parent exits. The default value for this option is -noparent_timer.
-[no]daemonize	[Do not] run as a daemon. The default value for this option is -daemonize.
-l	Do not run as a daemon.
-h	Print usage message.
-o help	Print usage message.

Note that if you use the default values for these options, the system will be configured so that all log messages of level `info` or less, and all trace messages for transaction events to file `/var/cluster/ha/log/fs2d_log`. When the file size reaches 10MB, this file will be moved to its namesake with the `.old` extension, and logging will roll over to a new file of the same name. A single message will not be split across files.

The following example shows an `/etc/config/fs2d.options` file that directs all fs2d logging information to `/var/adm/SYSLOG`, and all fs2d tracing information to `/var/cluster/ha/log/fs2d_ops1`. All log events are being logged, and the following trace events are being logged: rpcs, updates and transactions. When the size of the tracefile `/var/cluster/ha/log/fs2d_ops1` exceeds 100000000, this file is renamed to `/var/cluster/ha/log/fs2d_ops1.old` and a new file `/var/cluster/ha/log/fs2d_ops1` is created. A single message is not split across files.

```
-logevents all -loglevel trace -logdest syslog -trace rpcs -trace
updates -trace transactions -tracefile /var/cluster/ha/log/fs2d_ops1
-tracefilemax 100000000
```

The following example shows an `/etc/config/fs2d.options` file that directs all log and trace messages into one file, `/var/cluster/ha/log/fs2d_chaos6`, for which a maximum size of 100000000 is specified. -tracelog directs the tracing to the log file.

```
-logevents all -loglevel trace -trace rpcs -trace updates -trace
transactions -tracelog -logfile /var/cluster/ha/log/fs2d_chaos6
-logfilemax 100000000 -logdest logfile.
```

When you change the `fs2d.options` file, you must restart the FS2D processes with the `/etc/init.d/cluster restart` command for those changes to take affect.

Configuring /etc/config/cmond.options for FailSafe

The`/etc/config/cmond.options` file contains the list of parameters that the cluster monitor daemon (`cmond`) reads when the process is started. It also specifies the name of the file that logs `cmond` events. The cluster monitor daemon provides a framework for starting, stopping, and monitoring process groups. See the `cmond`(1M) man page for information on the cluster monitor daemon.

The following options can be set in the `cmond.options` file:

-L *loglevel*	Set log level to *loglevel*
-d	Run in debug mode
-l	Lazy mode, where `cmond` does not validate its connection to the cluster database
-t *napinterval*	The time interval in milliseconds after which `cmond` checks for liveliness of process groups it is monitoring
-s [*eventname*]	Log messages to `stderr`

A default `cmond.options` file is shipped with the following options. This default options file logs `cmond` events to the `/var/cluster/ha/log/cmond_log` file.

```
-L info -f /var/cluster/ha/log/cmond_log
```

Setting the coreplusid System Parameter

It is recommended that when you run FailSafe, you use the `systune`(1M) command to set the `coreplusid` flag to 1 on every node in the system. If this flag is set, IRIX will suffix all core files with a process pid. This prevents a core dump from being overwritten by another process core dump.

Setting NVRAM Variables

During the hardware installation of IRIS FailSafe nodes, two NVRAM variables must be set:

- The boot parameter AutoLoad must be set to **yes**. The IRIS FailSafe software requires the nodes to be automatically booted when they are reset or when the node is powered on.
- The SCSI IDs of the nodes in an IRIS FailSafe cluster, specified by the scsihostid variable, must be different. This variable is important only when a cluster is configured with shared SCSI storage. If a cluster has no shared storage or is using shared Fibre Channel storage, setting scsihostid is not important.

You can check the setting of these variables with these commands:

```
# nvram AutoLoad
Y
# nvram scsihostid
0
```

To set these variables, use these commands:

```
# nvram AutoLoad yes
# nvram scsihostid number
```

number is the SCSI ID you choose. A node uses its SCSI ID on all buses attached to it. Therefore, you must make sure that no device attached to a node has *number* as its SCSI unit number. If you change the value of the scsihostid variable, you must reboot the system for the change to take effect.

Creating XLV Logical Volumes and XFS Filesystems

In Chapter 2, "Planning IRIS FailSafe Configuration" you planned the XLV logical volumes and XFS filesystems to be used by highly available applications on the cluster. You can create them by following the instructions in the guide *IRIX Admin: Disks and Filesystems*.

Note: This section describes logical volume configuration using XLV logical volumes. For information on coexecution of FailSafe and CXFS filesystems (which use XVM logical volumes), see "Coexecution with CXFS", page 49. For information on creating CXFS filesystems, see *CXFS Software Installation and Administration Guide*. For information on creating XVM logical volumes, see *XVM Volume Manager Administrator's Guide*.

When you create the XLV logical volumes and XFS filesystems you need, remember these important points:

- If the shared disks are not in a RAID storage system, plexed XLV logical volumes should be created.
- Each XLV logical volume must be owned by the same node that is the primary node for the highly available applications that use the logical volume (see "Planning Logical Volumes"). To simplify the management of the *nodename*s (owners) of volumes on shared disks, follow these recommendations:
  - Work with the volumes on a shared disk from only one node in the cluster.
  - After you create all the volumes on one node, you can selectively change the *nodename* to the other node using `xlv_mgr`.
- If the XLV logical volumes you create are used as raw volumes (no filesystem) for storing database data, the database system may require that the device names (in `/dev/rxlv` and `/dev/xlv`) have specific owners, groups, and modes. If this is the case (see the documentation provided by the database vendor), use the `chown` and `chmod` commands (see the `chown`(1) and `chmod`(1) reference pages) to set the owner, group, and mode as required.
- No filesystem entries are made in `/etc/fstab` for XFS filesystems on shared disks; IRIS FailSafe software mounts the filesystems on shared disks. However, to simplify system administration, consider adding comments to `/etc/fstab` that list the XFS filesystems configured for IRIS FailSafe. Thus, a system administrator who sees mounted IRIS FailSafe filesystems in the output of the `df` command and looks for the filesystems in the `/etc/fstab` file will learn that they are filesystems managed by IRIS FailSafe.
- Be sure to create the mount point directory for each filesystem on all nodes.

Configuring Network Interfaces

The procedure in this section describes how to configure the network interfaces on the nodes in an IRIS FailSafe cluster. The example shown in Figure 3-1 is used in the procedure.

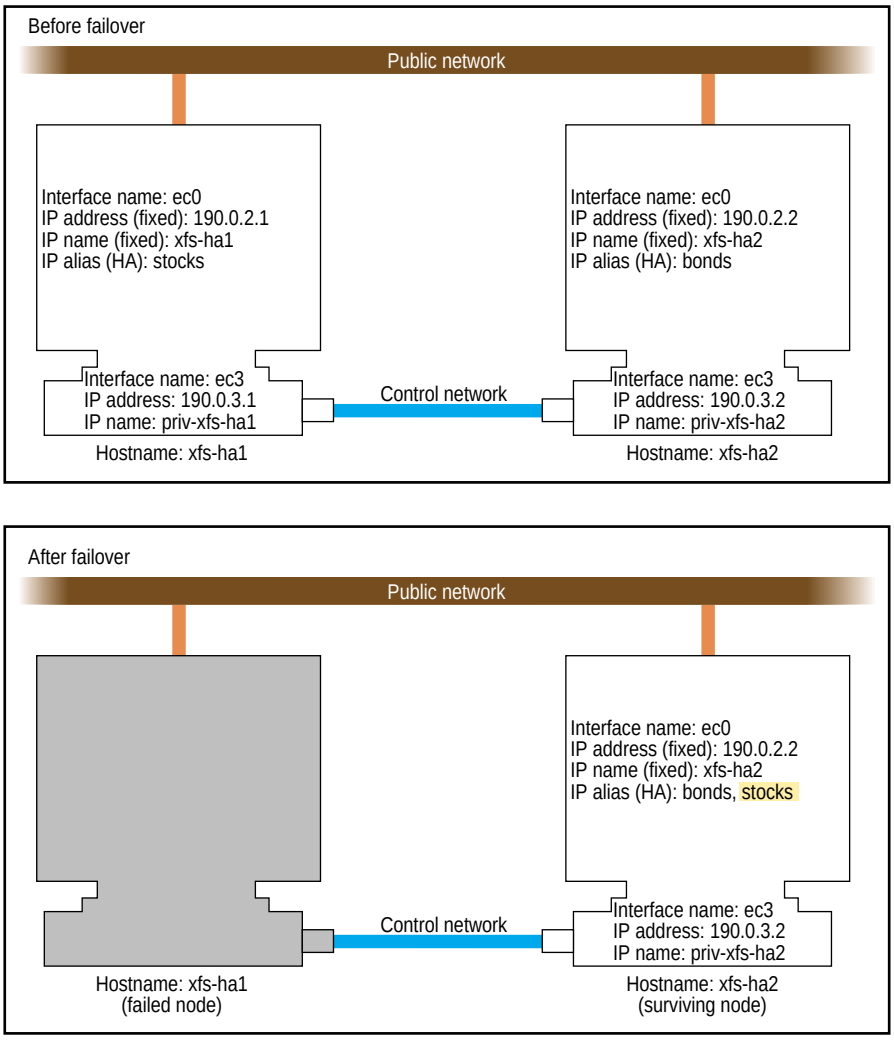

Figure 3-1 Example Interface Configuration

1. If possible, add every IP address, IP name, and IP alias for the nodes to `/etc/hosts` on one node.

For example:

```
190.0.2.1 xfs-ha1.company.com xfs-ha1
190.0.2.3 stocks
190.0.3.1 priv-xfs-ha1
190.0.2.2 xfs-ha2.company.com xfs-ha2
190.0.2.4 bonds
190.0.3.2 priv-xfs-ha2
```

Note: IP aliases that are used exclusively by highly available services are not added to the file `/etc/config/ipaliases.options`. Similarly, if all IP aliases are used only by highly available services, the `ipaliases chkconfig` flag should be `off`.

---

2. Add all of the IP addresses from Step 1 to */etc/hosts* on the other nodes in the cluster.

3. If there are IP addresses, IP names, or IP aliases that you did not add to `/etc/hosts` in Step 1 and 2, verify that NIS is configured on all nodes in the cluster by entering this command on each node:

   ```
   # chkconfig | grep yp
   ...
           yp              on
   ```

   If the output shows that `yp` is `off`, you must start NIS. See the *NIS Administrator's Guide* for details.

4. For IP addresses, IP names, and IP aliases that you did not add to `/etc/hosts` on the nodes in Steps 1 and 2, verify that they are in the NIS database by entering this command for each address:

   ```
   # ypmatch address hosts
   190.0.2.1 xfs-ha1.company.com xfs-ha1
   ```

   address is an IP address, IP name, or IP alias. If `ypmatch`(1M) reports that *address* doesn't match, it must be added to the NIS database. See the *NIS Administrator's Guide* for details.

5. On one node, add that node's interfaces and their IP addresses to the file `/etc/config/netif.options` (highly available IP addresses are not added to the `netif.options` file).

For the example in Figure 3-1, the public interface name and IP address lines are

```
if1name=ec0
if1addr=$HOSTNAME
```

$HOSTNAME is an alias for an IP address that appears in `/etc/hosts`.

If there are additional public interfaces, their interface names and IP addresses appear on lines like these:

```
if2name=
if2addr=
```

In the example, the control network name and IP address are

```
if3name=ec3
if3addr=priv-$HOSTNAME
```

The control network IP address in this example, priv-$HOSTNAME, is an alias for an IP address that appears in `/etc/hosts`.

6. If there are more than eight interfaces on the node, change the value of `if_num` to the number of interfaces. For less than eight interfaces (as in the example in Figure 3-1) the line looks like this:

   ```
   if_num=8
   ```

7. Repeat Steps 5 and 6 on the other nodes.

8. Edit the file `/etc/config/routed.options` on each node and add the `-q` option so that the routes are not shown over the control network (routing is turned off). An example of the content of `/etc/config/routed.options` on IRIX 6.5 nodes is:

   ```
   -h -Prdisc_interval=45 -q
   ```

   Note: The `-q` option is required for IRIS FailSafe to function correctly. This ensures that the heartbeat network does not get loaded with packets that are not related to the cluster.

9. Verify that IRIS FailSafe 2.*X* is `chkconfig`'d `off` on each node:

   ```
   # chkconfig | grep failsafe2
   ...
   ```

```
        failSafe2          off
...
```

If `failsafe2` is `on` on either node, enter this command on that node:

```
# chkconfig failSafe2 off
```

If Failsafe 1.X is present, you want to ensure that it is not configured `on` for any node, either. For each node, verify that IRIS FailSafe 1.X is `chkconfig`'d `off`:

```
# chkconfig | grep failsafe
...
        failsafe             off
...
```

If `failsafe` is `on` on any node, enter this command on that node:

```
# chkconfig failsafe off
```

10. Configure an e-mail alias on each node that sends the IRIS FailSafe e-mail notifications of cluster transitions to a user outside the IRIS FailSafe cluster and to a user on the other nodes in the cluster. For example, if there are two nodes called `xfs-ha1` and `xfs-ha2`, in `/usr/lib/aliases` on `xfs-ha1`, add

    ```
    fsafe_admin:operations@console.xyz.com,admin_user@xfs-ha2.xyz.com
    ```

    On `xfs-ha2`, add this line to `/usr/lib/aliases`:

    ```
    fsafe_admin:operations@console.xyz.com,admin_user@xfs-ha1.xyz.com
    ```

    The alias you choose, `fsafe_admin` in this case, is the value you will use for the mail destination address when you configure your system. In this example, `operations` is the user outside the cluster and `admin_user` is a user on each node.

11. If the nodes use NIS (`yp` is `chkconfig`'ed `on`) or the BIND domain name server (DNS), switching to local name resolution is recommended. On IRIS 6.5 systems, you should modify the `/etc/nsswitch.conf` file so that it reads as follows:

    ```
    hosts:                  files nis dns
    ```

Note: Exclusive use of NIS or DNS for IP address lookup for the nodes has been shown to reduce availability in situations where the NIS service becomes unreliable.

12. If FDDI is being used, finish configuring and verifying the new FDDI station, as explained in Chapter 2 of the FDDIXpress release notes and Chapter 2 of the *FDDIXPress Administration Guide*.
13. Reboot all nodes to put the new network configuration into effect.

Configuring the Serial Ports

The `getty` process for the tty ports to which the reset serial cables are connected must be turned off when a ring reset configuration is used. To do this, perform these steps on each node:

1. Determine which port is used for the reset serial line.
2. Open the file `/etc/inittab` for editing.
3. Find the line for the port by looking at the comments on the right for the port number from Step 1.
4. Change the third field of this line to `off`. For example:

```
t2:23:off:/sbin/getty -N ttyd2 co_9600          # port 2
```

5. Save the file.
6. Enter these commands to make the change take effect:

```
# killall getty
# init q
```

Note: If you configure a multinode cluster with the reset daemon running on an IRISconsole system, do not configure the reset port into the IRISconsole, because it may conflict with the reset daemon that the IRIS FailSafe system is running.

Installing an IRIS FailSafe Patch

The procedures in this section describe how to install a software path on a FailSafe 2.*X* system. The FailSafe patch should be installed on all nodes in the cluster.

This section includes the procedure for installing the FailSafe images and the FailSafe patch at the same time, and the procedure for installing just the FailSafe patch on an existing FailSafe cluster.

Installing FailSafe 2.*X* and FailSafe patch at the Same Time

When you install FailSafe 2.*X* images and an upgrade patch together, the cluster processes need to be stopped and started on each node after patch installation. This is because the FailSafe 2.*X* installation automatically starts the cluster processes and the patch installation does not automatically stop them, so the cluster processes will continue to run the unpatched shared libraries unless you restart them.

Follow these instructions to install FailSafe 2.*X* and an upgrade patch on each node:

1. Install FailSafe 2.*X* images on the node. This includes `cluster_admin`, `cluster_control`, `cluster_services`, `failsafe2`, `sysadm_base`, and `sysadm_failsafe2` products.
2. Install the FailSafe 2.*X* patch on the node.
3. In a Unix shell, stop all cluster processes on the node:

   ```
   # /etc/init.d/cluster stop
   ```

4. Verify that the cluster processes (`cad`, `cmond`, `crsd`, and `fs2d`) have stopped:

   ```
   # ps -ef | egrep '(cad|cmond|crsd|fs2d)'
   ```

5. Start cluster processes on the node:

   ```
   # /etc/init.d/cluster start
   ```

You are now ready to run the FailSafe Manager GUI or CLI to set up and begin using a FailSafe high availability cluster.

Installing a FailSafe patch on an Existing FailSafe 2.*X* Cluster

Using these instructions, you can install a FailSafe patch on each FailSafe 2.*X* node in turn, without shutting down the entire cluster and without interrupting the highly available services provided by the cluster.

Note: Before installing a FailSafe patch, you should read the patch release notes for the particular patchyou are installing. These release notes may contain special instructions that are not provided in this procedure.

To install a FailSafe patch on each node in your FailSafe cluster, follow these steps:

1. If you have the FailSafe Manager GUI client software installed on a machine that is not a node, first install the patch client subsystems on that machine. (The GUI client software subsystems are patchSG*xxxxxxx*.sysadm_base_sw.client, patchSG*xxxxxxx*.sysadm_failsafe2_sw.client, and patchSG*xxxxxxx*.sysadm_failsafe2_sw.desktop, where *xxxxxxx* is the patch number.)

2. Choose a node on which to install the FailSafe 2.*X* patch. Start up the FailSafe Manager GUI or CLI.

   For convenience, connect FailSafe Manager to the node that you are not upgrading. If you connect to the node that you are upgrading, then in a later step when you stop FailSafe HA services, FailSafe will no longer report accurate status to FailSafe Manager, and in another later step when you stop cluster services the FailSafe Manager GUI will lose its connection.

   You can use the following CLI command to select the node. This command assumes you have specified the cluster name:

   ```
   cmgr> set cluster <cluster name>
   ```

3. (Optional) If you wish to keep all resource groups running on the node during installation, take the resource groups offline using the "Detach" option (that is, detach the resource groups). If you do this, FailSafe will stop monitoring the resources, which will continue to run on the node, and will not have any control over the resource groups. Otherwise, in the next step the resources should migrate to the other node automatically, assuming the failover policy is defined that way.

   If you are using the FailSafe GUI, run the Take Resource Group Offline task and check the "Detach Only" checkbox.

If you are using the FailSafe CLI, execute the following command:

```
cmgr> admin offline_detach resource_group <group name>
```

4. Stop HA services on the node. (When FailSafe HA services stop, FailSafe will no longer be able to report current cluster and node state if the FailSafe Manager is connected to that node. To monitor the cluster state during installation, connect the FailSafe Manager to the node that you are not upgrading.)

   If you are using the FailSafe GUI, run the Stop FailSafe HA Services task, specifying the node you are patching in the "One Node Only" field.

   If you are using the FailSafe CLI, execute the following command:

   ```
   cmgr> stop ha_services on node <node name>
   ```

   If you skipped the previous optional step, FailSafe will attempt to migrate all resource groups off that node, but this will fail if there are no other available nodes in the resource group's failover domain. If this happens, either complete the previous step, or move the resource group to the other node:

   If you are using the FailSafe GUI, run the Move Resource Group task, specifying the node you are not patching in the "Failover Domain Node" field.

   If you are using the FailSafe CLI, execute the following command:

   ```
   cmgr> admin move resource_group <group name> to node <node name>
   ```

5. In a Unix shell on the node you are upgrading, stop all cluster processes:

   ```
   # /etc/init.d/cluster stop
   ```

   When you are using the FailSafe GUI, if the "connection lost" dialog appears, click No. If you wish to continue using the GUI, restart the GUI, connecting to the node you are not patching.

6. Verify that the cluster processes (`cad`, `cmond`, `crsd`, and `fs2d`) have stopped:

   ```
   # ps -ef | egrep '(cad|cmond|crsd|fs2d)'
   ```

7. Use `chkconfig`(1m) to turn off the `failsafe2` and `cluster` flags:

   ```
   # chkconfig failsafe2 off
   # chkconfig cluster off
   ```

8. Install the FailSafe 2.*X* patch on the node.

9. Use `chkconfig`(1m) to turn on the `failsafe2` and `cluster` flags:

   ```
   # chkconfig failsafe2 on
   # chkconfig cluster on
   ```

10. Start cluster processes on the node.

    ```
    # /etc/init.d/cluster start
    ```

11. Start HA services on the node.

    If you are using the FailSafe GUI and you are running the GUI in a Web browser, exit your browser, restart the Web server on the node you have just patched, and restart the GUI, connecting to the patched node.

    Run the Start FailSafe HA Services task, specifying the node that you just patched in the "One Node Only" field. If the GUI claims that FailSafe HA services are active on the cluster, then you are using an unpatched client; in this case, run the CLI command instead (in the next bullet), run the GUI on a patched client, or run the GUI in a Web browser from the patched node.

    If you are using the FailSafe CLI, execute the following command:

    ```
    cmgr> start ha_services on node <node name>
    ```

12. Monitor the resource groups and verify that they come back online on the upgraded node. This may take several minutes, depending on the types and numbers of resources in the groups.

    If you are using the FailSafe GUI, open the FailSafe Cluster View window. On the View menu, select Groups Owned by Nodes. Confirm that the resource group icons turn green (indicating online status).

    Note: When you restart HA services on the upgraded node, it can take several minutes for the node and cluster to return to normal Active state.

    If you are using the FailSafe CLI, execute the following command:

    ```
    cmgr> show status of resource_group <group name>
    ```

Repeat the above process for the other nodes. If you are using the GUI, remember to reconnect to the node that you have just upgraded. After completing the process for all nodes, you can continue to monitor and administer your upgraded cluster, defining additional new nodes if desired.

Installing Performance Co-Pilot Software

You can deploy PCP for FailSafe as a collector agent or as a monitor client:

- Collector agents are installed on *collector hosts*, which are the nodes in the FailSafe cluster itself from which you want to gather statistics. Typically, each node in a FailSafe cluster is designated as a collector host.
- A monitor client is installed on the ***monitor host***, which is typically a workstation that has a display and is running the IRIS Desktop.

Installing the Collector Host

To install PCP for FailSafe on the designated collector hosts, the following software components must already be installed:

- The `pcp_eoe.sw` subsystem from IRIX 6.5.6 or later
- IRIS FailSafe 2.1 or later
- PCP 2.1 or later

A collector license (`PCPCOL`) must also be installed on each of these nodes.

After this software is installed, you must install the following subsystems of PCP for FailSafe on each collector host. Table 3-1 lists the subsystems required for a collector host, and their approximate sizes:

Table 3-1 PCP for FailSafe Collector Subsystems

Subsystem	Size in Kbytes
`pcp_fsafe.man.pages`	40
`pcp_fsafe.man.relnotes`	32
`pcp_fsafe.sw.collector`	128

To install the required subsystems on a monitor host, do the following:

1. Mount the FailSafe CD-ROM by inserting it into an available drive. You can access a local CD-ROM drive, or a remote CD-ROM drive of another host over the network.

2. Log in as `root`.
3. Start the `inst`(1) command:

   ```
   # inst
   ```

4. Specify the installation location:

   - If you are installing from the local CD-ROM drive, enter the following:

     ```
     Inst> from /CDROM/dist
     ```

   - If you are installing from a remote drive, enter the following, where *host* is the host with the CD-ROM drive that contains a mounted FailSafe CD-ROM:

     ```
     Inst> from host:/CDROM/dist
     ```

5. Select the default subsystems in the `pcp_fsafe` package. The default subsystems are provided for easy installation onto multiple collector hosts.

   ```
   Inst> install default
   ```

6. Ensure that there are no conflicts:

   ```
   Inst> conflicts
   ```

7. Install the software:

   ```
   Inst> go
   ```

8. Change to the `/var/pcp/pmdas/fsafe` directory:

   ```
   # cd /var/pcp/pmdas/fsafe
   ```

9. Run the `Install` utility, which installs the FailSafe performance metrics into the PCP performance metrics namespace:

   ```
   # ./Install
   ```

10. Choose an appropriate configuration for installation of the `fsafe` Performance Metrics Domain Agent (PMDA):

`collector`	Collects performance statistics on this system
`monitor`	Allows this system to monitor local and/or remote systems

`both`	Allows collector and monitor configuration for this system

For example, to choose just the collector, enter the following:

```
Please enter c(ollector) or m(onitor) or b(oth) [b] c
```

Removing Performance Metrics from a Collector Host

If you wish to remove PCP for FailSafe from a collector host, you will need to remove the PCP for FailSafe metrics from the performance metrics namespace of that host. You can do this before removing the `pcp_fsafe` subsystem by performing the following commands:

1. Change to the `/var/pcp/pmdas/fsafe` directory:

   ```
   # cd /var/pcp/pmdas/fsafe
   ```

2. Run the `Remove` utility:

   ```
   # ./Remove
   ```

Installing the Monitor Host

To install PCP for FailSafe on a designated monitor host, the following software components must already installed on them:

- The `pcp_eoe.sw` subsystem of IRIX 6.5.6 or later, including the subsystem `pcp_eoe.sw.monitor`
- PCP 2.1 or later, including the subsystem `pcp.sw.monitor`

The monitor license (`PCPMON`) must also be installed on the monitor host.

After this software is installed, install the following subsystems of PCP for FailSafe on each collector host. Table 3-2 lists the subsystems required for a collector host, and their approximate sizes:

Table 3-2 PCP for FailSafe Monitor Subsystems

Subsystem	Size in Kbytes
`pcp_fsafe.man.pages`	40
`pcp_fsafe.man.relnotes`	32
`pcp_fsafe.sw.monitor`	516

To install the required subsystems for PCP for FailSafe on a monitor host, do the following:

1. Mount the PCP for FailSafe CD-ROM by inserting it into an available drive. You can access a local CD-ROM drive, or a remote CD-ROM drive of another host over the network.
2. Log in as `root`.
3. Start `inst`(1) :

   ```
   # inst
   ```

4. Specify the installation location:

   - If you are installing from the local CD-ROM drive, enter the following:

     ```
     Inst> from /CDROM/dist
     ```

   - If you are installing from a remote drive, enter the following, where *host* is the host with the CD-ROM drive that contains a mounted PCP for FailSafe CD-ROM:

     ```
     Inst> from host:/CDROM/dist
     ```

5. Select the required subsystems in the `pcp_fsafe` package for a monitor configuration:

   ```
   Inst> keep pcp_fsafe.sw.collector
   Inst> install pcp_fsafe.sw.monitor
   ```

6. Ensure that there are no conflicts before you install PCP for FailSafe:

   ```
   Inst> conflicts
   ```

7. Install the software:

```
Inst> go
```

IRIS FailSafe Administration Tools

This chapter describes IRIS FailSafe administration tools and their operation. The major sections in this chapter are as follows:

The IRIS FailSafe Cluster Manager Tools

You can perform the IRIS FailSafe administrative tasks using either of the following tools:

- The IRIS FailSafe Cluster Manager Graphical User Interface (GUI)
- The IRIS FailSafe Cluster Manager Command Line Interface (CLI)

Although these tools use the same underlying software to configure and monitor a FailSafe system, the GUI provides the following additional features, which are particularly important in a production system:

- Online help is provided with the **Help** button. You can also click any blue text to get more information about that concept or input field.
- The cluster state is shown visually for instant recognition of status, problems, and failovers.
- The state is updated dynamically for continuous system monitoring.
- All inputs are checked for correct syntax before attempting to change the cluster database information. In every task, the cluster configuration will not update until you click **OK**.
- Tasks and tasksets take you step-by-step through configuration and management operations, making actual changes to the cluster database as the you perform a task.
- The graphical tools can be run securely and remotely on any computer that has a Java virtual machine, including Windows® computers and laptops.

The IRIS FailSafe Cluster Manager CLI, on the other hand, is more limited in its functions. It enables you to configure and administer an IRIS FailSafe system using a command-line interface only on an IRIX system. It provides a minimum of help or formatted output and does not provide dynamic status except when queried. An experienced IRIS FailSafe administrator may find the Cluster Manager CLI to be convenient when performing basic IRIS FailSafe configuration tasks, isolated single tasks in a production environment, or when running scripts to automate some cluster administration tasks.

The cluster manager GUI uses underlying FailSafe commands to perform administration commands and update the configuration database. The Cluster Manager CLI uses the same underlying FailSafe administration commands as the Cluster Manager GUI.

Using the IRIS FailSafe Cluster Manager GUI

The IRIS FailSafe Cluster Manager GUI lets you configure, administer, and monitor a cluster using a graphical user interface. To ensure that the required privileges are available for performing all of the tasks, you should log in to the GUI as `root`. However, some or all privileges can be granted to any user by the system administrator using the Privilege Manager, part of the IRIX Interactive Desktop System Administration (`sysadmdesktop`) product. For more information, see the *Personal System Administration Guide*.

The Cluster Manager GUI uses an underlying FailSafe commands to perform administration commands and update the configuration database. The FailSafe CAD provides information to the GUI when there is a change in status or changes in the configuration database. The local CAD provides configuration database changes every 10 seconds; this can make GUI updates slow.

The Cluster Manager GUI consists of the FailSafe Cluster View and the FailSafe Manager and its tasks and tasksets. These interfaces are described in the following sections.

The FailSafe Cluster View

The FailSafe Cluster View window provides the following capabilities:

- Shows the relationships among the cluster items (nodes, resources groups, etc.)
- Gives access to every item's configuration and status details

- Shows health of the cluster
- Gives access to the FailSafe Manager and to the SYSLOG
- Gives access to Help information

From the FailSafe Cluster View, the user can click on any item to display key information about it. The items that can be viewed in this way are the following:

- Clusters
- Nodes
- Resource Types
- Resources
- Resource Groups
- Failover Policies

The FailSafe Manager

The FailSafe Manager provides access to the tasks that help you set up and administer your highly available cluster. The FailSafe Manager also provides access to the IRIS FailSafe Guided Configuration tasksets.

- Tasksets consist of a group of tasks collected together to accomplish a larger goal. For example, "Set Up a New Cluster" steps you through the process for creating a new cluster and allows you to launch the necessary tasks by simply clicking their titles.
- IRIS FailSafe tasksets let you set up and monitor all the components of a FailSafe cluster using an easy-to-use graphical user interface.

Starting the IRIS FailSafe Manager GUI

You can start the FailSafe Manager GUI by launching either the FailSafe Manager or the FailSafe Cluster View.

To launch the FailSafe Manager, use one of these methods:

- Choose "FailSafe Manager" from the FailSafe toolchest.

You will need to restart the toolchest after installing FailSafe to see the FailSafe entry on the toolchest display. Enter the following commands to restart the toolchest:

```
% killall toolchest
% /usr/bin/X11/toolchest &
```

In order for this to take effect, `sysadm_failsafe2.sw.desktop` must be installed on the client system, as described in the *IRIS FailSafe Installation and Maintenance Instructions*.

- Enter the following command line:

  ```
  % /usr/sbin/fstask
  ```

- In your Web browser, enter `http://`*server*`/FailSafeManager/` (where *server* is the name of node in the pool or cluster that you want to administer) and press Enter. At the resulting Web page, click on the shield icon.

  This method of launching FailSafe Manager works only if you have installed the Java Plug-in, exited all Java processes, restarted your browser, and enabled Java. If there is a long delay before the shield appears, you can click on the "non plug-in" link, but operational glitches may be the result of running in the browser-specific Java.

  You can use this method of launching FailSafe Manager if you want to administer the Cluster Manager GUI from a non-IRIX system. If you are running the Cluster Manager GUI on an IRIX system, the preferred method is to use toolchest or the `/usr/sbin/fstask` command.

To launch the FailSafe Cluster View, use one of these methods:

- Choose "FailSafe Cluster View" from the FailSafe toolchest.
- Enter the following command line:

  ```
  % /usr/sbin/fsdetail
  ```

The Cluster Manager GUI allows you to administer the entire cluster from a single point of administration. When FailSafe daemons have been activated in a cluster, you must be sure to connect to a node that is running all the FailSafe daemons to obtain the correct cluster status. When FailSafe daemons have not yet been activated in a cluster, you can connect to any node in the pool.

Opening the FailSafe Cluster View window

You can open the FailSafe Cluster View window using either of the following methods:

- Click the "FailSafe Cluster View" button at the bottom of the FailSafe Manager window.

  This is the preferred method of opening the FailSafe Cluster View window if you will have both the FailSafe Manager and the FailSafe Cluster View windows open at the same time, since it reuses the existing Java process to open the second window instead of starting a new one, which saves memory usage on the client.

- Open the FailSafe Cluster View window directly when you start the FailSafe Manager GUI, as described above in "Starting the IRIS FailSafe Manager GUI".

Viewing Cluster Item Details

To view the details on any cluster item, use the following procedure:

1. Open the FailSafe Cluster View Window.
2. Click the name or icon of any item.

The configuration and status details will appear in a separate window. To see the details in the same window, select Options. When you then click on the Show Details option, the status details will appear in the right side of the window.

Performing Tasks

To perform an individual task with the FailSafe GUI, do the following:

1. Click the name of a category in the lefthand column of the FailSafe Manager window.

   A list of individual tasksets and taskset topics appears in the righthand column.

2. Click the title of a task in the righthand column.

   The task window appears.

Note: You can click any blue text to get more information about that concept or input field.

3. Enter information in the appropriate fields and click **OK.** to complete the task. (Some tasks consist of more than one window; in these cases, click **Next** to go to the next window, complete the information there, and then click **OK.**

   A dialog box appears confirming the successful completion of the task and displaying additional tasks that you can launch.

4. Continue launching tasks as needed.

Using the FailSafe Tasksets

The FailSafe Manager GUI also provides tasksets to guide you through the steps necessary to complete a goal that encompasses several different tasks. Follow these steps to access the FailSafe tasksets:

1. Click the Guided Configuration category in the lefthand column of the FailSafe Manager window.

   A list of tasksets appears in the right hand column.

2. Click a taskset in the righthand column.

   A window appears and lists the series of tasks necessary to accomplish the desired goal.

3. Follow the steps shown, launching tasks by clicking them.

   As you click a task, its task window appears. After you complete all of the tasks listed, you can close the taskset window by double-clicking the upper left corner of its window or clicking Close if there is a Close button on the window.

Using the IRIS FailSafe Cluster Manager CLI

This section documents how to perform IRIS FailSafe administrative tasks by means of the IRIS FailSafe Cluster Manager CLI. In order to execute commands with the IRIS FailSafe Cluster Manager CLI, you should be logged in as root.

The Cluster Manager CLI uses the same underlying FailSafe commands as the Cluster Manager GUI.

To use the cluster manager, enter either of the following:

```
# /usr/cluster/bin/cluster_mgr
```

or

```
# /usr/cluster/bin/cmgr
```

After you have entered this command, you should see the following message and the cluster manager CLI command prompt:

```
Welcome to SGI Cluster Manager Command-Line Interface
cmgr>
```

Once the command prompt displays, you can enter the cluster manager commands.

At any time, you can enter `?` or `help` to bring up the CLI help display.

When you are creating or modifying a component of a FailSafe system, you can enter either of the following commands:

`cancel`	Abort the current mode and discard any changes you have made.
`done`	Commit the current definitions or modifications and return to the `cmgr` prompt.

Entering CLI Commands Directly

There are some Cluster Manager CLI command that you can execute directly from the command line, without entering `cmgr` mode, by using the `-c` option of the `cluster_mgr` command. These commands are `show`, `delete`, `admin`, `install`, `start`, `stop`, `test`, `help`, and `quit`. You can execute these commands directly using the following format:

```
cluster_mgr -c "command"
```

For example, you can execute a `show clusters` CLI command as follows:

```
% cluster_mgr -c "show clusters"

1 Cluster(s) defined
        eagan
```

Invoking the Cluster Manager CLI in “Prompt” Mode

The Cluster Manager CLI provides an option which displays prompts for the required inputs of administration commands that define and modify FailSafe components. You can run the CLI in prompt mode in either of the following ways:

- Specify a `-p` option when you enter the `cluster_mgr` (or `cmgr`) command, as in the following example:

  ```
  # cluster_mgr -p
  ```

- Execute a `set prompting on` command after you have brought up the CLI, as in the following example:

  ```
  cmgr> set prompting on
  ```

  This method of entering prompt mode allows you to toggle in and out of prompt mode as you execute individual CLI commands. To get out of prompt mode while you are running the CLI, enter the following CLI command:

  ```
  cmgr> set prompting
  ```

For example, if you are not in the prompt mode of the CLI and you enter the following command to define a node, you will see a single prompt, as indicated:

```
cmgr> define node A
Enter commands, when finished enter either "done" or "cancel"

A?
```

At this prompt, you enter the individual node definition commands in the following format (for full information on defining nodes, see "Defining a Node with the Cluster Manager CLI", page 100):

```
set hostname to B
 set nodeid to C
 set partition_id to D
 set reset_type to E
 set sysctrl_type to F
 set sysctrl_password to G
 set sysctrl_status to H
 set sysctrl_owner to I
 set sysctrl_device to J
 set sysctrl_owner_type to K
 set is_failsafe to L
```

```
set is_cxfs to M
set weight to N
add nic O
        set heartbeat to P
        set ctrl_msgs to Q
        set priority to R
remove nic S
```

Then, after you add a network interface, a prompt appears requesting the parameters for the network interface, which you enter similarly.

If you are running CLI in prompt mode, however, the display appears as follows (when you provide the appropriate inputs):

```
cmgr> define node cm1a
Enter commands, you may enter "done" or "cancel" at any time to exit

Node Name [cm1a]? cm1a

Hostname[optional]? cm1a
Is this a FailSafe node <true|false> ? true
Is this a CXFS node <true|false> ? false
Node ID ? 1
Reset type <powerCycle> ? (powerCycle)
Do you wish to define system controller info[y/n]:y
Sysctrl Type <msc|mmsc|l2>? (msc) msc
Sysctrl Password [optional]? ( )
Sysctrl Status <enabled|disabled>? enabled
Sysctrl Owner? cm2
Sysctrl Device? /dev/ttyd2
Sysctrl Owner Type <tty> [tty]?
Number of Network interfaces [2]? 2
NIC 1 - IP Address? cm1
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false>? true
NIC 1 - (use network for control messages) <true|false>? true
NIC 1 - Priority <1,2,...>? 1
NIC 2 - IP Address? cm2
NIC 2 Heartbeat HB (use network for heartbeats) <true|false>? true
NIC 2 - (use network for control messages) <true|false>? false
NIC 2 - Priority <1,2,...>? 2
```

CLI Startup Script

You can set the environment variable `CMGR_START_FILE` to point to a startup `cluster_mgr` script. The startup script that this variable specifies is executed when `cluster_mgr` is started (with or without the `-p` option). Only the `set` and `show` commands of the `cluster_mgr` are allowed in the `cluster_mgr` startup file.

The following is an example of a `cluster_mgr` startup script file called `cmgr_rc`:

```
set cluster test-cluster
show status of resource_group oracle_rg
```

To specify this file as the CLI startup script, execute the following command at the IRIX prompt:

```
$ setenv CMGR_START_FILE /cmgr_rc
```

Whenever Cluster Manager CLI is started, the `cmgr_rc` script is executed. The default cluster is set to `test-cluster` and the status of resource group `oracle_rg` in cluster `test-cluster` is displayed.

Using Input Files of CLI Commands

You can execute a series of Cluster Manager CLI commands by using the `-f` option of the `cluster_mgr` command and specifying an input file:

`cluster_mgr -f "`*input_file*"

The input file must contain Cluster Manager CLI commands and end with a `quit` command.

For example, the file `input.file` contains the following:

```
show clusters

show nodes in cluster beta3
quit
```

You can execute the following command, which will yield the indicated output:

```
% cluster_mgr -f input.file

1 Cluster(s) defined
        eagan
Cluster eagan has following 2 machine(s)
```

```
cm1
cm2
```

The `cluster_mgr` command provides a `-i` option to be used with the `-f` option. This is the "ignore" option which indicates that the Cluster Manager should not exit if a command fails while executing a script.

CLI Command Scripts

You can use the `-f` option of the `cluster_mgr` command to write a script of Cluster Manager CLI commands that you can execute directly. The script must contain the following line as the first line of the script.

```
#!/usr/cluster/bin/cluster_mgr -f
```

Note: When you use the `-i` option of the `cluster_mgr` command to indicate that the Cluster Manager should not exit if a command fails while executing a script, you must use the following syntax in the first line of the script file: `#!/usr/cluster/bin/cluster_mgr -if`. It is not necessary to use the `-if` syntax when using the `-i` option from the command line directly.

Each line of the script must be a valid `cluster_mgr command` line, similar to a `here` document. Because the Cluster Manager CLI will run through commands as if entered interactively, you must include `done` and `quit` lines to finish a multi-level command and exit out of the Cluster Manager CLI.

There are CLI template files of scripts that you can modify to configure the different components of your system. These files are located in the `/var/cluster/cmgr-templates` directory. For information on CLI templates, see "CLI Template Scripts", page 90.

The following shows an example of a CLI command script `cli.script`.

```
% more cli.script
#!/usr/cluster/bin/cluster_mgr -f

show clusters
show nodes in cluster beta3
quit

% cli.script
```

```
1 Cluster(s) defined
        eagan
Cluster eagan has following 2 machine(s)
        cm1
        cm2

%
```

CLI Template Scripts

Template files of CLI scripts that you can modify to configure the different components of your system are located in the `/var/cluster/cmgr-templates` directory.

Each template file contains list of `cluster_mgr` commands to create a particular object, as well as comments describing each field. The template also provides default values for optional fields.

The `var/cluster/cmgr-templates` directory contains following templates:

File name	Description
cmgr-create-cluster	Creation of a cluster
cmgr-create-failover_policy	Creation of failover policy
cmgr-create-node	Creation of node
cmgr-create-resource_group	Creation of Resource Group
cmgr-create-resource_type	Creation of resource type
cmgr-create-resource-*resource type*	CLI script template for creation of resource of type *resource type*

To create a FailSafe configuration, you can concatenate multiple templates into one file and execute the resulting CLI command script.

Note: If you concatenate information from multiple template scripts to prepare your cluster configuration, you must remove the `quit` at the end of each template script, except for the final `quit`. A `cluster_mgr` script must have only one `quit` line.

For example: For a 3 node configuration with an NFS resource group containing 1 volume, 1 filesystem, 1 IP address and 1 NFS resource, you would concatenate the following files, removing the `quit` at the end of each template script except the last one:

- 3 copies of the `cmgr-create-node` file
- 1 copy of the `cmgr-create-cluster` file
- 1 copy of the `cmgr-create-failover_policy` file
- 1 copy of the `cmgr-create-resource_group` file
- 1 copy of the `cmgr-create-resource-volume` file
- 1 copy of the `cmgr-create-resource-filesystem` file
- 1 copy of the `cmgr-create-resource-IP_address` file
- 1 copy of the `cmgr-create-resource-NFS` file

Invoking a Shell from within CLI

You can invoke a shell from within the Cluster Manager CLI. Enter the following command to invoke a shell:

```
cmgr> sh
```

To exit the shell and to return to the CLI, enter "exit" at the shell prompt.

IRIS FailSafe Configuration

This chapter describes administrative tasks you perform to configure the components of an IRIS FailSafe system. It describes how to perform tasks using the IRIS FailSafe Cluster Manager Graphical User Interface (GUI) and the IRIS FailSafe Cluster Manager Command Line Interface (CLI). The major sections in this chapter are as follows:

Note: It is recommended that all FailSafe administration be done from one node in the pool so that the latest copy of the database will be available even when there are network partitions.

Setting Configuration Defaults

Before you configure the components of an IRIS FailSafe system, you can set default values for some of the components that IRIS FailSafe will use when defining the components.

Default cluster	Certain cluster manager commands require you to specify a cluster. You can specify a default cluster to use as the default if you do not specify a cluster explicitly.
Default node	Certain cluster manager commands require you to specify a node. With the Cluster Manager CLI, you can specify a default node to use as the default if you do not specify a node explicitly.

Default resource type	Certain cluster manager commands require you to specify a resource type. With the Cluster Manager CLI, you can specify a default resource type to use as the default if you do not specify a resource type explicitly.

Setting Default Cluster with the Cluster Manager GUI

The GUI prompts you to enter the name of the default cluster when you have not specified one. Alternately, you can set the default cluster by clicking the "Select Cluster..." button at the bottom of the FailSafe Manager window.

When using the GUI, there is no need to set a default node or resource type.

Setting and Viewing Configuration Defaults with the Cluster Manager CLI

When you are using the Cluster Manager CLI, you can use the following commands to specify default values. The default values are in effect only for the current session of the Cluster Manager CLI.

Use the following command to specify a default cluster:

```
cmgr> set cluster A
```

Use the following command to specify a default node:

```
cmgr> set node A
```

Use the following command to specify a default resource type:

```
cmgr> set resource_type A
```

You can view the current default configuration values of the Cluster Manager CLI with the following command:

```
cmgr> show set defaults
```

Name Restrictions

When you specify the names of the various components of a FailSafe system, the name cannot begin with an underscore (_) or include any whitespace characters. In

addition, the name of any FailSafe component cannot contain a space, an unprintable character, or a *, ?, \, or #.

The following is the list of permitted characters for the name of a FailSafe component:

- alphanumeric characters
- /
- .
- - (hyphen)
- _ (underscore)
- :
- “
- =
- @
- ’

These character restrictions hold true whether you are configuring your system with the Cluster Manager GUI or the Cluster Manager CLI.

Configuring Timeout Values and Monitoring Intervals

When you configure the components of a FailSafe system, you configure various timeout values and monitoring intervals that determine the application downtown of a highly-available system when there is a failure. To determine reasonable values to set for your system, consider the following equations:

application downtime = failure detection + time to handle failure + failure recovery time

Failure detection depends on the type of failure that is detected:

- When a node goes down, there will be a node failure detection after the node timeout time, which is one of the IRIS FailSafe HA parameters that you can modify. All failures that translate into a node failure such as heartbeat failure and OS failure fall into this failure category. Node timeout time has a default value of

15 seconds. For information on modifying the node timeout value, see "IRIS FailSafe HA Parameters", page 107.

- When there is a resource failure, there will be a monitor failure of a resource. The time this will take is determined by the following:
  - The monitoring interval for the resource type
  - The monitor timeout for the resource type
  - The number of restarts defined for the resource type, if the restart mode is configured on

  For information on setting values for a resource type, see "Defining a Resource Type", page 130.

Reducing these values will result in a shorter failover time, but reducing these values could lead to significant increase in the FailSafe overhead on the system performance and could also lead to false failovers.

The time to handle a failure is something that the user cannot control. In general, this should take a few seconds.

The failure recovery time is determined by the total time it takes for FailSafe to perform the following:

- Execute the failover policy script (approximately five seconds).
- Run the stop action script for all resources in the resource group. This is not required for node failure; the failing node will be reset.
- Run the start action script for all resources in the resource group

Cluster Configuration

To set up an IRIS FailSafe system, you configure the cluster that will support the highly available services. This requires the following steps:

- Defining the local host
- Defining any additional nodes that are eligible to be included in the cluster
- Defining the cluster

The following subsections describe these tasks.

Defining Nodes

A *node* is a single UNIX image. Usually, a node is an individual computer. The term *node* is also used in this guide for brevity; this use of node does not have the same meaning as a node in an Origin system.

The *pool* is the entire set of *nodes* available for clustering.

The first node you define must be the local host, which is the host you have logged into to perform cluster administration.

When you are defining multiple nodes, it is advisable to wait for a minute or so between each node definition. When nodes are added to the configuration database, the contents of the configuration database currently on that node are marked as obsolete and the configuration database from a node in the current quorum is copied to the node being added. The node definition operation is completed when the new node configuration is added to the database, at which point the database configuration is synchronized. If you define two nodes one after another, the second operation might fail because the first database synchronization is not complete.

To add a logical node definition to the pool of nodes that are eligible to be included in a cluster, you must provide the following information about the node:

- The logical name of the node. This name can contain letters and numbers byt not spaces or pound signs. The name must be composed of no more than 255 characters. Any legal hostname is also a legal node name. For example, for a node whose hostname is `venus.eng.company.com`, you can use a node name of `venus`, `node1`, or whatever is most convenient.
- The hostname. This is the fully qualified name of the host, such as `server1.company.com`. Hostnames cannot begin with an underscore, include any whitespace, or be longer than 255 characters. This address should be the same as the output of the `hostname` command on the node you are defining. The IP address associated with this hostname should not be the same as any IP address you define as highly available when you define a FailSafe `IP_address` resource. FailSafe will not accept an IP address (such as `192.0.2.22`) for this input.
- Node ID, a 16-bit unsigned value (optionally specified by user). The default value is obtained from the CAD process. This number must be unique for each node in the pool and be in the range 1 through 32767.
- Partition ID. Uniquely defines a partition in a partitioned Origin 3000 system.

Note: Use the `mkpart`(1m) command to determine this value:
– the `-n` option lists the partition ID (which is 0 if the system is not partitioned).

– The `-l` option lists the bricks in the various partitions (use *rack#.slot#* format in the CLI).

For example:

```
# mkpart -n
Partition id = 1
# mkpart -l
partition: 3 = brick: 003c10 003c13 003c16 003c21 003c24 003c29 ...
partition: 1 = brick: 001c10 001c13 001c16 001c21 003c24 001c29 ...
```

You could enter one of the following for the **Partition ID** field:

```
1
001.10
```

If your system is not partitioned, leave this field empty.

To unset the partition ID, use a value of `0` or `none`.

- System controller information. If the node has a system controller and you want FailSafe to use the controller to reset the node, you must provide the following information about the system controller:
  – Type of system controller:
    - `msc` Module System Controller
    - `mmsc` Multimodule System Controller
    - `l2` L2 system controller for Origin 3000
  – The reset type of the system controller: only `powerCycle` is supported
  – System controller port password (optional)
  – Administrative status, which you can set to determine whether FailSafe can use the port: `enabled`, `disabled`
  – Logical node name of system controller owner (i.e. the system that is physically attached to the system controller)

  - Device name of port on owner node that is attached to the system controller
  - Type of owner device: `tty`

- Whether the node is a FailSafe node (Cluster Manager CLI only; a node defined with the FailSafe Cluster Manager GUI is a FailSafe node). For information on FailSafe and CXFS nodes, see "Coexecution with CXFS", page 49.
- Whether the node is a CXFS node (Cluster Manager CLI only; a node defined with the FailSafe Cluster Manager GUI is a FailSafe node). For information on FailSafe and CXFS nodes, see "Coexecution with CXFS", page 49.
- A list of control networks, which are the networks used for heartbeats, reset messages, and other FailSafe messages. For each network, provide the following:
  - Hostname or IP address. This address must not be the same as any IP address you define as highly available when you define a FailSafe `IP_address` resource, and it must be resolved in the `/etc/hosts` file.
  - Flags (`hb` for heartbeats, `ctrl` for control messages, `priority`). At least two control networks must use heartbeats, and at least one must use control messages.

    FailSafe requires multiple heartbeat networks. Usually a node sends heartbeat messages to another node on only one network at a time. However, there are times when a node might send heartbeat messages to another node on multiple networks simultaneously. This happens when the sender node does not know which networks are up and which others are down. This is a transient state and eventually the heartbeat network converges towards the highest priority network that is up. This is unlike FailSafe 1.2, where the heartbeat networks were tried sequentially one at a time.

    Note that at any time different pairs of nodes might be using different networks for heartbeats.

    Although all nodes in the FailSafe cluster should have two control networks, it is possible to define a node to add to the pool with one control network.

Defining a Node with the Cluster Manager GUI

To define a node with the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.

3. On the right side of the display click on the “Define a Node” task link to launch the task.
4. Enter the selected inputs on the screen. Click on “Next” at the bottom of the screen and continue inputting information on the second screen.

Defining a Node with the Cluster Manager CLI

Use the following command to add a logical node definition:

```
cmgr> define node A
```

Entering this command specifies the name of the node you are defining and puts you in a mode that enables you to define the parameters of the node. These parameters correspond to the items defined in "Defining Nodes", page 97. The following prompts appear:

```
Enter commands, you may enter "done" or "cancel" at any time to exit

A?
```

When this prompt of the node name appears, you enter the node parameters in the following format:

```
set hostname to B
 set nodeid to C
 set partition_id to D
 set reset_type to E
 set sysctrl_type to F
 set sysctrl_password to G
 set sysctrl_status to H
 set sysctrl_owner to I
 set sysctrl_device to J
 set sysctrl_owner_type to K
 set is_failsafe to L
 set is_cxfs to M
 set weight to N
 add nic O
         set heartbeat to P
         set ctrl_msgs to Q
         set priority to R
 remove nic S
```

You use the `add nic` *J* command to define the network interfaces. You use this command for each network interface to define. When you enter this command, the following prompt appears:

```
Enter network interface commands, when finished enter "done" or "cancel"
NIC - J?
```

When this prompt appears, you use the following commands to specify the flags for the control network:

```
set heartbeat to O
set ctrl_msgs to P
set priority to Q
```

After you have defined a network controller, you can use the following command from the node name prompt to remove it:

```
cmgr> remove nic R
```

When you have finished defining a node, enter `done`.

The following example defines a FailSafe node called `cm1a`, with one controller:

```
cmgr> define node cm1a
Enter commands, you may enter "done" or "cancel" at any time to exit

cm1a? set hostname to cm1a
cm1a? set nodeid to 1
cm1a? set reset_type to powerCycle
cm1a? set sysctrl_type to msc
cm1a? set sysctrl_password to [ ]
cm1a? set sysctrl_status to enabled
cm1a? set sysctrl_owner to cm2
cm1a? set sysctrl_device to /dev/ttyd2
cm1a? set sysctrl_owner_type to tty
cm1a? set is_failsafe to true
cm1a? set is_cxfs to true
cm1a? add nic cm1
Enter network interface commands, when finished enter "done"
or "cancel"

NIC - cm1 > set heartbeat to true
NIC - cm1 > set ctrl_msgs to true
NIC - cm1 > set priority to 0
```

```
NIC - cm1 > done
cm1a? done
cmgr>
```

If you have invoked the Cluster Manager CLI with the `-p` option or you entered the "set prompting on" command, the display appears as in the following example:

```
cmgr> define node cm1a
Enter commands, when finished enter either "done" or "cancel"

Hostname[optional]? cm1a
Is this a FailSafe node <true|false> ? true
Is this a CXFS node <true|false> ? false
Node ID ? 1
Reset type <powerCycle> ? (powerCycle)
Do you wish to define system controller info[y/n]:y
Sysctrl Type <msc|mmsc|l2>? (msc) msc
Sysctrl Password [optional]? ( )
Sysctrl Status <enabled|disabled>? enabled
Sysctrl Owner? cm2
Sysctrl Device? /dev/ttyd2
Sysctrl Owner Type <tty> [tty]?
Number of Network interfaces [2]? 2
NIC 1 - IP Address? cm1
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false>? true
NIC 1 - (use network for control messages) <true|false>? true
NIC 1 - Priority <1,2,...>? 1
NIC 2 - IP Address? cm2
NIC 2 Heartbeat HB (use network for heartbeats) <true|false>? true
NIC 2 - (use network for control messages) <true|false>? false
NIC 2 - Priority <1,2,...>? 2
```

Converting a CXFS Node to FailSafe

You can reconfigure an existing CXFS node so that it applies to FailSafe.

Converting a CXFS Node to FailSafe with the Cluster Manager GUI

To use the Cluster Manager GUI to reconfigure a CXFS node so that it can be used with both CXFS and FailSafe, perform the following procedure. If you have not

installed CXFS on the server to which you connected with the GUI, this task does not appear on the GUI menu.

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.
3. On the right side of the display click on the "Convert a CXFS Node to FailSafe" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

The node will apply to both CXFS and FailSafe.

Converting a CXFS Node to Failsafe with the Cluster Manager CLI

To convert an existing CXFS node so that it also applies to FailSafe, use the `modify` command of `cmgr` to change the setting.

Note: You cannot turn off FailSafe or CXFS for a node if the respective high availability (HA) or CXFS services are active. You must first stop the services for the node.

Example using prompting:

```
cmgr> modify node cxfs6
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ? (cxfs6.americas.sgi.com)
Is this a FailSafe node  ? (false) true
Is this a CXFS node  ? (true)
Node ID[optional] ? (13203)
Reset type  ? (powerCycle)
Do you wish to modify system controller info[y/n]:n
Number of Network Interfaces ? (1)
NIC 1 - IP Address ? (cxfs6)
NIC 1 - Heartbeat HB (use network for heartbeats)  ? (true)
NIC 1 - (use network for control messages)  ? (true)
NIC 1 - Priority <1,2,...> ? (1)
Node Weight ? (0)
```

```
Successfully modified node cxfs6
```

Example without prompting:

```
cmgr> modify node cxfs6
Enter commands, when finished enter either "done" or "cancel"

cxfs6 ? set is_FailSafe to true
cxfs6 ? done

Successfully modified node cxfs6
```

Modifying Nodes

After you have defined a node, you can modify the noder with the Cluster Manager GUI or the Cluster Manager CLI.

Modifying a Node with the Cluster Manager GUI

To modify a node with the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.
3. On the right side of the display click on the "Modify a Node Definition" task link to launch the task.
4. Modify the node parameters.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Modifying a Node with the Cluster Manager CLI

You can use the following command to modify an existing node. After entering this command, you can execute any of the commands you use to define a node.

```
cmgr> modify node A
```

Deleting Nodes

After you have defined a node, you delete the cluster with the Cluster Manager GUI or the Cluster Manager CLI. You must remove a node from a cluster before you can delete the node.

Deleting a Node with the Cluster Manager GUI

To delete a node with the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.
3. On the right side of the display click on the "Delete a Node" task link to launch the task.
4. Enter the name of the node to delete.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Deleting a Node with the Cluster Manager CLI

After defining a node, you can delete it with the following command:

```
cmgr> delete node A
```

You can delete a node only if the node is not currently part of a cluster. This means that first you must modify a cluster that contains the node so that it no longer contains that node before you can delete it.

Displaying Nodes

After you define nodes, you can perform the following display tasks:

- display the attributes of a node
- display the nodes that are members of a specific cluster
- display all the nodes that have been defined

You can perform any of these tasks with the IRIS FailSafe Cluster Manager GUI or the IRIS FailSafe Cluster Manager CLI.

Displaying Nodes with the Cluster Manager GUI

The Cluster Manager GUI provides a convenient graphic display of the defined nodes of a cluster and the attributes of those nodes through the FailSafe Cluster View. You can launch the FailSafe Cluster View directly, or you can bring it up at any time by clicking on "FailSafe Cluster View" at the bottom of the "FailSafe Manager" display.

From the View menu of the FailSafe Cluster View, you can select "Nodes in Pool" to view all nodes defined in the FailSafe pool. You can also select "Nodes In Cluster" to view all nodes that belong to the default cluster. Click any node's name or icon to view detailed status and configuration information about the node.

Displaying Nodes with the Cluster Manager CLI

After you have defined a node, you can display the node's parameters with the following command:

```
cmgr> show node A
```

A `show node` command on node cm1a would yield the following display:

```
cmgr> show node cm1
Logical Machine Name: cm1
Hostname: cm1
Node Is FailSafe: true
Node is CXFS: false
Nodeid: 1
Reset type: powerCycle
System Controller: msc
System Controller status: enabled
System Controller owner: cm2
System Controller owner device: /dev/ttyd2
System Controller owner type: tty
ControlNet Ipaddr: cm1
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 0
```

You can see a list of all of the nodes that have been defined with the following command:

```
cmgr> show nodes in pool
```

You can see a list of all of the nodes that have defined for a specified cluster with the following command:

```
cmgr> show nodes [in cluster A]
```

If you have specified a default cluster, you do not need to specify a cluster when you use this command and it will display the nodes defined in the default cluster.

IRIS FailSafe HA Parameters

There are several parameters that determine the behavior of the nodes in a cluster of an IRIS FailSafe system.

The IRIS FailSafe parameters are as follows:

- The tie-breaker node, which is the logical name of a machine used to compute the FailSafe membership in situations where 50% of the nodes in a cluster can talk to each other. If you do not specify a tie-breaker node, the node with the lowest node ID number is used.

  The tie-breaker node is a cluster-wide parameter.

  It is recommended that you configure a tie-breaker node even if there is an odd number of nodes in the cluster, since one node may be deactivated, leaving an even number of nodes to determine membership.

  In a heterogeneous cluster, where the nodes are of different sizes and capabilities, the largest node in the cluster with the most important application or the maximum number of resource groups should be configured as the tie-breaker node.

- Node timeout, which is the timeout period, in milliseconds. If no heartbeat is received from a node in this period of time, the node is considered to be dead and is not considered part of the FailSafe membership. It has a default value of 15000 milliseconds.

  The node timeout must be at least 5 seconds. In addition, the node timeout must be at least 10 times the heartbeat interval for proper FailSafe operation; otherwise, false failovers may be triggered.

  Node timeout is a cluster-wide parameter.

- The heartabeat interval, which is the interval, in milliseconds, between heartbeat messages. This interval must be greater than 500 milliseconds and it must not be greater than one-tenth the value of the node timeout period. This interval is set to

one second, by default. Heartbeat interval is a cluster-wide parameter. It has a default value of 1000 milliseconds.

The higher the number of heartbeats (smaller heartbeat interval), the greater the potential for slowing down the network. Conversely, the fewer the number of heartbeats (larger heartbeat interval), the greater the potential for reducing availability of resources.

- The node wait time, in milliseconds, which is the time a node waits for other nodes to join the cluster before declaring a new FailSafe membership. If the value is not set for the cluster, FailSafe assumes the value to be the node timeout times the number of nodes.
- The powerfail mode, which indicates whether a special power failure algorithm should be run when no response is received from a system controller after a reset request. This can be set to ON or OFF. Powerfail is a node-specific parameter, and should be defined for the machine that performs the reset operation.

Resetting IRIS FailSafe Parameters with the Cluster Manager GUI

To set IRIS FailSafe parameters with the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.
3. On the right side of the display click on the "Set FailSafe HA Parameters" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Resetting IRIS FailSafe Parameters with the Cluster Manager CLI

You can modify the FailSafe parameters with the following command:

```
cmgr> modify ha_parameters [on node A] [in cluster B]
```

If you have specified a default node or a default cluster, you do not have to specify a node or a cluster in this command. FailSafe will use the default.

```
Enter commands, You may enter "done" or "cancel" at any time to exit
```

A?

When this prompt of the node or cluster name appears, you enter the FailSafe parameters you wish to modify in the following format:

```
set node_timeout to A
set heartbeat to B
set run_pwrfail to C
set node_wait to D
set tie_breaker to E
```

Setting `tie_breaker` to " " unsets the `tie_breaker` value. Unsetting the `tie_breaker` is equivalent to not setting the value in the first place. In this case, FailSafe will use the node with the lowest node ID as the tie breaker node.

Defining a Cluster

A *cluster* is a collection of one or more *nodes* coupled with each other by networks and other similar interconnects. In IRIS FailSafe, a cluster is identified by a simple name. A given node may be a member of only one cluster.

Defining a cluster takes a long time to complete. When you define a cluster, a default logging configuration is created and default HA parameters are created in the CDB. Resource types in the cluster are created for the FailSafe plugins installed in the node using the `/usr/cluster/bin/cdb-create-resource-type` script. Resource types that were not created when the cluster was configured can be added later using the `resource type install` command.

To define a cluster, you must provide the following information:

- The logical name of the cluster, with a maximum length of 255 characters.
- Whether the node is a FailSafe cluster (Cluster Manager CLI only; a cluster defined with the FailSafe Cluster Manager GUI is a FailSafe cluster). For information on FailSafe and CXFS clusters, see "Coexecution with CXFS", page 49.
- Whether the cluster is a CXFS cluster (Cluster Manager CLI only; a cluster defined with the FailSafe Cluster Manager GUI is a FailSafe cluster). For information on FailSafe and CXFS clusters, see "Coexecution with CXFS", page 49.
- The mode of operation: *normal* (the default) or *experimental*. Experimental mode allows you to configure a FailSafe cluster in which resource groups do not fail

over when a node failure is detected. This mode can be useful when you are tuning node timeouts or heartbeat values. When a cluster is configured in normal mode, FailSafe fails over resource groups when it detects failure in a node or resource group.

- (Optional) The e-mail address to use to notify the system administrator when problems occur in the cluster (for example, `root@system`)
- (Optional) The e-mail program to use to notify the system administrator when problems occur in the cluster (for example, `/usr/sbin/Mail`).

  Specifying the e-mail program is optional and you can specify only the notification address in order to receive notifications by mail. If an address is not specified, notification will not be sent.

Adding Nodes to a Cluster

After you have added nodes to the pool and defined a cluster, you must provide the names of the nodes to include in the cluster.

Defining a Cluster with the Cluster Manager GUI

To define a cluster with the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on “Guided Configuration”.
3. On the right side of the display click on “Set Up a New Cluster” to launch the task link.
4. In the resulting window, click each task link in turn, as it becomes available. Enter the selected inputs for each task.
5. When finished, click “OK” to close the taskset window.

Defining a Cluster with the Cluster Manager CLI

When you define a cluster with the CLI, you define and cluster and add nodes to the cluster with the same command.

Use the following cluster manager CLI command to define a cluster:

```
cmgr> define cluster A
```

Entering this command specifies the name of the node you are defining and puts you in a mode that allows you to add nodes to the cluster. The following prompt appears:

```
cluster A?
```

When this prompt appears during cluster creation, you can specify nodes to include in the cluster and you can specify an e-mail address to direct messages that originate in this cluster.

You specify nodes to include in the cluster with the following command:

```
cluster A? add node C
cluster A?
```

You can add as many nodes as you want to include in the cluster.

You specify an e-mail program to use to direct messages with the following command:

```
cluster A? set notify_cmd to B
cluster A?
```

You specify an e-mail address to direct messages with the following command:

```
cluster A? set notify_addr to B
cluster A?
```

You specify a mode for the FailSafe cluster (normal or experimental) with the following command:

```
cluster A? set ha_mode to D
cluster A?
```

You specify a whether the cluster is a FailSafe cluster with the following command:

```
cluster A? set is_failsafe to E
cluster A?
```

You specify a whether the cluster is a CXFS cluster with the following command:

```
cluster A? set is_cxfs to E
cluster A?
```

When you are finished defining the cluster, enter `done` to return to the `cmgr` prompt.

Converting a CXFS Cluster to FailSafe

You can reconfigure an existing CXFS cluster so that it can be used with FailSafe as well.

When using the GUI, if you want an existing CXFS cluster to apply only to FailSafe, you must delete the CXFS cluster and redefine it from scratch as a FailSafe cluster. This is not necessary when using the CLI, which allows you to reconfigure a cluster so that it is a FailSafe cluster, a CXFS cluster, or a CXFS and FailSafe cluster.

Converting a CXFS Cluster to FailSafe with the Cluster Manager GUI

To use the Cluster Manager GUI to reconfigure a CXFS cluster so that it can be used with both CXFS and FailSafe, perform the following procedure. If you have not installed CXFS on the server to which you connected with the GUI, this task does not appear on the GUI menu.

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.
3. On the right side of the display click on the "Convert a CXFS Cluster to FailSafe" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

The cluster will apply to both CXFS and FailSafe.

Converting a CXFS Cluster to Failsafe with the Cluster Manager CLI

To convert a cluster with `cmgr`, use the `modify node` command then the following commands:

```
set is_failsafe to true|false
set is_cxfs to true|false
```

For example, to convert CXFS cluster `TEST` so that it also applies to FailSafe, enter the following:

```
cmgr> modify cluster TEST
Enter commands, when finished enter either "done" or "cancel"

TEST ?set is_failsafe to true
```

The cluster must support all of the functionalities (FailSafe and/or CXFS) that are turned on for its nodes; that is, if your cluster is of type `CXFS` , then you cannot modify a node that is part of the cluster so that the node is of type `FailSafe` or `CXFS and FailSafe`. However, the nodes do not have to support all the functionalities of the cluster; that is, you can have a node of type `CXFS` in a cluster of type `CXFS and FailSafe`.

Modifying Clusters

After you have defined a cluster, you can modify the attributes of the cluster. The process of modifying a cluster is similar to the process of creating a cluster.

Modifying a Cluster with the Cluster Manager GUI

To modify a cluster with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.
3. On the right side of the display click on the "Modify a Cluster Definition" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Modifying a Cluster with the Cluster Manager CLI

To modify an existing cluster, enter the following command:

```
cmgr> modify cluster A
```

Entering this command specifies the name of the cluster you are modifying and puts you in a mode that allows you to modify the cluster. The following prompt appears:

```
cluster A?
```

When this prompt appears, you can modify the cluster definition with the following commands:

```
cluster A? set notify_addr to B

cluster A? set notify_cmd to C
cluster A? add node D
cluster A? remove node D
cluster A?
```

When you are finished modifying the cluster, enter `done` to return to the `cmgr` prompt.

Deleting Clusters

After you have defined a cluster, you can delete the cluster. You cannot delete a cluster that contains nodes; you must move those nodes out of the cluster first.

Deleting a Cluster with the Cluster Manager GUI

To delete a cluster with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.
3. On the right side of the display click on the "Delete a Cluster" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Deleting a Cluster with the Cluster Manager CLI

You can delete a defined cluster with the following command:

```
cmgr> delete cluster A
```

Displaying Clusters

You can display defined clusters with the Cluster Manager GUI or the Cluster Manager CLI.

Displaying a Cluster with the Cluster Manager GUI

The Cluster Manager GUI provides a convenient display of a cluster and its components through the FailSafe Cluster View. You can launch the FailSafe Cluster View directly, or you can bring it up at any time by clicking on the "FailSafe Cluster View" prompt at the bottom of the "FailSafe Manager" display.

From the View menu of the FailSafe Cluster View, you can choose elements within the cluster to examine. To view details of the cluster, click on the cluster name or icon. Status and configuration information will appear in a new window. To view this information within the FailSafe Cluster View window, select Options. When you then click on the Show Details option, the status details will appear in the right side of the window.

Displaying a Cluster with the Cluster Manager CLI

After you have defined a cluster, you can display the nodes in that cluster with the following command:

```
cmgr> show cluster A
```

You can see a list of the clusters that have been defined with the following command:

```
cmgr> show clusters
```

Resource Configuration

A *resource* is a single physical or logical entity that provides a service to clients or other resources. A resource is generally available for use on two or more *node*s in a *cluster*, although only one node controls the resource at any given time. For example, a resource can be a single disk volume, a particular network address, or an application such as a web node.

Defining Resources

Resources are identified by a *resource name* and a *resource type*. A resource name identifies a specific instance of a resource type. A resource type is a particular class of resource. All of the resources in a given resource type can be handled in the same way for the purposes of *failover*. Every resource is an instance of exactly one resource type.

A resource type is identified with a simple name. A resource type can be defined for a specific logical node, or it can be defined for an entire cluster. A resource type that is defined for a node will override a cluster-wide resource type definition of the same name; this allows an individual node to override global settings from a cluster-wide resource type definition.

The IRIS FailSafe software includes many predefined resource types. If these types fit the application you want to make into a highly available service, you can reuse them. If none fit, you can define additional resource types.

To define a resource, you provide the following information:

- The name of the resource to define, with a maximum length of 255 characters.
- The type of resource to define. The IRIS FailSafe system includes some pre-defined resource types, including `NFS`, `Netscape_web`, `statd`, `MAC_Address`, `IP_Address`, `Oracle_DB`, `INFORMIX_DB`, `volume` and `filesystem`. You can define your own resource type as well.
- The name of the cluster that contains the resource.
- The logical name of the node that contains the resource (optional). If you specify a node, a local version of the resource will be defined on that node.
- Resource type-specific attributes for the resource. Each resource type may require specific parameters to define for the resource, as described in the following subsections.

You can define up to 100 resources in a FailSafe configuration.

Volume Resource Attributes

The volume resource is the XLV volume used by the resources in the resource group.

Note: IRIS FailSafe assumes that CXFS filesystems are highly-available because they do not require a FailSafe failover in order to be made available on another node in the cluster. Therefore, FailSafe does not directly start, stop, or monitor CXFS filesystems or XVM volumes, and CXFS filesystems and XVM volumes should not be added to the FailSafe resource groups.

When you define a volume resource, the resource name should be the name of the XLV volume. Do not specify the XLV device file name as the resource name. For example, the resource name for a volume might be `xlv_vol` but not `/dev/xlv/xlv_vol` or `/dev/dsk/xlv/xlv_vol`.

When an XLV volume is assembled on a node, a file is created in `/dev/xlv`. Even when you configure a volume resource in a FailSafe cluster, you can view that volume from only one node at a time, unless a failover has occurred.

You may be able to view a volume name in `/dev/xlv` on two different nodes after failover because when an XLV volume is shut down, the filename is not removed from that directory. Hence, more than one node may have the volume filename in its directory. However, only one node at a time will have the volume assembled. Use `xlv_mgr`(1M) to see which machine has the volume assembled.

When you define a volume, you can optionally specify the following parameters:

- The user name (login name) of the owner of the XLV device file. `root` is the default owner for XLV device files.
- The group name of the XLV device file. The `sys` group is the default group name for XLV device files.
- The device file permissions, specified in octal notation. 600 mode is the default value for XLV device file permissions.

Filesystem Resource Attributes

The `filesystem` resource must be an XFS filesystem.

Note: IRIS FailSafe assumes that CXFS filesystems are highly-available because they do not require a FailSafe failover in order to be made available on another node in the cluster. Therefore, FailSafe does not directly start, stop, or monitor CXFS filesystems or XVM volumes, and CXFS filesystems and XVM volumes should not be added to the FailSafe resource groups.

Any XFS filesystem that must be highly available should be configured as a filesystem resource. All XFS filesystems that you use as a filesystem resource must be created on XLV volumes on shared disks.

When you define a filesystem resource, the name of the resource should be the mount point of the filesystem. For example, an XFS filesystem created on an XLV volume `xlv_vol` and is mounted on the `/shared1` directory will have the resource name `/shared1`.

When you define a filesystem, you must specify all of the following parameters:

- The name of the xlv volume associated with the filesystem. For example, for the filesystem created on the XLV volume `xlv_vol` the volume name attribute will be `xlv_vol` as well.
- The mount options to be used for mounting the filesystem, which are the mount options that have to be passed to the `-o` option of the `mount`(1M) command. The list of available options is provided in `fstab`(4).
- The monitoring level to be used for the filesystem. A monitoring level of 1 specifies to check whether the filesystem exists in `/etc/mtab`, as described in the `mtab`(4) man page. A monitoring level of 2 specifies to check whether the filesystem is mounted using the `stat`(1M) command. Monitoring level 2 is a more intrusive check that is more reliable if it completes on time. Some loaded systems have been known to have problems with this level check.

IP_address Resource Attributes

The `IP_address` resources are the IP addresses used by clients to access the highly available services within the resource group. These IP addresses are moved from one node to another along with the other resources in the resource group when a failure is detected.

You specify the resource name of an `IP_address` resource in "." notation. IP names that require name resolution should not be used. For example, 192.26.50.1 is a valid resource name of the `IP_address` resource type.

The IP address you define as a FailSafe resource must not be the same as the IP address of a node hostname or the IP address of a node's control network.

When you define an `IP_address` resource, you can optionally specifying the following parameters. If you specify any of these parameters, you must specify all of them.

- The broadcast address for the IP address
- The network mask of the IP address
- A comma-separated list of interfaces on which the IP address can be configured. This ordered list is a superset of all the interfaces on all nodes where this IP address might be allocated. You can specify multiple interfaces to configure local restart of the IP address, if those interfaces are on the same node.

  The order of the list of interfaces determines the priority order for determining which IP address will be used for local restarts of the node.

MAC Address Resource Attributes

The MAC address is the Link level (MAC) address of the network interface. If MAC addresses are to be failed over, dedicated network interfaces are required.

The resource name of a MAC address is the MAC address of the interface. You can obtain MAC addresses by using the `ha_macconfig2`(1M) command.

When you define a MAC address for an interface, you must specify the interface that has to be re-MACed.

Currently, only ethernet interfaces are capable of undergoing the reMAC process.

NFS Resource Attributes

An NFS resource is any NFS filesystem that you configure as highly available. This resource definition has a dependency on the filesystem and statd resource type.

The resource name of the NFS resource is the NFS export mount point. Since the name must be a valid filesystem name, if must start with a "/", as, for example, `/disk1`.

When you define an NFS resource, you must specify the following parameters:

- The filesystem that is used as input to the `mount`(1M) command, which must be an existing filesystem resource
- The export options for the file system used in the `exportfs`(1M) command
- The filesystem resource dependency of the NFS resource, which must be the name of a pre-defined `filesystem` resource

statd_unlimited Resource Attributes

The `statd_unlimited` resource is only applicable when defined in a resource group that contains NFS resources. The `statd_unlimited` resource is used to provide highly available file locking and recovery, `lockf`(3C), `fcntl`(2), and `flock`(3B). The `statd_unlimited` resource type allows you to provide NFS lock failover for an unlimited number of resource groups in a cluster.

The resource name of a `statd_unlimited` resource defines the `statmon` (NFS lock) directory for IRIS FailSafe. This is a directory on a pre-existing highly available filesystem, which is part of a resource group. Only one `statd_unlimited` resource needs to be added to a resource group to provide NFS failover support for all the filesystems defined in the same resource group. The directory is usually of the form *filesystem*`/statmon`, as, for example, `/disk1/statmon`.

Using the `statd_unlimited` resource type allows you to have as many `statmon` directories as you require. For this reason, when you add filesystems to a FailSafe configuration that are to be exported and require locking, or if you want to have many exports spread across resource groups, you should configure the filesystems using the `statd_unlimited` resource type rather than the `statd` resource type.

The `statd_unlimited` resource has a dependency on the `NFS` resource type.

When you configure a `statd_unlimited` resource, you specify the following parameters:

- The NFS export point associated with the `statmon` directory, which can be a directory or a filesystem under which you have put the `statmon` directory
- The `NFS` resource dependency of the statd resource

statd Resource Attributes

The `statd` resource is provided for compatibility with older release of IRIS FailSafe 2.*X*.

The `statd` resource is only applicable when defined in a resource group that contains NFS resources. The `statd` resource is used to provide highly available file locking and recovery, `lockf`(3C), `fcntl`(2), and `flock`(3B).

The resource name of a `statd` resource defines the `statmon` (NFS lock) directory for IRIS FailSafe. This is a directory on a pre-existing highly available filesystem, which is part of a resource group. Only one `statd` resource needs to be added to a resource group to provide NFS failover support for all the filesystems defined in the same resource group. The directory is usually of the form *filesystem*`/statmon`, as, for example, `/disk1/statmon`.

The `statd` resource has a dependency on the `IP_address` and `filesystem` resource type.

When you configure a `statd` resource, you specify the following parameters:

- The highly available interface address for NFS clients
- The resource dependencies of the `statd` resource
  - The `IP_address` dependency
  - The `filesystem` dependency

Netscape_web Resource Attributes

You configure any Netscape Web server that must be highly available as a `Netscape_web` resource. The server can be a Netscape FastTrack or Enterprise server. This resource definition has a dependency on an `IP_address` resource type. We recommend that you add your own filesystem dependency; this is not a required dependency since the contents for a web server could be replicated across multiple nodes.

You specify the resource name of a Netscape_web resource as any string that uniquely defines this resource within the context of the cluster.

When you define a Netscape_web resource, you must specify the following parameters:

- The port number of the port on which the web server will listen
- The location of the web server's start and stop commands

- The monitor level, which defines the type of monitoring action performed by the monitor script: a monitor level of 1 monitors the webserver process; a monitor level of 2 monitors the webserver by requesting a server response
- The home page directory, which defines the location of the web server's home page directory.
- The Web IP address, which is the IP address of the server host. This must be an existing `IP_address` resource.
- The resource dependency of the Netscape_web resource, which is an `IP_address` resource type.

Adding Dependency to a Resource

One resource can be dependent on one or more other resources; if so, it will not be able to start (that is, be made available for use) unless the dependent resources are started as well. Dependent resources must be part of the same *resource group*.

Like resources, a resource type can be dependent on one or more other resource types. If such a dependency exists, at least one instance of each of the dependent resource types must be defined. For example, a resource type named `Netscape_web` might have resource type dependencies on a resource types named `IP_address` and `volume`. If a resource named `ws1` is defined with the `Netscape_web` resource type, then the resource group containing `ws1` must also contain at least one resource of the type `IP_address` and one resource of the type `volume`.

As you define resources, you can define which resources are dependent on which other resources. For example, a web server may depend on a both an IP address and a file system. In turn, a file system may depend on a volume.

You cannot make resources mutually dependent. For example, if resource A is dependent on resource B, then you cannot make resource B dependent on resource A. In addition, you cannot define cyclic dependencies. For example, if resource A is dependent on resource B, and resource B is dependent on resource C, then resource C cannot be dependent on resource A.

When you add a dependency to a resource definition, you provide the following information:

- The name of the existing resource to which you are adding a dependency
- The resource type of the existing resource to which you are adding a dependency

- The name of the cluster that contains the resource
- Optionally, the logical node name of the node in the cluster that contains the resource. If specified, resource dependencies are added to the node's definition of the resource. If this is not specified, resource dependencies are added to the cluster-wide resource definition.
- The resource name of the resource dependency
- The resource type of the resource dependency

Defining a Resource with the Cluster Manager GUI

To define a resource with the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Resources & Resource Types" category.
3. On the right side of the display click on the "Define a New Resource" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.
6. On the right side of the display, click on the "Add/Remove Dependencies for a Resource Definition" to launch the task.
7. Enter the selected inputs.
8. Click on "OK" at the bottom of the screen to complete the task.

When you use this command to define a resource, you define a cluster-wide resource that is not specific to a node. For information on defining a node-specific resource, see "Defining a Node-Specific Resource", page 127.

Defining a Resource with the Cluster Manager CLI

Use the following CLI command to define a clusterwide resource:

```
cmgr> define resource A [of resource_type B] [in cluster C]
```

Entering this command specifies the name and resource type of the resource you are defining within a specified cluster. If you have specified a default cluster or a default

resource type, you do not need to specify a resource type or a cluster in this command and the CLI will use the default.

When you use this command to define a resource, you define a clusterwide resource that is not specific to a node. For information on defining a node-specific resource, see "Defining a Node-Specific Resource", page 127.

The following prompt appears:

```
resource A?
```

When this prompt appears during resource creation, you can enter the following commands to specify the attributes of the resource you are defining and to add and remove dependencies from the resource:

```
resource A? set key to value
resource A? add dependency E of type F
resource A? remove dependency E of type F
```

The attributes you define with the `set` *key* `to` *value* command will depend on the type of resource you are defining, as described in "Defining Resources", page 116.

For detailed information on how to determine the format for defining resource attributes, see "Specifying Resource Attributes with Cluster Manager CLI", page 124.

When you are finished defining the resource and its dependencies, enter `done` to return to the `cmgr` prompt.

The following section of a `cmgr` script defines a resource of resource type `statd_unlimited`:

```
define resource /hafs1/nfs/statmon of resource_type statd_unlimited in cluster nfs-cluster
        set ExportPoint to /hafs1/subdir
done
```

Specifying Resource Attributes with Cluster Manager CLI

To see the format in which you can specify the user-specific attributes that you need to set for a particular resource type, you can enter the following command to see the full definition of that resource type:

```
cmgr> show resource_type A in cluster B
```

For example, to see the *key* attributes you define for a resource of resource type `volumes`, enter the following command:

```
cmgr> show resource_type volume in cluster chaos
```

At the bottom of the resulting display, the following appears:

```
...
Type specific attribute: devname-group
        Data type: string
        Default value: sys
Type specific attribute: devname-owner
        Data type: string
        Default value: root
Type specific attribute: devname-mode
        Data type: string
        Default value: 600
...
```

This display reflects the format in which you can specify the group id, the device owner, and the device file permissions for the volume. The `devname-group` key specifies the group id of the xlv device file, the `devname_owner` key specifies the owner of the xlv device file, and the `devname_mode` key specifies the device file permissions.

For example, to set the group id to `sys`, enter the following command:

```
resource A? set devname-group to sys
```

This remainder of this section summarizes the attributes you specify for the predefined IRIS FailSafe resource types with the *set key to value* command of the Cluster Manger CLI.

When you define a `volume` resource, you specify the following attributes as keys:

`devname-group`	Group id of the xlv device file
`devname_owner`	Owner of the xlv device file
`devname_mode`	Device file permissions

When you define a `filesystem` resource, you specify the following attributes as keys:

`volume-name`	Name of the xlv volume associated with the filesystem
`mount-options`	Mount options to be used for mounting the filesystem

When you define an `IP_address` resource, you specify the following attributes:

`NetworkMask`	The subnet mask of the IP address
`interfaces`	A comma-separated list of interfaces on which the IP address can be configured
`BroadcastAddress`	The broadcast address for the IP address

When you define a MAC address resource, you specify the following attribute as a key:

`interface-name`	Name of the interface that has to be re-MACed

When you define an NFS resource, you specify the following attributes as keys:

`export-info`	The export options for the filesystem used in the `exportfs`(1M) command
`filesystem`	The filesystem that is used as input to the `mount`(1M) command

When you define a `statd_unlimited` resource, you specify the following attribute as a key:

`ExportPoint`	NFS export point associated with the `statmon` directory.

When you define a `statd` resource, you specify the following attribute as a key:

`InterfaceAddress`	Name of the interface that NFS clients will use

When you define a `Netscape_web` resource, you specify the following attributes as keys:

`monitor-level`	The monitor level, which defines the type of monitoring action performed by the monitor script
`port-number`	The port number of the port on which the web server will listen
`admin-scripts`	The location of the web server's start and stop commands
`default-page-location`	The location of the web server's default web page

`web-ipaddr`	The IP address of the highly available interface for the web server

Defining a Node-Specific Resource

You can redefine an existing resource with a resource definition that applies only to a particular node. Only existing clusterwide resources can be redefined; resources already defined for a specific node cannot be redefined.

You use this feature when you configure heterogeneous clusters for an `IP_address` resource. For example, `IP_address` 192.26.50.2 can be configured on a gigabit ethernet interface eg0 on an Origin node and on a 100baseT interface ef0 on another Origin node. The clusterwide resource definition for 192.26.50.2 will have the `interfaces` field set to ef0 and the node-specific definition for the first node will have eg0 as the `interfaces` field.

Defining a Node-Specific Resource with the Cluster Manager GUI

Using the Cluster Manager GUI, you can take an existing clusterwide resource definition and redefine it for use on a specific node in the cluster:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Resources & Resource Types" category.
3. On the right side of the display click on the "Redefine a Resource For a Specific Node" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.

Defining a Node-Specific Resource with the Cluster Manager CLI

You can use the Cluster Manager CLI to redefine a clusterwide resource to be specific to a node just as you define a clusterwide resource, except that you specify a node on the `define resource` command.

Use the following CLI command to define a node-specific resource:

```
cmgr> define resource A of resource_type B on node C [in cluster D]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

Modifying Resources

After you have defined resources, you can modify and delete them.

You can modify only the type-specific attributes for a resource. You cannot rename a resource once it has been defined.

Note: There are some resource attributes whose modification does not take effect until the resource group containing that resource is brought online again. For example, if you modify the export options of a resource of type NFS, the modifications do not take effect immediately; they take effect when the resource is brought online.

Modifying Resources with the Cluster Manager GUI

To modify a resource with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Resources & Resource Types" category.
3. On the right side of the display click on the "Modify a Resource Definition" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Modifying Resources with the Cluster Manager CLI

Use the following CLI command to modify a resource:

```
cmgr> modify resource A of resource_type B [in cluster C]
```

Entering this command specifies the name and resource type of the resource you are modifying within a specified cluster. If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

You modify a resource using the same commands you use to define a resource.

Deleting Resources

After you have defined resources, you can delete them.

Deleting Resources with the Cluster Manager GUI

To delete a resource with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Resources & Resource Types" category.
3. On the right side of the display click on the "Delete a Resource" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Deleting Resources with the Cluster Manager CLI

You can use the following command to delete a resource definition:

```
cmgr> delete resource A of resource_type B [in cluster D]
```

Displaying Resources

You can display resources in various ways. You can display the attributes of a particular defined resource, you can display all of the defined resources in a specified resource group, or you can display all the defined resources of a specified resource type.

Displaying Resources with the Cluster Manager GUI

The Cluster Manager GUI provides a convenient display of resources through the FailSafe Cluster View. You can launch the FailSafe Cluster View directly, or you can bring it up at any time by clicking on the "FailSafe Cluster View" prompt at the bottom of the "FailSafe Manager" display.

From the View menu of the FailSafe Cluster View, select Resources to see all defined resources. The status of these resources will be shown in the icon (green indicates online, grey indicates offline). Alternately, you can select "Resources of Type" from

the View menu to see resources organized by resource type, or you can select "Resources by Group" to see resources organized by resource group.

Displaying Resources with the Cluster Manager CLI

Use the following command to view the parameters of a defined resource:

```
cmgr> show resource A of resource_type B
```

Use the following command to view all of the defined resources in a resource group:

```
cmgr> show resources in resource_group A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

Use the following command to view all of the defined resources of a particular resource type in a specified cluster:

```
cmgr> show resources of resource_type A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

Defining a Resource Type

The IRIS FailSafe software includes many predefined resource types. Resource types in the cluster are created for the FailSafe plugins installed in the node using the `/usr/cluster/bin/cdb-create-resource-type` script. Resource types that were not created when the cluster was configured can be added later using the `resource type install` command, as described in "Installing (Loading) a Resource Type on a Cluster", page 142.

If these predefined resource types fit the application you want to make into a highly available service, you can reuse them. If none fits, you can define additional resource types.

Complete information on defining resource types is provided in the *IRIS FailSafe Version 2 Programmer's Guide*. This manual provides a summary of that information.

To define a new resource type, you must have the following information:

- Name of the resource type, with a maximum length of 255 characters.

- Name of the cluster to which the resource type will apply
- Node on which the resource type will apply, if the resource type is to be restricted to a specific node
- Order of performing the action scripts for resources of this type in relation to resources of other types:
  - Resources are started in the increasing order of this value
  - Resources are stopped in the decreasing order of this value

  See the *IRIS FailSafe Version 2 Programmer's Guide* for a full description of the order ranges available.
- Restart mode, which can be one of the following values:
  - 0 = Do not restart on monitoring failures
  - 1 = Restart a fixed number of times
- Number of local restarts (when restart mode is 1).

  The local restart flag enables local failover.
  - If local restart is enabled and the resource monitor script fails, the SRMD executes the restart script for the resource.
  - If the restart script is successful, SRMD continues to monitor the resource.
  - If the restart script fails or the restart count is exhausted, SRMD sends a resource group monitoring error to FSD. FSD itself is not involved in local failover.

  When a resource is restarted, all other resources in the resource group are not restarted. It is not possible to do a local restart of a resource using the Cluster Manager GUI or the Cluster Manager CLI.

  If you find that you need to reset the restart counter for a resource type, you can put the resource group in maintenance mode and remove it from maintenance mode. This process will restart counters for all resources in the resource group. For information on putting a resource group in maintenance mode, see "Stop Monitoring of a Resource Group (Maintenance Mode)", page 196.

- Location of the executable script. This is always `/var/cluster/ha/resource_types/`*rtname*, where *rtname* is the resource type name.
- Monitoring interval, which is the time period (in milliseconds) between successive executions of the `monitor` action script; this is only valid for the `monitor` action script.
- Starting time for monitoring. When the resource group is made in online in a node, IRIS FailSafe will start monitoring the resources after the specified time period (in milliseconds).
- Action scripts to be defined for this resource type, You must specify scripts for `start`, `stop`, `exclusive`, and `monitor`, although the `monitor` script may contain only a return-success function if you wish. If you specify 1 for the restart mode, you must specify a `restart` script.
- Type-specific attributes to be defined for this resource type. The action scripts use this information to start, stop, and monitor a resource of this resource type. For example, NFS requires the following resource keys:
  - `export-point`, which takes a value that defines the export disk name. This name is used as input to the `exportfs`(1M) command. For example:

    ```
    export-point = /this_disk
    ```

  - `export-info`, which takes a value that defines the export options for the filesystem. These options are used in the `exportfs`(1M) command. For example:

    ```
    export-info = rw,wsync,anon=root
    ```

  - `filesystem`, which takes a value that defines the raw filesystem. This name is used as input to the `mount`(1M)) command. For example:

    ```
    filesystem = /dev/xlv/xlv_object
    ```

To define a new resource type, you use the Cluster Manager GUI or the Cluster Manager CLI.

Defining a Resource Type with the Cluster Manager GUI

To define a resource type with the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.

2. On the left side of the display, click on the "Resources & Resource Types" category.
3. On the right side of the display click on the "Define a Resource Type" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.

Defining a Resource Type with the Cluster Manager CLI

The following steps show the use of `cmgr` (which is the same command as `cluster_mgr`(1m)) interactively to define a resource type called `newresourcetype`.

1. Log in as `root`.
2. Execute the `cmgr` command using the `-p` option to prompt you for information (the command name can be abbreviated to `cmgr`):

```
# /usr/cluster/bin/cmgr -p
Welcome to SGI Cluster Manager Command-Line Interface

cmgr>
```

3. Use the `set` subcommand to specify the default cluster used for `cmgr` operations. In this example, we use a cluster named `TEST`:

```
cmgr> set cluster TEST
```

Note: If you prefer, you can specify the cluster name as needed with each subcommand.

---

4. Use the `define resource_type` subcommand. By default, the resource type will apply across the cluster; if you wish to limit the resource type to a specific node, enter the node name when prompted. If you wish to enable restart mode, enter 1 when prompted.

Note: The following example only shows the prompts and answers for two action scripts (`start` and `stop`) for a new resource type named `newresourcetype`.

```
cmgr> define resource_type newresourcetype

(Enter "cancel" at any time to abort)

Node[optional]?
Order ? 300
Restart Mode ? (0)

DEFINE RESOURCE TYPE OPTIONS

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:1

No current resource type actions

Action name ? start
Executable timeout (in milliseconds) ? 40000

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:1
```

```
Current resource type actions:
        start

Action name stop
Executable timeout?  (in milliseconds) 40000

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:3

No current type specific attributes

Type Specific Attribute ? integer-att
Datatype ? integer
Default value[optional] ? 33

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:3

Current type specific attributes:
        Type Specific Attribute - 1: integer-att
```

```
Type Specific Attribute ? string-att
Datatype ? string
Default value[optional] ? rw

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:5

No current resource type dependencies

Dependency name ? filesystem

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:7

Current resource type actions:
        Action - 1: start
        Action - 2: stop

Current type specific attributes:
        Type Specific Attribute - 1: integer-att
        Type Specific Attribute - 2: string-att
```

```
No current resource type dependencies

Resource dependencies to be added:
        Resource dependency - 1: filesystem

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:9
Successfully defined resource_type newresourcetype

cmgr> show resource_types

template
MAC_address
newresourcetype
IP_address
filesystem
volume

cmgr> exit
#
```

Defining a Node-Specific Resource Type

You can redefine an existing resource type with a resource definition that applies only to a particular node. Only existing clusterwide resource types can be redefined; resource types already defined for a specific node cannot be redefined.

A resource type that is defined for a node overrides a cluster-wide resource type definition with the same name; this allows an individual node to override global settings from a clusterwide resource type definition. You can use this feature if you

want to have different script timeouts for a node or you want to restart a resource on only one node in the cluster.

For example, the `IP_address` resource has local restart enabled by default. If you would like to have an `IP_address` type without local restart for a particular node, you can make a copy of the `IP_address` clusterwide resource type with all of the parameters the same except for restart mode, which you set to 0.

Defining a Node-Specific Resource Type with the Cluster Manager GUI

Using the Cluster Manager GUI, you can take an existing clusterwide resource type definition and redefine it for use on a specific node in the cluster. Perform the following tasks:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Resources & Resource Types" category.
3. On the right side of the display click on the "Redefine a Resource Type For a Specific Node" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.

Defining a Node-Specific Resource Type with the Cluster Manager CLI

With the Cluster Manager CLI, you redefine a node-specific resource type just as you define a cluster-wide resource type, except that you specify a node on the `define resource_type` command.

Use the following CLI command to define a node-specific resource type:

```
cmgr> define resource_type A on node B [in cluster C]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

Adding Dependencies to a Resource Type

Like resources, a resource type can be dependent on one or more other resource types. If such a dependency exists, at least one instance of each of the dependent resource types must be defined. For example, a resource type named `Netscape_web`

might have resource type dependencies on a resource type named `IP_address` and `volume`. If a resource named `ws1` is defined with the `Netscape_web` resource type, then the resource group containing `ws1` must also contain at least one resource of the type `IP_address` and one resource of the type `volume`.

When using the Cluster Manager GUI, you add or remove dependencies for a resource type by selecting the "Add/Remove Dependencies for a Resource Type" from the "Resources & Resource Types" display and providing the indicated input. When using the Cluster Manager CLI, you add or remove dependencies when you define or modify the resource type.

Modifying Resource Types

After you have defined resource types, you can modify them. The process of modifying a resource type is similar to the process of defining a resource type.

Modifying Resource Types with the Cluster Manager GUI

To modify a resource type with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Resources & Resource Types" category.
3. On the right side of the display click on the "Modify a Resource Type Definition" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Modifying Resource Types with the Cluster Manager CLI

Use the following CLI command to modify a resource:

```
cmgr> modify resource_type A [in cluster B]
```

Entering this command specifies the resource type you are modifying within a specified cluster. If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

You modify a resource type using the same commands you use to define a resource type.

The CLI can display the current values of the resource type timeouts, allowing you to modify any of the action timeouts.

The following example shows how to use the CLI to increase the `statd` resource type monitor executable timeout from 40 seconds to 60 seconds.

```
# cmgr
Welcome to SGI Cluster Manager Command-Line Interface

cmgr> modify resource_type statd in cluster test-cluster
Enter commands, when finished enter either "done" or "cancel"

resource_type statd ? modify action monitor
Enter action parameters, when finished enter "done" or "cancel"

Current action monitor parameters:
        exec_time : 40000ms
        monitor_interval : 20000ms
        monitor_time : 50000ms

Action - monitor ? set exec_time to 60000
Action - monitor ? done
resource_type statd ? done
Successfully modified resource_type statd
```

The following examples show how to modify the resource type timeouts with the CLI in prompt mode.

```
# cmgr -p
Welcome to SGI Cluster Manager Command-Line Interface

cmgr> modify resource_type statd in cluster test-cluster


(Enter "cancel" at any time to abort)

Node[optional] ?
Order ? (411)
Restart Mode ? (0)
```

```
MODIFY RESOURCE TYPE OPTIONS

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:0
Current resource type actions:
        stop
        exclusive
        start
        restart
        monitor

Action name ? monitor
Executable timeout (in milliseconds) ? (40000ms) 60000
Monitoring Interval (in milliseconds) ? (20000ms)
Start Monitoring Time (in milliseconds) ? (50000ms)

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:9
Successfully modified resource_type statd
```

Deleting Resource Types

After you have defined resource types, you can delete them.

Deleting Resource Types with the Cluster Manager GUI

To delete a resource type with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Resources & Resource Types" category.
3. On the right side of the display click on the "Delete a Resource Type" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Deleting Resource Types with the Cluster Manager CLI

You can use the following command to delete a resource type:

```
cmgr> delete resource_type A [in cluster B]
```

Installing (Loading) a Resource Type on a Cluster

When you define a cluster, FailSafe installs a set of resource type definitions that you can use that include default values. If you need to install additional standard Silicon Graphics-supplied resource type definitions on the cluster, or if you delete a standard resource type definition and wish to reinstall it, you can load that resource type definition on the cluster.

The resource type definition you are installing cannot exist on the cluster.

Installing a Resource Type with the Cluster Manager GUI

To install a resource type using the GUI, select the "Load a Resource" task from the "Resources & Resource Types" task page and enter the resource type to load.

Installing a Resource Type with the Cluster Manager CLI

Use the following CLI command to install a resource type on a cluster:

```
cmgr> install resource_type A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

Displaying Resource Types

After you have defined a resource types, you can display them.

Displaying Resource Types with the Cluster Manager GUI

The Cluster Manager GUI provides a convenient display of resource types through the FailSafe Cluster View. You can launch the FailSafe Cluster View directly, or you can bring it up at any time by clicking on the "FailSafe Cluster View" prompt at the bottom of the "FailSafe Manager" display.

From the View menu of the FailSafe Cluster View, select Types to see all defined resource types. You can then click on any of the resource type icons to view the parameters of the resource type.

Displaying Resource Types with the Cluster Manager CLI

Use the following command to view the parameters of a defined resource type in a specified cluster:

```
cmgr> show resource_type A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

Use the following command to view all of the defined resource types in a cluster:

```
cmgr> show resource_types [in cluster A]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

Use the following command to view all of the defined resource types that have been installed:

```
cmgr> show resource_types installed
```

Defining a Failover Policy

Before you can configure your resources into a resource group, you must determine which failover policy to apply to the resource group. To define a failover policy, you provide the following information:

- The name of the failover policy, with a maximum length of 63 characters, which must be unique within the pool.
- The name of an existing failover script.
- The initial failover domain, which is an ordered list of the nodes on which the resource group may execute. The administrator supplies the initial failover domain when configuring the failover policy; this is input to the failover script, which generates the runtime failover domain.
- The failover attributes, which modify the behavior of the failover script.

Complete information on failover policies and failover scripts, with an emphasis on writing your own failover policies and scripts, is provided in the *IRIS FailSafe Version 2 Programmer's Guide*.

Failover Domain

A *failover domain* is the `ordered` list of nodes on which a given *resource group* can be allocated. The nodes listed in the failover domain `must` be within the same cluster; however, the failover domain does not have to include every node in the cluster. The failover domain can be also used to statically load balance the resource groups in a cluster.

Examples:

- In a four–node cluster, a set of two nodes that have access to a particular XLV volume may be the failover domain of the resource group containing that XLV volume.
- In a cluster of nodes named venus, mercury, and pluto, you could configure the following initial failover domains for resource groups RG1 and RG2:
  - mercury, venus, pluto for RG1
  - pluto, mercury for RG2

The administrator defines the *initial failover domain* when configuring a failover policy. The initial failover domain is used when a cluster is first booted. The ordered list specified by the initial failover domain is transformed into a *run-time failover domain* by the *failover script*. With each failure, the failover script takes the current run-time failover domain and potentially modifies it (for the `ordered` failover script, the order will not change); the initial failover domain is never used again. Depending on the run-time conditions such as load and contents of the failover script, the initial and run-time failover domains may be identical.

For example, suppose that the cluster contains three nodes named `N1`, `N2`, and `N3`; that node failure is not the reason for failover; and that the initial failover domain is as follows:

```
N1 N2 N3
```

The runtime failover domain will vary based on the failover script:

- If `ordered`:

  ```
  N1 N2 N3
  ```

- If `round-robin`:

  ```
  N2 N3 N1
  ```

- If a customized failover script, the order could be any permutation, based on the contents of the script:

  ```
  N1 N2 N3
  N1 N3 N2
  N2 N3 N1
  N2 N1 N3
  N3 N2 N1
  N3 N1 N2
  ```

FailSafe stores the run-time failover domain and uses it as input to the next failover script invocation.

Failover Attributes

A *failover attribute* is a value that is passed to the failover script and used by IRIS FailSafe for the purpose of modifying the run-time failover domain used for a specific resource group.

You can specify the following classes of failover attributes:

- Required attributes: either `Auto_Failback` or `Controlled_Failback` (mutually exclusive)
- Optional attributes:
  - `Auto_Recovery` or `InPlace_Recovery` (mutually exclusive).
  - `Critical_RG`
  - `Node_Failures_Only`

Note: The starting conditions for the attributes differs by class:

- For required attributes, the starting condition is that a node joins the FailSafe membership when the cluster is already providing highly available services
- For optional attributes, the starting condition is that highly available services are started and the resource group is running in only one node in the cluster

---

Table 5-1 describes each attribute.

Table 5-1 Failover Attributes

Class	Name	Description
Required	`Auto_Failback`	Specifies that the resource group is made online based on the failover policy when the node joins the cluster. This attribute is best used when some type of load balancing is required. You must specify either this attribute or the `Controlled_Failback` attribute.
	`Controlled_Failback`	Specifies that the resource group remains on the same node when a node joins the cluster. This attribute is best used when client/server applications have expensive recovery mechanisms, such as databases or any application that uses `tcp` to communicate. You must specify either this attribute or the `Auto_Failback` attribute.

Class	Name	Description
Optional	`Auto_Recovery`	Specifies that the resource group is made online based on the failover policy even when an exclusivity check shows that the resource group is running on a node. This attribute is optional and is mutually exclusive with the `InPlace_Recovery` attribute. If you specify neither of these attributes, IRIS FailSafe will use this attribute by default if you have specified the `Auto_Failback` attribute.
	`InPlace_Recovery`	Specifies that the resource group is made online on the same node where the resource group is running. This attribute is optional and is mutually exclusive with the `Auto_Recovery` attribute. If you specify neither of these attributes, IRIS FailSafe will use this attribute by default if you have specified the `Controlled_Failback` attribute.
	`Critical_RG`	Allows monitor failure recovery to succeed even when there are resource group release failures. When resource monitoring fails, FailSafe attempts to move the resource group to another node in the application failover domain. If FailSafe fails to release the resources in the resource group, FailSafe puts the Resource group into `srmd executable error` status. If the `Critical_RG` failover attribute is specified in the failover policy of the resource group, FailSafe will reset the node where the release operation failed and move the resource group to another node based on failover policy.
	`Node_Failures_Only`	Allows failover only when there are node failures. This attribute does not have an impact on resource restarts in the local node. The failover does not occur when there is a resource monitoring failure in the resource group. This attribute is useful for customers who are using a hierarchical storage management system such as DMF; in this situation, a customer may want to have resource monitoring failures reported without automatic recovery, allowing operators to perform the recovery action manually if necessary.

See the *IRIS FailSafe Version 2 Programmer's Guide* for a full discussions of example failover policies.

Failover Scripts

A *failover script* generates the run-time failover domain and returns it to the FailSafe process. The FailSafe process applies the failover attributes and then selects the first node in the returned failover domain that is also in the current FailSafe membership.

The `ordered` failover script is provided with the IRIS FailSafe release. The `ordered` script never changes the initial domain; when using this script, the initial and runtime domains are equivalent.

The `round-robin` failover script is also provided with the IRIS FailSafe release. The `round-robin` script selects the resource group owner in a round-robin (circular) fashion. This policy can be used for resource groups that can be run in any node in the cluster.

Failover scripts are stored in the `/var/clusters/ha/policies` directory. If the `ordered` script does not meet your needs, you can define a new failover script and place it in the `/var/clusters/ha/policies` directory. When you are using the FailSafe GUI, the GUI automatically detects your script and presents it to you as a choice for you to use. You can configure the cluster database to use your new failover script for the required resource groups. For information on defining failover scripts, see the *IRIS FailSafe Version 2 Programmer's Guide*.

Defining a Failover Policy with the Cluster Manager GUI

To define a failover policy using the GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Failover Policies & Resource Groups" category.
3. On the right side of the display click on the "Define a Failover Policy" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.

Defining a Failover Policy with the Cluster Manager CLI

To define a failover policy, enter the following command at the `cmgr` prompt to specify the name of the failover policy:

```
cmgr> define failover_policy A
```

The following prompt appears:

```
failover_policy A?
```

When this prompt appears you can use the following commands to specify the components of a failover policy:

```
failover_policy A? set attribute to B
failover policy A? set script to C
failover policy A? set domain to D
failover_policy A?
```

When you define a failover policy, you can set as many attributes and domains as your setup requires, but executing the `add attribute` and `add domain` commands with different values. The CLI also allows you to specify multiple domains in one command of the following format:

```
failover_policy A? set domain to A B C ...
```

The components of a failover policy are described in detail in the *IRIS FailSafe Version 2 Programmer's Guide* and in summary in "Defining a Failover Policy", page 144.

When you are finished defining the failover policy, enter `done` to return to the `cmgr` prompt.

Modifying Failover Policies

After you have defined a failover policy, you can modify or delete it.

Modifying Failover Policies with the Cluster Manager GUI

To modify a failover policy with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.

2. On the left side of the display, click on the “Failover Policies & Resource Groups” category.
3. On the right side of the display click on the “Modify a Failover Policy Definition” task link to launch the task.
4. Enter the selected inputs.
5. Click on “OK” at the bottom of the screen to complete the task, or click on “Cancel” to cancel.

Modifying Failover Policies with the Cluster Manager CLI

Use the following CLI command to modify a failover policy:

```
cmgr> modify failover_policy A
```

You modify a failover policy using the same commands you use to define a failover policy.

Deleting Failover Policies

After you have defined a failover policy, you can delete it.

Deleting Failover Policies with the Cluster Manager GUI

To delete a failover policy with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the “Failover Policies & Resource Groups” category.
3. On the right side of the display click on the “Delete a Failover Policy” task link to launch the task.
4. Enter the selected inputs.
5. Click on “OK” at the bottom of the screen to complete the task, or click on “Cancel” to cancel.

Deleting Failover Policies with the Cluster Manager CLI

You can use the following command to delete a failover policy definition:

```
cmgr> delete failover_policy A
```

Displaying Failover Policies

You can use IRIS FailSafe to display any of the following:

- The components of a specified failover policy
- All of the failover policies that have been defined
- All of the failover policy attributes that have been defined
- All of the failover policy scripts that have been defined

Displaying Failover Policies with the Cluster Manager GUI

The Cluster Manager GUI provides a convenient display of failover policies through the FailSafe Cluster View. You can launch the FailSafe Cluster View directly, or you can bring it up at any time by clicking on the "FailSafe Cluster View" prompt at the bottom of the "FailSafe Manager" display.

From the View menu of the FailSafe Cluster View, select Failover Policies to see all defined failover policies.

Displaying Failover Policies with the Cluster Manager CLI

Use the following command to view the parameters of a defined failover policy:

```
cmgr> show failover_policy A
```

Use the following command to view all of the defined failover policies:

```
cmgr> show failover policies
```

Use the following command to view all of the defined failover policy attributes:

```
cmgr> show failover_policy attributes
```

Use the following command to view all of the defined failover policy scripts:

```
cmgr> show failover_policy scripts
```

Defining Resource Groups

Resources are configured together into *resource group*s. A resource group is a collection of interdependent resources. If any individual resource in a resource group becomes unavailable for its intended use, then the entire resource group is considered unavailable. Therefore, a resource group is the unit of failover for IRIS FailSafe.

For example, a resource group could contain all of the resources that are required for the operation of a web node, such as the web node itself, the IP address with which it communicates to the outside world, and the disk volumes containing the content that it serves.

When you define a resource group, you specify a *failover policy*. A failover policy controls the behavior of a resource group in failure situations.

To define a resource group, you provide the following information:

- The name of the resource group, with a maximum length of 63 characters.
- The name of the cluster to which the resource group is available
- The resources to include in the resource group, and their resource types
- The name of the failover policy that determines which node will take over the services of the resource group on failure

FailSafe does not allow resource groups that do not contain any resources to be brought online.

You can define up to 100 resources configured in any number of resource groups.

Defining a Resource Group with the Cluster Manager GUI

To define a resource group with the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on “Guided Configuration”.
3. On the right side of the display click on “Set Up Highly Available Resource Groups” to launch the task link.
4. In the resulting window, click each task link in turn, as it becomes available. Enter the selected inputs for each task.

5. When finished, click "OK" to close the taskset window.

Defining a Resource Group with the Cluster Manager CLI

To configure a resource group, enter the following command at the `cmgr` prompt to specify the name of a resource group and the cluster to which the resource group is available:

```
cmgr> define resource_group A [in cluster B]
```

Entering this command specifies the name of the resource group you are defining within a specified cluster. If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

The following prompt appears:

```
Enter commands, you may enter "done" or "cancel" at any time to exit
resource_group A?
```

When this prompt appears you can use the following commands to specify the resources to include in the resource group or remove from the resource group, and to specify the failover policy to apply to the resource group:

```
resource_group A? add resource B of resource_type C
resource_group A? remove resource D of resource_type E
resource_group A? set failover_policy to F
```

After you have set the failover policy and you have finished adding resources to the resource group, enter `done` to return to the `cmgr` prompt.

For a full example of resource group creation using the Cluster Manager CLI see "Resource Group Creation Example", page 161.

Modifying Resource Groups

After you have defined resource groups, you can modify the resource groups. You can change the failover policy of a resource group by specifying a new failover policy associated with that resource group, and you can add or delete resources to the existing resource group. Note, however, that since you cannot have a resource group online that does not contain any resources, FailSafe does not allow you to delete all resources from a resource group once the resource group is online. Likewise, FailSafe does not allow you to bring a resource group online if it has no resources. Also,

resources must be added and deleted in atomic units; this means that resources which are interdependent must be added and deleted together.

Modifying Resource Groups with the Cluster Manager GUI

To modify a resource group with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Failover Policies & Resource Groups" category.
3. On the right side of the display click on the "Modify a Resource Group Definition" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

To add or delete resources to a resource group definition with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Failover Policies & Resource Groups" category.
3. On the right side of the display click on the "Add/Remove Resources in Resource Group" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Modifying Resource Groups with the Cluster Manager CLI

Use the following CLI command to modify a resource group:

```
cmgr> modify resource_group A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default. You modify a resource group using the same commands you use to define a resource group:

```
resource_group A? add resource B of resource_type C
resource_group A? remove resource D of resource_type E
resource_group A? set failover_policy to F
```

Deleting Resource Groups

After you have defined resource groups, you can delete the resource groups.

Deleting Resource Groups with the Cluster Manager GUI

To delete a resource group with the Cluster Manager GUI, perform the following procedure:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Failover Policies & Resource Groups" category.
3. On the right side of the display click on the "Delete a Resource Group" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task, or click on "Cancel" to cancel.

Deleting Resource Groups with the Cluster Manager CLI

You can use the following command to delete a resource group definition:

```
cmgr> delete resource_group A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

Displaying Resource Groups

You can display the parameters of a defined resource group, and you can display all of the resource groups defined for a cluster.

Displaying Resource Groups with the Cluster Manager GUI

The Cluster Manager GUI provides a convenient display of resource groups through the FailSafe Cluster View. You can launch the FailSafe Cluster View directly, or you can bring it up at any time by clicking on the "FailSafe Cluster View" prompt at the bottom of the "FailSafe Manager" display.

From the View menu of the FailSafe Cluster View, select Groups to see all defined resource groups.

To display which nodes are currently running which groups, select "Groups owned by Nodes." To display which groups are running which failover policies, select "Groups by Failover Policies."

Displaying Resource Groups with the Cluster Manager CLI

Use the following command to view the parameters of a defined resource group:

```
cmgr> show resource_group A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

Use the following command to view all of the defined failover policies:

```
cmgr> show resource_groups [in cluster A]
```

FailSafe System Log Configuration

IRIS FailSafe maintains system logs for each of the FailSafe daemons. You can customize the system logs according to the level of logging you wish to maintain.

A log group is a set of processes that log to the same log file according to the same logging configuration. All FailSafe daemons make one log group each. FailSafe maintains the following log groups:

`cli`	Commands log
`crsd`	Cluster reset services (`crsd`) log

`diags`	Diagnostics log
`ha_agent`	HA monitoring agents (`ha_ifmx2`) log
`ha_cmsd`	FailSafe membership daemon (`ha_cmsd`) log
`ha_fsd`	FailSafe daemon (`ha_fsd`) log
`ha_gcd`	Group communication daemon (`ha_gcd`) log
`ha_ifd`	network interface monitoring daemon (`ha_ifd`) log
`ha_script`	Action and Failover policy scripts log
`ha_srmd`	System resource manager (`ha_srmd`) log

Log group configuration information is maintained for all nodes in the pool for the `cli` and `crsd` log groups or for all nodes in the cluster for all other log groups. You can also customize the log group configuration for a specific node in the cluster or pool.

When you configure a log group, you specify the following information:

- The log level, specified as character strings with the CUI and numerically (1 to 19) with the CLI, as described below
- The log file to log to
- The node whose specified log group you are customizing (optional)

The log level specifies the verbosity of the logging, controlling the amount of log messages that FailSafe will write into an associated log group's file. There are 10 debug level. Table 5-2 shows the logging levels as you specify them with the GUI and the CLI.

Table 5-2 Log Levels

GUI level	CLI level	Meaning
Off	0	No logging
Minimal	1	Logs notification of critical errors and normal operation
Info	2	Logs minimal notification plus warning
Default	5	Logs all Info messages plus additional notifications

GUI level	CLI level	Meaning
Debug0	10	
...		Debug0 through Debug9 (11 -19 in CLI) log increasingly more debug information, including data structures. Many megabytes of disk space can be consumed on the server when debug levels are used in a log configuration.
Debug9	19	

Note: Notifications of critical errors and normal operations are always sent to `/var/adm/SYSLOG`. Changes you make to the log level for a log group do not affect `SYSLOG`.

The FailSafe software appends the node name to the name of the log file you specify. For example, when you specify the log file name for a log group as `/var/cluster/ha/log/cli`, the file name will be `/var/cluster/ha/log/cli_`*nodename*.

The default log file names are as follows.

`/var/cluster/ha/log/cmsd_`*nodename*

log file for the FailSafe membership services daemon in node `nodename`

`/var/cluster/ha/log/gcd_`*nodename*

log file for group communication daemon in node `nodename`

`/var/cluster/ha/log/srmd_`*nodename*

log file for system resource manager daemon in node `nodename`

`/var/cluster/ha/log/failsafe_`*nodename*

log file for failsafe daemon, a policy implementor for resource groups, in node `nodename`

`/var/cluster/ha/log/`*agent_nodename*

log file for monitoring agent named `agent` in node `nodename`. For example, `ifd_nodename` is the log file for the interface daemon

monitoring agent that monitors interfaces and IP addresses and performs local failover of IP addresses.

`/var/cluster/ha/log/crsd_`*nodename*

log file for reset daemon in node `nodename`

`/var/cluster/ha/log/`script_*nodename*

log file for scripts in node `nodename`

`/var/cluster/ha/log/cli_`*nodename*

log file or internal administrative commands in node `nodename` invoked by the Cluster Manager GUI and Cluster Manager CLI

For information on using log groups in system recovery, see Chapter 9, "IRIS FailSafe Recovery".

Configuring Log Groups with the Cluster Manager GUI

To configure a log group with the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Clusters" category.
3. On the right side of the display click on the "Set Log Configuration" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.

Configuring Log Groups with the Cluster Manager CLI

You can configure a log group with the following CLI command:

```
cmgr> define log_group A [on node B] [in cluster C]
```

You specify the node if you wish to customize the log group configuration for a specific node only. If you have specified a default cluster, you do not have to specify a cluster in this command; FailSafe will use the default.

The following prompt appears:

```
Enter commands, you may enter "done" or "cancel" at any time to exit
log_group A?
```

When this prompt of the node name appears, you enter the log group parameters you wish to modify in the following format:

```
log_group A? set log_level to A
log_group A? add log_file A
log_group A? remove log_file A
```

When you are finished configuring the log group, enter `done` to return to the `cmgr` prompt.

Modifying Log Groups with the Cluster Manager CLI

Use the following CLI command to modify a log group:

```
cmgr> modify log_group A on [node B] [in cluster C]
```

You modify a log group using the same commands you use to define a log group.

Displaying Log Group Definitions with the Cluster Manager GUI

To display log group definitions with the Cluster Manager GUI, run “Set Log Configuration” and choose the log group to display from the rollover menu. The current log level and log file for that log group will be displayed in the task window, where you can change those settings if you desire.

Displaying Log Group Definitions with the Cluster Manager CLI

Use the following command to view the parameters of a defined resource:

```
cmgr> show log_groups
```

This command shows all of the log groups currently defined, with the log group name, the logging levels and the log files.

For information on viewing the contents of the log file, see Chapter 9, "IRIS FailSafe Recovery".

Resource Group Creation Example

Use the following procedure to create a resource group using the Cluster Manager CLI:

1. Determine the list of resources that belong to the resource group you are defining. The list of resources that belong to a resource group are the resources that move from one node to another as one unit.

   A resource group that provides NFS services would contain a resource of each of the following types:

   - `IP_address`
   - `volume`
   - `filesystem`
   - `NFS`

     All resource and resource type dependencies of resources in a resource group must be satisfied. For example, the `NFS` resource type depends on the `filesystem` resource type, so a resource group containing a resource of `NFS` resource type should also contain a resource of `filesystem` resource type.

2. Determine the failover policy to be used by the resource group.
3. Use the template `cluster_mgr` script available in the `/var/cluster/cmgr-templates/cmgr-create-resource_group` file.

   This example shows a script that creates a resource group with the following characteristics:

   - the resource group is named `nfs-group`
   - the resource group is in cluster `HA-cluster`
   - the resource group uses the failover policy
   - the resource group contains `IP_Address`, `volume`, `filesystem`, and `NFS` resources

     The following script can be used to create this resource group:

```
define resource_group nfs-group in cluster HA-cluster
        set failover_policy to n1_n2_ordered
        add resource 192.0.2.34 of resource_type IP_address
```

```
        add resource havol1 of resource_type volume
        add resource /hafs1 of resource_type filesystem
        add resource /hafs1 of resource_type NFS
done
```

4. Run this script using the `-f` option of the `cluster_mgr`(1M) command.

IRIS FailSafe Configuration Examples

This chapter provides an example of a FailSafe configuration that uses a three-node cluster, and some variations of that configuration. In addition, this chapter provides instructions for exporting CXFS filesystems in a FailSafe configuration. It includes the following sections:

FailSafe Example with Three-Node Cluster

The following illustration shows a three-node FailSafe cluster. This configuration consists of a pool containing nodes `N1`, `N2`, `N3`, and `N4`. Nodes `N1`, `N2`, and `N3` make up the FailSafe cluster. The nodes in this cluster share disks, and are connected to an El-8 serial port multiplexer, which is also connected to the private control network.

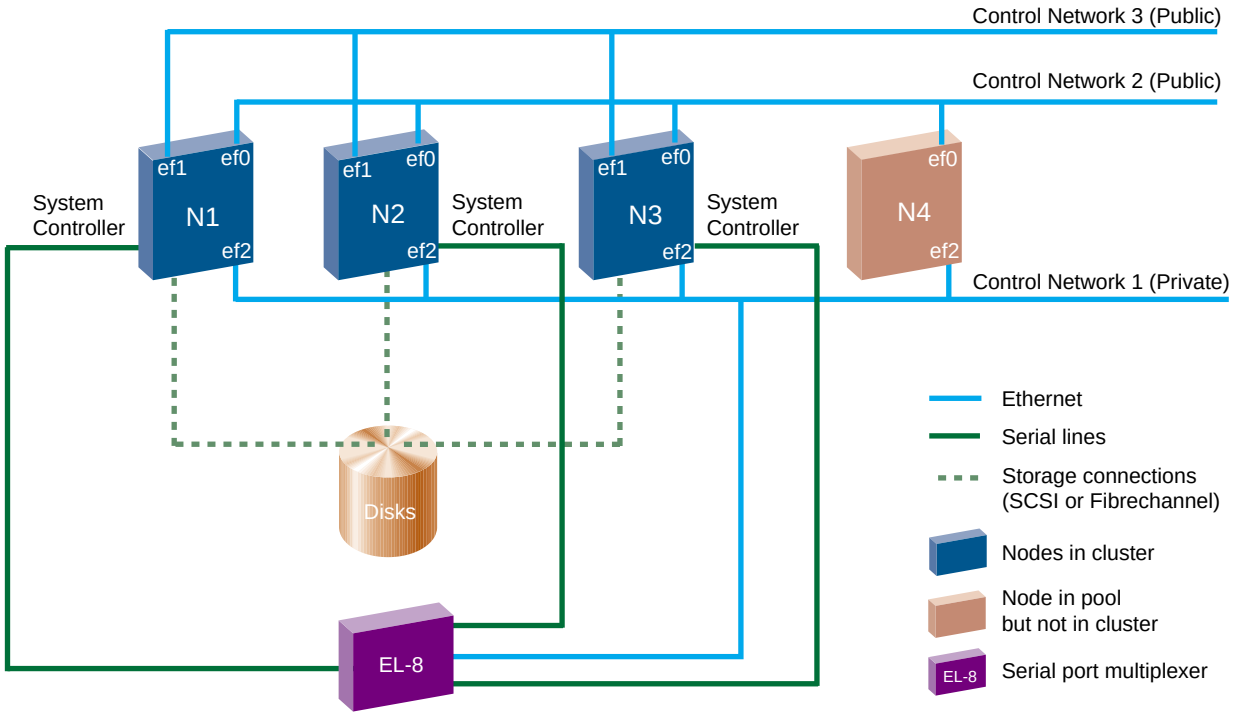

Figure 6-1 FailSafe Configuration Example

Examples of FailSafe configurations that use this setup are provided in the following sections.

FailSafe cmgr Script to Configure Example

This section provides an example `cmgr` script that defines a FailSafe three-node cluster as shown in Figure 6-1. For general information on CLI scripts see "CLI Command Scripts", page 89. For information on the CLI template files that you can use to create your own configuration script, see "CLI Template Scripts", page 90.

This cluster has two resource groups, `RG1` and `RG2`.

Resource group `RG1` contains the following resources:

IP address	192.26.50.1
filesystem	`/ha1`
volume	`ha1_vol`
NFS	`/ha1/export`

Resource group `RG1` has a failover policy of `FP1`. `FP1` has the following components:

script	`ordered`
attributes	`Auto_Failback`
	`Auto_Recovery`
failover domain	`N1, N2, N3`

Resource group `RG2` contains the following resources:

IP address	192.26.50.2
filesystem	`/ha2`
volume	`ha2_vol`
NFS	`/ha2/export`

Resource group `RG2` has a failover policy of `FP2`. `FP2` has the following components:

script	`round-robin`
attributes	`Controlled_Failback`
	`Inplace_Recovery`
failover domain	`N2, N3`

The `cmgr` script to define this configuration is as follows:

```
#!/usr/cluster/bin/cluster_mgr -f
define node N1
        set hostname to N1
        set is_failsafe to true
        set sysctrl_type to msc
        set sysctrl_status to enabled
```

```
        set sysctrl_password to none
        set sysctrl_owner to N4
        set sysctrl_device to /dev/ttydn001
        set sysctrl_owner_type to tty
        add nic ef2-N1
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
        add nic ef0-N1
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 2
        done
        add nic ef1-N1
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 3
        done
done


define node N2
        set hostname to N2
        set is_failsafe to true
        set sysctrl_type to msc
        set sysctrl_status to enabled
        set sysctrl_password to none
        set sysctrl_owner to N4
        set sysctrl_device to /dev/ttydn002
        set sysctrl_owner_type to tty
        add nic ef2-N2
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
        add nic ef0-N2
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 2
        done
```

```
        add nic ef1-N2
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 3
        done
done

define node N3
        set hostname to N3
        set is_failsafe to true
        set sysctrl_type to msc
        set sysctrl_status to enabled
        set sysctrl_password to none
        set sysctrl_owner to N4
        set sysctrl_device to /dev/ttydn003
        set sysctrl_owner_type to tty
        add nic ef2-N3
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
        add nic ef0-N3
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 2
        done
        add nic ef1-N3
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 3
        done
done


define node N4
        set hostname to N4
        set is_failsafe to true
        add nic ef2-N4
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
```

```
        done
        add nic ef0-N4
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 2
        done
done
define cluster TEST
        set is_failsafe to true
        set notify_cmd to /usr/bin/mail
        set notify_addr to failsafe_sysadm@company.com
        add node N1
        add node N2
        add node N3
done

define failover_policy fp1
        set attribute to Auto_Failback
        set attribute to Auto_Recovery
        set script to ordered
        set domain to N1 N2 N3
done


define failover_policy fp2
        set attribute to Controlled_Failback
        set attribute to Inplace_Recovery
        set script to round-robin
        set domain to N2 N3
done

define resource 192.26.50.1 of resource_type IP_address in cluster TEST
        set NetworkMask to 0xffffff00
        set interfaces to ef0,ef1
        set BroadcastAddress to 192.26.50.255
done

define resource ha1_vol of resource_type volume in cluster TEST
        set devname-owner to root
        set devname-group to sys
        set devname-mode to 666
```

```
done

define resource /ha1 of resource_type filesystem in cluster TEST
        set volume-name to ha1_vol
        set mount-options to rw,noauto
        set monitoring-level to 2
done

modify resource /ha1 of resource_type filesystem in cluster TEST
        add dependency ha1_vol of type volume
done

define resource /ha1/export of resource_type NFS in cluster TEST
        set export-info to rw,wsync
        set filesystem to /ha1
done

modify resource /ha1/export of resource_type NFS in cluster TEST
        add dependency /ha1 of type filesystem
done
define resource_group RG1 in cluster TEST
        set failover_policy to fp1
        add resource 192.26.50.1 of resource_type IP_address
        add resource ha1_vol of resource_type volume
        add resource /ha1 of resource_type filesystem
        add resource /ha1/export of resource_type NFS
done


define resource 192.26.50.2 of resource_type IP_address in cluster TEST
        set NetworkMask to 0xffffff00
        set interfaces to ef0
        set BroadcastAddress to 192.26.50.255
done

define resource ha2_vol of resource_type volume in cluster TEST
        set devname-owner to root
        set devname-group to sys
        set devname-mode to 666
done
```

```
define resource /ha2 of resource_type filesystem in cluster TEST
        set volume-name to ha2_vol
        set mount-options to rw,noauto
        set monitoring-level to 2
done

modify resource /ha2 of resource_type filesystem in cluster TEST
        add dependency ha2_vol of type volume
done

define resource /ha2/export of resource_type NFS in cluster TEST
        set export-info to rw,wsync
        set filesystem to /ha2
done

modify resource /ha2/export of resource_type NFS in cluster TEST
        add dependency /ha2 of type filesystem
done

define resource_group RG2 in cluster TEST
        set failover_policy to fp2
        add resource 192.26.50.2 of resource_type IP_address
        add resource ha2_vol of resource_type volume
        add resource /ha2 of resource_type filesystem
        add resource /ha2/export of resource_type NFS
done


quit
```

Modifying FailSafe Cluster to include a CXFS Filesystem

The following procedural example modifies the sample FailSafe configuration illustrated in Figure 6-1 so that it includes highly-available NFS services on a CXFS filesystem.

Note: IRIS FailSafe assumes that CXFS filesystems are highly-available because they do not require a FailSafe failover in order to be made available on another node in the cluster. Therefore, FailSafe does not directly start, stop, or monitor CXFS filesystems or XVM volumes, and CXFS filesystems and XVM volumes should not be added to the FailSafe resource groups.

To modify the FailSafe configuration to include a CXFS filesystem, perform the following steps:

1. Convert the cluster `TEST` for CXFS use. For information on converting FailSafe clusters to CXFS, see the *CXFS Software Installation and Administration Guide*.
2. Convert the nodes `N1` and `N2` for CXFS use. For information on converting FailSafe nodes to CXFS, see the *CXFS Software Installation and Administration Guide*. Start CXFS services on the nodes.
3. Create a new resource type `NFS1`. This is the same as resource type `NFS` but without a filesystem dependency. To create this resource type you can perform the following steps:
   a. Using `cmgr`, execute the following:

      ```
      show resource_type NFS in cluster TEST
      ```

      The parameters of resource type `NFS` will be displayed.

   b. Define resource type `NFS1` using the same configuration information that was displayed for resource type NFS, but do not copy the filesystem dependency.
4. Define a new failover policy, `FP3`, with the following attributes:

AFD	`N1`, `N2`
script	`ordered`
attribute	`Inplace_Recovery`

5. Create a resource group `RG3` with failover policy `FP3`, resource `IP3` of resource type `IP_address`, and resource `/cxfs/exports` of resource type `NFS1`.
6. Mount `/cxfs` on nodes `N1` and `N2`. For information on defining a CXFS filesystem with an XVM volume and for information on mounting CXFS filesystems, see *CXFS Software Installation and Administration Guide*.
7. Bring resource group `RG3` online in cluster `TEST`.

Local Failover of IP Address

You can configure a FailSafe system to fail over an IP address to a second interface within the same host. To do this, you specify multiple interfaces for resources of `IP_address` resource type. You can also specify different interfaces for supporting a heterogeneous cluster. For information on specifying IP address resources, see "IP_address Resource Attributes", page 118.

The following example configures local failover of an IP address. It uses the configuration illustrated in Figure 6-1.

1. Define an IP address resource with two interfaces:

```
define resource 192.26.50.1 of resource_type IP_address in cluster TEST
          set NetworkMask to 0xffffff00
          set interfaces to ef0,ef1
          set BroadcastAddress to 192.26.50.255
done
```

   IP address 192.26.50.1 will be locally failed over from interface `ef0` to interface `ef1` when there is an `ef0` interface failure.

   In nodes `N1`, `N2`, and `N3`, either `ef0` or `ef1` should configure up automatically, when the node boots up. Both `ef0` and `ef1` are physically connected to the same subnet 192.26.50. Only one network interface connected to the same network should be configured up in a node.

2. Modify the `/etc/conf/netif.options` file to configure the `ef0` and `ef1` interfaces:

```
if1name-ef0
if1addr=192.26.50.10

if2name=ef1
if2addr=192.26.50.11
```

3. The `etc/init.d/network` script should configure the network interface `ef1` down in all nodes `N1`, `N2`, and `N3`. Add the following line to the file:

```
ifconfig ef1 down
```

Exporting CXFS Filesystems

To export a CXFS filesystem in a FailSafe configuration, you must create a special NFS resource type to be used with CXFS filesystems. This is necessary because FailSafe should no manipulate or monitor the CXFS filesystems that are exported. Therefore, the new resource type will not include a filesystem dependency.

Perform the following steps to export CXFS filesystems in a FailSafe configuration:

1. Make sure that you have loaded the NFS 2.1 release as the NFS agent software.
2. Create a new NFS resource type for CXFS filesystems. In this procedure, this new resource type is called `NFS_CXFS`.

   You can create a new resource type similar to the existing NFS resource type with the following procedure:

   a. Copy the `var/cluster/ha/resource_types/NFS` directory to a directory with the name of the new resource type, in this case `var/cluster/ha/resource_types/NFS_CXFS`.

   b. After creating the new `NFS_CXFS` directory, modify all the scripts in the directory to specify `NFS_CXFS` for `LOCAL_TEST_KEY`. It is recommended that you change NFS references in the script (for example, in variable names) to reflect the new resource type name of `NFS_CXFS`; however, you should not change NFS references in the log messages.
3. After you have created the new `NFS_CXFS` resource type, eliminate the filesystem dependency from the resource type, using either the GUI or `cmgr`, as follows:

```
perf34 40# cmgr
Welcome to SGI Cluster Manager Command-Line Interface

cmgr> modify resource_type NFS_CXFS in cluster "testcluster"
Enter commands, when finished enter either "done" or "cancel"

resource_type NFS_CXFS ? remove dependency filesystem
resource_type NFS_CXFS ? done
Successfully modified resource_type NFS_CXFS
```

4. Modify the monitor script for the new resource type, `/var/cluster/ha/resource_types/NFS_CXFS/monitor`. The difference between the standard NFS monitor script and the monitor script for the new resource type is that when we export CXFS filesystems, we do not want FailSafe

to check if the filesystem is mounted and to exit with `HA_CMD_FAILED` if it is not. The CXFS monitoring script itself will determine what action should take place if the filesystem becomes unmounted.

The following section of the monitor script has the line beginning with `exit_script` commented out. The status of the commands will be written to log, but the script will not exit.

```
# Check to see if the filesystem is mounted
        HA_CMD="/sbin/mount | grep $fs  >> /dev/null 2>&1"
        ha_execute_cmd "check to see if $fs is mounted"
        if [ $? -ne 0 ]; then
            ${HA_LOG} "NFS: $fs not mounted";
            ha_write_status_for_resource ${resource} ${HA_CMD_FAILED};
#           exit_script $HA_CMD_FAILED;
```

Note: SGI does not currently advocate lock failover of NFS exported CXFS filesystems. If you choose to use locking you will have to create a new `statd_unlimited` resource type with a dependency of `NFS_CXFS` instead of `NFS`. The original `statd` resource type can not be used for an exported CXFS filesystem.

IRIS FailSafe System Operation

This chapter describes administrative tasks you perform to operate and monitor an IRIS FailSafe system. It describes how to perform tasks using the IRIS FailSafe Cluster Manager Graphical User Interface (GUI) and the IRIS FailSafe Cluster Manager Command Line Interface (CLI). The major sections in this chapter are as follows:

Note: It is recommended that all FailSafe administration be done from one node in the pool so that the latest copy of the database will be available even when there are network partitions.

Setting System Operation Defaults

Several commands that you perform on a running system allow you the option of specifying a node or cluster. You can specify a node or a cluster to use as the default if you do not specify the node or cluster explicitly.

Setting Default Cluster with Cluster Manager GUI

The Cluster Manager GUI prompts you to enter the name of the default cluster when you have not specified one. Alternately, you can set the default cluster by clicking the "Select Cluster..." button at the bottom of the FailSafe Manager window.

When using the Cluster Manager GUI, there is no need to set a default node.

Setting Defaults with Cluster Manager CLI

When you are using the Cluster Manager CLI, you can use the following commands to specify default values. Use either of the following commands to specify a default cluster:

```
cmgr> set cluster A
cmgr> set node A
```

System Operation Considerations

Once a FailSafe command is started, it may partially complete even if you interrupt the command by typing `Ctrl-c`. If you halt the execution of a command this way, you may leave the cluster in an indeterminate state and you may need to use the various status commands to determine the actual state of the cluster and its components.

Activating (Starting) IRIS FailSafe

After you have configured your IRIS FailSafe system and run diagnostic tests on its components, you can activate the highly available services by starting FailSafe. You can start FailSafe on a systemwide basis, on all of the nodes in a cluster, or on a specified node only.

Caution: When you start HA services on a subset of the nodes, you should make sure that resource groups are not running in other nodes in the cluster. For example, if a cluster contains nodes N1, N2, and N3 and HA services are started on nodes N1 and N2 but not on node N3, you should make sure that resource groups are not running on node N3. FailSafe will not perform exclusivity checks on nodes where HA services are not started.

When you start HA services, the following actions are performed:

1. All nodes in the cluster in the CDB are enabled
2. FailSafe returns success to the user after modifying the CDB
3. The local CMOND gets notification from fs2d/CDBD
4. The local CMOND starts all HA processes (CMSD, GCD, SRMD, FSD) and IFD.
5. CMOND sets `failsafe2` `chkconfig` flag to `on`.

Activating IRIS FailSafe with the Cluster Manager GUI

To start FailSafe services using the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.
3. On the right side of the display click on the "Start FailSafe HA Services" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.

Activating IRIS FailSafe with the Cluster Manager CLI

To activate IRIS FailSafe in a cluster, use the following command:

```
cmgr> start ha_services [on node A] [for cluster B]
```

System Status

While the IRIS FailSafe system is running, you can monitor the status of the IRIS FailSafe components to determine the state of the component. FailSafe allows you to view the system status in the following ways:

- You can keep continuous watch on the state of a cluster using the `cluster_status` command, or the FailSafe Cluster View of the Cluster Manager GUI.

- You can query the status of an individual resource group, node, or cluster using either the Cluster Manager GUI or the Cluster Manager CLI.
- You can use the `haStatus` script provided with the Cluster Manager CLI to see the status of all clusters, nodes, resources, and resource groups in the configuration.

The following sections describe the procedures for performing each of these tasks.

Monitoring System Status with the `cluster_status` command

You can use the `cluster_status` command to monitor the cluster using a curses(3X) interface. For example, the following shows a two-node cluster configured for FailSafe only with a NFS resource group and `cluster_status` help text displayed:

```
# /var/cluster/cmgr-scripts/cluster_status
+ Cluster=nfs-cluster  FailSafe=ACTIVE CXFS=Not Configured              10:57:57
   Nodes =   hans2    hans1
FailSafe =      UP       UP
    CXFS =

ID ]             ResourceGroup           Owner           State            Error
 0 ]                nfs-group1           hans1          Online         No error

                     +-------+ cluster_status Help +--------+
                     |  on s  - Toggle Sound on event       |
                     |  on r  - Toggle Resource Group View  |
                     |  on c  - Toggle CXFS View            |
                     |     h  - Toggle help screen          |
                     |     i  - View Resource Group detail  |
                     |     q  - Quit cluster_status         |
                     +--- Press 'h' to remove help window --+

--------------------------------------------------------------------------------
cmd('h' for help) >
```

The above shows that a sound will be activated when a node or the cluster changes status. You can override the `s` setting by invoking `cluster_status` with the `-m` (mute) option.

Monitoring System Status with the Cluster Manager GUI

The easiest way to keep a continuous watch on the state of a cluster is to use the FailSafe Cluster View of the Cluster Manager GUI. You can launch the FailSafe Cluster View directly from the FailSafe toolchest.

In the FailSafe Cluster View window, problems system components are experiencing appear as blinking red icons. Components in transitional states also appear as blinking icons. If there is a problem in a resource group or node, the FailSafe Cluster View icon for the cluster turns red and blinks, as well as the resource group or node icon.

The full color legend for component states in the FailSafe Cluster View is as follows:

grey	healthy but not online or active
green	healthy and active or online
blinking green	transitioning to green
blinking red	problems with component
black and white outline	resource type
grey with yellow wrench	maintenance mode, may or may not be currently monitored by FailSafe

If you minimize the FailSafe Cluster View window, the minimized-icon shows the current state of the cluster. When the cluster has FailSafe HA services active and there is no error, the icon shows a green cluster. When the cluster goes into error state, the icon shows a red cluster. When the cluster has FailSafe HA services inactive, the icon shows a grey cluster.

Monitoring Resource and Reset Serial Line with the Cluster Manager CLI

You can use the CLI to query the status of a resource or to ping the system controller at a node, as described in the following subsections.

Querying Resource Status with the Cluster Manager CLI

To query a resource status, use the following CLI command:

```
cmgr> show status of resource A of resource_type B [in cluster C]
```

If you have specified a default cluster, you do not need to specify a cluster when you use this command and it will show the status of the indicated resource in the default cluster.

Pinging a System Controller with the Cluster Manager CLI

To perform a ping operation on a system controller by providing the device name, use the following CLI command:

```
cmgr> admin ping dev_name A of dev_type B with sysctrl_type C
```

Resource Group Status

To query the status of a resource group, you provide the name of the resource group and the cluster which includes the resource group. Resource group status includes the following components:

- resource group state
- resource group error state
- resource owner

These components are described in the following subsections.

If a node that contains a resource group online has a status of UNKNOWN, the status of the resource group will not be available or ONLINE-READY.

Resource Group State

A resource group state can be one of the following:

ONLINE	FailSafe is running on the local nodes. The resource group is allocated on a node in the cluster and is being monitored by IRIS FailSafe. It is fully allocated if there is no error; otherwise, some resources may not be allocated or some resources may be in error state.
ONLINE-PENDING	FailSafe is running on the local nodes and the resource group is in the process of being allocated. This is a transient state.
OFFLINE	The resource group is not running or the resource group has been detached, regardless of whether

	FailSafe is running. When FailSafe starts up, it will not allocate this resource group.
OFFLINE-PENDING	FailSafe is running on the local nodes and the resource group is in the process of being released (becoming offline). This is a transient state.
ONLINE-READY	FailSafe is not running on the local node. When FailSafe starts up, it will attempt to bring this resource group online. No FailSafe process is running on the current node is this state is returned.
ONLINE-MAINTENANCE	The resource group is allocated in a node in the cluster but it is not being monitored by IRIS FailSafe. If a node failure occurs while a resource group in ONLINE-MAINTENANCE state resides on that node, the resource group will be moved to another node and monitoring will resume. An administrator may move a resource group to an ONLINE-MAINTENANCE state for upgrade or testing purposes, or if there is any reason that IRIS FailSafe should not act on that resource for a period of time.
INTERNAL ERROR	An internal FailSafe error has occurred and FailSafe does not know the state of the resource group. Error recovery is required. This could result from a memory error, bugs in a program, or communication problems.
DISCOVERY (EXCLUSIVITY)	The resource group is in the process of going online if FailSafe can correctly determine whether any resource in the resource group is already allocated on all nodes

	in the resource group's application failure domain. This is a transient state.
INITIALIZING	FailSafe on the local node has yet to get any information about this resource group. This is a transient state.

Resource Group Error State

When a resource group is ONLINE, its error status is continually being monitored. A resource group error status can be one of the following:

NO ERROR	Resource group has no error.
INTERNAL ERROR - NOT RECOVERABLE	Notify Silicon Graphics if this condition arises.
NODE UNKNOWN	Node that had the resource group online is in unknown state. This occurs when the node is not part of the cluster. The last known state of the resource group is ONLINE, but the system cannot talk to the node.
SRMD EXECUTABLE ERROR	The start or stop action has failed for a resource in the resource group.
SPLIT RESOURCE GROUP (EXCLUSIVITY)	FailSafe has determined that part of the resource group was running on at least two different nodes in the cluster.
NODE NOT AVAILABLE (EXCLUSIVITY)	FailSafe has determined that one of the nodes in the resource group's application failure domain was not in the membership. FailSafe cannot bring the resource group online until that node is removed from the application failure domain or HA services are started on that node.
MONITOR ACTIVITY UNKNOWN	In the process of turning maintenance mode on or off, an error occurred. FailSafe can no longer determine if monitoring is enabled or disabled. Retry the operation. If the error continues, report the error to Silicon Graphics.

NO AVAILABLE NODES	A monitoring error has occurred on the last valid node in the FailSafe membership.

Resource Owner

The resource owner is the logical node name of the node that currently owns the resource.

Monitoring Resource Group Status with the Cluster Manager GUI

You can use the FailSafe ClusterView to monitor the status of the resources in a FailSafe configuration. You can launch the FailSafe Cluster View directly, or you can bring it up at any time by clicking on "FailSafe Cluster View" at the bottom of the "FailSafe Manager" display.

From the View menu, select "Resources in Groups" to see the resources organized by the groups they belong to, or select "Groups owned by Nodes" to see where the online groups are running. This view lets you observe failovers as they occur.

Querying Resource Group Status with the Cluster Manager CLI

To query a resource group status, use the following CLI command:

```
cmgr> show status of resource_group A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster when you use this command and it will show the status of the indicated resource group in the default cluster.

Node Status

To query the status of a node, you provide the logical node name of the node. The node status can be one of the following:

UP	This node is part of the FailSafe membership.
DOWN	This node is not part of the FailSafe membership (no heartbeats) and this node has been reset. This is a transient state.
UNKNOWN	This node is not part of the FailSafe membership (no heartbeats) and this nod has not been reset (reset attempt has failed).
INACTIVE	HA services have not been started on this node.

When you start HA services, node states transition from INACTIVE to UP. It may happen that a node state may transition from INACTIVE to UNKNOWN to UP.

Monitoring Node Status with the `cluster_status` command

You can use the `cluster_status` command to monitor the status of the nodes in the cluster.

Monitoring Cluster Status with the Cluster Manager GUI

You can use the FailSafe ClusterView to monitor the status of the clusters in a FailSafe configuration. You can launch the FailSafe Cluster View directly, or you can bring it up at any time by clicking on "FailSafe Cluster View" at the bottom of the "FailSafe Manager" display.

From the View menu, select "Groups owned by Nodes" to monitor the health of the default cluster, its resource groups, and the group's resources.

Querying Node Status with the Cluster Manager CLI

To query node status, use the following CLI command:

```
cmgr> show status of node A
```

Pinging the System Controller with the Cluster Manager CLI

When FailSafe is running, you can determine whether the system controller on a node is responding with the following Cluster Manger CLI command:

```
cmgr> admin ping node A
```

This command uses the FailSafe daemons to test whether the system controller is responding.

You can verify reset connectivity on a node in a cluster even when the FailSafe daemons are not running by using the `standalone` option of the `admin ping` command of the CLI:

```
cmgr> admin ping standalone node A
```

This command does not go through the FailSafe daemons, but calls the `ping` command directly to test whether the system controller on the indicated node is responding.

Cluster Status

To query the status of a cluster, you provide the name of the cluster. The cluster status can be one of the following:

ACTIVE	There is a valid FailSafe membership and this node is part of the cluster.
INACTIVE	There is no FailSafe membership. HA services have not been started in the cluster.
UNKNOWN	HA services have been started in this cluster. This node is not part of the membership. There may be a valid FailSafe membership.

When you start HA services in the cluster, the cluster state transitions from INACTIVE to ACTIVE.

Querying Cluster Status with the Cluster Manager GUI

You can use the ClusterView of the Cluster Manager GUI to monitor the status of the clusters in a FailSafe system.

Querying Cluster Status with the Cluster Manager CLI

To query node and cluster status, use the following CLI command:

```
cmgr> show status of cluster A
```

Viewing System Status with the haStatus CLI Script

The `haStatus` script provides status and configuration information about clusters, nodes, resources, and resource groups in the configuration. This script is installed in the `/var/cluster/cmgr-scripts` directory. You can modify this script to suit your needs. See the `haStatus` (1M) man page for further information about this script.

The following examples show the output of the different options of the `haStatus` script.

```
# haStatus -help
Usage: haStatus [-a|-i] [-c clustername]
where,
 -a prints detailed cluster configuration information and cluster
status.
 -i prints detailed cluster configuration information only.
 -c can be used to specify a cluster for which status is to be printed.
 ''clustername'' is the name of the cluster for which status is to be
printed.
# haStatus
Tue Nov 30 14:12:09 PST 1999
Cluster test-cluster:
        Cluster state is ACTIVE.
Node hans2:
        State of machine is UP.
Node hans1:
        State of machine is UP.
Resource_group nfs-group1:
        State: Online
        Error: No error
        Owner: hans1
        Failover Policy: fp_h1_h2_ord_auto_auto
        Resources:
                /hafs1  (type: NFS)
                /hafs1/nfs/statmon      (type: statd)
                150.166.41.95   (type: IP_address)
```

```
                /hafs1  (type: filesystem)
                havol1  (type: volume)
# haStatus -i
Tue Nov 30 14:13:52 PST 1999
Cluster test-cluster:
Node hans2:
        Logical Machine Name: hans2
        Hostname: hans2.engr.sgi.com
        Is FailSafe: true
        Is CXFS: false
        Nodeid: 32418
        Reset type: powerCycle
        System Controller: msc
        System Controller status: enabled
        System Controller owner: hans1
        System Controller owner device: /dev/ttyd2
        System Controller owner type: tty
        ControlNet Ipaddr: 192.26.50.15
        ControlNet HB: true
        ControlNet Control: true
        ControlNet Priority: 1
        ControlNet Ipaddr: 150.166.41.61
        ControlNet HB: true
        ControlNet Control: false
        ControlNet Priority: 2
Node hans1:
        Logical Machine Name: hans1
        Hostname: hans1.engr.sgi.com
        Is FailSafe: true
        Is CXFS: false
        Nodeid: 32645
        Reset type: powerCycle
        System Controller: msc
        System Controller status: enabled
        System Controller owner: hans2
        System Controller owner device: /dev/ttyd2
        System Controller owner type: tty
        ControlNet Ipaddr: 192.26.50.14
        ControlNet HB: true
        ControlNet Control: true
        ControlNet Priority: 1
```

```
        ControlNet Ipaddr: 150.166.41.60
        ControlNet HB: true
        ControlNet Control: false
        ControlNet Priority: 2
Resource_group nfs-group1:
        Failover Policy: fp_h1_h2_ord_auto_auto
                Version: 1
                Script: ordered
                Attributes: Auto_Failback Auto_Recovery
                Initial AFD: hans1 hans2
        Resources:
                /hafs1  (type: NFS)
                /hafs1/nfs/statmon      (type: statd)
                150.166.41.95   (type: IP_address)
                /hafs1  (type: filesystem)
                havol1  (type: volume)
Resource /hafs1 (type NFS):
        export-info: rw,wsync
        filesystem: /hafs1
        Resource dependencies
        statd /hafs1/nfs/statmon
        filesystem /hafs1
Resource /hafs1/nfs/statmon (type statd):
        InterfaceAddress: 150.166.41.95
        Resource dependencies
        IP_address 150.166.41.95
        filesystem /hafs1
Resource 150.166.41.95 (type IP_address):
        NetworkMask: 0xffffff00
        interfaces: ef1
        BroadcastAddress: 150.166.41.255
        No resource dependencies
Resource /hafs1 (type filesystem):
        volume-name: havol1
        mount-options: rw,noauto
        monitoring-level: 2
        Resource dependencies
        volume havol1
Resource havol1 (type volume):
        devname-group: sys
        devname-owner: root
```

```
        devname-mode: 666
        No resource dependencies
Failover_policy fp_h1_h2_ord_auto_auto:
        Version: 1
        Script: ordered
        Attributes: Auto_Failback Auto_Recovery
        Initial AFD: hans1 hans2
# haStatus -a
Tue Nov 30 14:45:30 PST 1999
Cluster test-cluster:
        Cluster state is ACTIVE.
Node hans2:
        State of machine is UP.
        Logical Machine Name: hans2
        Hostname: hans2.engr.sgi.com
        Is FailSafe: true
        Is CXFS: false
        Nodeid: 32418
        Reset type: powerCycle
        System Controller: msc
        System Controller status: enabled
        System Controller owner: hans1
        System Controller owner device: /dev/ttyd2
        System Controller owner type: tty
        ControlNet Ipaddr: 192.26.50.15
        ControlNet HB: true
        ControlNet Control: true
        ControlNet Priority: 1
        ControlNet Ipaddr: 150.166.41.61
        ControlNet HB: true
        ControlNet Control: false
        ControlNet Priority: 2
Node hans1:
        State of machine is UP.
        Logical Machine Name: hans1
        Hostname: hans1.engr.sgi.com
        Is FailSafe: true
        Is CXFS: false
        Nodeid: 32645
        Reset type: powerCycle
        System Controller: msc
```

```
        System Controller status: enabled
        System Controller owner: hans2
        System Controller owner device: /dev/ttyd2
        System Controller owner type: tty
        ControlNet Ipaddr: 192.26.50.14
        ControlNet HB: true
        ControlNet Control: true
        ControlNet Priority: 1
        ControlNet Ipaddr: 150.166.41.60
        ControlNet HB: true
        ControlNet Control: false
        ControlNet Priority: 2
Resource_group nfs-group1:
        State: Online
        Error: No error
        Owner: hans1
        Failover Policy: fp_h1_h2_ord_auto_auto
                Version: 1
                Script: ordered
                Attributes: Auto_Failback Auto_Recovery
                Initial AFD: hans1 hans2
        Resources:
                /hafs1  (type: NFS)
                /hafs1/nfs/statmon      (type: statd)
                150.166.41.95   (type: IP_address)
                /hafs1  (type: filesystem)
                havol1  (type: volume)
Resource /hafs1 (type NFS):
        State: Online
        Error: None
        Owner: hans1
        Flags: Resource is monitored locally
        export-info: rw,wsync
        filesystem: /hafs1
        Resource dependencies
        statd /hafs1/nfs/statmon
        filesystem /hafs1
Resource /hafs1/nfs/statmon (type statd):
        State: Online
        Error: None
        Owner: hans1
```

```
        Flags: Resource is monitored locally
        InterfaceAddress: 150.166.41.95
        Resource dependencies
        IP_address 150.166.41.95
        filesystem /hafs1
Resource 150.166.41.95 (type IP_address):
        State: Online
        Error: None
        Owner: hans1
        Flags: Resource is monitored locally
        NetworkMask: 0xffffff00
        interfaces: ef1
        BroadcastAddress: 150.166.41.255
        No resource dependencies
Resource /hafs1 (type filesystem):
        State: Online
        Error: None
        Owner: hans1
        Flags: Resource is monitored locally
        volume-name: havol1
        mount-options: rw,noauto
        monitoring-level: 2
        Resource dependencies
        volume havol1
Resource havol1 (type volume):
        State: Online
        Error: None
        Owner: hans1
        Flags: Resource is monitored locally
        devname-group: sys
        devname-owner: root
        devname-mode: 666
        No resource dependencies
# haStatus -c test-cluster
Tue Nov 30 14:42:04 PST 1999
Cluster test-cluster:
        Cluster state is ACTIVE.
Node hans2:
        State of machine is UP.
Node hans1:
        State of machine is UP.
```

```
Resource_group nfs-group1:
        State: Online
        Error: No error
        Owner: hans1
        Failover Policy: fp_h1_h2_ord_auto_auto
        Resources:
                /hafs1  (type: NFS)
                /hafs1/nfs/statmon      (type: statd)
                150.166.41.95   (type: IP_address)
                /hafs1  (type: filesystem)
                havol1  (type: volume)
```

Resource Group Failover

While a IRIS FailSafe system is running, you can move a resource group online to a particular node, or you can take a resource group offline. In addition, you can move a resource group from one node in a cluster to another node in a cluster. The following subsections describe these tasks.

Bringing a Resource Group Online

Before you bring a resource group online for the first time, you should run the diagnostic tests on that resource group. Diagnostics check system configurations and perform some validations that are not performed when you bring a resource group online.

To bring a resource group online, you specify the name of the resource and the name of the cluster which contains the node.

You cannot bring a resource group online if the resource group has no members, and you cannot bring a resource group online if the resource group is currently running in the cluster.

To bring a resource group fully online, HA services must be active. When HA services are active, an attempt is made to allocate the resource group in the cluster. However, you can also execute a command to bring the resource group online when HA services are not active. When HA services are not active, the resource group is marked to be brought online when HA services become active; the resource group is then in an `ONLINE-READY` state. Failsafe tries to bring a resource group in an `ONLINE-READY` state online when HA services are started.

You can disable resource groups from coming online when HA services are started by using the FailSafe GUI or CLI to take the resource group offline, as described in "Taking a Resource Group Offline", page 193.

Caution: Before bringing a resource group online in the cluster, you must be sure that the resource group is not running on a disabled node (where HA services are not running). Bringing a resource group online while it is running on a disabled node could cause data corruption. For information on detached resource groups, see "Taking a Resource Group Offline", page 193.

Bringing a Resource Group Online with the Cluster Manager GUI

To bring a resource group online using the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Failover Policies & Resource Groups" category.
3. On the right side of the display click on the "Bring a Resource Group Online" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.

Bringing a Resource Group Online with the Cluster Manager CLI

To bring a resource group online, use the following CLI command:

```
cmgr> admin online resource_group A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster when you use this command.

Taking a Resource Group Offline

When you take a resource group offline, FailSafe takes each resource in the resource group offline in a predefined order. If any single resource gives an error during this process, the process stops, leaving all remaining resources allocated.

You can take a FailSafe resource group offline in any of three ways:

- Take the resource group offline. This physically stops the processes for that resource group and does not reset any error conditions. If this operation fails, the resource group will be left online in an error state.
- Force the resource group offline. This physically stops the processes for that resource group but resets any error conditions. This operation cannot fail.
- Detach the resource groups. This causes FailSafe to stop monitoring the resource group, but does not physically stop the processes on that group. FailSafe will report the status as offline and will not have any control over the group. This operation should rarely fail.

If you do not need to stop the resource group and do not want FailSafe to monitor the resource group while you make changes but you would still like to have administrative control over the resource group (for instance, to move that resource group to another node), you can put the resource group in maintenance mode using the "Suspend Monitoring a Resource Group" task on the GUI or the `admin maintenance_on` command of the CLI, as described in "Stop Monitoring of a Resource Group (Maintenance Mode)", page 196.

If the FSD daemon is not running or not ready to accept client requests, executing this command disables the resource group in the configuration database only. The resource group remains online and the command fails.

Caution: Detaching a resource group leaves the resources in the resource group running at the node where it was online. After stopping HA services on that node, you should not bring the resource group online onto another node in the cluster, as this may cause data corruption.

Taking a Resource Group Offline with the Cluster Manager GUI

To take a resource group offline using the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Failover Policies & Resource Groups" category.
3. On the right side of the display click on the "Take a Resource Group Offline" task link to launch the task.

4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.

Taking a Resource Group Offline with the Cluster Manager CLI

To take a resource group offline, use the following CLI command:

```
cmgr> admin offline resource_group A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster in this command and the CLI will use the default.

To take a resource group offline with the force option in effect, use the following CLI command:

```
cmgr> admin offline_force resource_group A [in cluster B]
```

To detach a resource group, use the following CLI command:

```
cmgr> admin offline_detach resource_group A [in cluster B]
```

Moving a Resource Group

While IRIS FailSafe is active, you can move a resource group to another node in the same cluster. When you move a resource group, you specify the following:

- The name of the resource group.
- The logical name of the destination node (optional). When you do not provide a logical destination name, FailSafe chooses the destination based on the failover policy.
- The name of the cluster that contains the nodes.

Note: When you move a resource group in an active system, you may find the unexpected behavior that the command appears to have succeeded, but the resource group remains online on the same node in the cluster. This can occur if the resource group fails to start on the node to which you are moving it. In this case, FailSafe will fail over the resource group to the next node in the application failover domain, which may be the node on which the resource group was originally running. Since FailSafe kept the resource group online, the command succeeds.

Moving a Resource Group with the Cluster Manager GUI

To move a resource group using the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Failover Policies & Resource Groups" category.
3. On the right side of the display click on the "Move a Resource Group" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.

Moving a Resource Group with the Cluster Manager CLI

To move a resource group to another node, use the following CLI command:

```
cmgr> admin move resource_group A [in cluster B] [to node C]
```

Stop Monitoring of a Resource Group (Maintenance Mode)

You can temporarily stop FailSafe from monitoring a specific resource group, which puts the resource group in maintenance mode. The resource group remains on its same node in the cluster but is no longer monitored by IRIS FailSafe for resource failures.

You can put a resource group into maintenance mode if you do not want FailSafe to monitor the group for a period of time. You may want to do this for upgrade or testing purposes, or if there is any reason that IRIS FailSafe should not act on that resource group. When a resource group is in maintenance mode, it is not being monitored and it is not highly available. If the resource group's owner node fails, FailSafe will move the resource group to another node and resume monitoring.

When you put a resource group into maintenance mode, resources in the resource group are in ONLINE-MAINTENANCE state. The ONLINE-MAINTENANCE state for the resource is seen only on the node that has the resource online. All other nodes will show the resource as ONLINE. The resource group, however, should appear as being in ONLINE-MAINTENANCE state in all nodes.

Putting a Resource Group into Maintenance Mode with the Cluster Manager GUI

To put a resource group into maintenance mode using the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Failover Policies & Resource Groups" category.
3. On the right side of the display click on the "Suspend Monitoring a Resource Group" task link to launch the task.
4. Enter the selected inputs.

Resume Monitoring of a Resource Group with the Cluster Manager GUI

To resume monitoring a resource group using the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Failover Policies & Resource Groups" category.
3. On the right side of the display click on the "Resume Monitoring a Resource Group" task link to launch the task.
4. Enter the selected inputs.

Putting a Resource Group into Maintenance Mode with the Cluster Manager CLI

To put a resource group into maintenance mode, use the following CLI command:

```
cmgr> admin maintenance_on resource_group A [in cluster B]
```

If you have specified a default cluster, you do not need to specify a cluster when you use this command.

Resume Monitoring of a Resource Group with the Cluster Manager CLI

To move a resource group back online from maintenance mode, use the following CLI command:

```
cmgr> admin maintenance_off resource_group A [in cluster B]
```

Deactivating (Stopping) IRIS FailSafe

You can stop the execution of IRIS FailSafe on a systemwide basis, on all the nodes in a cluster, or on a specified node only.

Deactivating a node or a cluster is a complex operation that involves several steps and can take several minutes. Aborting a deactivate operation can leave the nodes and the resources in an intended state.

When deactivating HA services on a node or for a cluster, the operation may fail if any resource groups are not in a stable clean state. Resource groups which are in transition will cause any deactivate HA services command to fail. In many cases, the command may succeed at a later time after resource groups have settled into a stable state.

After you have successfully deactivated a node or a cluster, the node or cluster should have no resource groups and all HA services should be gone.

Serially stopping HA services on every node in a cluster is not the same as stopping HA services for the entire cluster. If the former case, an attempt is made to keep resource groups online and highly available while in the latter case resource groups are moved offline, as described in the following sections.

When you stop HA services, the FailSafe daemons perform the following actions:

1. A shutdown request is sent to FailSafe (FSD)
2. FSD releases all resource groups and puts them in ONLINE-READY state
3. All nodes in the cluster in the configuration database are disabled (one node at a time and the local node last)
4. FailSafe waits until the node is removed from the FailSafe membership before disabling the node
5. The shutdown is successful only when all nodes are not part of the FailSafe membership
6. CMOND receives notification from the configuration database when nodes are disabled
7. The local CMOND sends SIGTERM to all HA processes and IFD
8. All HA processes clean up and exit with “don’t restart” code

9. All other CMSD daemons remove the disabled node from the FailSafe membership

Deactivating HA Services on a Node

The operation of deactivating a node tries to move all resource groups from the node to some other node and then tries to disable the node in the cluster, subsequently killing all HA processes.

When HA services are stopped on a node, all resource groups owned by the node are moved to some other node in the cluster that is capable of maintaining these resource groups in a highly available state. This operation will fail if there is no node that can take over these resource groups. This condition will always occur if the last node in a cluster is shut down when you deactivate HA services on that node.

In this circumstance, you can specify the `force` option to shut down the node even if resource groups cannot be moved or released. This will normally leave resource groups allocated in a non-highly-available state on that same node. Using the `force` option might result in the node getting reset. In order to guarantee that the resource groups remain allocated on the last node in a cluster, all online resource groups should be detached.

If you wish to move resource groups offline that are owned by the node being shut down, you must do so prior to deactivating the node.

Deactivating HA Services in a Cluster

The operation of deactivating a cluster attempts to release all resource groups and disable all nodes in the cluster, subsequently killing all HA processes.

When a cluster is deactivated and the FailSafe HA services are stopped on that cluster, resource groups are moved offline or deallocated. If you want the resource groups to remain allocated, you must detach the resource groups before attempting to deactivate the cluster.

Serially stopping HA services on every node in a cluster is not the same as stopping HA services for the entire cluster. If the former case, an attempt is made to keep resource groups online and highly available while in the latter case resource groups are moved offline.

Deactivating FailSafe with the Cluster Manager GUI

To stop FailSafe services using the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.
3. On the right side of the display click on the "Stop FailSafe HA Services" task link to launch the task.
4. Enter the selected inputs.
5. Click on "OK" at the bottom of the screen to complete the task.

Deactivating FailSafe with the Cluster Manager CLI

To deactivate IRIS FailSafe in a cluster and stop FailSafe processing, use the following command:

```
cmgr> stop ha_services [on node A] [for cluster B][force]
```

Resetting Nodes

You can use FailSafe to reset nodes in a cluster. This sends a reset command to the system controller port on the specified node. When the node is reset, other nodes in the cluster will detect this and remove the node from the active cluster, reallocating any resource groups that were allocated on that node onto a backup node. The backup node used depends on how you have configured your system.

Once the node reboots, it will rejoin the cluster. Some resource groups might move back to the node, depending on how you have configured your system.

Resetting a Node with the Cluster Manager GUI

To reset a FailSafe node using the Cluster Manager GUI, perform the following steps:

1. Select FailSafe Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Nodes & Cluster" category.

3. On the right side of the display click on the "Reset a Node" task link to launch the task.
4. Enter the node to reset.
5. Click on "OK" at the bottom of the screen to complete the task.

Resetting a Node with the Cluster Manager CLI

When FailSafe is running, you can reboot a node with the following Cluster Manger CLI command:

```
cmgr> admin reset node A
```

This command uses the FailSafe daemons to reset the specified node.

You can reset a node in a cluster even when the FailSafe daemons are not running by using the `standalone` option of the `admin reset` command of the CLI:

```
cmgr> admin reset standalone node A
```

This command does not go through the FailSafe daemons.

Backing Up and Restoring Configuration With Cluster Manager CLI

The Cluster Manager CLI provides scripts that you can use to backup and restore your configuration: `cdbDump` and `cdbRestore`. These scripts are installed in the `/var/cluster/cmgr-scripts` directory. You can modify these scripts to suit your needs.

The `cdbDump` script, as provided, creates compressed tar files of the `/var/cluster/cdb/cdb.db#` directory and the `/var/cluster/cdb.db` file.

The `cdbRestore` script, as provided, restores the compressed tar files of the `/var/cluster/cdb/cdb.db#` directory and the `/var/cluster/cdb.db` file.

When you use the `cdbDump` and `cdbRestore` scripts, you should follow the following procedures:

- Run the `cdbDump` and `cdbRestore` scripts only when no administrative commands are running. This could result in an inconsistent backup.

- You must backup the configuration of each node in the cluster separately. The configuration information is different for each node, and all node-specific information is stored locally only.
- Run the backup procedure whenever you change your configuration.
- The backups of all nodes in the pool taken at the same time should be restored together.
- Cluster and FailSafe process should not be running when you restore your configuration.

Note: In addition to the above restrictions, you should not perform a `cdbDump` while information is changing in the CDB. Check SYSLOG for information to help determine when CDB activity is occurring. As a rule of thumb, you should be able to perform a `cdbDump` if at least 15 minutes have passed since the last node joined the cluster or the last administration command was run.

Log File Management

You should rotate the log files at least weekly so that your disk will not become full.

The following sections provide example scripts. You may want to consider placing an entry in the `root` crontab to run such scripts periodically.

For information about log levels, see "FailSafe System Log Configuration", page 156.

Rotating All Log Files

You can use a script such as the following to copy all files to a new location.

```
#!/bin/sh

DATE=‘/sbin/date +’%U-%a’‘
LOG_DIR="/var/cluster/ha/log"
HOST=‘/usr/bsd/hostname -s‘
LOG_FILES="cad_log cmond_log fs2d_log"
LOG_HFILES="cli cmsd crsd failsafe gcd ifd script srmd clconfd"

LOG_ARCH=$LOG_DIR"/Old-Log"
```

```
if [ ! -d $LOG_ARCH ] ; then
   mkdir $LOG_ARCH
fi

for file in $LOG_FILES
do

  rm -f ${LOG_ARCH}/${file}-${DATE}
  cp ${LOG_DIR}/${file} ${LOG_ARCH}/${file}-${DATE}
  echo "Log Rotation at ‘date‘" > ${LOG_DIR}/${file}
done

for file in $LOG_HFILES
do

  rm -f ${LOG_ARCH}/${file}_${HOST}-${DATE}
  cp ${LOG_DIR}/${file}_${HOST} ${LOG_ARCH}/${file}_${HOST}-${DATE}
  echo "Log Rotation at ‘date‘" > ${LOG_DIR}/${file}_${HOST}
done
```

The script can be executed as a cron job to regularly clean up log files. This script rotates log files when HA services are active in the FailSafe cluster. Default log levels do not create large log files.

Testing IRIS FailSafe Configuration

This chapter explains how to test the IRIS FailSafe system configuration using the Cluster Manager GUI and the Cluster Manager CLI. For general information on using the Cluster Manager GUI and the Cluster Manager CLI, see Chapter 4, "IRIS FailSafe Administration Tools".

The sections in this chapter are as follows:

Overview of FailSafe Diagnostic Commands

Table 8-1 shows the tests you can perform with IRIS FailSafe diagnostic commands:

Table 8-1 FailSafe Diagnostic Test Summary

Diagnostic Test	Checks Performed
resource	Checks that the resource type parameters are set Check that the parameters are syntactically correct Validates that the parameters exist
resource group	Tests all resources defined in the resource group
failover policy	Checks that the failover policy exists Checks that the failover domain contains a valid list of hosts
network connectivity	Checks that the control interfaces are on the same network Checks that the nodes can communicate with each other
serial connection	Checks that the nodes can reset each other

All transactions are logged to the diagnostics file `diags_`*nodename* in the log directory.

You should test resource groups before starting FailSafe HA services or starting a resource group. These tests are designed to check for resource inconsistencies which could prevent the resource group from starting successfully.

Performing Diagnostic Tasks with the Cluster Manager GUI

To test the components of a FailSafe system using the Cluster Manager GUI, perform the following steps:

1. Select Task Manager on the FailSafe Toolchest.
2. On the left side of the display, click on the "Diagnostics" category.
3. Select one of the diagnostics tasks that appear on the right side of the display: "Test Connectivity," "Test Resources," or "Test Failover Policy."

Testing Connectivity with the Cluster Manager GUI

When you select the "Test Connectivity" task from the Diagnostics display, you can test the network and serial connections on the nodes in your cluster by entering the requested inputs. You can test all of the nodes in the cluster at one time, or you can specify an individual node to test.

Testing Resources with the Cluster Manager GUI

When you select the "Test Resources" task from the Diagnostics display, you can test the resources on the nodes in your cluster by entering the requested inputs. You can test resources by type and by group. You can test the resources of a resource type or in a resource group on all of the nodes in the cluster at one time, or you can specify an individual node to test. Resource tests are performed only on nodes in the resource group's application failover domain.

Testing Failover Policies with the Cluster Manager GUI

When you select the "Test Failover Policy" task from the Diagnostics display, you can test whether a failover policy is defined correctly. This test checks the failover policy by validating the policy script, failover attributes, and whether the application failover domain consists of valid nodes from the cluster.

Performing Diagnostic Tasks with the Cluster Manager CLI

The following subsections described how to perform diagnostic tasks on your system using the Cluster Manager CLI commands.

Testing the Serial Connections with the Cluster Manager CLI

You can use the Cluster Manager CLI to test the serial connections between the IRIS FailSafe nodes. This test pings each specified node through the serial line and produces an error message if the ping is not successful. Do not execute this command while FailSafe is running.

When you are using the Cluster Manager CLI, use the following command to test the serial connections for the machines in a cluster

cmgr> **test serial in cluster** *A* [**on node** *B* **node** *C ...*]

This test yields an error message when it encounters its first error, indicating the node that did not respond. If you receive an error message after executing this test, verify the cable connections of the serial cable from the indicated node's serial port to the remote power control unit or the system controller port of the other nodes and run the test again.

The following shows an example of the `test serial` CLI command:

```
# cluster_mgr
Welcome to IRIS FailSafe Cluster Manager Command-Line Interface

cmgr> test serial in cluster eagan on node cm1
Success: testing serial...
Success: Ensuring Node Can Get IP Addresses For All Specified Hosts
Success: Number of IP addresses obtained for <cm1> = 1
Success:        The first IP address for <cm1> = 128.162.19.34
Success: Checking serial lines via crsd (crsd is running)
Success: Successfully checked serial line
Success: Serial Line OK
Success: overall exit status:success, tests failed:0, total tests executed:1
```

The following shows an example of an attempt to run the `test serial` CLI command while FailSafe is running (causing the command to fail to execute):

```
cmgr> test serial in cluster eagan on node cm1
Error: Cannot run the serial tests, diagnostics has detected FailSafe (ha_cmsd) is running
```

```
Failed to execute FailSafe tests/diagnostics ha

test command failed
cmgr>
```

Testing Network Connectivity with the Cluster Manager CLI

You can use the Cluster Manager CLI to test the network connectivity in a cluster. This test checks if the specified nodes can communicate with each other through each configured interface in the nodes. This test will not run if FailSafe is running.

When you are using the Cluster Manager CLI, use the following command to test the network connectivity for the machines in a cluster

cmgr> **test connectivity in cluster** *A* [**on node** *B* **node** *C* ...]

The following shows an example of the `test connectivity` CLI command:

```
cmgr> test connectivity in cluster eagan on node cm1
Success: testing connectivity...
Success: checking that the control IP_addresses are on the same networks
Success: pinging address cm1-priv interface ef0 from host cm1
Success: pinging address cm1 interface ef1 from host cm1
Success: overall exit status:success, tests failed:0, total tests
executed:1
```

This test yields an error message when it encounters its first error, indicating the node that did not respond. If you receive an error message after executing this test, verify that the network interface has been configured up, using the `ifconfig` command, for example:

/usr/etc/ifconfig ec3

```
ec3: flags=c63<UP,BROADCAST,NOTRAILERS,RUNNING,FILTMULTI,MULTICAST>
        inet 190.0.3.1 netmask 0xffffff00 broadcast 190.0.3.255
```

The UP in the first line of output indicates that the interface is configured up.

If the network interface is configured up, verify that the network cables are connected properly and run the test again.

Testing Resources with the Cluster Manager CLI

You can use the Cluster Manager CLI to test any configured resource by resource name or by resource type.

The Cluster Manager CLI uses the following syntax to test a resource by name:

```
cmgr> test resource A of resource_type B in cluster C [on node D node E ...]
```

The following shows an example of testing a resource by name:

```
cmgr> test resource /disk1 of resource_type filesystem in cluster eagan on machine cm1
Success: *** testing node resources on node cm1 ***
Success: *** testing all filesystem resources on node cm1 ***
Success: testing resource /disk1 of resource type filesystem on node cm1
Success: overall exit status:success, tests failed:0, total tests executed:1
```

The Cluster Manager CLI uses the following syntax to test a resource by resource type:

```
cmgr> test resource_type A in cluster B [on node C node D...]
```

The following shows an example of testing resources by resource type:

```
cmgr> test resource_type filesystem in cluster eagan on machine cm1
Success: *** testing node resources on node cm1 ***
Success: *** testing all filesystem resources on node cm1 ***
Success: testing resource /disk4 of resource type filesystem on node cm1
Success: testing resource /disk5 of resource type filesystem on node cm1
Success: testing resource /disk2 of resource type filesystem on node cm1
Success: testing resource /disk3 of resource type filesystem on node cm1
Success: testing resource /disk1 of resource type filesystem on node cm1
Success: overall exit status:success, tests failed:0, total tests executed:5
```

You can use the CLI to test volume and filesystem resources in `destructive` mode. This provides a more thorough test of filesystems and volumes. CLI tests will not run in destructive mode if FailSafe is running.

The Cluster Manager CLI uses the following syntax for the commands that test resources in destructive mode:

```
cmgr> test resource A of resource_type B in cluster C [on node D node C ...] destructive
```

The following sections describe the diagnostic tests available for resources.

Testing Logical Volumes

You can use the Cluster Manager CLI to test the logical volumes in a cluster. This test checks if the specified volume is configured correctly.

When you are using the Cluster Manager CLI, use the following command to test a logical volume:

```
cmgr> test resource A of resource_type volume on cluster B [on node C node D ...]
```

The following example tests a logical volume:

```
cmgr> test resource alternate of resource_type volume on cluster eagan
Success: *** testing node resources on node cm1 ***
Success: *** testing all volume resources on node cm1 ***
Success: running resource type volume tests on node cm1
Success: *** testing node resources on node cm2 ***
Success: *** testing all volume resources on node cm2 ***
Success: running resource type volume tests on node cm2
Success: overall exit status:success, tests failed:0, total tests executed:2
cmgr>
```

The following example tests a logical volume in destructive mode:

```
cmgr> test resource alternate of resource_type volume on cluster eagan destructive
Warning: executing the tests in destructive mode
Success: *** testing node resources on node cm1 ***
Success: *** testing all volume resources on node cm1 ***
Success: running resource type volume tests on node cm1
Success: successfully assembled volume: alternate
Success: *** testing node resources on node cm2 ***
Success: *** testing all volume resources on node cm2 ***
Success: running resource type volume tests on node cm2
Success: successfully assembled volume: alternate
Success: overall exit status:success, tests failed:0, total tests executed:2
cmgr>
```

Testing Filesystems

You can use the Cluster Manager CLI to test the filesystems configured in a cluster. This test checks if the specified filesystem is configured correctly and, in addition, checks whether the volume the filesystem will reside on is configured correctly.

When you are using the Cluster Manager CLI, use the following command to test a filesystem:

```
cmgr> test resource A of resource_type filesystems on cluster B [on node C node D ...]
```

The following example tests a filesystem. This example first uses a CLI `show` command to display the filesystems that have been defined in a cluster.

```
cmgr> show resources of resource_type filesystem in cluster eagan
/disk4 type filesystem
/disk5 type filesystem
/disk2 type filesystem
/disk3 type filesystem
/disk1 type filesystem
cmgr> test resource /disk4 of resource_type filesystem in cluster eagan on node cm1
Success: *** testing node resources on node cm1 ***
Success: *** testing all filesystem resources on node cm1 ***
Success: successfully mounted filesystem: /disk4
Success: overall exit status:success, tests failed:0, total tests executed:1
cmgr>
```

The following example tests a filesystem in destructive mode:

```
cmgr> test resource /disk4 of resource_type filesystem in cluster eagan on node cm1
destructive
Warning: executing the tests in destructive mode
Success: *** testing node resources on node cm1 ***
Success: *** testing all filesystem resources on node cm1 ***
Success: successfully mounted filesystem: /disk4
Success: overall exit status:success, tests failed:0, total tests executed:1
cmgr>
```

Testing NFS Filesystems

You can use the Cluster Manager CLI to test the NFS filesystems configured in a cluster. This test checks if the specified NFS filesystem is configured correctly and, in addition, checks whether the volume the NFS filesystem will reside on is configured correctly.

When you are using the Cluster Manager CLI, use the following command to test an NFS filesystem:

```
cmgr> test resource A of resource_type NFS on cluster B [on node C node D ...]
```

The following example tests an NFS filesystem:

```
cmgr> test resource /disk4 of resource_type NFS in cluster eagan
Success: *** testing node resources on node cm1 ***
Success: *** testing all NFS resources on node cm1 ***
Success: *** testing node resources on node cm2 ***
Success: *** testing all NFS resources on node cm2 ***
Success: overall exit status:success, tests failed:0, total tests executed:2
cmgr>
```

Testing statd Resources

You can use the Cluster Manager CLI to test the statd resources configured in a cluster. When you are using the Cluster Manager CLI, use the following command to test an NFS filesystem:

```
cmgr> test resource A of resource_type statd on cluster B [on node C node D ...]
```

The following example tests a statd resource:

```
cmgr> test resource /disk1/statmon of resource_type statd in cluster eagan
Success: *** testing node resources on node cm1 ***
Success: *** testing all statd resources on node cm1 ***
Success: *** testing node resources on node cm2 ***
Success: *** testing all statd resources on node cm2 ***
Success: overall exit status:success, tests failed:0, total tests executed:2
cmgr>
```

Testing Netscape-web Resources

You can use the Cluster Manager CLI to test the Netscape Web resources configured in a cluster.

When you are using the Cluster Manager CLI, use the following command to test a Netscape-web resource:

```
cmgr> test resource A of resource_type Netscape_web on cluster B [on node C node D ...]
```

The following example tests a Netscape-web resource. In this example, the Netscape-web resource on node cm2 failed the diagnostic test.

```
cmgr> test resource nss-enterprise of resource_type Netscape_web in cluster eagan
Success: *** testing node resources on node cm1 ***
Success: *** testing all Netscape_web resources on node cm1 ***
```

```
Success: *** testing node resources on node cm2 ***
Success: *** testing all Netscape_web resources on node cm2 ***
Warning: resource nss-enterprise has invalid script /var/netscape/suitespot/https-ha85 location
Warning: /var/netscape/suitespot/https-ha85/config/magnus.conf must contain the
"Port" parameter
Warning: /var/netscape/suitespot/https-ha85/config/magnus.conf must contain the
"Address" parameter
Warning: resource nss-enterprise of type Netscape_web failed
Success: overall exit status:failed, tests failed:1, total tests executed:2
Failed to execute FailSafe tests/diagnostics ha
test command failed
cmgr>
```

Testing Resource Groups

You can use the Cluster Manager CLI to test a resource group. This test cycles through the resource tests for all of the resources defined for a resource group. Resource tests are performed only on nodes in the resource group's application failover domain.

The Cluster Manager CLI uses the following syntax for the commands that test resource groups:

```
cmgr> test resource_group A in cluster B [on node C node D ...]
```

The following example tests a resource group. This example first uses a CLI `show` command to display the resource groups that have been defined in a cluster.

```
cmgr> show resource_groups in cluster eagan
Resource Groups:
        nfs2
        informix
cmgr> test resource_group nfs2 in cluster eagan on machine cm1
Success: *** testing node resources on node cm1 ***
Success: testing resource /disk4 of resource type NFS on node cm1
Success: testing resource /disk3 of resource type NFS on node cm1
Success: testing resource /disk3/statmon of resource type statd on node cm1
Success: testing resource 128.162.19.45 of resource type IP_address on node cm1
Success: testing resource /disk4 of resource type filesystem on node cm1
Success: testing resource /disk3 of resource type filesystem on node cm1
Success: testing resource dmf1 of resource type volume on node cm1
Success: testing resource dmfjournals of resource type volume on node cm1
Success: overall exit status:success, tests failed:0, total tests executed:16
cmgr>
```

Testing Failover Policies with the Cluster Manager CLI

You can use the Cluster Manager CLI to test whether a failover policy is defined correctly. This test checks the failover policy by validating the policy script, failover attributes, and whether the application failover domain consists of valid nodes from the cluster.

The Cluster Manager CLI uses the following syntax for the commands that test a failover policy:

cmgr> **test failover_policy** *A* **in cluster** *B* [**on node** *C* **node** *D* ...]

The following example tests a failover policy. This example first uses a CLI `show` command to display the failover policies that have been defined in a cluster.

```
cmgr> show failover_policies
Failover Policies:
        reverse
        ordered-in-order
cmgr> test failover_policy reverse in cluster eagan
Success: *** testing node resources on node cm1 ***
Success: testing policy reverse on node cm1
Success: *** testing node resources on node cm2 ***
Success: testing policy reverse on node cm2
Success: overall exit status:success, tests failed:0, total tests executed:2
cmgr>
```

IRIS FailSafe Recovery

This chapter provides information on FailSafe system recovery, and includes sections on the following topics:

Overview of FailSafe System Recovery

When a FailSafe system experiences problems, you can use some of the FailSafe features and commands to determine where the problem is.

FailSafe provides the following tools to evaluate and recover from system failure:

- Log files
- Commands to monitor status of system components
- Commands to start, stop, and fail over highly available services

Keep in mind that the FailSafe logs may not detect system problems that do not translate into FailSafe problems. For example, if a CPU goes bad, or hardware maintenance is required, FailSafe may not be able to detect and log these failures.

In general, when evaluating system problems of any nature on a FailSafe configuration, you should determine whether you need to shut down a node to address those problems. When you shut down a node, perform the following steps:

1. Stop FailSafe services on that node

2. Shut down the node to perform needed maintenance and repair
3. Start up the node
4. Start FailSafe services on that node

It is important that you explicitly stop FailSafe services before shutting down a node, where possible, so that FailSafe does not interpret the node shutdown as node failure. If FailSafe interprets the service interruption as node failure, there could be unexpected ramifications, depending on how you have configured your resource groups and your application failover domain.

When you shut down a node to perform maintenance, you may need to change your FailSafe configuration to keep your system running.

FailSafe Log Files

IRIS FailSafe maintains system logs for each of the FailSafe daemons. You can customize the system logs according to the level of logging you wish to maintain.

For information on setting logging for CAD, CMOND, and CDBD/fs2d, see "Configuring System Files", page 58. For information on setting up log configurations, see "FailSafe System Log Configuration", page 156 in Chapter 5, "IRIS FailSafe Configuration".

Log messages can be of the following types:

Normal

Normal messages report on the successful completion of a task. An example of a normal message is as follows:

```
Wed Sep 2 11:57:25.284
<N ha_gcd cms 10185:0>
Delivering TOTAL membership (S# 1, GS# 1)
```

The `<N` notation indicates a normal message.

Error/Warning

Error or warning messages indicate that an error has occurred or may occur soon. These messages may result from using the wrong command or improper syntax. An example of a warning message is as follows:

```
Wed Sep 2 13:45:47.199
<W crsd crs 9908:0 crs_config.c:634>
CI_ERR_NOTFOUND, safer - no such node
```

	The <W notation indications a warning. <E indicates an error.
Syslog Messages	All normal and error messages are also logged to `syslog`. Syslog messages include the symbol <CI> in the header to indicate they are cluster-related messages. An example of a syslog message is as follows: `Wed Sep 2 12:22:57 6X:safe syslog: <<CI> ha_cmsd misc 10435:0> CI_FAILURE, I am not part of the enabled cluster anymore`
Debug	Debug messages appear in the log group file when the logging level is set to debug0 or higher (using the GUI) or 10 or higher (using the CLI). **Note:** Many megabytes of disk space can be consumed on the server when debug levels are used in a log configuration.

Examining the log files should enable you to see the nature of the system error. Noting the time of the error and looking at the log files to note the activity of the various daemons immediately before error occurred, you may be able to determine what situation existed that caused the failure.

FailSafe Membership and Resets

In looking over the actions of a FailSafe system on failure to determine what has gone wrong and how processes have transferred, it is important to consider the concept of FailSafe membership. When failover occurs, the runtime failover domain can include only those nodes that are in the FailSafe membership.

FailSafe Membership and Tie-Breaker Node

Nodes can enter into the FailSafe membership only when they are not disabled and they are in a known state. This ensures that data integrity is maintained because only nodes within the FailSafe membership can access the shared storage. If nodes outside the membership and not controlled by FailSafe were able to access the shared storage, two nodes might try to access the same data at the same time, a situation that would

result in data corruption. For this reason, disabled nodes do not participate in the membership computation. Note that no attempt is made to reset nodes that are configured disabled before confirming the FailSafe membership.

FailSafe membership in a cluster is based on a quorum majority. For a cluster to be enabled, more than 50% of the nodes in the cluster must be in a known state, able to talk to each other, using heartbeat control networks. This quorum determines which nodes are part of the FailSafe membership that is formed.

If there are an even number of nodes in the cluster, it is possible that there will be no majority quorum; there could be two sets of nodes, each consisting of 50% of the total number of node, unable to communicate with the other set of nodes. In this case, FailSafe uses the node that has been configured as the tie-breaker node when you configured your FailSafe parameters. If no tie-breaker node was configured, FailSafe uses the enabled node with the lowest node id number.

For information on setting tie-breaker nodes, see "IRIS FailSafe HA Parameters", page 107 in Chapter 5, "IRIS FailSafe Configuration".

The nodes in a quorum attempt to reset the nodes that are not in the quorum. Nodes that can be reset are declared DOWN in the membership, nodes that could not be reset are declared UNKNOWN. Nodes in the quorum are UP.

If a new majority quorum is computed, a new membership is declared whether any node could be reset or not.

If at least one node in the current quorum has a current membership, the nodes will proceed to declare a new membership if they can reset at least one node.

If all nodes in the new tied quorum are coming up for the first time, they will try to reset and proceed with a new membership only if the quorum includes the tie-breaker node.

If a tied subset of nodes in the cluster had no previous membership, then the subset of nodes in the cluster with the tie-breaker node attempts to reset nodes in the other subset of nodes in the cluster. If at least one node reset succeeds, a new membership is confirmed.

If a tied subset of nodes in the cluster had previous membership, the nodes in one subset of nodes in the cluster attempt to reset nodes in the other subset of nodes in the cluster. If at least one node reset succeeds, a new membership is confirmed. The subset of nodes in the cluster with the tie-breaker node resets immediately, the other subset of nodes in the cluster attempts to reset after some time.

Resets are done through system controllers connected to tty ports through serial lines. Periodic serial line monitoring never stops. If the estimated serial line monitoring failure interval and the estimated heartbeat loss interval overlap, we suspect a power failure at the node being reset.

No Membership Formed

When no FailSafe membership is formed, you should check the following areas for possible problems:

- Is the FailSafe membership daemon, `ha_cmsd` running? Is the database daemon, `fs2d`, running?
- Can the nodes communicate with each other?
  - Are the control networks configured as heartbeat networks?
- Can the control network addresses be pinged from peer nodes?
- Are the quorum majority or tie rules satisfied?

  Look at the `cmsd` log to determine membership status.
- If a reset is required, are the following conditions met?
  - Is the node control daemon, `crsd`, up and running?
  - Is the reset serial line in good health?

    You can look at the `crsd` log for the node you are concerned with, or execute an `admin ping` and `admin reset` command on the node to check this.

Status Monitoring

FailSafe allows you to monitor and check the status of specified clusters, nodes, resources, and resource groups. You can use this feature to isolate where your system is encountering problems.

With the FailSafe Cluster Manager GUI Cluster View, you can monitor the status of the FailSafe components continuously through their visual representation. Using the FailSafe Cluster Manger CLI, you can display the status of the individual components by using the `show` command.

For information on status monitoring and on the meaning of the states of the FailSafe components, see "System Status", page 177 of Chapter 7, "IRIS FailSafe System Operation".

Dynamic Control of FailSafe Services

FailSafe allows you to perform a variety of administrative tasks that can help you troubleshoot a system with problems without bringing down the entire system. These tasks include the following:

- You can add or delete nodes from a cluster without affecting the FailSafe services and the applications running in the cluster
- You can add or delete a resource group without affecting other online resource groups
- You can add or delete resources from a resource group while it is still online
- You can change FailSafe parameters such as the heartbeat interval and the node timeout and have those values take immediate affect while the services are up and running
- You can start and stop FailSafe services on specified nodes
- You can move a resource group online, or take it offline
- You can stop the monitoring of a resource group by putting the resource group into maintenance mode. This is not an expensive operation, as it does not stop and start the resource group, it just puts the resource group in a state where it is not available to FailSafe.
- You can reset individual nodes

For information on how to perform these tasks, see Chapter 5, "IRIS FailSafe Configuration" and Chapter 7, "IRIS FailSafe System Operation".

Recovery Procedures

The following sections describe various recovery procedures you can perform when different failsafe components fail. Procedures for the following situations are provided:

Cluster Error Recovery

Follow this procedure if status of the cluster is UNKNOWN in all nodes in the cluster.

1. Check to see if there are control networks that have failed (see "Control Network Failure Recovery", page 226).
2. At least 50% of the nodes in the cluster must be able to communicate with each other to have an active cluster (Quorum requirement). If there are not sufficient nodes in the cluster that can communicate with each other using control networks, stop HA services on some of the nodes so that the quorum requirement is satisfied.
3. If there are no hardware configuration problems, detach all resource groups that are online in the cluster (if any), stop HA services in the cluster, and restart HA services in the cluster.

The following `cluster_mgr` command detaches the resource group `web-rg` in cluster `web-cluster`:

```
cmgr> admin detach resource_group web-rg in cluster web-cluster
```

To stop HA services in the cluster `web-cluster` and ignore errors (`force` option), use the following `cluster_mgr` command:

```
cmgr> stop ha_services for cluster web-cluster force
```

To start HA services in the cluster `web-cluster`, use the following `cluster_mgr` command:

```
cmgr> start ha_services for cluster web-cluster
```

Node Error recovery

When a node is not able to talk to the majority of nodes in the cluster, the SYSLOG will display a message that the CMSD is in a lonely state. Another problem you may see is that a node is getting reset or going to an unknown state.

Follow this procedure to resolve node errors:

1. Check to see if the control networks in the node are working (see "Control Network Failure Recovery", page 226).
2. Check to see if the serial reset cables to reset the node are working (see "Serial Cable Failure Recovery", page 227).
3. Verify that the sgi-cmsd port is the same in all nodes in the cluster.
4. Check the node configuration; it should be consistent and correct.
5. Check `syslog` and `cmsd` logs for errors. If a node is not joining the cluster, check the logs of the nodes that are part of the cluster.
6. If there are no hardware configuration problems, stop HA services in the node and restart HA services.

   To stop HA services in the node `web-node3` in the cluster `web-cluster`, ignoring errors (`force` option), use the following `cluster_mgr` command

   ```
   cmgr> stop ha_services in node web-node3 for cluster web-cluster
   force
   ```

   To start HA services in the node `web-node3` in the cluster `web-cluster`, use the following `cluster_mgr` command:

   ```
   cmgr> start ha_services in node web-node3 for cluster web-cluster
   ```

Resource Group Maintenance and Error Recovery

To do simple maintenance on an application that is part of the resource group, use the following procedure. This procedure stops monitoring the resources in the resource group when maintenance mode is on. You need to turn maintenance mode off when application maintenance is done.

Caution: If there is node failure on the node where resource group maintenance is being performed, the resource group is moved to another node in the failover policy domain.

1. To put a resource group `web-rg` in maintenance mode, use the following `cluster_mgr` command:

   ```
   cmgr> admin maintenance_on resource_group web-rg in cluster
   web-cluster
   ```

2. The resource group state changes to ONLINE_MAINTENANCE. Do whatever application maintenance is required. (Rotating application logs is an example of simple application maintenance).

3. To remove a resource group `web-rg` from maintenance mode, use the following `cluster_mgr` command:

   ```
   cmgr> admin maintenance_off resource_group web-rg in cluster
   web-cluster
   ```

   The resource group state changes back to ONLINE.

You perform the following procedure when a resource group is in an ONLINE state and has an SRMD EXECUTABLE ERROR.

1. Look at the SRM logs (default location: `/var/cluster/ha/logs/srmd_nodename`) to determine the cause of failure and the resource that has failed. Search for the `ERROR` string in the SRMD log file:

   ```
   Wed Nov 3 04:20:10.135
   <E ha_srmd srm 12127:1 sa_process_tasks.c:627>
   CI_FAILURE, ERROR: Action (start) for resource (192.0.2.45) of type
   (IP_address) failed with status (failed)
   ```

2. Check the script logs on that same node for `IP_address` start script errors.

3. Fix the cause of failure. This might require changes to resource configuration or changes to resource type stop/start/failover action timeouts.

4. After fixing the problem, move the resource group offline with the `force` option and then move the resource group online in the cluster.

The following `cluster_mgr` command moves the resource group `web-rg` in the cluster `web-cluster` offline and ignores any errors:

```
cmgr> admin offline resource_group web-rg in cluster web-cluster
force
```

The following `cluster_mgr` command moves the resource group `web-rg` in the cluster `web-cluster` online:

```
cmgr> admin online resource_group web-rg in cluster web-cluster
```

The resource group `web-rg` should be in an ONLINE state with no error.

You use the following procedure when a resource group is not online but is in an error state. Most of these errors occur as a result of the exclusivity process. This process, run when a resource group is brought online, determines if any resources are already allocated somewhere in the failure domain of a resource group. Note that exclusivity scripts return that a resource is allocated on a node if the script fails in any way. In other words, unless the script can determine that a resource is not present, it returns a value indicating that the resource is allocated.

Some possible error states include: SPLIT RESOURCE GROUP (EXCLUSIVITY), NODE NOT AVAILABLE (EXCLUSIVITY), NO AVAILABLE NODES in failure domain. See "Resource Group Status", page 180 in Chapter 7, "IRIS FailSafe System Operation" for explanations of resource group error codes.

1. Look at the `failsafe` and SRMD logs (default directory: `/var/cluster/ha/logs`, files: `failsafe`_*nodename*, `srmd`_*nodename*) to determine the cause of the failure and the resource that failed.

   For example, say the task of moving a resource group online results in a resource group with error state SPLIT RESOURCE GROUP (EXCLUSIVITY). This means that parts of a resource group are allocated on at least two different nodes. One of the failsafe logs will have the description of which nodes are believed to have the resource group partially allocated:

```
[Resource Group:name]:Exclusivity failed -- RUNNING on nodename and nodename
```

```
[Resource Group:name]:Exclusivity failed -- PARTIALLY RUNNING on nodename and
PARTIALLY RUNNING on nodename
```

   At this point, look at the `srmd` logs on each of these nodes for exclusive script errors to see what resources are believed to be allocated. In some cases, a

misconfigured resource will show up as a resource which is allocated. This is especially true for `Netscape_web` resources.

2. Fix the cause of the failure. This might require changes to resource configuration or changes to resource type start/stop/exclusivity timeouts.
3. After fixing the problem, move the resource group offline with the `force` option and then move the resource group online.

Perform the following checks when a resource group shows a "no more nodes in AFD" error:

1. All nodes in the AFD are not in the membership. Check CMSD logs for errors.
2. Check the SRMC/script logs on all nodes in AFD for start/monitor script errors.

There are a few double failures that can occur in the cluster which will cause resource groups to remain in a non-highly-available state. At times a resource group might get stuck in an offline state. A resource group might also stay in an error state on a node even when a new node joins the cluster and the resource group can migrate to that node to clear the error.

When these circumstances arise, the correct action should be as follows:

1. Try to move the resource group online if it is offline.
2. If the resource group is stuck on a node, detach the resource group, then bring it online again. This should clear many errors.
3. If detaching the resource group does not work, force the resource group offline, then bring it back online.
4. If commands appear to be hanging or not working properly, detach all resource groups, then shut down the cluster and bring all resource groups back online.

See "Taking a Resource Group Offline", page 193 for information on detaching resource groups and forcing resource groups offline.

Resource Error Recovery

You use this procedure when a resource that is not part of a resource group is in an ONLINE state with error. This can happen when the addition or removal of resources from a resource group fails.

1. Look at the SRM logs (default location: `/var/cluster/ha/logs/srmd_`*nodename*) to determine the cause of failure and the resource that has failed.
2. Fix the cause of failure. This might require changes to resource configuration or changes to resource type stop/start/failover action timeouts.
3. After fixing the problem, move the resource offline with the `force` option of the Cluster Manager CLI `admin offline` command:

   ```
   cmgr> admin offline_force resource web-srvr of resource_type
   Netscape_Web in cluster web-cluster
   ```

   Executing this command removes the error state of resource `web-srvr` of type `Netscape_Web`, making it available to be added to a resource group.

   You can also use the Cluster Manager GUI to clear the error state for the resource. To do this, you select the "Recover a Resource" task from the "Resources and Resource Types" category of the FailSafe Manager.

Control Network Failure Recovery

Control network failures are reported in `cmsd` logs. The default location of `cmsd` log is `/var/cluster/ha/logs/cmsd_`*node name*. Follow this procedure when the control network fails:

1. Use the `ping`(1M) command to check whether the control network IP address is configured in the node.
2. Check node configuration to see whether the control network IP addresses are correctly specified.

   The following `cluster_mgr` command displays node configuration for `web-node3:`

   ```
   cmgr> show node web-node3
   ```

3. If IP names are specified for control networks instead of IP addresses in XX.XX.XX.XX notation, check to see whether IP names can be resolved using DNS. It is recommended that IP addresses are used instead of IP names.
4. Check whether the heartbeat interval and node timeouts are correctly set for the cluster. These HA parameters can seen using `cluster_mgr show ha_parameters` command.

Serial Cable Failure Recovery

Serial cables are used for resetting a node when there is a node failure. Serial cable failures are reported in `crsd` logs. The default location for the `crsd` log is `/var/cluster/ha/log/crsd_`*nodename*.

1. Check the node configuration to see whether serial cable connection is correctly configured.

   The following `cluster_mgr` command displays node configuration for `web-node3`

   ```
   cmgr> show node web-node3
   ```

   Use the `cluster_mgr admin ping` command to verify the serial cables.

   ```
   cmgr> admin ping node web-node3
   ```

The above command reports serial cables problems in node `web-node3`.

CDB Sync Failure

If the CDB sync is failing, follow this procedure:

1. Check for the following message in SYSLOG on the target node:

   ```
   Starting to receive CDB sync series from machine <node1's node id>
   ...
   Finished receiving CDB sync series from machine <node1's node id>
   ```

2. Check for control network or portmapper/rpcbind problems.
3. Check the node definition in the CDB.
4. Check the SYSLOG and CDBD/fs2d logs on the source node.

CDB Maintenance and Recovery

When the entire configuration database (CDB) must be reinitialized, execute the following command:

```
# /usr/cluster/bin/cdbreinit /var/cluster/cdb/cdb.db
```

This command will restart all cluster processes. The contents of the configuration database will be automatically synchronized with other nodes if other nodes in the pool are available.

Otherwise, the CDB will need to be restored from backup at this point. For instructions on backing up and restoring the CDB, see "Backing Up and Restoring Configuration With Cluster Manager CLI", page 201 in Chapter 7, "IRIS FailSafe System Operation".

IRIS FailSafe Cluster Manager GUI and CLI Inconsistencies

If the FailSafe Cluster Manager GUI is displaying information that is inconsistent with the FailSafe `cluster_mgr` command, restart cad process on the node to which Cluster Manager GUI is connected to by executing the following command:

```
# killall cad
```

The cluster administration daemon is restarted automatically by the `cmond` process.

GUI does not Report Information

If the FailSafe Cluster Manager GUI is not reporting configuration information and status, perform the following checks:

1. Check the information using the Cluster Manager CLI. If `cmgr` is reporting correct information, there is a GUI update problem.
2. If there is a GUI update problem, kill CAD on that node. Wait for a couple of minutes to see whether CAD gets correct information. Check the CAD logs on that node for errors.
3. If the problem is not a GUI update problem, check the CLI logs on that node for errors.
4. If the status information is incorrect, check the CMSD or FSD logs on that node.

Using the cdbreinit Command

When the configuration databases are not in sync on all the nodes in the cluster, you can run the `cdbreinit` command to recover. The `cdbreinit` command should be run on the node which is not in sync.

Perform the following steps.

Note: Perform each step on all the nodes before proceeding to the next step in the recovery procedure.

1. Stop FailSafe services in the cluster using the GUI or `cluster_mgr`.
2. Stop cluster processes on all nodes in the pool:

   ```
   /etc/init.d/cluster stop
   killall fs2d
   ```

3. Run `cdbreinit` on the node where the CDB is not in sync.
4. Start cluster processes on all nodes in the pool:

   ```
   /etc/init.d/cluster start
   ```

5. Wait a couple of minutes for the CDB to sync. There will be CDB sync long messages in the `SYSLOG` on the node.
6. Start FailSafe services in the cluster.

Upgrading and Maintaining Active Clusters

When a IRIS FailSafe system is running, you may need to perform various administration procedures without shutting down the entire cluster. This chapter provides instructions for performing upgrade and maintenance procedures on active clusters. It includes the following procedures:

Adding a Node to an Active Cluster

Use the following procedure to add a node to an active cluster. This procedure begins with the assumption that `cluster_admin`, `cluster_control`, `cluster_ha`, and `failsafe2` products are already installed in this node.

1. Check control network connections from the node to the rest of the cluster using `ping`(1M) command. Note the list of control network IP addresses.
2. Check the serial connections to reset this node. Note the name of the node that can reset this node.
3. Run node diagnostics. For information on FailSafe diagnostic commands, see Chapter 8, "Testing IRIS FailSafe Configuration".
4. Make sure `sgi-cad`, `sgi-crsd`, `sgi-cmsd`, and `sgi-gcd` entries are present in the `/etc/services` file. The port numbers for these processes should match the port numbers in other nodes in the cluster.

   Example entries:

```
sgi-cad         7200/tcp      # SGI cluster admin daemon
sgi-crsd        7500/udp      # SGI cluster reset services daemon
sgi-cmsd        7000/udp      # SGI FailSafe membership Daemon
sgi-gcd         8000/udp      # SGI group communication Daemon
```

5. Check if cluster processes (`cad`, `cmond`, `crsd`) are running.

   # **ps -ef | grep cad**

   If cluster processes are not running, run the `cdbreinit` command.

```
# /usr/cluster/bin/cdbreinit /var/cluster/cdb/cdb.db
 Killing fs2d...
 Removing database header file /var/cluster/cdb/cdb.db...
 Preparing to delete database directory /var/cluster/cdb/cdb.db# !!
 Continue[y/n]y
 Removing database directory /var/cluster/cdb/cdb.db#...
 Deleted CDB database at /var/cluster/cdb/cdb.db
 Recreating new CDB database at /var/cluster/cdb/cdb.db with cdb-exitop...
  fs2d
  Created standard CDB database in /var/cluster/cdb/cdb.db

  Please make sure that ‘‘sgi-cad’’ service is added to /etc/services file
  If not, add the entry and restart cluster processes.
  Please refer to IRIS FailSafe administration manual for more
  information.

  Modifying CDB database at /var/cluster/cdb/cdb.db with cluster_ha-exitop...
  Modified standard CDB database in /var/cluster/cdb/cdb.db

  Please make sure that ‘‘sgi-cmsd’’ and ‘‘sgi-gcd’’ services are added
  to /etc/services file before starting HA services.
  Please refer to IRIS FailSafe administration manual for more
  information.

  Starting cluster control processes with cluster_control-exitop...

  Please make sure that ‘‘sgi-crsd’’ service is added to /etc/services file
  If not, add the entry and restart cluster processes.
  Please refer to IRIS FailSafe administration manual for more
  information.
```

```
Started cluster control processes
Restarting cluster admin processes with failsafe-exitop...
```

6. Use `cluster_mgr` template (`/var/cluster/cmgr-templates/cmgr-create-node`) or `cluster_mgr` command to define the node.

   Note: This node must be defined from one of nodes that is already in the cluster.

7. Use the `cluster_mgr` command to add the node to the cluster.

   For example: The following `cluster_mgr` command adds the node `web-node3` to the cluster `web-cluster`:

   ```
   cmgr> modify cluster web-cluster
   Enter commands, when finished enter either ''done'' or ''cancel''

   web-cluster ? add node web-node3
   web-cluster ? done
   ```

8. You can start HA services on this node using the `cluster_mgr` command. For example, the following `cluster_mgr` command starts HA services on node `web-node3` in cluster `web-cluster`:

   ```
   cmgr> start ha_services on node web-node3 in cluster web-cluster
   ```

9. Remember to add this node to the failure domain of the relevant failover policy. In order to do this, the entire failover policy must be re-defined, including the additional node in the failure domain.

Deleting a Node from an Active Cluster

Use the following procedure to delete a node from an active cluster. This procedure begins with the assumption that the node status is UP.

1. If resource groups are online on the node, use the `cluster_mgr` command to move them to another node in the cluster.

   To move the resource groups to another node in the cluster, there should be another node available in the failover policy domain of the resource group. If you want to leave the resource groups running in the same node, use the

cluster_mgr command to detach the resource group. For example, the following command would leave the resource group web-rg running in the same node in the cluster web-cluster.

```
cmgr> admin detach resource_group ''web-rg'' in cluster web-cluster
```

2. Delete the node from the failure domains of any failover policies which use the node. In order to do this, the entire failover policy must be re-defined, deleting the affected node from the failure domain.

3. To stop HA services on the node web-node3, use the following cluster_mgr command. This command will move all the resource groups online on this node to other nodes in the cluster if possible.

   ```
   cmgr> stop ha_services on node web-node3 for cluster web-cluster
   ```

   If it is not possible to move resource groups that are online on node web-node3, the above command will fail. The force option is available to stop HA services in a node even in the case of an error. Should there be any resources which can not be moved offline or deallocated properly, a side-effect of the offline force command will be to leave these resources allocated on the node.

Perform Steps 4, 5, 6, and 7 if the node must be deleted from the configuration database.

4. Delete the node from the cluster. To delete node web-node3 from web-cluster configuration, use the following cluster_mgr command:

   ```
   cmgr> modify cluster web-cluster
   Enter commands, when finished enter either ''done'' or ''cancel''


   web-cluster ? remove node web-node3
   web-cluster ? done
   ```

5. Remove node configuration from the configuration database.

   The following cluster_mgr command deletes the web-node3 node definition from the configuration database.

   ```
   cmgr> delete node web-node3
   ```

6. Stop all cluster processes and delete the configuration database.

The following commands stop cluster processes on the node and delete the configuration database.

```
# /etc/init.d/cluster stop
# killall fs2d
# cdbdelete /var/cluster/cdb/cdb.db
```

7. Disable cluster and HA processes from starting when the node boots. The following commands perform those tasks:

```
# chkconfig cluster off
# chkconfig failsafe2 off
```

Changing Control Networks in a Cluster

Use the following procedure to change the control networks in a currently active cluster. This procedure is valid for a two-node cluster consisting of nodes `node1` and `node2`. In this procedure, you must complete each step before proceeding to the next step.

Note: Do not perform any other administration operations during this procedure.

---

1. From any node, stop HA services on the cluster. Make sure all HA processes have exited on both nodes.
2. From `node2`, stop the cluster processes on `node2`:

```
# /etc/init.d/cluster stop
# killall fs2d
```

   Make sure the `fs2d` process have been killed on `node2`.

3. From `node1`, modify the `node1` and `node2` definition. Use the following `cmgr` commands:

```
cmgr> modify node node1
Enter commands, when finished enter either "done" or "cancel"
node1?> remove nic old nic address
node1> add nic nnew nic address
NIC - new nic address set heartbeat to ...
NIC - new nic address set ctrl_msgs to ...
NIC - new nic address set priority to ...
```

```
NIC - new nic address done
node1? done
```

Repeat the same procedure to modify node2.

4. From node1, check if the node1 and node2 definitions are correct. Using cmgr on node1, execute the following commands to view the node definitions:

```
cmgr> show node node1
cmgr> show node node2
```

5. On both node1 and node2, modify the network interface IP addresses in /etc/config/netif.options and execute ifconfig to configure the new IP addresses on node1 and node2. Verify that the IP addresses match the node definitions in the CDB.

6. From node1, stop the cluster process on node1:

```
# /etc/init.d/cluster stop
# killall fs2d
```

Make sure the fs2d process have been killed on node1.

7. From node2, execute the following command to start cluster process on node2:

```
# /usr/cluster/bin/cdbreinit /var/cluster/cdb/cdb.db
```

Answer **y** to the prompt the appears.

8. From node1, start cluster processes on node1:

```
# /etc/init.d/cluster start
```

The following messages should appear in the SYSLOG on node2:

```
Starting to receive CDB sync series from machine <node1 node id>
...
Finished receiving CDB sync series from machine <node1 node id>
```

Wait for approximately sixty seconds for the sync to complete.

9. From any node, start HA services in the cluster.

Upgrading OS Software in an Active Cluster

When you upgrade your OS software in an active cluster, you perform the upgrade on one node at a time.

If the OS software upgrade does not require reboot or does not impact the FailSafe software, there is no need to use the OS upgrade procedure. If you do not know whether the upgrade will impact FailSafe software or if the OS upgrade requires a machine reboot, follow the upgrade procedure described below.

The following procedure upgrades the OS software on node `web-node3`.

1. If resource groups are online on the node, use a `cluster_mgr` command to move them another node in the cluster. To move the resource group to another node in the cluster, there should be another node available in the failover policy domain of the resource group.

   The following `cluster_mgr` command moves resource group `web-rg` to another node in the cluster `web-cluster:`

   ```
   cmgr> admin move resource_group web-rg in cluster web-cluster
   ```

2. To stop HA services on the node `web-node3`, use the following `cluster_mgr` command. This command will move all the resource groups online on this node to other nodes in the cluster if possible.

   ```
   cmgr> stop ha_services on node web-node3 for cluster web-cluster
   ```

   If it is not possible to move resource groups that are online on node `web-node3`, the above command will fail. You can use the `force` option to stop HA services in a node even in the case of an error.

3. Perform the OS upgrade in the node `web-node3`.

4. After the OS upgrade, make sure cluster processes (`cmond`, `cad`, `crsd`) are running.

5. Restart HA services on the node. The following `cluster_mgr` command restarts HA services on the node:

   ```
   cmgr> start ha_services on node web-node3 for cluster web-cluster
   ```

   Make sure the resource groups are running on the most appropriate node after restarting HA services.

Upgrading FailSafe Software in an Active Cluster

When you upgrade FailSafe software in an active cluster, you upgrade one node at a time in the cluster.

The following procedure upgrades FailSafe on node `web-node3`.

1. If resource groups are online on the node, use a `cluster_mgr` command to move them another node in the cluster. To move the resource group to another node in the cluster, there should be another node available in the failover policy domain of the resource group.

   The following `cluster_mgr` command moves resource group `web-rg` to another node in the cluster `web-cluster:`

   ```
   cmgr> admin move resource_group web-rg in cluster web-cluster
   ```

2. To stop HA services on the node `web-node3`, use the following `cluster_mgr` command. This command will move all the resource groups online on this node to other nodes in the cluster if possible.

   ```
   cmgr> stop ha_services on node web-node3 for cluster web-cluster
   ```

   If it is not possible to move resource groups that are online on node `web-node3`, the above command will fail. You can use the `force` option to stop HA services in a node even in the case of an error.

3. Stop all cluster processes running on the node.

   ```
   # /etc/init.d/cluster stop
   ```

4. Perform the FailSafe upgrade in the node `web-node3`.

5. After the FailSafe upgrade, check whether cluster processes (`cmond`, `cad`, `crsd`) are running. If not, restart cluster processes:

   ```
   # chkconfig cluster on; /etc/init.d/cluster start
   ```

6. Restart HA services on the node. The following `cluster_mgr` command restarts HA services on the node:

   ```
   cmgr> start ha_services on node web-node3 for cluster web-cluster
   ```

   Make sure the resource groups are running on the most appropriate node after restarting HA services.

Adding New Resource Groups or Resources in an Active Cluster

The following procedure describes how to add a resource group and resources to an active cluster. To add resources to an existing resource group, perform resource configuration (Step 4), resource diagnostics (Step 5) and add resources to the resource group (Step 6).

1. Identify all the resources that have to be moved together. These resources running on a node should be able to provide a service to the client. These resources should be placed in a resource group. For example, Netscape webserver `mfg-web`, its IP address 192.26.50.40, and the filesystem `/shared/mfg-web` containing the web configuration and document pages should be placed in the same resource group (for example, `mfg-web-rg`).

2. Configure the resources in all nodes in the cluster where the resource group is expected to be online. For example, this might involve configuring netscape web server `mfg-web` on nodes `web-node1` and `web-node2` in the cluster.

3. Create a failover policy. Determine the type of failover attribute required for the resource group. The `cluster_mgr` template (`/var/cluster/cmgr-templates/cmgr-create-failover_policy`) can be used to create the failover policy.

4. Configure the resources in configuration database. There are `cluster_mgr` templates to create resources of various resource types in `/var/cluster/cmgr-templates` directory. For example, the volume resource, the `/shared/mfg-web` filesystem, the 192.26.50.40 IP_address resource, and the `mfg-web` Netscape_web resource have to be created in the configuration database. Create the resource dependencies for these resources.

5. Run resource diagnostics. For information on the diagnostic commands, see Chapter 8, "Testing IRIS FailSafe Configuration".

6. Create resource group and add resources to the resource group. The `cluster_mgr` template (`/var/cluster/cmgr-templates/cmgr-create-resource_group`) can be used to create resource group and add resources to resource group.

   All resources that are dependent on each other should be added to the resource group at the same time. If resources are added to an existing resource group that is online in a node in the cluster, the resources are also made online on the same node.

Adding a New Hardware Device in an Active Cluster

When you add hardware devices to an active cluster, you add them one node at a time.

To add hardware devices to a node in an active cluster, follow the same procedure as when you upgrade OS software in an active cluster, as described in "Upgrading OS Software in an Active Cluster", page 237. In summary:

- You must move the resource groups offline and stop HA services in the node before adding the hardware device.
- After adding the hardware device, make sure cluster processes are running and start HA services on the node.

To include the new hardware device in the configuration database, you must modify your resource configuration and your node configuration, where appropriate.

Performance Co-Pilot for FailSafe

This chapter tells you how to use Performance Co-Pilot (PCP) for FailSafe to monitor the availability of an IRIX FailSafe cluster. For information about installing PCP for FailSafe, see "Installing Performance Co-Pilot Software", page 74.

PCP provides the following:

- An agent for exporting FailSafe heartbeat and resource monitoring statistics to the PCP framework
- 3-D visualization tools for displaying these statistics in an intuitive presentation

The visualization of statistics provides valuable information about the availability of nodes and resources monitored by FailSafe. For example, it can highlight a reduction in monitoring response times that may indicate problems in availability of services provided by the cluster.

Because PCP for FailSafe is an extension to the PCP framework, you can use other PCP tools to analyze or present FailSafe monitoring statistics, and record PCP for FailSafe metrics as archives for deferred analysis. You can also use PCP to gather statistics about CPU and memory utilization, network and disk activity, and other performance metrics for each node in the cluster.

Using the Visualization Tools

To view statistics about the FailSafe cluster, use the `hbvis`(1) and `rmvis`(1) commands.

The `hbvis`(1) command constructs a display showing the distribution of heartbeat response times for every node in the cluster. Figure 11-1 shows an example display.

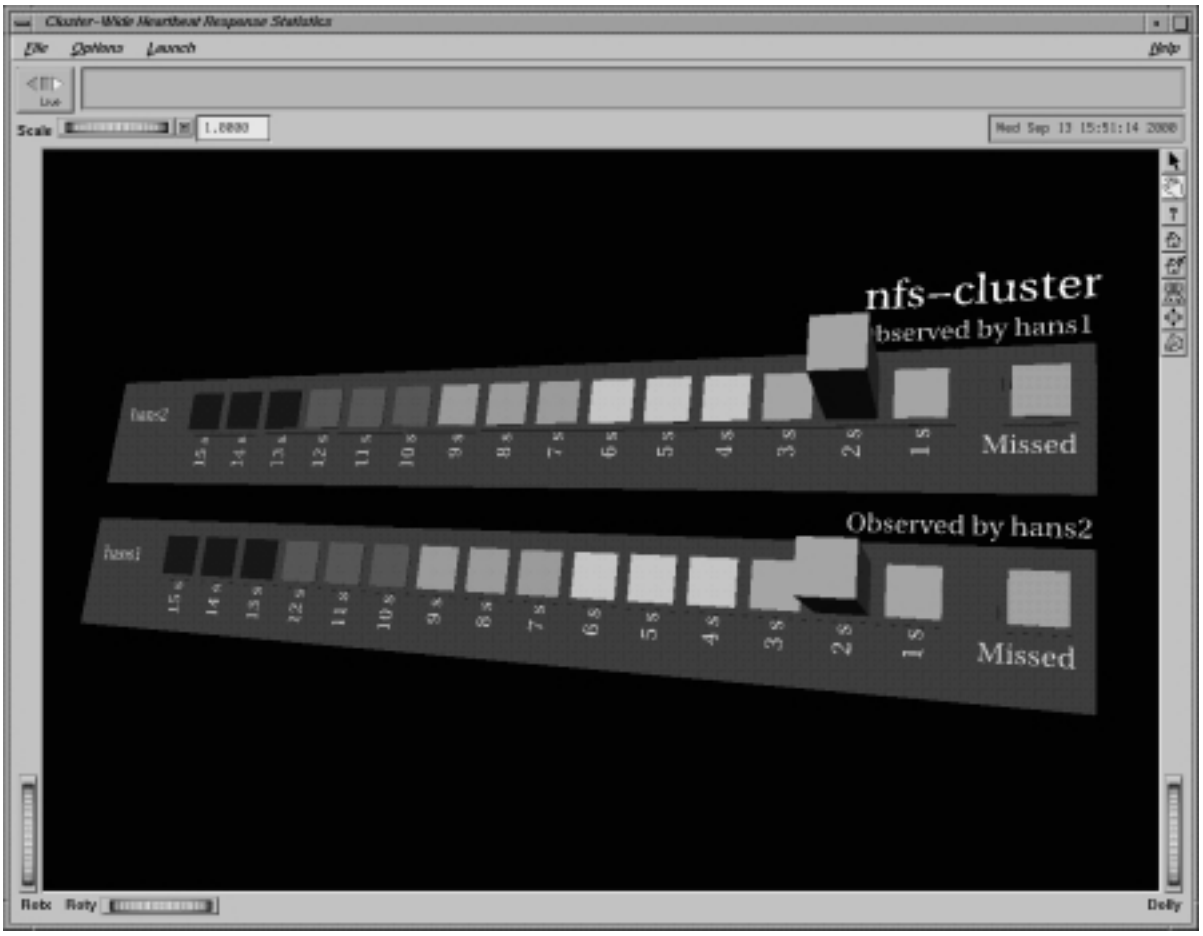

Figure 11-1 Heartbeat Response Statistics

Key features of the display include the frequency of heartbeat responses that arrive at particular intervals within the timeout period, and the frequency of heartbeat responses that have been missed (determined not to have arrived). The bar representing the frequency of missed heartbeat responses changes color to indicate the urgency of problems with availability of a node.

The `rmvis`(1) command constructs a display of the resource monitoring response times for resources monitored on every node of the cluster. Figure 11-2 shows an example display.

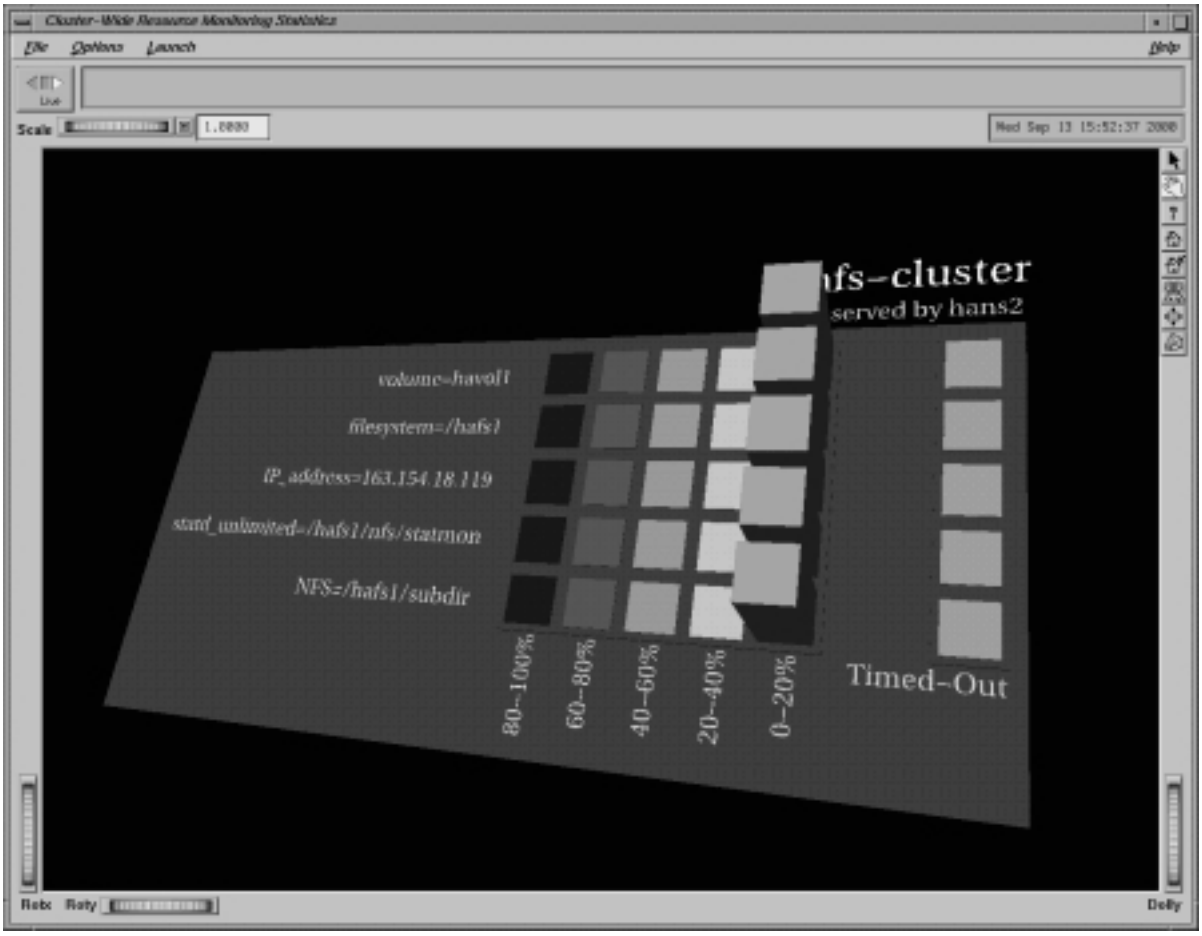

Figure 11-2 Resource Monitoring Statistics

The display is similar in concept to that of `hbvis`(1), showing the frequency of resource monitoring responses that arrive within the timeout period, and the frequency of responses that have timed out. The bar representing the frequency of

resource responses that have timed out also changes color to indicate the urgency of problems with the availability of particular resources.

If a node has failed or a resource has failed over, its statistics will disappear from the display.

To run a visualization tool on the monitor host, use the `-h` option to specify an available collector host in the cluster (*host*):

```
% hbvis -h host
```

or

```
% rmvis -h host
```

The collector host specified can be **any** collector host that is a member of the cluster for which you wish to view statistics.

There are various options available to alter the display provided by `hbvis`(1) and `rmvis`(1):

`-H` *hostfile* — Provides a file that lists the nodes that are to appear in the visualization. This is useful in limiting the number of nodes in the display, because it takes more time to construct the display for clusters with more nodes.

`-t` *interval* — Assigns the sampling time of the visualization. There may be circumstances where extending the period of the sampling time may provide better application responsiveness, particularly for clusters with many nodes. Because FailSafe maintains the statistics, `hbvis`(1) and `rmvis`(1) will always show the latest statistics available for the sampling time selected. For details about the *interval* option, see the `pmview`(1) and `PCPIntro`(1) man pages.

`-r` — Selects the FailSafe metrics that present a sampling of statistics taken from the time of the last statistical reset. This enables `hbvis`(1) and `rmvis`(1) to improve the sensitivity of the visualization when abrupt changes appear in the FailSafe monitoring statistics.

Without the `-r` option, the statistics presented are from a sampling of FailSafe metrics collected from the time `ha_cmsd`(1m) and/or `ha_srmd`(1m) was last restarted.

`-R` — Starts a new statistical sampling.

`-v` — (`hbvis`(1) only) Provides a visualisation of heartbeat statistics for each node in the cluster, from the point of view of the selected collector host

only. (The collector host is selected using the `-h` option). There is a graphical representation of heartbeat statistics for each node in the cluster as observed by the selected collector host.

`-w` (`hbvis`(1) only) Provides a visualisation of the aggregate of heartbeat statistics for all nodes in the cluster, from the point of view of the selected collector host only. (The collector host is selected using the `-h` option). There is a only one graphical representation of heartbeat statistics for the entire cluster as observed by the selected collector host.

For a complete description of options, see the `hbvis`(1) and `rmvis`(1) man pages.

`hbvis`(1) and `rmvis`(1) use the command `pmview`(1) to display the 3-D visualization of FailSafe performance metrics. For a description of the various menu commands and controls in the visualization window, consult the man pages for `pmview`(1).

PCP for FailSafe Performance Metrics

PCP tools such as `pmlogger`(1), `pmchart`(1), and `pminfo`(1) can use the metrics exported by PCP for FailSafe.

Appendix C, "Metrics Exported by PCP for FailSafe", page 267, provides a description of PCP for FailSafe metrics. You can also display a description of metrics by using the following command:

```
% pminfo -tT -h host
```

(If you are logged in to a collector host, you can leave out the `-h` option).

Troubleshooting

A grey display (that is, no colored rectangle bars appear on the node's grey baseplane) when using `hbvis`(1) or `rmvis`(1) may indicate one of the following:

- The node is down.

  If you wish to see only the nodes that are up, create a file containing a list of nodes that are to be displayed and pass it as an option to `hbvis`(1)/`rmvis`(1) using the `-H` option (or the environment variable PCP_FSAFE_NODES) so that a new picture of the cluster can be generated. Please refer to the `hbvis`(1)/`rmvis`(1) man pages for more details on the `-H` option.

- The collector daemons have been killed on that node.

  To solve this problem, restart `pmdafsafe`(1) in one of the following ways:

  – If `pmcd`(1) is still running, send `pmcd`(1) the `SIGHUP` signal by entering the following::

    ```
    # killall -HUP pmcd
    ```

  – If `pmcd`(1) is not running, restart PCP by entering the following:

    ```
    # /etc/init.d/pcp start
    ```

- The timeout and sampling settings are too short.

  To change the sampling time, use the time controls available in the `pmview`(1) window=. By default, this is 2 seconds; you may need to lengthen the sampling period if you are getting an unsatisfactory display.

  Alternatively, there may be timeout issues between `pmdafsafe`(1) and `pmcd`(1), or between `pmcd`(1) and `pmview`(1). Refer to the man pages for `pmcd`(1) and `PCPIntro`(1) for information on how to change the timeout settings for the various PCP tools.

- The resource has failed over (for `rmvis`(1)).

  In this case, restart `rmvis`(1) so that a new picture of the cluster can be generated

Updating from IRIS FailSafe 1.2 to IRIS FailSafe 2.X

IRIS FailSafe 2.X is not a new release of the IRIS FailSafe 1.2 product but, instead, is a new set of files and scripts that provides many additional possibilities for the size and complexity of a highly available system. If you wish to migrate an IRIS FailSafe 1.2 system to an IRIS FailSafe 2.X system to take advantage of these features, you must upgrade your system configuration. There is no upgrade installation option to automatically upgrade FailSafe 1.2 to FailSafe 2.X.

This chapter provides a description of the procedures you perform to upgrade a system form IRIS FailSafe 1.2 to IRIS FailSafe 2.X. It includes the following sections:

Hardware Changes

There are no hardware changes that are required when you upgrade a system to FailSafe 2.X. A FailSafe 1.2 system will be a dual-hosted storage with reset ring two-node configuration in FailSafe 2.X.

With FailSafe 2.X, you can test the hardware configuration with FailSafe diagnostic commands. See Chapter 8, "Testing IRIS FailSafe Configuration" for instructions on using FailSafe to test the connections. These diagnostics are not run automatically when you start FailSafe 2.X; you must run them manually.

You can also use the `admin ping` CLI command to test the serial reset line in FailSafe 2.X. This command replaces the `ha_spng` command you used with FailSafe 1.2.

FailSafe 1.2 command to test serial reset lines:

```
# /usr/etc/ha_spng -i 1 -d msc -f /dev/ttyd2
# echo $status
```

FailSafe 2.X CLI command to test serial reset lines:

```
cmgr> admin ping dev_name /dev/ttyd2 of dev_typetty with sysctrl_type msc
```

See Chapter 4, "IRIS FailSafe Administration Tools" for information on using CLI commands.

Software Changes

FailSafe 2.X consists of a different set of files than FailSafe 1.2. FailSafe 1.2 and FailSafe 2.X can exist on the same node, but you cannot run both versions of FailSafe at the same time.

FailSafe 1.2 contains a configuration file, `ha.conf`. In FailSafe 2.X, configuration information is contained in a configuration database at `/var/cluster/cdb/cdb.db` that is kept in all nodes in the pool. You create the configuration database using the Cluster Manager CLI or the Cluster Manager GUI.

The FailSafe 2.X configuration database is automatically copied to all nodes in the pool. The FailSafe 2.X configuration is kept in all nodes in the pool.

Configuration Changes

You must reconfigure your FailSafe 1.2 system by using the FailSafe 2.X Cluster Manager GUI or the FailSafe 2.X Cluster Manager CLI to configure the system as a FailSafe 2.X system. For information on using these administration tools, see Chapter 4, "IRIS FailSafe Administration Tools".

To update a FailSafe 1.2 configuration, consider how the FailSafe 1.2 configuration maps onto the concept of resource groups:

- A dual-active FailSafe 1.2 configuration contains two resource groups, one for each node.
- An active/standby FailSafe 1.2 configuration contains one resource group, consisting of an entire node (the active node).

Each resource group contains all the applications that were primary on each node and backed up by the other node.

When you configure a FailSafe 2.X system, you perform the following steps:

1. Add nodes to the pool
2. Create cluster
3. Add nodes to the cluster
4. Set HA parameters

   FailSafe 2.X can be started at this point, if desired.
5. Create resources
6. Create failover policy
7. Create resource groups
8. Add resources to resource groups
9. Put resource groups online

These steps are captured in the task sets on the Guided Configuration page of FailSafe Manager in the FailSafe Cluster Manager GUI. These task sets lead you through these configuration steps.

For a configuration example that compares FailSafe 1.2 configuration to FailSafe 2.X configuration, see "Upgrade Examples", page 251.

Scripts

All FailSafe 1.2 scripts must be rewritten for FailSafe 2.X. The *IRIS FailSafe Version 2 Programmer's Guide* provides detailed information on FailSafe 2.X scripts as well as detailed instructions for migrating FailSafe 1.2 scripts to their FailSafe 2.X functional equivalent.

Operational Comparison

In FailSafe 1.2, the unit of failover is the node. In FailSafe 2.X, the unit of failover is the resource group. Because of this, the concepts of node failover, node failback, and

even node state do longer apply to FailSafe 2.X. In addition, all FailSafe scripts differ between the two releases.

Table A-1, page 250 summarizes the differences between the releases.

Table A-1 Differences Between IRIS FailSafe 1.2 and 2.X

IRIS FailSafe 1.2	IRIS FailSafe 2.X
`ha.conf` configuration file	Configuration database at `/var/cluster/cdb/cdb/db`. The database is automatically copied to all nodes in the pool. Much of the data contained in the 1.2 `ha.conf` file will be used in the 2.X database, but the format is completely different. You will configure the database using the Cluster Manager graphical user interface or the `cluster_mgr` command.
Node states (standby, normal, degraded, booting or up)	Resource Group states (online, offline, pending, maintenance, error)

Scripts:
`giveaway, giveback`
`takeover, takeback`
`check`
(no equivalent)
	Scripts:
	`stop`
	`start`
	`monitor`
	`exclusive`, `probe`, `restart`
	Failover script
	Failover attributes

All common functions and variables are kept in the `/var/ha/actions/common.vars` file	All common functions and variables are kept in the `/var/cluster/ha/common_scripts/scriptlib` file
Configuration information is read using the `ha_cfginfo` command	Configuration information is read using the `ha_get_info()` and `ha_get_field()` shell functions
Software links specify application ordering	Software links are not used for ordering
Scripts use `/sbin/sh`	Scripts use `/sbin/ksh`
Scripts require configuration checksum verification	There is no configuration checksum verification in the scripts
Scripts require resource ownership	Action scripts have no notion of resource ownership

IRIS FailSafe 1.2	IRIS FailSafe 2.X
Scripts do not run in parallel	Multiple instances of action scripts can be run at the same time
Each service had its own log in `/var/ha/logs`	Action scripts use cluster logging and all scripts log to the same file using the `ha_cilog` command
There were two units of failover, one for each node in the cluster	There is a unit of failover (a resource group) for each highly available service

Upgrade Examples

In order to upgrade a FailSafe 1.2 system to a FailSafe 2.X system, you must examine your `ha.conf` file to determine how to define the equivalent parameters in the FailSafe 2.X configuration database.

The following sections show upgrade examples for the following tasks:

- Defining a Node
- Defining a Cluster
- Setting HA Parameters
- Defining a Resource: XLV Volume
- Defining a Resource: XFS Filesystem
- Defining a Resource: IP Address

For upgrade examples of the following tasks, see the *IRIS FailSafe Version 2 Programmer's Guide*, where customized resources and scripts are described.

- Defining a Resource Type
- Defining a Failover Policy
- Writing FailSafe Scripts

Defining a Node

The following example shows node definition in the FailSafe 1.2 `ha.conf` file. Parameters that you will need to use when configuring a FailSafe 2.X system are indicated in bold.

```
Node node1
{
interface node1-fxd
{
name = rns0
ip-address = 54.3.252.6
netmask = 255.255.255.0
broadcast-addr = 54.3.252.6
}
heartbeat
{
hb-private-ipname = 192.0.2.3
hb-public-ipname = 54.3.252.6
hb-probe-time = 6
hb-timeout = 6
hb-lost-count = 4
}
reset-tty = /dev/ttyd2

sys-ctlr-type = MSC
}
```

In this configuration example, you will use the following values when you define the same node in FailSafe 2.X:

node name:	node 1
primary network interface:	node 1
type of system controller:	msc
system control device name:	/dev/ttyd2
control networks:	192.0.2.3, 54.3.252.6

Use the following `cmgr` command to use these values to define a node in FailSafe 2.X. Note that there are additional parameters you will need to specify when you define this node.

```
cmgr> define node node1
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional]? node1
Is this a FailSafe node <true|false> ? true
Is this a CXFS node <true|false> ? false
Node ID ? 10
Reset type <powerCycle> ? (powerCycle)
Do you wish to define system controller info[y/n]:y
Sysctrl Type <msc|mmsc>? (msc) msc
Sysctrl Password [optional]? ( )
Sysctrl Status <enabled|disabled>? enabled
Sysctrl Owner? node2
Sysctrl Device? /dev/ttyd2
Sysctrl Owner Type <tty> [tty]?
Number of Network interfaces [2]? 2
NIC 1 - IP Address? 192.0.2.3
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false>? true
NIC 1 - (use network for control messages) <true|false>? true
NIC 1 - Priority <1,2,...>? 1
...
```

As this `ha.conf` node definition shows, in FailSafe 1.2 you defined parameters to set the values that determined how often to send monitoring messages and how long of a time period without a response would indicate a failure when you defined a node. For information on setting monitoring values in FailSafe 2.X, see "Setting HA Parameters", page 254.

Defining a Cluster

Although FailSafe 1.2 does not require the definition of clusters, you specify a parameter in the `ha.conf` file that FailSafe 2.X uses in its cluster definition: the e-mail address to use to notify the system administrator when problems occur in the cluster.

The `ha.conf` file includes the following:

```
system configuration
{
mail-dest-addr = root@localhost
...
}
```

When you define a cluster in FailSafe 2.X, you can use this as the e-mail address to use for problem notification.

There are other things you must provide in addition to this parameter when you define a FailSafe 2.X cluster, such as the e-mail program to use for this notification and, of course, the nodes to include in the cluster. Use the following `cmgr` command to define a cluster:

```
cmgr> define cluster apache-cluster
Enter commands, you may enter "done" or "cancel" at any time to exit

cluster apache-cluster? set notify_addr to root@localhost
cluster A? done
```

Use the following `cmgr` command to add nodes to the cluster:

```
cmgr> modify cluster apache-cluster
Enter commands, you may enter "done" or "cancel" at any time to exit

cluster apache-cluster? add node node1
cluster A? done
```

Setting HA Parameters

The following example shows the sections of a FailSafe 1.2 `ha.conf` file that are used to set monitoring and timeout values. Parameters that you will need to use when configuring a FailSafe 2.X system are indicated in bold.

```
system-configuration
{
      pwrfail = true
      ...
}


Node node1
{
...
heartbeat
{
       hb-private-ipname = 192.0.2.3
       hb-public-ipname = 54.3.252.6
       hb-probe-time = 6
```

```
        hb-timeout = 6
        hb-lost-count = 4
}
...
}
```

As this `ha.conf` node-definition shows, in FailSafe 1.2 you defined `hb-probe-time`, `hb-timeout`, and `hb-lost-count` parameters to set the values that determined how often to send monitoring messages and how long of a time period without a response would indicate a failure. FailSafe 2.X uses a different method for monitoring the nodes in a cluster than FailSafe 1.2 uses, sending out continuous messages to the other nodes in a cluster and, in turn, maintaining continuous monitoring of the messages the other nodes are sending.

Because of the different monitoring methods between the two systems, there is no one-to-one correspondence between the values you set in the `ha.conf` file and the timeout and heartbeat intervals you set in FailSafe 2.X when you set FailSafe HA parameters. However, if you wish to maintain approximately the same time interval before which your system determines that failure has occurred, you can use the following formula to determine the value to which you should set your node timeout interval:

node timeout = (probetime + timeout) * lostcount

This formula should account for the same total node-to-node communication time.

All FailSafe 2.X timeouts are in milliseconds, and can be changed when FailSafe 2.X is running. Timeouts can be specified for the cluster for a specific node in the cluster.

There is no long-timeout value in FailSafe 2.X. The long-timeout value equivalent is set with the resource type start and stop action monitor timeouts. The resource type start, monitor, and stop action timeouts can be changed using the Cluster Manager GUI or the Cluster Manager CLI.

Use the following `cmgr` command to modify the HA parameters for `node1` in FailSafe 2.X:

```
 cmgr> modify ha_parameters on node node1 in cluster apache-cluster


Enter commands, when finished enter either "done" or "cancel"

node1 ? set node_timeout to 24000
node1 ? set heartbeat to 6000
```

```
node1 ? set run_pwrfail to true
node1 ? done
```

Defining a Resource: XLV Volume

The following example shows a volume definition in the FailSafe 1.2 `ha.conf` file. Parameters that you will need to use when configuring the same volume as a volume resource in a FailSafe 2.X system are indicated in bold.

```
volume apache-vol
{
   server-node = node1
   backup-node = node2
   devname = apache-vol
   devname-owner = root
   devname-group = sys
   devname-mode = 600
}
```

In this configuration example, you will use the following values when you define the same volume in FailSafe 2.X:

volume name:	apache-vol
user name of device file owner:	root
group name of device file:	sys
device file permissions:	600

To create an XLV volume resource, use the following `cmgr` commands:

```
cmgr> define resource apache-vol of resource_type volume in cluster apache-cluster
Enter commands, when finished enter either "done" or "cancel"

resource apache-vol? set devname-owner to root
resource apache-vol? set devname-group to sys
resource apache-vol? set devname-mode to 600
resource apache-vol? done
```

Defining a Resource: XFS Filesystem

The following example shows an XFS filesystem definition in the FailSafe 1.2 `ha.conf` file. Parameters that you will need to use when configuring the same filesystem as a filesystem resource in a FailSafe 2.X system are indicated in bold.

```
filesystem apache-fs
{
   mount-point = /apache-fs
   mount-info
{
   fs-type = xfs
   volume-name = apache-vol
   mode = rw, noauto
}
}
```

In this configuration example, you will use the following values when you define the same filesystem in FailSafe 2.X:

resource name (mount point):	/apache-vol
xlv volume:	apache-vol
mount options:	rw, noauto

To create a filesystem resource, use the following `cmgr` commands:

```
cmgr> define resource /apache-fs of resource_type filesystem in cluster apache-cluster
Enter commands, when finished enter either "done" or "cancel"

resource /apache-fs? set volme-name to apache-vol
resource /apache-fs? set mount-options to "rw,noauto"
resource /apache-fs? done
```

Defining a Resource: IP Address

The following example shows an IP address definition in the FailSafe 1.2 `ha.conf` file. Parameters that you will need to use when configuring the same IP address as a highly available resource in a FailSafe 2.X system are indicated in bold.

```
interface-pair FDDI_1
{
   primary-interface = node-fxd
```

```
    secondary-interface = node2-fxd
    re-mac = false
    netmask = 0xffffff00
    broadcast-addr = 54.3.252.255

    ip-aliases = ( 54.3.252.7 )
}
```

In this configuration example, you will use the following values when you define the same IP Address in FailSafe 2.X:

Resource name:	54.3.252.7
broadcast address:	54.3.252.255
network mask:	0xffffff00

To create an IP address resource, use the following `cmgr` commands:

```
cmgr> define resource 54.3.252.7 of resource_type IP_address in cluster apache-cluster
Enter commands, when finished enter either "done" or "cancel"

resource 54.3.252.7? set interfaces to rns0
resource 54.3.252.7? set NetworkMask to 0xffffff00
resource 54.3.252.7? set BroadcasetAddress to 54.3.252.255
resource 54.3.252.7? done
```

Additional FailSafe 2.X Tasks

After you have defined your nodes, clusters, and resources, you define your resource groups, a task which has no equivalent in FailSafe 1.2. When you define a resource group, you specify the resources that will be included in the resource group and the failover policy that determines which node will take over the services of the resource group on failure.

For information on defining resource groups, see "Defining Resource Groups", page 152 in Chapter 5, "IRIS FailSafe Configuration".

After you have configured you system, you can start FailSafe services, as described in "Activating (Starting) IRIS FailSafe", page 176 in Chapter 7, "IRIS FailSafe System Operation".

Status

In FailSafe 1.2, you produced a display of the system status with the `ha_admin -a` command. In FailSafe 2.X, you can display the system status in the following ways:

- You can keep continuous watch on the state of a cluster using the Cluster View of the Cluster Manager GUI.
- You can query the status of an individual resource group, node, or cluster using either the Cluster Manager GUI or the Cluster Manager CLI.
- You can use the `/var/cluster/cmgr-scripts/ha Status` script provided with the Cluster Manager CLI to see the status of all clusters, nodes, resources, and resource groups in the configuration.

For information on performing these tasks, see "System Status", page 177 in Chapter 7, "IRIS FailSafe System Operation".

IRIS FailSafe 2.1 Software

This appendix summarizes software to be installed on systems used for IRIS FailSafe 2.1.

Note: "Installing Required Software", page 54 contains step-by-step instructions for installing the software.

This appendix consists of these sections:

Subsystems on the IRIS FailSafe 2.1 CD

The IRIS FailSafe 2.1 base CD requires about 10 MB.

Table B-1, page 262 lists IRIS FailSafe 2.1 subsystems on the IRIS FailSafe 2.1 CD.

Table B-1 IRIS FailSafe 2.1 CD

Purpose	System
IRIS FailSafe 2.1	*failsafe2* *failsafe2.idb* *failsafe2.man* *failsafe2.sw* *failsafe2.books* (InSight versions of customer manuals)
FailSafe system administration	*sysadm_failsafe2* *sysadm_failsafe2.idb* *sysadm_failsafe2.man* *sysadm_failsafe2.sw*

Note: Users must install base system administration (*sysadm_base*), cluster administration (*cluster_admin*), cluster control (*cluster_control*), cluster services (*cluster_services*), java (*java_eoe*), and Java Plug-in (*java_plugin*) from the IRIX CD set.

The EL-8+ multiplexer driver subsystems are *el_serial*, *el_serial.man*, and *el_serial.sw*, which are on a CD accompanying the EL-8+ multiplexer.

Subsystems to Install on Servers and Workstations in an IRIS FailSafe 2.1 Pool

Table B-2, page 262 lists subsystems required for servers and workstations in the pool. The pool is the entire set of servers available for clustering (nodes). It includes servers and the workstation(s) used for administering the cluster

Table B-2 Subsystems Required for Nodes in the Pool (Servers and GUI Client(s))

Product	Images and Subsystems	Prerequisites
Base system administration	*sysadm_base.sw.dso*	None
Base system administration server	*sysadm_base.sw.server*	*sysadm_base.sw.dso*

Product	Images and Subsystems	Prerequisites
IRIS FailSafe 2.1 administration server	*sysadm_failsafe2.sw.server*	*sysadm_base.sw.server* *cluster_admin.sw.base* *cluster_services.sw.cli* *cluster_control.sw.cli* *failsafe2.sw.cli*
Cluster administration	*cluster_admin.sw* *cluster_control.sw*	*sysadm_base.sw.dso*
Web-based administration	*sysadm_failsafe2.sw.web*	*sysadm_failsafe2.sw.client* *sysadm_failsafe2.sw.server* *sysadmbase.sw.client* *java_eoe.sw*, version 3.1.1 Web server
EL-8+ multiplexer driver (from CD included with multiplexer)	*el_serial* *el_serial.man* *el_serial.sw*	

Additional Subsystems for Nodes in an IRIS FailSafe 2.1 Cluster

Table B-3, page 263 lists additional subsystems required for each server that is a node in the cluster. A cluster is one or more nodes coupled with each other by networks. A node is a single UNIX image, usually, an individual server. A node can be a member of only one cluster.

Table B-3 Additional Subsystems Required for Nodes in the Cluster

Product	Images and Subsystems	Prerequisites
Highly available clustering software	*cluster_services.sw*	*cluster_admin.sw* *cluster_control.sw*
IRIS FailSafe 2.1 software	*failsafe2.sw*	*cluster_services.sw*

Additional Subsystems to Install on Administrative Workstations

On a workstation used to run the GUI client, you must install subsystems depending on the type of workstation. The following sections provide a list of the subsystems to install on the following:

- IRIX Administrative Workstations
- Non-IRIX Administrative Workstations

Subsystems for IRIX Administrative Workstations

On a workstation used to run the GUI client from an IRIX desktop, such as IRISconsole, install subsystems listed in Table B-4, page 264.

Table B-4 Subsystems Required for IRIX Administrative Workstations

Product	Subsystems	Prerequisites
IRIS FailSafe 2.1 Cluster Manager GUI	*sysadm_failsafe2.sw.client* *sysadm_failsafe2.sw.desktop*	sysadm_base.sw.client *java_eoe.sw*, version 3.1.1
Java Plug-in: required only if the workstation is used to launch the GUI client from a Web browser that supports Java	*java_plugin.sw* *java_plugin.sw32*	Web browser that supports Java

If the Java Plug-in is not installed when the IRIS FailSafe 2.1 Cluster Manager GUI is run from a browser, the browser is redirected to *http://java.sun.com/products/plugin/1.1/plugin-install.html*.

Subsystems for Non-IRIX Administrative Workstations

From a non-IRIX workstation, the GUI can be launched from a web browser that supports Java. On a workstation used to run the GUI client from a non-IRIX workstation, install subsystems listed in Table B-5, page 265.

Table B-5 Subsystems Required for Non-IRIX Administrative Workstations

Product	Subsystem	Prerequisite
Java Plug-in	Download Java Plug-in from *http://java.sun.com/products/plugin/1.1/plugin-install.html*	Web browser that supports Java

If the Java Plug-in is not installed when the IRIS FailSafe 2.1 Cluster Manager GUI is run from a browser, the browser is redirected to the Web site listed in Table B-5, page 265.

Metrics Exported by PCP for FailSafe

lists the metrics implemented by `pmdafsafe`(1).

`fsafe.srm.all.*` metrics are the same as the `fsafe.srm.*` metrics, except that the latest values obtained for all resources will be available, even if `ha_srmd`(1M) or any of the resources themselves are not available.

Table C-1 PCP Metrics

Metric	Description
`fsafe.srm.status` `fsafe.srm.all.status`	Latest status of a monitoring event performed on a resource, for all resources configured to be monitored on this node.
`fsafe.srm.timeout` `fsafe.srm.all.timeout`	The prescribed timeout, in milliseconds, for monitoring a resource.
`fsafe.srm.probes` `fsafe.srm.all.probes`	Number of times a resource has been monitored, for all resources configured to be monitored on this node, since the time `ha_smrd`(1M) has started.
`fsafe.srm.recent.probes`	Number of times a resource has been monitored, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.timeouts` `fsafe.srm.all.timeouts`	Number of resource monitoring events that have timed out before declaring that resource as failed, for all resources configured to be monitored on this node, since the time the resources have last been available.
`fsafe.srm.recent.timeouts`	Number of resource monitoring events that have timed out before declaring that resource as failed, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).

Metric	Description
`fsafe.srm.min_resp` `fsafe.srm.all.min_resp`	Approximate minimum time, in milliseconds, taken to complete a monitoring event on a resource, for all resources configured to be monitored
`fsafe.srm.max_resp` `fsafe.srm.all.max_resp`	Approximate maximum time, in milliseconds, taken to complete a monitoring event on a resource, for all resources configured to be monitored on this node.
`fsafe.srm.last_resp` `fsafe.srm.all.last_resp`	Approximate time, in milliseconds, taken in completing the most recent monitoring event on a resource, for all resources configured to be monitored on this node.
`fsafe.srm.cumm_timeouts` `fsafe.srm.all.cumm_timeouts`	Cumulative number of resource monitoring events that have timed out, for all resources configured to be monitored on this node, since the time `ha_smrd`(1M) has started.
`fsafe.srm.recent.cumm_timeouts`	Cumulative number of resource monitoring events that have timed out, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.histo_20` `fsafe.srm.all.histo_20`	Fraction of monitoring events that have been received within 0- 20% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since the time `ha_smrd`(1M) has started.
`fsafe.srm.recent.histo_20`	Fraction of monitoring events that have been received within 0- 20% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.histo_40` `fsafe.srm.all.histo_40`	Fraction of monitoring events that have been received within 20- 40% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since the time `ha_smrd`(1M) has started.

Metric	Description
`fsafe.srm.recent.histo_40`	Fraction of monitoring events that have been received within 20- 40% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.histo_60` `fsafe.srm.all.histo_60`	Fraction of monitoring events that have been received within 40- 60% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since the time `ha_smrd(1M)` has started.
`fsafe.srm.recent.histo_60`	Fraction of monitoring events that have been received within 40- 60% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.histo_80` `fsafe.srm.all.histo_80`	Fraction of monitoring events that have been received within 60- 80% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since the time `ha_smrd`(1M) has started.
`fsafe.srm.recent.histo_80`	Fraction of monitoring events that have been received within 60- 80% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.histo_100` `fsafe.srm.all.histo_100`	Fraction of monitoring events that have been received within 80- 100% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since the time `ha_smrd`(1M) has started.

Metric	Description
`fsafe.srm.recent.histo_100`	Fraction of monitoring events that have been received within 80- 100% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.frac_timeouts` `fsafe.srm.all.frac_timeouts`	Fraction of monitoring events that have timed out before declaring that resource as failed, for all resources configured to be monitored on this node, since the time the resources have last been available.
`fsafe.srm.recent.frac_timeouts`	Fraction of monitoring events that have timed out, before declaring that resource as failed, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.frac_cumm_timeouts` `fsafe.srm.all.frac_cumm_timeouts`	Fraction of cumulative number of monitoring events that have timed out, for all resources configured to be monitored on this node, since the time `ha_smrd`(1M) has started.
`fsafe.srm.recent.frac_cumm_timeouts`	Fraction of cumulative number of monitoring events that have timed out, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.recent.timestamp`	The time when a new collection of statistics was started for the `fsafe.srm.recent.*` metrics, after issuing a store to the metric `fsafe.control.reset_srm`.
`fsafe.config.clustername`	The name of this cluster.
`fsafe.config.hostname`	The name of all hosts in the cluster specified by `fsafe.config.clustername`.
`fsafe.config.nnodes`	Number of nodes in the cluster specified by `fsafe.config.clustername`.
`fsafe.config.cms.interval`	The cluster heartbeat event interval, in milliseconds.

Metric	Description
`fsafe.config.cms.timeout`	The heartbeat event timeout for all nodes in the cluster, in milliseconds.
`fsafe.config.cms.nbuckets`	The number of heartbeat event response intervals per node, where each interval covers a time equal to the heartbeat event interval (`fsafe.config.cms.interval`) for segments of time until the heartbeat event timeout (`fsafe.config.cms.timeout`).
`fsafe.control.debug`	Debugging flags for the `fsafe` PMDA when a decimal integer value is stored to this metric. It ultimately affects what information is put into the `fsafe` PMDA's log (normally at `/var/adm/pcplog/fsafe.log`). Reading this metric will return the currently assigned debugging flags as a decimal integer.
`fsafe.control.reset_cms`	Resets data collection statistics for all metrics gathered from `ha_cmsd`(1M). When this metric is stored to, the data provided is ignored; it is the act of storing to this metric which causes the reset. Reading this metric will return zero (0).
`fsafe.control.reset_srm`	Resets data collection statistics for all metrics gathered from `ha_srmd`(1M). When this metric is stored, the data provided is ignored; it is the act of storing to this metric which causes the reset. Reading this metric will return zero (0).
`fsafe.control.retry`	Sets the number of retries permitted when contacting `ha_cmsd`(1M) or `ha_srmd`(1M), and when the daemons indicate that they are busy.

Metric	Description
	Depending on which metrics are being read, and which daemon is required to obtain values for the required metrics, values for some metrics may not be available, possibly producing the message "`Try again. Information not currently available.`" This metric can be adjusted in order to increase the number of retries permitted when collecting metrics, before giving up and displaying this message. A retry is performed once every 100 ms (approximately). Note that setting this metric does not alter how the fsafe PMDA handles more serious errors from `ha_cmsd`(1M) or `ha_srmd`(1M). Reading this metric will return the current retry count.
`fsafe.cms.expected`	The number of heartbeat events expected to have been received for each node in the cluster (excluding the collector host), since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.expected`	The number of heartbeat events expected to have been received for each node in the cluster (excluding the collector host), since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.received`	The number of heartbeat events actually received for each node in the cluster (excluding the collector host), since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.received`	The number of heartbeat events actually received for each node in the cluster (excluding the collector host), since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.missed`	The number of heartbeat events determined not to have been received for each node in the cluster (excluding the collector host), since the time `ha_cmsd`(1M) has started.

Metric	Description
`fsafe.cms.recent.missed`	The number of heartbeat events determined not to have been received for each node in the cluster (excluding the collector host), since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.histo`	Histogram of heartbeat event response times for events that have occurred within discrete heartbeat response intervals for each node in the cluster (excluding the collector host), since the time `ha_cmsd`(1M) has started. The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (`fsafe.config.cms.interval`), for a number of intervals up to the configured heartbeat event timeout (`fsafe.config.cms.timeout`).
`fsafe.cms.recent.histo`	Histogram of heartbeat event response times for events that have occurred within discrete heartbeat response intervals for each node in the cluster (excluding the collector host), since a data collection reset (via `fsafe.control.reset_cms`). The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (`fsafe.config.cms.interval`), for a number of intervals up to the configured heartbeat event timeout (`fsafe.config.cms.timeout`).
`fsafe.cms.frac_received`	Fraction of heartbeat events received over all expected events for each node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.frac_received`	Fraction of heartbeat events received over all expected events for each node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.frac_missed`	Fraction of heartbeat events determined not to have been received over all expected events for each node in the cluster, since the time `ha_cmsd`(1M) has started.

Metric	Description
`fsafe.cms.recent.frac_missed`	Fraction of heartbeat events determined not to have been received over all expected events for each node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.recent.timestamp`	The time when a new collection of statistics was started for the `fsafe.cms.recent.*` metrics, after issuing a store to the metric `fsafe.control.reset_cms`.
`fsafe.cms.pernode.expected`	The number of heartbeat events expected to have been received for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.pernode.expected`	The number of heartbeat events expected to have been received for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.pernode.received`	The number of heartbeat events actually received for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.pernode.received`	The number of heartbeat events actually received for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.pernode.missed`	The number of heartbeat events determined not to have been received for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.pernode.missed`	The number of heartbeat events determined not to have been received for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.pernode.histo`	Histogram of heartbeat event response times for events that have occurred within discrete heartbeat response intervals for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.

Metric	Description
	The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (`fsafe.config.cms.interval`), for a number of intervals up to the configured heartbeat event timeout (`fsafe.config.cms.timeout`).
`fsafe.cms.recent.pernode.histo`	Histogram of heartbeat event response times for events that have occurred within discrete heartbeat response intervals for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
	The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (`fsafe.config.cms.interval`), for a number of intervals up to the configured heartbeat event timeout (`fsafe.config.cms.timeout`).
`fsafe.cms.pernode.frac_received`	Fraction of heartbeat events received over all expected events for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.pernode.frac_received`	Fraction of heartbeat events received over all expected events for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.pernode.frac_missed`	Fraction of heartbeat events determined not to have been received over all expected events for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.pernode.frac_missed`	Fraction of heartbeat events determined not to have been received over all expected events for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).

Glossary

action scripts

The set of scripts that determine how a resource is started, monitored, and stopped. There must be a set of action scripts specified for each resource type. The possible set of action scripts is: `exclusive`, `start`, `stop`, `monitor`, and `restart`.

cluster

The set of nodes in the pool that have been defined as a cluster. A cluster is identified by a simple name; this name must be unique within the pool. All nodes in the cluster are also in the pool. However, all nodes in the pool are not necessarily in the cluster; that is, the cluster may consist of a subset of the nodes in the pool. There is only one cluster per pool.

cluster administrator

The person responsible for managing and maintaining a cluster.

cluster configuration database

Contains configuration information about all resources, resource types, resource groups, failover policies, nodes, and the cluster.

cluster process group

A group of application instances in a distributed application that cooperate to provide a service.

For example, distributed lock manager instances in each node would form a process group. By forming a process group, they can obtain membership and reliable, ordered, atomic communication services. There is no relationship between a UNIX process group and a cluster process group.

collector host

The nodes in the FailSafe cluster itself from which you want to gather statistics, on which PCP for FailSafe has installed the collector agents.

control messages

Messages that cluster software sends between the nodes to request operations on or distribute information about nodes and resource groups. IRIS FailSafe sends control messages for the purpose of ensuring that nodes and groups remain highly available. Control messages and heartbeat messages are sent through a node's network interfaces that have been attached to a control network. A node can be attached to multiple control networks.

control network

The network that connects nodes through their network interfaces (typically Ethernet) such that FailSafe can maintain a cluster's high availability by sending heartbeat messages and control messages through the network to the attached nodes. FailSafe uses the highest priority network interface on the control network; it uses a network interface with lower priority when all higher-priority network interfaces on the control network fail.

A node must have at least one control network interface for heartbeat messages and one for control messages (both heartbeat and control messages can be configured to use the same interface). A node can have no more than eight control network interfaces.

database

See *cluster configuration database*

dependency list

See *resource dependency* or *resource type dependency*.

failover

The process of allocating a *resource group* to another *node* according to a *failover policy* A failover may be triggered by the failure of a resource, a change in the FailSafe membership (such as when a node fails or starts), or a manual request by the administrator.

failover attribute

A string that affects the allocation of a resource group in a cluster. The administrator must specify system-defined attributes (such as `Auto_Failback` or `Controlled_Failback`), and can optionally supply site-specific attributes.

failover domain

The ordered list of nodes on which a particular *resource group* can be allocated. The nodes listed in the failover domain must be within the same cluster; however, the failover domain does not have to include every node in the cluster. The administrator defines the *initial failover domain* when creating a failover policy. This list is transformed into the *run-time failover domain* by the *failover script* the run-time failover domain is what is actually used to select the failover node. FailSafe stores the run-time failover domain and uses it as input to the next failover script invocation. The initial and run-time failover domains may be identical, depending upon the contents of the failover script. In general, FailSafe allocates a given resource group to the first node listed in the run-time failover domain that is also in the FailSafe membership; the point at which this allocation takes place is affected by the *failover attributes*.

failover policy

The method used by FailSafe to determine the destination node of a failover. A failover policy consists of a *failover domain*, *failover attributes*, and a *failover script*. A failover policy name must be unique within the *pool*.

failover script

A failover policy component that generates a *run-time failover domain* and returns it to the FailSafe process. The process applies the failover attributes and then selects the first node in the returned failover domain that is also in the current FailSafe membership.

FailSafe membership

The list of FailSafe nodes in a cluster on which FailSafe can make resource groups online. It differs from the CXFS membership. For more information about CXFS, see *CXFS Software Installation and Administration Guide*.

FailSafe database

See *cluster configuration database*

`fs2d` membership

Also known as *user-space membership*. The group of nodes in the pool that are accessible to `fs2d` and therefore can receive cluster configuration database updates; this may be a subset of the nodes defined in the pool.

heartbeat messages

Messages that cluster software sends between the nodes that indicate a node is up and running. Heartbeat messages and *control messages* are sent through a node's network interfaces that have been attached to a control network. A node can be attached to multiple control networks.

heartbeat interval

Interval between heartbeat messages. The node timeout value must be at least 10 times the heartbeat interval for proper FailSafe operation (otherwise false failovers may be triggered). The higher the number of heartbeats (smaller heartbeat interval), the greater the potential for slowing down the network. Conversely, the fewer the number of heartbeats (larger heartbeat interval), the greater the potential for reducing availability of resources.

initial failover domain

The ordered list of nodes, defined by the administrator when a failover policy is first created, that is used the first time a cluster is booted. The ordered list specified by the initial failover domain is transformed into a *run-time failover domain* by the *failover script*; the run-time failover domain is used along with failover attributes to determine the node on which a resource group should reside. With each failure, the failover script takes the current run-time failover domain and potentially modifies it; the initial failover domain is never used again. Depending on the run-time conditions and contents of the failover script, the initial and run-time failover domains may be identical. See also *run-time failover domain*.

key/value attribute

A set of information that must be defined for a particular resource type. For example, for the resource type `filesystem` one key/value pair might be *mount_point=/fs1* where *mount_point* is the key and *fs1* is the value specific to the particular resource being defined. Depending on the value, you specify either a `string` or `integer` data type. In the previous example, you would specify `string` as the data type for the value *fs1*.

log configuration

A log configuration has two parts: a *log level* and a *log file*, both associated with a *log group*. The cluster administrator can customize the location and amount of log output, and can specify a log configuration for all nodes or for only one node. For example, the `crsd` log group can be configured to log detailed level-10 messages to the

`/var/cluster/ha/log/crsd-foo` log only on the node `foo` and to write only minimal level-1 messages to the `crsd` log on all other nodes.

log file

A file containing notifications for a particular *log group*. A log file is part of the *log configuration* for a log group. By default, log files reside in the `/var/cluster/ha/log` directory, but the cluster administrator can customize this. Note: FailSafe logs both normal operations and critical errors to `/var/adm/SYSLOG`, as well as to individual logs for specific log groups.

log group

A set of one or more FailSafe processes that use the same log configuration. A log group usually corresponds to one daemon, such as `gcd`.

log level

A number controlling the number of log messages that FailSafe will write into an associated log group's log file. A log level is part of the log configuration for a log group.

monitor host

A workstation that has a display and is running the IRIS Desktop, on which PCP for FailSafe has installed the monitor client.

node

A single IRIX kernel image. Usually, a node is an individual computer. This use of the term node does not have the same meaning as a node in an Origin system.

node ID

A 16-bit positive integer that uniquely defines a node. During node definition, FailSafe will assign a node ID if one has not been assigned by the cluster administrator. Once assigned, the node ID cannot be modified.

node timeout

If no heartbeat is received from a node in this period of time, the node is considered to be dead. The node timeout value must be at least 10 times the heartbeat interval for proper FailSafe operation (otherwise false failovers may be triggered).

notification command

The command used to notify the cluster administrator of changes or failures in the cluster, nodes, and resource groups. The command must exist on every node in the cluster.

offline resource group

A resource group that is not highly available in the cluster. To put a resource group in offline state, FailSafe stops the group (if needed) and stops monitoring the group. An offline resource group can be running on a node, yet not under FailSafe control. If the cluster administrator specifies the *detach only* option while taking the group offline, then FailSafe will not stop the group but will stop monitoring the group.

online resource group

A resource group that is highly available in the cluster. When FailSafe detects a failure that degrades the resource group availability, it moves the resource group to another node in the cluster. To put a resource group in online state, FailSafe starts the group (if needed) and begins monitoring the group. If the cluster administrator specifies the *attach only* option while bringing the group online, then FailSafe will not start the group but will begin monitoring the group.

owner host

A system that can control a node remotely, such as power-cycling the node. At run time, the owner host must be defined as a node in the pool.

owner TTY name

The device file name of the terminal port (TTY) on the *owner host* to which the system controller serial cable is connected. The other end of the cable connects to the node with the system controller port, so the node can be controlled remotely by the owner host.

pool

The entire set of nodes that are coupled to each other by networks and are defined as nodes in FailSafe. The nodes are usually close together and should always serve a common purpose. A replicated cluster configuration database is stored on each node in the pool.

All nodes that can be added to a cluster are part of the pool, but not all nodes in the pool must be part of the cluster. There is only one pool. Other pools may exist, but each is disjoint from the other. They share no node or cluster definitions.

port password

The password for the system controller port, usually set once in firmware or by setting jumper wires. (This is not the same as the node's root password.)

powerfail mode

When powerfail mode is turned `on`, FailSafe tracks the response from a node's system controller as it makes reset requests to a node. When these requests fail to reset the node successfully, FailSafe uses heuristics to try to estimate whether the machine has been powered down. If the heuristic algorithm returns with success, FailSafe assumes the remote machine has been reset successfully. When powerfail mode is turned `off`, the heuristics are not used and FailSafe may not be able to detect node power failures.

process membership

A list of process instances in a cluster that form a process group. There can multiple process groups per node.

resource

A single physical or logical entity that provides a service to clients or other resources. For example, a resource can be a single disk volume, a particular network address, or an application such as a web server. A resource is generally available for use over time on two or more nodes in a *cluster*, although it can be allocated to only one node at any given time. Resources are identified by a *resource name* and a *resource type*. Dependent resources must be part of the same *resource group* and are identified in a *resource dependency list*

resource dependency

The condition in which a resource requires the existence of other resources.

resource dependency list

A list of resources upon which a resource depends. Each resource instance must have resource dependencies that satisfy its resource type dependencies before it can be added to a resource group.

resource group

A collection of resources. A resource group is identified by a simple name; this name must be unique within a cluster. Resource groups cannot overlap; that is, two resource groups cannot contain the same resource. All interdependent resources must be part of the same resource group. If any individual resource in a resource group becomes unavailable for its intended use, then the entire resource group is considered unavailable. Therefore, a resource group is the unit of failover.

resource keys

Variables that define a resource of a given resource type. The action scripts use this information to start, stop, and monitor a resource of this resource type.

resource name

The simple name that identifies a specific instance of a *resource type*. A resource name must be unique within a given resource type.

resource type

A particular class of *resource*. All of the resources in a particular resource type can be handled in the same way for the purposes of *failover*. Every resource is an instance of exactly one resource type. A resource type is identified by a simple name; this name must be unique within a cluster. A resource type can be defined for a specific node or for an entire cluster. A resource type that is defined for a node overrides a cluster-wide resource type definition with the same name; this allows an individual node to override global settings from a cluster-wide resource type definition.

resource type dependency

A set of resource types upon which a resource type depends. For example, the *filesystem* resource type depends upon the *volume* resource type, and the *Netscape_web* resource type depends upon the *filesystem* and *IP_address* resource types.

resource type dependency list

A list of resource types upon which a resource type depends.

run-time failover domain

The ordered set of nodes on which the resource group can execute upon failures, as modified by the *failover script*. The run-time failover domain is used along with

failover attributes to determine the node on which a resource group should reside. See also *initial failover domain*.

start/stop order

Each resource type has a start/stop order, which is a nonnegative integer. In a resource group, the start/stop orders of the resource types determine the order in which the resources will be started when FailSafe brings the group online and will be stopped when FailSafe takes the group offline. The group's resources are started in increasing order, and stopped in decreasing order; resources of the same type are started and stopped in indeterminate order. For example, if resource type *volume* has order 10 and resource type *filesystem* has order 20, then when FailSafe brings a resource group online, all volume resources in the group will be started before all file system resources in the group.

system controller port

A port located on a node that provides a way to power-cycle the node remotely. Enabling or disabling a system controller port in the cluster configuration database (CDB) tells FailSafe whether it can perform operations on the system controller port. (When the port is enabled, serial cables must attach the port to another node, the owner host.) System controller port information is optional for a node in the pool, but is required if the node will be added to a cluster; otherwise resources running on that node never will be highly available.

tie-breaker node

A node identified as a tie-breaker for FailSafe to use in the process of computing the FailSafe membership for the cluster, when exactly half the nodes in the cluster are up and can communicate with each other. If a tie-breaker node is not specified, FailSafe will use the node with the lowest node ID in the cluster as the tie-breaker node.

type-specific attribute

Required information used to define a resource of a particular resource type. For example, for a resource of type *filesystem* you must enter attributes for the resource's volume name (where the file system is located) and specify options for how to mount the file system (for example, as readable and writable).

user-space membership

See `fs2d` *membership*

Index

H

I

L

M

N

W

X