IRIS FailSafe™ 2.0 Administrator's Guide

Document Number 007-3901-002

CONTRIBUTORS

Written by Steven Levine, Susan Ellis
Illustrated by Dany Galgani
Production by Amy Swenson, Glen Traefald
Engineering contributions by Ashwinee Khaladkar, Michael Nishimoto, Wesley Smith, Bill Sparks, Paddy Sreenivasan, Dan Stekloff, Rebecca Underwood, Manish Verma
St. Peter's Basilica image courtesy of ENEL SpA and InfoByte SpA. Disk Thrower image courtesy of Xavier Berenguer, Animatica.

IRIS FailSafe™ 2.0 Administrator's Guide
Document Number 007-3901-002

What's New in This Guide

This revision contains the following:

- Updated information on configuring log files in"FailSafe System Log Configuration"
- Updated example of *haStatus* CLI script output in "Viewing System Status with the haStatus CLI Script"
- Resource group creation example in "Resource Group Creation Example"
- Clarification of character restrictions of FailSafe component names
- A description of the */etc/config/cmond.options* configuration file in "Configuring /etc/config/cmond.options for FailSafe"
- A caution has been added advising the administrator that data corruption can occur if a resource group is brought online while it is running on a disabled node (where HA services are not running)
- Clarification of use of **-f** and **-i** options used together in first line of a FailSafe *cmgr* script
- A description of the new "experimental mode" you can use when you define or modify a cluster in "Defining a Cluster"
- Updated summary description of failover policies
- Some minor clarifications throughout manual
- General minor updates to text describing GUI screens

Record of Revision

Version	Description
002	December 1999 Published in conjunction with FailSafe 2.0 rollup patch. Supports IRIX 6.5.2 and later.

Contents

List of Figures

List of Tables

About This Guide

This guide describes the configuration and administration of an IRIS FailSafe™ 2.0 high-availability system.

This guide was prepared in conjunction with Release 2.0 of the IRIS FailSafe product.

Audience

The *IRIS FailSafe 2.0 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 CHALLENGE® or Origin™ servers, as well as optional CHALLENGE Vault, Origin Vault, Fibre Channel RAID and JBOD, or CHALLENGE RAID storage systems, whichever is used in the IRIS FailSafe configuration. Good knowledge of XLV and XFS™ is also required.

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 2.0 System," introduces the components of the IRIS FailSafe system and explains its hardware and software architecture.
- Chapter 2, "Planning IRIS FailSafe 2.0 Configuration," describes how to plan the configuration of an IRIS FailSafe cluster.
- Chapter 3, "Installing IRIS FailSafe 2.0 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 2.0 Administration Tools," describes the cluster manager tools you can use to administer and IRIS FailSafe 2.0 system.

- Chapter 5, “IRIS FailSafe 2.0 Configuration,” explains how to perform the administrative tasks to configure a FailSafe 2.0 system.
- Chapter 6, “IRIS FailSafe 2.0 System Operation,” explains how to perform the administrative tasks to operate and monitor a FailSafe 2.0 system.
- Chapter 7, “Testing IRIS FailSafe 2.0 Configuration,” describes how to test the configured IRIS FailSafe system.
- Chapter 8, “IRIS FailSafe 2.0 Recovery,” describes the log files used by FailSafe and how to evaluate problems in a FailSafe system.
- Chapter 9, “Upgrading and Maintaining Active Clusters,” describes some procedures you may need to perform without shutting down a FailSafe cluster.
- Appendix A, “Updating from IRIS FailSafe 1.2 to IRIS FailSafe 2.0,” provides a description of the procedures you perform to upgrade a system from IRIS FailSafe 1.2 to IRIS FailSafe 2.0.
- Appendix B, “IRIS FailSafe 2.0 Software,” summarizes the systems to install on each component of a cluster or node.

Related Documentation

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

- *IRIS FailSafe 2.0 Programmer’s Guide*
- *IRIS FailSafe 2.0 Oracle Administrator’s Guide* (IRIS FailSafe Oracle® option)
- *IRIS FailSafe 2.0 INFORMIX Administrator’s Guide* (IRIS FailSafe INFORMIX® option)
- *IRIS FailSafe 2.0 Netscape Server Administrator’s Guide* (IRIS FailSafe Netscape Web® option)
- *IRIS FailSafe 2.0 NFS Administrator’s Guide* (IRIS FailSafe NFS™ option)

The IRIS FailSafe reference pages are as follows:

- *cbeutil*(1M)
- *cdbBackup*(1M)
- *cdbRestore*(1M)
- *cdbutil*(1M)

- *cluster_mgr*(1M)
- *crsd*(1M)
- *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)
- *ha_exec2*(1M)
- *failsafe*(7M)

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

Table i IRIS FailSafe Release Notes

Release Note	Product
failsafe2	IRIS FailSafe 2.0
failsafe2_nfs	IRIS FailSafe NFS
failsafe2_www	IRIS FailSafe Netscape Web
failsafe2_ifmx_db	IRIS FailSafe INFORMIX
failsafe2_orcl_db	IRIS FailSafe Oracle
failsafe2_sybs_db	IRIS FailSafe Sybase

Table i (continued) IRIS FailSafe Release Notes

Release Note	Product
cluster_ha	Cluster high availability services
cluster_admin	Cluster administration services
cluster_control	Cluster node control services

Conventions Used in This Guide

These type conventions and symbols are used in this guide:

Bold Literal command-line arguments and literal parameter values

Italics Command names, filenames, new terms, the names of *inst* subsystems, manual/book titles, variable command-line arguments, and variables to be supplied by the user in examples, code, and syntax statements

`Fixed-width type`
Examples of command output that is displayed in windows on your monitor and of the contents of files

`Bold fixed-width type`
Commands and text that you are to type literally in response to shell and command prompts

`#` IRIX® shell prompt for the superuser (*root*)

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 frontmatter of the manual. In printed manuals, the document number can be found on the back cover.)

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 Amphitheater 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 2.0 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.

- High-availability 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 high-availability services. The IRIS FailSafe 2.0 release provides high-availability services for a cluster that contains multiple nodes (N-node configuration). Using IRIS FailSafe 2.0, you can configure a high-availability 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 high-availability 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 high-availability service.

In an IRIS FailSafe high-availability system, 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 high-availability system can be used during normal operation to run other applications that are not necessarily high-availability services. All high-availability services are owned and accessed by one node at a time.

High-availability 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 2.0, 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 2.0 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 high-availability services fall into two groups: high-availability resources and high-availability applications. High-availability 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 high-availability applications. This optional software includes

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

IRIS FailSafe 2.0 provides a framework for making additional applications into high-availability services. If you want to add high-availability 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 2.0 Programmer's Guide* (007-3900-001).

IRIS FailSafe 2.0 System Components and Concepts

An IRIS FailSafe 2.0 system is a *cluster* system, which consists of various components with defined interrelationships. In order to use IRIS FailSafe 2.0 to configure and monitor high-availability 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 2.0 system; all system administration tasks are based on these concepts.

Cluster Node (or Node)

A *cluster node* is a single UNIX® image. Usually, a cluster 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.

Pool

A *pool* is the entire set of *nodes* involved with a group of clusters. The group of clusters are usually close together and should always serve a common purpose. A replicated database is stored on each node in the pool.

Cluster

A *cluster* is a collection of one or more *nodes* coupled with each other by networks or other similar interconnects. In IRIS FailSafe 2.0, a cluster is identified by a simple name. A given node may be a member of only one cluster. All nodes in a cluster are also in the pool; however, all nodes in the pool are not necessarily in the cluster.

Node Membership

A *node membership* is the list of nodes in a cluster on which IRIS FailSafe can allocate resource groups.

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*. A resource name must be unique for a given 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 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 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.

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 IRIS FailSafe software includes many predefined resource types. If these types fit the application you want to make into a high-availability service, you can reuse them. If none fits, you can create additional resource types by following the procedures described in the *IRIS FailSafe 2.0 Programmer's Guide*.

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 for a resource group named *WebGroup.*

Table 1-1 Example Resource Group

Resource	Resource Type
vol1	*volume*
/fs1	*filesystem*
199.10.48.22	*IP_address*
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 for IRIS FailSafe.

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*. IRIS 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 node 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 IRIS FailSafe to determine the destination node of a failover. A failover policy consists of the following:

- Failover domain
- Failover attributes
- Failover script

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

For each resource group that you define for an IRIS FailSafe 2.0 system, you can specify the failover policy to apply to that group to determine which nodes will take over under what circumstances. FailSafe 2.0 includes pre-defined failover policies, but you can define your own failover algorithm as well.

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 *runtime failover domain* by the *failover script*; IRIS FailSafe uses the runtime failover domain along with failover attributes and the node membership to determine the node on which a resource group should reside. IRIS FailSafe stores the runtime failover domain and uses it as input to the next failover script invocation. Depending on the runtime conditions and contents of the failover script, the initial and runtime failover domains may be identical.

In general, IRIS FailSafe allocates a given resource group to the first node listed in the runtime failover domain that is also in the node 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 *AutoFailback* or *ControlledFailback*), and can optionally supply site-specific attributes.

Failover Script

A *failover script* is a shell script that generates a *runtime failover domain* and returns it to the IRIS FailSafe process. The IRIS FailSafe process applies the failover attributes and then selects the first node in the returned failover domain that is also in the current node membership.

The *ordered* failover script is provided with the IRIS FailSafe release. This script does not change the order of the *initial failover domain*. If this script does not meet your needs, you can create a new failover script using the information in this guide.

Additional IRIS FailSafe 2.0 Features

IRIS FailSafe 2.0 provides the following features to increase the flexibility and ease of operation of a high-availability system:

- dynamic management
- fine grain failover
- local restarts

These features are summarized in the following sections.

Dynamic Management

FailSafe 2.0 allows you to perform a variety of administrative tasks while the system is running:

- Dynamically managed application monitoring

  FailSafe 2.0 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 2.0 system.

- Dynamically managed FailSafe resources

  FailSafe 2.0 allows you to add resources while the FailSafe system is online.

- Dynamically managed FailSafe upgrades

  FailSafe 2.0 allows you to upgrade FailSafe software on one node at a time without taking down the entire FailSafe cluster.

Fine Grain Failover

Using FailSafe 2.0, you can specify *fine-grain failover*. Fine-grain failover is a 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, where possible. Fine-grain failover is possible in FailSafe 2.0 because the unit of failover is the resource group, and not the entire node.

Local Restarts

FailSafe 2.0 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 2.0 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 high-availability 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 2.0 Administration Tools.”

For information on IRIS FailSafe configuration and administration tasks, see Chapter 5, “IRIS FailSafe 2.0 Configuration” and Chapter 6, “IRIS FailSafe 2.0 System Operation.”

Hardware Components of an IRIS FailSafe 2.0 Cluster

Figure 1-1 shows an example of IRIS FailSafe 2.0 hardware components, in this case for a two-node system.

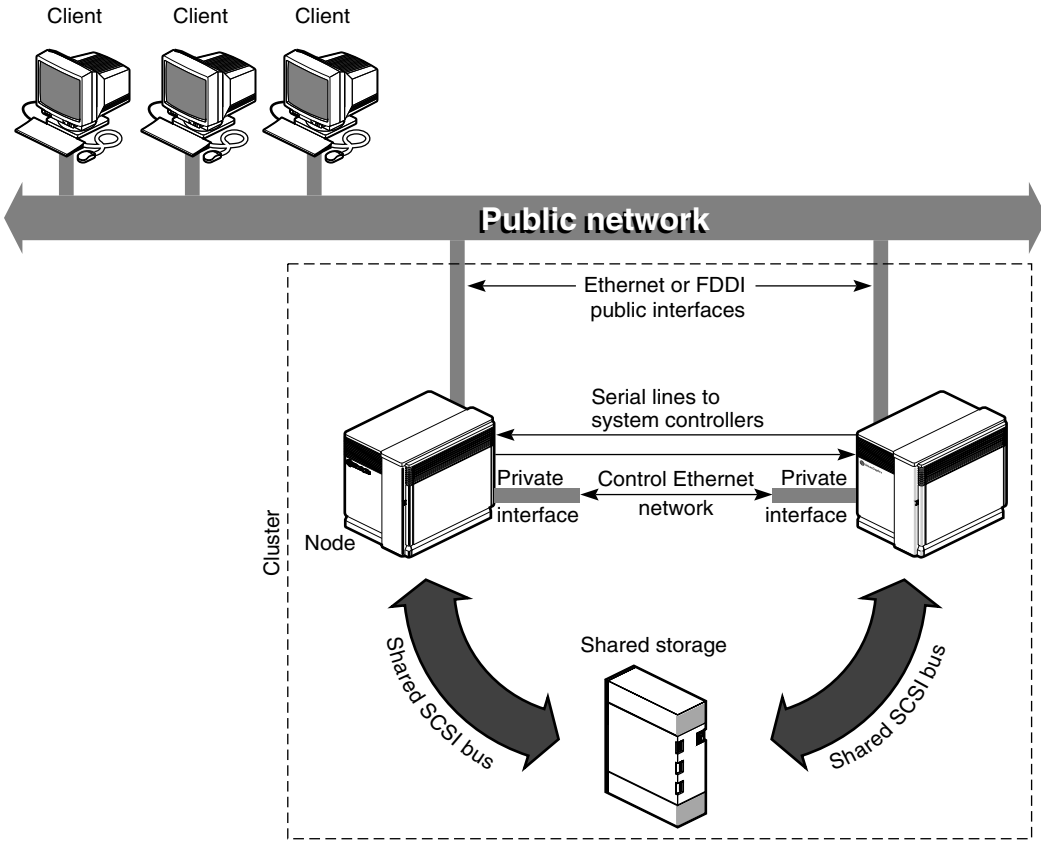

Figure 1-1 Sample IRIS FailSafe System Components

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

- Up to eight CHALLENGE/ Origin nodes
- More than two interfaces on each node to control networks (Ethernet or FDDI for CHALLENGE nodes and Ethernet or FDDI for Origin nodes)

  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:

  – CHALLENGE Vault peripheral enclosures with SCSI disks (CHALLENGE and Origin nodes)

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

  – FibreVault peripheral enclosures with SCSI disks (Origin nodes only)

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

An IRIS FailSafe 2.0 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 2.0 Supported Configurations

IRIS FailSafe 2.0 supports the following high-availability 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 2.0 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.

The following sections provide descriptions of the different IRIS FailSafe 2.0 configurations.

Basic Two-Node Configuration

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

- All high-availability 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 high-availability services.
- High-availability 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 high-availability 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 2.0 system handles or eliminates these points of failure.

Network Interfaces and IP Addresses

Clients access the high-availability services provided by the IRIS FailSafe 2.0 cluster using IP addresses. Each high-availability service can use multiple IP addresses. The IP addresses are not tied to a particular high-availability service; they can be shared by all the high-availability services in the cluster.

IRIS FailSafe 2.0 uses the IP aliasing mechanism to support multiple IP addresses on a single network interface. Clients can use a high-availability 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 2.0 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 2.0 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 *high-availability IP addresses*.

High-availability 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. High-availability 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 2.0, 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" in Chapter 2 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 2.0 can include shared SCSI-based storage in the form of one or more CHALLENGE RAID storage systems (for CHALLENGE or Origin nodes) or it can include Origin FibreVault RAID storage systems. It can also include CHALLENGE Vaults (for CHALLENGE or Origin2000 nodes only) or FibreVault peripheral enclosures with SCSI disks (Origin nodes only) with plexed disks. All data for high-availability applications must be stored in XLV logical volumes on shared disks. If high-availability applications use filesystems, XFS filesystems must be used.

For CHALLENGE RAID or 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 CHALLENGE Vault or 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.

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.

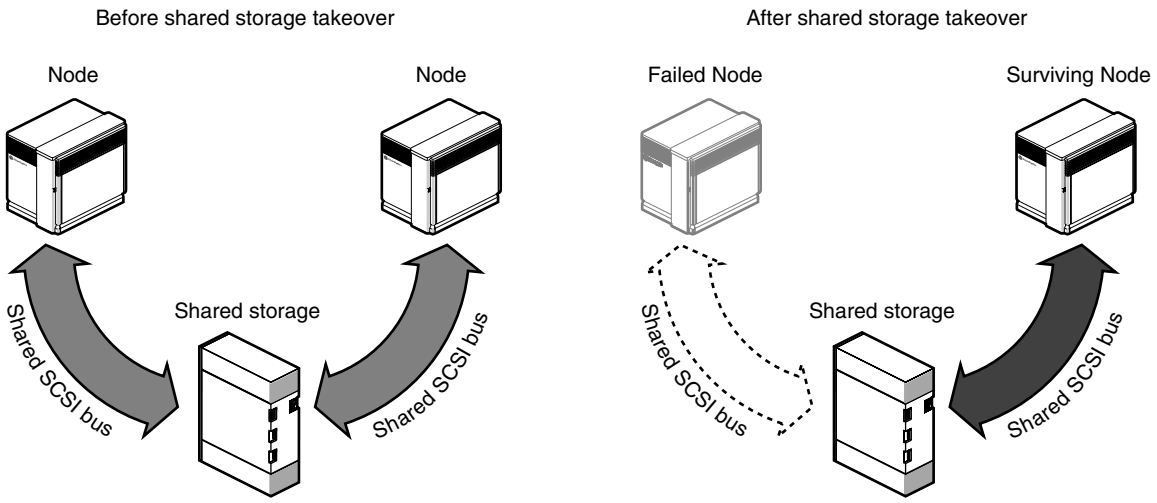

Figure 1-2 Disk Storage Failover on a Two-Node System

High-Availability 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 high-availability resources or high-availability application is detected by IRIS FailSafe 2.0 software, all high-availability resources in the affected resource group on the failed node are failed over to a different node and the high-availability applications on the failed node are stopped. When these operations are complete, the high-availability applications are started on the backup node.

All information about high-availability 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 2.0 Configuration." Monitoring scripts detect the failure of a high-availability application.

The IRIS FailSafe software provides a framework for making applications high-availability services. By writing scripts and configuring the system in accordance with those scripts, you can turn client/server applications into high-availability applications. For information, see the *IRIS FailSafe 2.0 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 high-availability service is no longer operating), a different node performs a failover of the high-availability services that are being provided on the node with the failure (called the *failed node*). Failover allows all of the high-availability services, including those provided by the failed node, to remain available within the cluster.

A failure in a high-availability service can be detected by IRIS FailSafe 2.0 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 high-availability resource group running on the node
- moves the high-availability 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 2.0 software running on a different node, the node detecting the failure performs these operations:

- using the serial connection between the nodes, reboots 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 high-availability services again depends on the failover policy you define. For information on defining failover policies, see "Defining a Failover Policy" on page 111 in Chapter 5.

Normally, a node that experiences a failure automatically reboots and resumes providing high-availability services. This scenario works well for transient errors (as well as for planned outages for equipment and software upgrades). However, if there are persistent errors, automatic reboot can cause recovery and an immediate failover again. To prevent this, the IRIS FailSafe 2.0 software checks how long the rebooted node has been up since the last time it was started. If the interval is less than five minutes (by default), the IRIS FailSafe software automatically does a chkconfig failsafe off on the failed node and does not start up the IRIS FailSafe 2.0 software on this node. It also writes error messages to */var/adm/SYSLOG* and to the appropriate log file.

Overview of Configuring and Testing a New IRIS FailSafe 2.0 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 2.0 terms by reviewing this chapter.
2. Plan the configuration of high-availability applications and services on the cluster using Chapter 2, "Planning IRIS FailSafe 2.0 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 2.0 Software and Preparing the System."
4. Define the IRIS FailSafe configuration as explained in Chapter 5, "IRIS FailSafe 2.0 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.

Chapter 2

Planning IRIS FailSafe 2.0 Configuration

This chapter explains how to plan the configuration of high-availability services on your IRIS FailSafe 2.0 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 2.0 cluster, and based on that, how the disks and interfaces must be set up to meet the needs of the high-availability 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 high-availability service?

  The IRIS FailSafe 2.0 NFS option enables you to provide exported NFS filesystems as high-availability services. Similarly, the IRIS FailSafe 2.0 Web option is used for the Netscape FastTrack and Enterprise Servers, the IRIS FailSafe 2.0 INFORMIX option is used for Informix databases, and the IRIS FailSafe 2.0 Oracle option is used for Oracle databases. To offer other applications as high-availability services, you must develop a set of shell scripts that provide switch over and switch back functionality. Developing these scripts is described in the *IRIS FailSafe 2.0 Programmer's Guide*.

- Which node will be the primary node for each high-availability 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 high-availability 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 2.0; filesystems must be XFS filesystems. The choice of volumes or filesystems depends on the application that is going to use the disk space.

- Which IP addresses will be used by clients of high-availability 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 depended 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 high-availability IP addresses on each network interface will be available to clients of the high-availability services?

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

- Which IP addresses on primary nodes are going to be available to clients of the high-availability services?
- For each high-availability IP address that is available on a primary node to clients of high-availability services, which interface on the other nodes will be assigned that IP address after a failover?

  Every high-availability IP address used by a high-availability service must be mapped to at least one interface in each node that can take over the resource group service. The high-availability 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 2.0 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 high-availability 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 CHALLENGE RAID 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 2.0, and collect the information you need to perform the configuration tasks described in Chapter 3, “Installing IRIS FailSafe 2.0 Software and Preparing the System,” and Chapter 5, “IRIS FailSafe 2.0 Configuration.”

Disk Configuration

The first subsection below describes the disk configuration issues that must be considered when planning an IRIS FailSafe 2.0 system. It explains the basic configurations of shared and non-shared disks and how they are reconfigured by IRIS FailSafe 2.0 after a failover. The second subsection explains how disk configurations are specified when you configure the IRIS FailSafe 2.0 system.

Planning Disk Configuration

For each disk in an IRIS FailSafe 2.0 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 2.0 cluster must follow these requirements:

- The system disk must be a non-shared disk.
- The IRIS FailSafe 2.0 software, in particular the directory */var/ha*, must be on a non-shared disk.

Choosing to make a disk shared or non-shared depends on the needs of the high-availability services that use the disk. Each high-availability 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 2.0 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 2.0 cluster, each of which has a non-shared disk with two resource groups. When non-shared disks are used by high-availability 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.

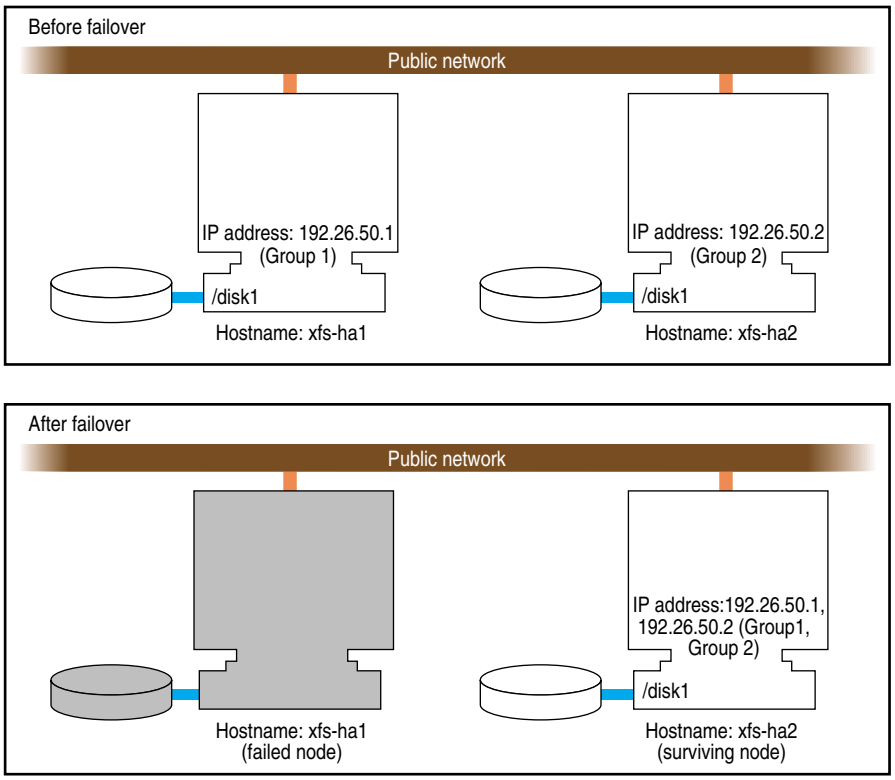

Figure 2-1 Non-Shared Disk Configuration and Failover

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 high-availability resource groups until a failover occurs.

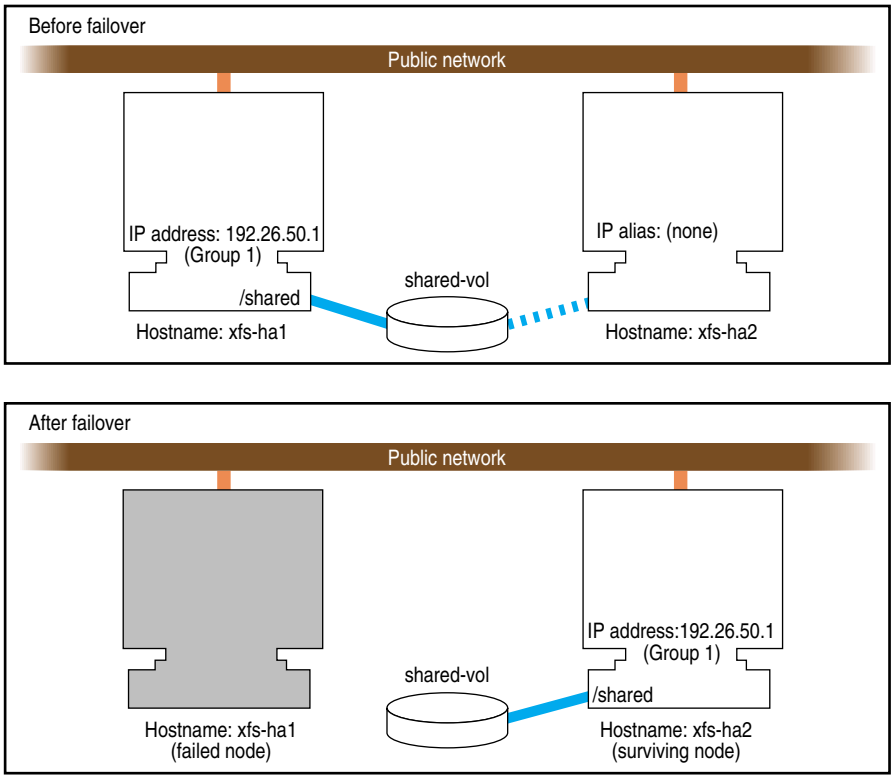

Figure 2-2 Shared Disk Configuration for Active/Backup Use

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.

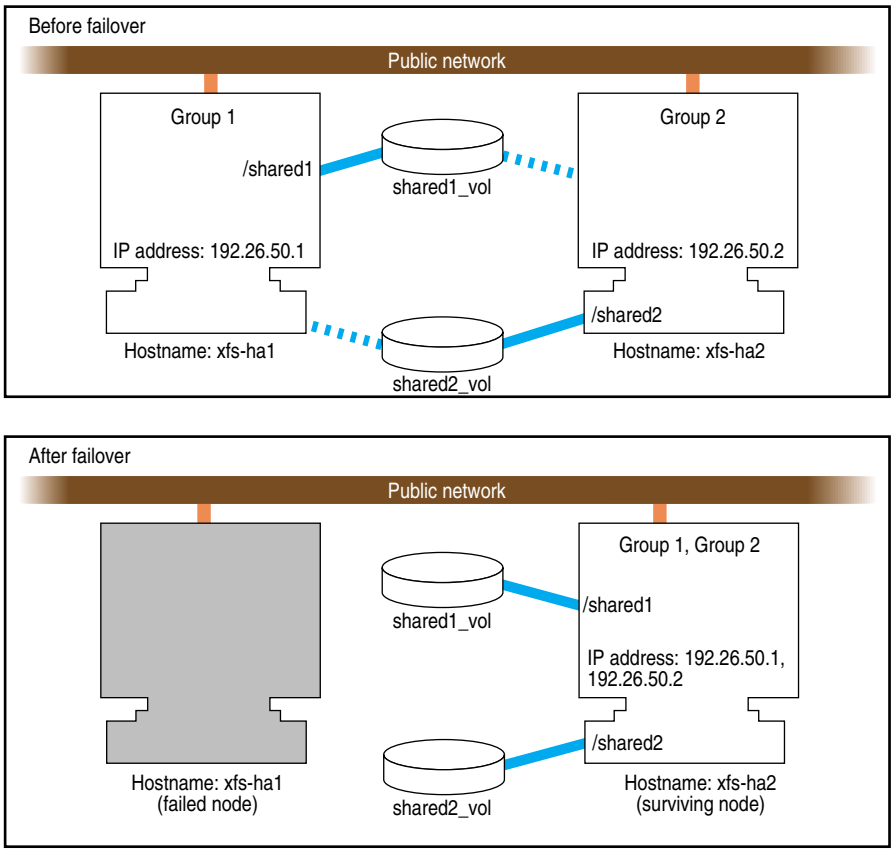

Figure 2-3 Shared Disk Configuration For Dual-Active Use

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 high-availability 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 2.0 system. Only shared disks (actually, the XLV logical volumes on shared disks) are specified at configuration. See the section "Configuration Parameters for Logical Volumes" on page 31 for details.

Logical Volume Configuration

The first subsection below describes logical volume issues that must be considered when planning an IRIS FailSafe 2.0 system. The second subsection gives an example of an XLV logical volume configuration on an IRIS FailSafe 2.0 system. The third subsection explains the aspects of the configuration that must be specified for an IRIS FailSafe 2.0 system.

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 2.0 configuration, you must follow these rules:

- All data that is used by high-availability 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 2.0 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" in Chapter 3 for more information about XLV volume ownership.)
- Each disk in a CHALLENGE/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 2.0 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 for IRIX 6.4 and IRIX 6.5
- *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 6.4 and IRIX 6.5. On the peer ha2, however, the same raw character volume appears as */dev/rxlv/ha1.vol1* on IRIX 6.4 or 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 2.0 software does not operate correctly.

The IRIS FailSafe 2.0 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 6.4 or 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” in Chapter 3 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 2.0 system. The second subsection gives an example of an XFS filesystem configuration on an IRIS FailSafe 2.0 system. The third subsection explains the aspects of the configuration that must be specified for an IRIS FailSafe 2.0 system.

Planning Filesystems

The IRIS FailSafe 2.0 software supports the automatic failover of XFS filesystems on shared disks. Shared disks must be in CHALLENGE/Fibre Channel Vault or RAID storage systems that are shared between the nodes in the two-node IRIS FailSafe 2.0 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 2.0 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 2.0 cluster must be created on either mirrored disks (using the XLV plexing software) or on the CHALLENGE/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 2.0 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 2.0 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 2.0 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 2.0 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. 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" in Chapter 3 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 2.0 system. The second subsection gives an example of the configuration of network interfaces and IP addresses on an IRIS FailSafe 2.0 system. The third subsection explains the aspects of the configuration that must be specified for an IRIS FailSafe 2.0 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 high-availability 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. The primary IP address does not fail over.
- The hostname of a node cannot be a high-availability IP address.
- All IP addresses used by clients to access high-availability services must be part of the resource group to which the HA service belongs.
- If re-MACing is required, all of the high-availability IP addresses must have the same backup interface.

- Making good choices for high-availability IP addresses is important; these are the "hostnames" that will be used by users of the high-availability services, not the true hostnames of the nodes.
- Make a plan for publicizing the high-availability IP addresses to the user community, since users of high-availability services must use high-availability IP addresses instead of the output of the *hostname* command.
- High-availability IP addresses should not be configured in the */etc/config/netif.options* file.

Follow the procedure below to determine whether re-MACing is required (see the section "Network Interfaces and IP Addresses" in Chapter 1 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 2.0 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

Chapter 3

Installing IRIS FailSafe 2.0 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 2.0. These procedures assume that you have done the planning described in Chapter 2, "Planning IRIS FailSafe 2.0 Configuration."

The major sections in this chapter are as follows:

Overview of Configuring Nodes for IRIS FailSafe 2.0

Performing the system administration procedures required to prepare nodes for IRIS FailSafe 2.0 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 XLV logical volumes and XFS filesystems required by the high-availability 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 2.0 software starts up when they are rebooted.

To complete the configuration of nodes for IRIS FailSafe 2.0, you must configure the components of the IRIS FailSafe 2.0 system, as described in Chapter 5, "IRIS FailSafe 2.0 Configuration."

Installing Required Software

Note: The IRIS FailSafe 2.0 base CD requires about 25 MB.

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. For any system running IRIX 6.2, check for the latest POSIX patch set for IRIX pthreads support (included in later versions of IRIX).

5. 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_ha.sw.cli*
   - *cluster_control.sw.cli*
   - *failsafe2.sw.cli*
   - *sysadm_failsafe2.sw.server*
   - *cluster_admin.sw*
   - *cluster_control.sw*

   Note: For IRIX 6.2 systems, or IRIX systems 6.3 through 6.5 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 (IRIX 6.3 and later).

   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 2.0 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*

6. Install additional software on cluster nodes:

   On each node that is part of the cluster, install the following software, in the order given. This software is required for cluster nodes in addition to that listed in Step 5.

   - *cluster_ha.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*

7. 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 2.0 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 2.0 Cluster Manager GUI does not use them.

     If the Java plug-in is not installed when the IRIS FailSafe 2.0 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 2.0 Manager GUI is run from a browser, the browser is redirected to this site.

8. On the appropriate servers, install other optional software the customer may have ordered, such as storage management or network board software.
9. 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 2.0 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*.
10. Install recommended patches for IRIS FailSafe 2.0 as shown at http://failsafe.

For IRIS FailSafe 2.0, you must set the AutoLoad variable to Yes; this can be done when you set host SCSI IDs, as explained in "Setting NVRAM Variables" on page 46.

Note: For reference, Appendix B, "IRIS FailSafe 2.0 Software," summarizes systems to install on each component of a cluster or node.

Configuring System Files

When you install the FailSafe Software, there are some system file considerations you must take into account. This section describes the required and optional changes you make to the following files for every node in the pool:

- */etc/services*
- */etc/config/cad.options*
- */etc/config/fs2d.options*
- */etc/config/cmond.options*

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 Cluster 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/f32d.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*
*S*et 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 NVRAM Variables

During the hardware installation of IRIS FailSafe 2.0 nodes, two NVRAM variables must be set:

- The boot parameter AutoLoad must be set to **yes**. The IRIS FailSafe 2.0 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 2.0 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 you planned the XLV logical volumes and XFS filesystems to be used by high-availability applications on the cluster. You can create them by following the instructions in the guide *IRIX Admin: Disks and Filesystems*.

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

- If the shared disks are not in a CHALLENGE 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 high-availability applications that use the logical volume (see "Planning Logical Volumes" in Chapter 2). 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/rdsk/xlv* and */dev/dsk/xlv* on IRIX 6.2 and in */dev/rxlv* and */dev/xlv* on IRIX 6.4 and IRIX 6.5) 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 2.0 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 2.0. Thus, a system administrator who sees mounted IRIS FailSafe 2.0 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 2.0.
- Be sure to create the mount point directory for each filesystem on both nodes.

Configuring Network Interfaces

The procedure in this section describes how to configure the network interfaces on the nodes in an IRIS FailSafe 2.0 cluster. The example shown in Figure 3-1 is used in the procedure.

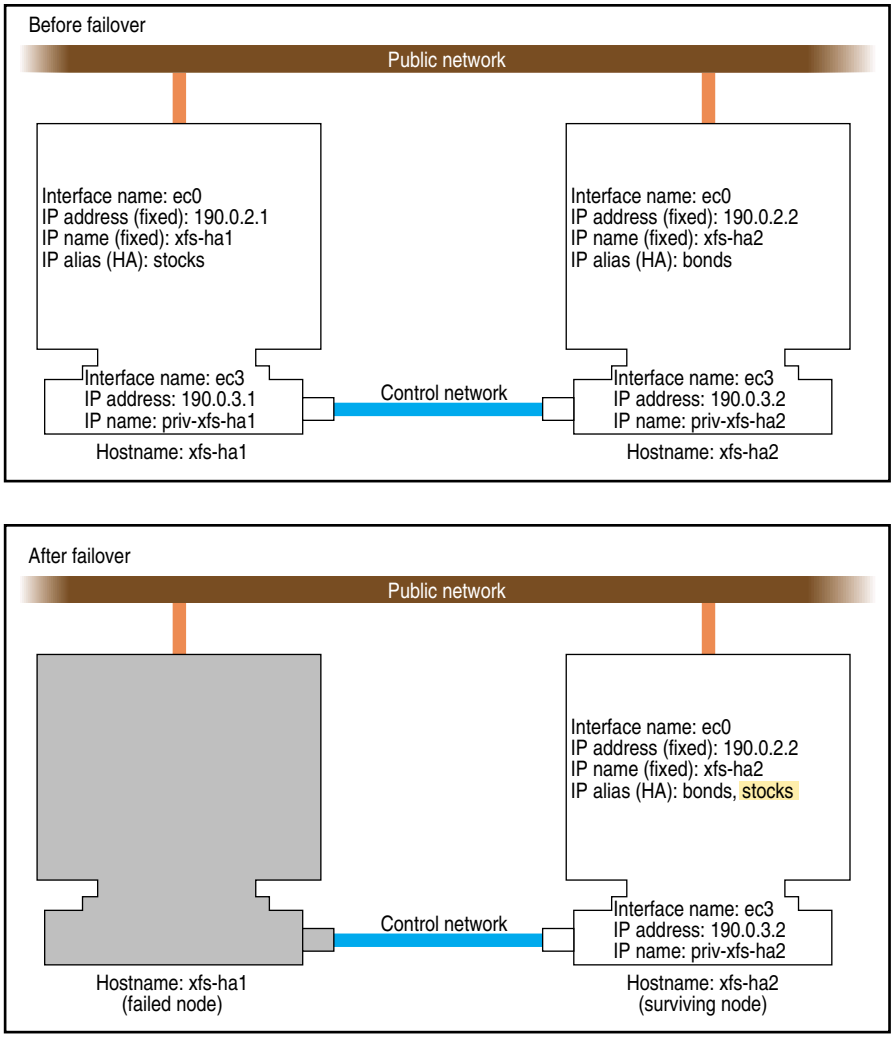

Figure 3-1 Example Interface Configuration

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 high-availability services are not added to the file */etc/config/ipaliases.options*. Similarly, if all IP aliases are used only by high-availability 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 Steps 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* (high availability 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 6.2 nodes is

   ```
   -h -q
   ```

   An example of the content of */etc/config/routed.options* on IRIX 6.4 or IRIX 6.5 nodes is

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

   Note: The `-q` option is required for IRIS FailSafe 2.0 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 2.0 is *chkconfig*'d **off** on each node:

```
# chkconfig | grep failsafe2
...
        failSafe2          off
...
```

If *Failsafe 2.0* 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 2.0 e-mail notifications of cluster transitions to a user outside the IRIS FailSafe 2.0 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
```

On IRIX 6.2 or 6.4 systems, create or modify the */etc/resolv.conf* file so that `local` is listed first for the hostresorder keyword (the order of `nis` and `bind` is up to you):

```
hostresorder local nis bind
```

Note: Exclusive use of NIS or DNS for IP address lookup for the cluster 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 both 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 2.0 system is running.

Chapter 4

IRIS FailSafe 2.0 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 2.0 administrative tasks using either of the following tools:

- The IRIS FailSafe 2.0 Cluster Manager Graphical User Interface (GUI)
- The IRIS FailSafe 2.0 Cluster Manager Command Line Interface (CLI)

Although these tools use the same underlying software to configure and monitor a FailSafe 2.0 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.

Using the IRIS FailSafe 2.0 Cluster Manager GUI

The IRIS FailSafe 2.0 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 Adminstration Guide*.

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
- Cluster 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 high availability 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 2.0 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.

  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 */usr/sbin/fstask*.

  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.

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 2.0 Cluster Manager CLI

This section documents how to perform IRIS FailSafe 2.0 administrative tasks by means of the IRIS FailSafe 2.0 Cluster Manager CLI. In order to execute commands with the IRIS FailSafe Cluster Manager CLI, you should be logged in as root.

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 IRIS FailSafe 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 2.0 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" on page 74):

```
set hostname to B
set nodeid to C
set sysctrl_type to D
set sysctrl_password to E
set sysctrl_status to F
set sysctrl_owner to G
set sysctrl_device to H
set sysctrl_owner_type to I
add nic J
```

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 A
Enter commands, when finished enter either "done" or "cancel"

Node Name [A]?

Hostname?
Node ID [0]?
Sysctrl Type <chalL|msc|mmsc>?
Sysctrl Password [ ]?
Sysctrl Status <enabled|disabled>?
Sysctrl Owner?
Sysctrl Device?
Sysctrl Owner Type <tty> ?
Number of Controllers [2]?
Controller IP Address?
Controller Heartbeat HB (use network for heartbeats) <true|false>?
Controller (use network for control messages) <true|false>?
Controller Priority <1,2,...>?
```

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" on page 65.

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

%
```

For a complete example of a CLI command script that configures a cluster, see "FailSafe Configuration Example CLI Script" on page 127 in Chapter 5, "IRIS FailSafe 2.0 Configuration."

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 2.0 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.

Chapter 5

IRIS FailSafe 2.0 Configuration

This chapter describes administrative tasks you perform to configure the components of an IRIS FailSafe 2.0 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:

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.

Cluster Configuration

To set up an IRIS FailSafe 2.0 system, you configure the cluster that will support the high-availability 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 Cluster Nodes

A *cluster node* is a single UNIX image. Usually, a cluster 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 are also 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, with a maximum length of 255 characters.
- The public network. You must provide the hostname of the public network, which is the network used by clients to access highly available services. This address should be the same as the output of the *hostname* command on the node you are defining. 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 not be in XX.XX.XX.XX notation.
- Node ID, a 16-bit unsigned value (optionally specified by user).
- 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: *chalL*, *msc*, *mmsc*
  - 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*

- 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 2.0 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.
5. Click on “OK” at the bottom of the screen to complete the task, or click on “Cancel” to cancel.

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 Cluster Nodes" on page 71. The following prompts appear:

```
Enter commands, when finished enter either "done" or "cancel"

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 sysctrl_type to D
set sysctrl_password to E
set sysctrl_status to F
set sysctrl_owner to G
set sysctrl_device to H
set sysctrl_owner_type to I
add nic J
```

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 K

set ctrl_msgs to L

set priority to M
```

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

```
cmgr> remove nic N
```

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

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

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

cm1a? set hostname to cm1a
cm1a? set nodeid to 1
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? 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, the display appears as in the following example:

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

Node Name [cm1a]? cm1a

Hostname? cm1a
Node ID [0]? 1
Sysctrl Type <chalL|msc|mmsc>? msc
Sysctrl Password [ ]?
Sysctrl Status <enabled|disabled>? enabled
Sysctrl Owner? cm2
Sysctrl Device? /dev/ttyd2
Sysctrl Owner Type <tty> [tty]?
Number of Controllers [2]? 2
Controller IP Address? cm1
Controller Heartbeat HB (use network for heartbeats) <true|false>? true
Controller (use network for control messages) <true|false>? true
Controller Priority <1,2,...>? 0
Controller IP Address? cm2
Controller Heartbeat HB (use network for heartbeats) <true|false>? true
Controller (use network for control messages) <true|false>? false
Controller Priority <1,2,...>? 1
```

Modifying and Deleting Cluster Nodes

After you have defined a cluster node, you can modify or 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.

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 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 Cluster Nodes

After you define cluster 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 Node Name: cm1
Hostname: cm1
Nodeid: 1
Reset type: reset
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 node 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 cluster membership.

  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 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.

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

- 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, when finished enter either "done" or "cancel"
```

A?

When this prompt of the node 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 tie_breaker to D
```

Defining a Cluster

A *cluster* is a collection of one or more *nodes* coupled with each other by networks or other similar interconnects. In IRIS FailSafe 2.0, a cluster is identified by a simple name. A given node may be a member of only one cluster.

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

- The logical name of the cluster, with a maximum length of 255 characters.
- 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 email address to use to notify the system administrator when problems occur in the cluster (for example, *root@system*)
- (Optional) The email program to use to notify the system administrator when problems occur in the cluster (for example, */usr/sbin/Mail*).

  Specifying the email 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 email 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 email program to use to direct messages with the following command:

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

You specify an email address to direct messages with the following command:

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

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

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

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

Modifying and Deleting Clusters

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

Modifying and Deleting 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.

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.

Modifying and Deleting 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 B
cluster A? add node C
cluster A? remove node D
cluster A?
```

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

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 *nodes* 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 high-availability service, you can reuse them. If none fit, you can define additional resource types.

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 v*olume*. 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*.

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 2.0 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.

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. 666 mode is the default value for XLV device file permissions.

Filesystem Resource Attributes

The *filesystem* resource must be an XFS filesystem.

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 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, 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. Hence, in a mixed Origin/CHALLENGE cluster, an IP address might be placed on ef0 on an Origin and et0 on a CHALLENGE. In this case the *interfaces* field would be *ef0,et0* or *et0,ef0*.

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 resource dependencies of the NFS resource, which must be the names of pre-defined resources
  - The filesystem dependency
  - The statd dependency

statd Resource Attributes

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*.

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" on page 96.

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" on page 96.

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" on page 86.

For detailed information on how to determine the format for defining resource attributes, see "Specifying Resource Attributes with Cluster Manager CLI" on page 94.

When you are finished defining the resource and its dependencies, enter *done* to return to the *cmgr* prompt.

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, 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, 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, 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, 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 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 cluster node cannot be redefined.

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 and Deleting 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 and Deleting 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.

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.

Modifying and Deleting 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.

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. If these types fit the application you want to make into a high-availability 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 2.0 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 2.0 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).
- 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 cluster 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. The *probe* script is optional.

- 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 *cluster_mgr* interactively to define a resource type called **`test_rt`**.

1. Log in as **root**.

2. Execute the *cluster_mgr* command using the **-p** option to prompt you for information (the command name can be abbreviated to *cmgr*):

```
# /usr/cluster/bin/cluster_mgr -p
Welcome to IRIS FailSafe Cluster Manager Command-Line Interface

cmgr>
```

3. Use the *set* subcommand to specify the default cluster used for *cluster_mgr* 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 *test_rt*.

```
cmgr> define resource_type test_rt

(Enter "cancel" at any time to abort)

Node[optional]?
Order ? 300
Restart Mode ? (0)

DEFINE RESOURCE TYPE OPTIONS

        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 Time? 40000
Monitoring Interval? 0
Start Monitoring Time? 0

        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:
        Action - 1: start

Action name stop
Executable Time? 40000
Monitoring Interval? 0
Start Monitoring Time? 0

        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

        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: export-point

Type Specific Attribute ? string-att
Datatype ? string
Default value[optional] ? rw

        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

        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

        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 created resource_type test_rt
```

```
cmgr> show resource_types

NFS
template
Netscape_web
test_rt
statd
Oracle_DB
MAC_address
IP_address
INFORMIX_DB
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 cluster node cannot be redefined.

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 and Deleting Resource Types

After you have defined a resource types, you can modify and delete them.

Modifying and Deleting 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.

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.

Modifying and Deleting 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.

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 2.0 Programmer's Guide*.

Failover Scripts

A *failover script* helps determine the node that is chosen for a failed resource group. The failover script takes the initial failover domain and transforms it into the runtime failover domain. Depending upon the contents of the script, the initial and the runtime domains may be identical.

The *ordered* failover script is provided with the IRIS FailSafe 2.0 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 2.0 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 IRIS FailSafe database to use your new failover script for the required resource groups. For information on defining failover scripts, see the *IRIS FailSafe 2.0 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 used to statically load balance the resource groups in a cluster.

Examples:

- In a four-node cluster, two nodes might share an XLV volume. The failover domain of the resource group containing the XLV volume will be the two nodes that share the XLV volume.
- If you have a cluster of nodes named venus, mercury, and pluto, you could configure the following initial failover domains for resource groups RG1 and RG2:
  - venus, mercury, pluto for RG1
  - pluto, mercury for RG2

When you define a failover policy, you specify the *initial failover domain*. 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 *runtime failover domain* by the *failover script*. 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.

IRIS 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 a failover attribute of *Auto_Failback*, *Controlled_Failback, Auto_Recovery,* or *InPlace_Recovery. Auto_Failback* and *Controlled_Failback* are mutually exclusive, but you must specify one or the other. *Auto_Recovery* and *InPlace_Recovery* are mutually exclusive, but whether you specify one or the other is optional.

A failover attribute of *Auto_Failback* specifies that the resource group will be run on the first available node in the runtime failover domain. If the first node fails, the next available node will be used; when the first node reboots, the resource group will return to it. This attribute is best used when some type of load balancing is required.

A failover attribute of *Controlled_Failback* specifies that the resource group will be run on the first available node in the runtime failover domain, and will remain running on that node until it fails. If the first node fails, the next available node will be used; the resource group will remain on this new node even after the first node reboots.This attribute is best used when client/server applications have expensive recovery mechanisms, such as databases or any application that uses *tcp* to communicate.

The recovery attributes *Auto_Recovery* and *InPlace_Recovery* determine the node on which a resource group will be allocated when its state changes to online and a member of the group is already allocated (such as when volumes are present). *Auto_Recovery* specifies that the failover policy will be used to allocate the resource group; this is the default recovery attribute if you have specified the *Auto_Failback* attribute. *InPlace_Recovery* specifies that the resource group will be allocated on the node that already contains part of the resource group; this is the default recovery attribute if you have specified the *Controlled_Failback* attribute.

See the *IRIS FailSafe 2.0 Programmer's Guide* for a full discussions of example failover policies.

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 2.0 Programmer's Guide* and in summary in "Defining a Failover Policy" on page 111.

When you are finished defining the failover policy, enter *done* to return to the *cmgr* prompt.

Modifying and Deleting Failover Policies

After you have defined a failover policy, you can modify or delete it.

Modifying and Deleting 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.

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.

Modifying and Deleting 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.

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 2.0.

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, when finished enter either "done" or "cancel"

resource_group A?
```

When this prompt appears you can use the following commands to specify the resources to include in the resource group and the failover policy to apply to the resource group:

```
resource_group A? add resource B of resource_type C
resource_group A? set failover_policy to D
```

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" on page 126.

Modifying and Deleting Resource Groups

After you have defined resource groups, you can modify and delete 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 and Deleting Resource Groups with the Cluster Manager GUI

To modify a failure 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 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.

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.

Modifying and Deleting 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 failover policy:

```
resource_group A? add resource B of resource_type C
resource_group A? set failover_policy to D
```

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	Cluster 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 5-1 shows the logging levels as you specify them with the GUI and the CLI.

Table 5-1 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
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 cluster 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 8, "IRIS FailSafe 2.0 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, when finished enter either "done" or "cancel"
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 8, "IRIS FailSafe 2.0 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.

FailSafe Configuration Example CLI Script

The following Cluster Manager CLI script provides an example which shows how to configure a cluster in the cluster database. The script illustrates the CLI commands that you execute when you define a cluster. You will use the parameters of your own system when you configure your cluster. After you create a CLI script, you can set the execute permissions and execute the script directly.

For general information on CLI scripts see "CLI Command Scripts" on page 64 in Chapter 4, "IRIS FailSafe 2.0 Administration Tools." For information on the CLI template files that you can use to create your own configuration script, see "CLI Template Scripts" on page 65 in Chapter 4, "IRIS FailSafe 2.0 Administration Tools."

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


##############################################################
#                                                            #
# Sample cmgr script to create a 2-node cluster in the cluster  #
# database (cdb).                                            #
# This script is created using cmgr template files under     #
#          /var/cluster/cmgr-scripts directory.              #
# The cluster has 2 resource groups:                         #
# 1. nfs-group - Has 2 NFS, 2 filesystem, 2 volume, 1 statd and #
#                1 IP_address resources.                     #
# 2. web-group - Has 1 Netscape_web and 1 IP_address resources. #
#                                                            #
# NOTE: After running this script to define the cluster in the  #
# cdb, the user has to enable the two resource groups using the #
# cmgr admin online resource_group command.                  #
#                                                            #
##############################################################

#
# Create the first node.
# Information to create a node is obtained from template script:
#/var/cluster/cmgr-templates/cmgr-create-node
#

#
```

```
#
# logical name of the node. It is recommended that logical name of the
# node be output of hostname(1) command.
#
define node sleepy
#
# Hostname of the node. This is optional. If this field is not
# specified,logical name of the node is assumed to be hostname.
# This value has to be
# the output of hostname(1) command.
#
      set hostname to sleepy
#
# Node identifier. Node identifier is a 16 bit integer that uniquely
# identifies the node. This field is optional. If value is
# not provided,cluster software generates node identifier.
# Example value: 1
      set nodeid to 101
#
# Description of the system controller of this node.
# System controller can be "chalL" or "msc" or "mmsc". If the node is a
# Challenge DM/L/XL, then system controller type is "chalL". If the
# node is Origin 200 or deskside Origin 2000, then the system
# controller type is "msc". If the node is rackmount Origin 2000, the
# system controller type is "mmsc".
# Possible values: msc, mmsc, chalL
#
       set sysctrl_type to msc
#
# You can enable or disable system controller definition. Users are
# expected to enable system controller definition after verify the
# serial reset cables connected to this node.
# Possible values: enabled, disabled
#
       set sysctrl_status to enabled
#
# The system controller password for doing privileged system controller
# commands.
# This field is optional.
#
       set sysctrl_password to none
#
# System controller owner. The node name of the machine that is
# connected using serial cables to system controller of this node.
# System controller node also has to be defined in the CDB.
```

```
#
        set sysctrl_owner to grumpy
#
# System controller device. The absolute device path name of the tty
# to which the serial cable is connected in this node.
# Example value: /dev/ttyd2
#
        set sysctrl_device to /dev/ttyd2
#
# Currently, the system controller owner can be connected to the system
# controller on this node using "tty" device.
# Possible value: tty
#
        set sysctrl_owner_type to tty
#
# List of control networks. There can be multiple control networks
# specified for a node. HA cluster software uses these control
# networks for communication between nodes.  At least two control
# networks should be specified for heartbeat messages and one
# control network for FailSafe control messages.
# For each control network for the node, please add one more
# control network section.
#
# Name of control network IP address. This IP address must
# be configured on the network interface in /etc/config/netif.options
# file in the node.
# It is recommended that the IP address in internet dot notation
# is provided.
# Example value: 192.26.50.3
#
        add nic 192.26.50.14
#
# Flag to indicate if the control network can be used for sending
# heartbeat messages.
# Possible values: true, false
#
            set heartbeat to true
#
# Flag to indicate if the control network can be used for sending
# FailSafe control messages.
# Possible values: true, false
#
            set ctrl_msgs to true
#
```

```
# Priority of the control network. Higher the priority value, lower the
# priority of the control network.
# Example value: 1
#
           set priority to 1
#
# Control network information complete
#
       done
#
# Add more control networks information here.
#

# Name of control network IP address. This IP address must be
# configured on the network interface in /etc/config/netif.options
# file in the node.
# It is recommended that the IP address in internet dot
# notation is provided.
# Example value: 192.26.50.3
#
       add nic 150.166.41.60
#
# Flag to indicate if the control network can be used for sending
# heartbeat messages.
# Possible values: true, false
#
           set heartbeat to true
#
# Flag to indicate if the control network can be used for sending
# FailSafe control messages.
# Possible values: true, false
#
           set ctrl_msgs to false
#
# Priority of the control network. Higher the priority value, lower the
# priority of the control network.
# Example value: 1
#
           set priority to 2
#
# Control network information complete
#
       done
#
```

```
# Node definition complete
#
done


#
# Create the second node.
# Information to create a node is obtained from template script:
# /var/cluster/cmgr-templates/cmgr-create-node
#

#
#
# logical name of the node. It is recommended that logical name of
# the node be output of hostname(1) command.
#
define node grumpy
#
# Hostname of the node. This is optional. If this field is not
# specified,logical name of the node is assumed to be hostname.
# This value has to be
# the output of hostname(1) command.
#
       set hostname to grumpy
#
# Node identifier. Node identifier is a 16 bit integer that uniquely
# identifies the node. This field is optional. If value is
# not provided,cluster software generates node identifier.
# Example value: 1
       set nodeid to 102
#
# Description of the system controller of this node.
# System controller can be "chalL" or "msc" or "mmsc". If the node is a
# Challenge DM/L/XL, then system controller type is "chalL". If the
# node is Origin 200 or deskside Origin 2000, then the system
# controller type is "msc". If the node is rackmount Origin 2000,
# the system controller type is "mmsc".
# Possible values: msc, mmsc, chalL
#
        set sysctrl_type to msc
#
```

```
# You can enable or disable system controller definition. Users are
# expected to enable system controller definition after verify the
# serial reset cables connected to this node.
# Possible values: enabled, disabled
#
        set sysctrl_status to enabled
#
# The system controller password for doing privileged system controller
# commands.
# This field is optional.
#
        set sysctrl_password to none
#
# System controller owner. The node name of the machine that is
# connected using serial cables to system controller of this node.
# System controller node also has to be defined in the CDB.
#
        set sysctrl_owner to sleepy
#
# System controller device. The absolute device path name of the tty
# to which the serial cable is connected in this node.
# Example value: /dev/ttyd2
#
        set sysctrl_device to /dev/ttyd2
#
# Currently, the system controller owner can be connected to the system
# controller on this node using "tty" device.
# Possible value: tty
#
        set sysctrl_owner_type to tty
#
# List of control networks. There can be multiple control networks
# specified for a node. HA cluster software uses these control
# networks for communication between nodes.  At least two control
# networks should be specified for heartbeat messages and one
# control network for FailSafe control messages.
# For each control network for the node, please add one more
# control network section.
#
```

```
# Name of control network IP address. This IP address must be
# configured on the network interface in /etc/config/netif.options
# file in the node.
# It is recommended that the IP address in internet dot notation
# is provided.
# Example value: 192.26.50.3
#
       add nic 192.26.50.15
#
# Flag to indicate if the control network can be used for sending
# heartbeat messages.
# Possible values: true, false
#
           set heartbeat to true
#
# Flag to indicate if the control network can be used for sending
# FailSafe control messages.
# Possible values: true, false
#
           set ctrl_msgs to true
#
# Priority of the control network. Higher the priority value, lower the
# priority of the control network.
# Example value: 1
#
           set priority to 1
#
# Control network information complete
#
       done
#
# Add more control networks information here.
#

# Name of control network IP address. This IP address must be
# configured on the network interface in /etc/config/netif.options
# file in the node.
# It is recommended that the IP address in internet dot notation
# is provided.
# Example value: 192.26.50.3
#
       add nic 150.166.41.61
#
```

```
# Flag to indicate if the control network can be used for sending
# heartbeat messages.
# Possible values: true, false
#
            set heartbeat to true
#
# Flag to indicate if the control network can be used for sending
# FailSafe control messages.
# Possible values: true, false
#
            set ctrl_msgs to false
#
# Priority of the control network. Higher the priority value, lower the
# priority of the control network.
# Example value: 1
#
            set priority to 2
#
# Control network information complete
#
        done
#
# Node definition complete
#
done


#
# Define (create) the cluster.
# Information to create the cluster is obtained from template script:
#       /var/cluster/cmgr-templates/cmgr-create-cluster
#

#
# Name of the cluster.
#
define cluster failsafe-cluster
#
# Notification command for the cluster. This is optional. If this
# field is not specified,  /usr/bin/mail command is used for
# notification. Notification is sent when there is change in status of
# cluster, node and resource group.
#
       set notify_cmd to /usr/bin/mail
#
```

```
# Notification address for the cluster. This field value is passed as
# argument to the notification command. Specifying the notification
# command is optional and user can specify only the notification
# address in order to receive notifications by mail. If address is
# not specified, notification will not be sent.
# Example value: failsafe_alias@sysadm.company.com
      set notify_addr to robinhood@sgi.com princejohn@sgi.com
#
# List of nodes added to the cluster.
# Repeat the following line for each node to be added to the cluster.
# Node should be already defined in the CDB and logical name of the
# node has to be specified.
       add node sleepy
#
# Add more nodes to the cluster here.
#
       add node grumpy

#
# Cluster definition complete
#
done


#
# Create failover policies
# Information to create the failover policies is obtained from
# template script:
#       /var/cluster/cmgr-templates/cmgr-create-cluster
#

#
# Create the first failover policy.
#

#
# Name of the failover policy.
#
define failover_policy sleepy-primary
#
```

```
# Failover policy attribute. This field is mandatory.
# Possible values: Auto_Failback, Controlled_Failback, Auto_Recovery,
# InPlace_Recovery
#

        set attribute to Auto_Failback

        set attribute to Auto_Recovery

#
# Failover policy script. The failover policy scripts have to
# be present in
# /var/cluster/ha/policies directory. This field is mandatory.
# Example value: ordered (file name not the full path name).
        set script to ordered
#
# Failover policy domain. Ordered list of nodes in the cluster
# separated by spaces. This field is mandatory.
#
        set domain to sleepy grumpy
#
# Failover policy definition complete
#
done

#
# Create the second failover policy.
#

#
# Name of the failover policy.
#
define failover_policy grumpy-primary
#
# Failover policy attribute. This field is mandatory.
# Possible values: Auto_Failback, Controlled_Failback, Auto_Recovery,
# InPlace_Recovery
#

        set attribute to Auto_Failback

        set attribute to InPlace_Recovery

#
```

```
# Failover policy script. The failover policy scripts have
# to be present in
# /var/cluster/ha/policies directory. This field is mandatory.
# Example value: ordered (file name not the full path name).
       set script to ordered
#
# Failover policy domain. Ordered list of nodes in the cluster
# separated by spaces. This field is mandatory.
#
       set domain to  grumpy sleepy
#
# Failover policy definition complete
#
done


#
# Create the IP_address resources.
# Information to create an IP_address resource is obtained from:
#       /var/cluster/cmgr-templates/cmgr-create-resource-IP_address
#

#
# If multiple resources of resource type IP_address have to be created,
# repeat the following IP_address definition template.
#
# Name of the IP_address resource.  The name of the resource has to
# be IP address in the internet "." notation. This IP address is used
# by clients to access highly available resources.
# Example value: 192.26.50.140
#
define resource 150.166.41.179 of resource_type IP_address in cluster
failsafe-cluster

#
# The network mask for the IP address. The network mask value is used
# to configure the IP address on the network interface.
# Example value: 0xffffff00
       set NetworkMask to 0xffffff00
#
# The ordered list of interfaces that can be used to configure the IP
# address.The list of interface names are separated by comma.
# Example value: ef0, ef1
       set interfaces to ef1
#
```

```
# The broadcast address for the IP address.
# Example value: 192.26.50.255
        set BroadcastAddress to 150.166.41.255

#
# IP_address resource definition for the cluster complete
#
done

#
# Name of the IP_address resource.  The name of the resource has to be
# IP address in the internet "." notation. This IP address is used by
# clients to access highly available resources.
# Example value: 192.26.50.140
#
define resource 150.166.41.99 of resource_type IP_address in cluster
failsafe-cluster

#
# The network mask for the IP address. The network mask value is used
# to configure the IP address on the network interface.
# Example value: 0xffffff00
        set NetworkMask to 0xffffff00
#
# The ordered list of interfaces that can be used to configure the IP
# address.
# The list of interface names are separated by comma.
# Example value: ef0, ef1
        set interfaces to ef1
#
# The broadcast address for the IP address.
# Example value: 192.26.50.255
        set BroadcastAddress to 150.166.41.255

#
# IP_address resource definition for the cluster complete
#
done


#
```

```
# Create the volume resources.
# Information to create a volume resource is obtained from:
#       /var/cluster/cmgr-templates/cmgr-create-resource-volume
#

#
# If multiple resources of resource type volume have to be created,
# repeat the following volume definition template.
#
# Name of the volume.  The name of the volume has to be XLV volume.
# Example value: HA_vol (not /dev/xlv/HA_vol)
#
define resource bagheera of resource_type volume in cluster
failsafe-cluster

#
# The user name of the xlv device file name. This field is optional. If
# this field is not specified, value “root” is used.
# Example value: oracle
        set devname-owner to root
#
# The group name of the xlv device file name. This field is optional.
# If this field is not specified, value “sys” is used.
# Example value: oracle
        set devname-group to sys
#
# The xlv device file permissions. This field is optional. If this
# field is not specified, value “666” is used. The file permissions
# have to be specified in octal notation. See chmod(1) for more
# information.
# Example value: 666
        set devname-mode to 666

#
# Volume resource definition for the cluster complete
#
done

#
```

```
# Name of the volume.  The name of the volume has to be XLV volume.
# Example value: HA_vol (not /dev/xlv/HA_vol)
#
define resource bhaloo of resource_type volume in cluster
failsafe-cluster

#
# The user name of the xlv device file name. This field is optional. If
# this field is not specified, value "root" is used.
# Example value: oracle
       set devname-owner to root
#
# The group name of the xlv device file name. This field is optional.
# If this field is not specified, value "sys" is used.
# Example value: oracle
       set devname-group to sys
#
# The xlv device file permissions. This field is optional. If this
# field is
# not specified, value "666" is used. The file permissions have to
# be specified in octal notation. See chmod(1) for more information.
# Example value: 666
       set devname-mode to 666

#
# Volume resource definition for the cluster complete
#
done


#
# Create the filesystem resources.
# Information to create a filesystem resource is obtained from:
#       /var/cluster/cmgr-templates/cmgr-create-resource-filesystem
#

#
# filesystem resource type is for XFS filesystem only.
# If multiple resources of resource type filesystem have to be created,
# repeat the following filesystem definition template.
#
```

```
# Name of the filesystem.  The name of the filesystem resource has
# to be absolute path name of the filesystem mount point.
# Example value: /shared_vol
#
define resource /haathi of resource_type filesystem in cluster
failsafe-cluster

#
# The name of the volume resource corresponding to the filesystem. This
# resource should be the same as the volume dependency, see below.
# This field is mandatory.
# Example value: HA_vol
       set volume-name to bagheera
#
# The options to be used when mounting the filesystem. This field is
# mandatory. For the list of mount options, see fstab(4).
# Example value: "rw"
       set mount-options to rw
#
# The monitoring level for the filesystem. This field is optional. If
# this field is not specified, value "1" is used.
# Monitoring level can be
# 1 - Checks if filesystem exists in the mtab file (see mtab(4)). This
# is a lightweight check compared to monitoring level 2.
# 2 - Checks if the filesystem is mounted using stat(1m) command.
#
       set monitoring-level to 2
done

#
# Add filesystem resource type dependency
#
modify resource /haathi of resource_type filesystem in cluster
failsafe-cluster
#
# The filesystem resource type definition also contains a resource
# dependency on a volume resource.
# This field is mandatory.
# Example value: HA_vol
       add dependency bagheera of type volume
#
# filesystem resource definition for the cluster complete
#
```

```
done

#
# Name of the filesystem.  The name of the filesystem resource has
# to be absolute path name of the filesystem mount point.
# Example value: /shared_vol
#
define resource /sherkhan of resource_type filesystem in cluster
failsafe-cluster

#
# The name of the volume resource corresponding to the filesystem. This
# resource should be the same as the volume dependency, see below.
# This field is mandatory.
# Example value: HA_vol
        set volume-name to bhaloo
#
# The options to be used when mounting the filesystem. This field is
# mandatory.For the list of mount options, see fstab(4).
# Example value: "rw"
        set mount-options to rw
#
# The monitoring level for the filesystem. This field is optional. If
# this field is not specified, value "1" is used.
# Monitoring level can be
# 1 - Checks if filesystem exists in the mtab file (see mtab(4)). This
# is a lightweight check compared to monitoring level 2.
# 2 - Checks if the filesystem is mounted using stat(1m) command.
#
        set monitoring-level to 2
done

#
# Add filesystem resource type dependency
#
modify resource /sherkhan of resource_type filesystem in cluster
failsafe-cluster
#
# The filesystem resource type definition also contains a resource
# dependency on a volume resource.
# This field is mandatory.
# Example value: HA_vol
        add dependency bhaloo of type volume
#
```

```
# filesystem resource definition for the cluster complete
#
done


#
# Create the statd resource.
# Information to create a filesystem resource is obtained from:
#       /var/cluster/cmgr-templates/cmgr-create-resource-statd
#

#
# If multiple resources of resource type statd have to be created,
# repeat the following filesystem definition template.
#
# Name of the statd.  The name of the resource has to be the location
# of the NFS/lockd directory.
# Example value: /disk1/statmon
#

define resource /haathi/statmon of resource_type statd in cluster
failsafe-cluster

#
# The IP address on which the NFS clinets connect, this resource should
# be the same as the IP_address dependency, see below.
# This field is mandatory.
# Example value: 128.1.2.3
        set InterfaceAddress to 150.166.41.99
done


#
# Add the statd resource type dependencies
#
modify resource /haathi/statmon of resource_type statd in cluster
failsafe-cluster
#
# The statd resource type definition also contains a resource
# dependency on a IP_address resource.
# This field is mandatory.
# Example value: 128.1.2.3
        add dependency 150.166.41.99 of type IP_address
#
```

```
# The statd resource type definition also contains a resource
# dependency on a filesystem resource. It defines the location of
# the NFS lock directory filesystem.
# This field is mandatory.
# Example value: /disk1
        add dependency /haathi of type filesystem
#
# statd resource definition for the cluster complete
#
done


#
# Create the NFS resources.
# Information to create a NFS resource is obtained from:
#       /var/cluster/cmgr-templates/cmgr-create-resource-NFS
#

#
# If multiple resources of resource type NFS have to be created, repeat
# the following NFS definition template.
#
# Name of the NFS export point. The name of the NFS resource has to be
# export path name of the filesystem mount point.
# Example value: /disk1
#
define resource /haathi of resource_type NFS in cluster
failsafe-cluster

#
# The export options to be used when exporting the filesystem. For the
# list of export options, see exportfs(1M).
# This field is mandatory.
# Example value: "rw,wsync,anon=root"
        set export-info to rw
#
# The name of the filesystem resource corresponding to the export
# point. This resource should be the same as the filesystem dependency,
# see below.
# This field is mandatory.
# Example value: /disk1
        set filesystem to /haathi
done

#
```

```
# Add the resource type dependency
#
modify resource /haathi of resource_type NFS in cluster
failsafe-cluster
#
# The NFS resource type definition also contains a resource dependency
# on a filesystem resource.
# This field is mandatory.
# Example value: /disk1
        add dependency /haathi of type filesystem
#
# The NFS resource type also contains a pseudo resource dependency
# on a statd resource. You really must have a statd resource associated
# with a NFS resource, so the NFS locks can be failed over.
# This field is mandatory.
# Example value: /disk1/statmon
        add dependency /haathi/statmon of type statd

#
# NFS resource definition for the cluster complete
#
done

#
# Name of the NFS export point. The name of the NFS resource has to be
# export path name of the filesystem mount point.
# Example value: /disk1
#
define resource /sherkhan of resource_type NFS in cluster
failsafe-cluster

#
# The export options to be used when exporting the filesystem. For the
# list of export options, see exportfs(1M).
# This field is mandatory.
# Example value: "rw,wsync,anon=root"
        set export-info to rw
#
# The name of the filesystem resource corresponding to the export
# point. This
# resource should be the same as the filesystem dependency, see below.
# This field is mandatory.
# Example value: /disk1
        set filesystem to /sherkhan
```

```
done

#
# Add the resource type dependency
#
modify resource /sherkhan of resource_type NFS in cluster
failsafe-cluster
#
# The NFS resource type definition also contains a resource dependency
# on a filesystem resource.
# This field is mandatory.
# Example value: /disk1
        add dependency /sherkhan of type filesystem
#
# The NFS resource type also contains a pseudo resource dependency
# on a statd resource. You really must have a statd resource associated
# with a NFS resource, so the NFS locks can be failed over.
# This field is mandatory.
# Example value: /disk1/statmon
        add dependency /haathi/statmon of type statd

#
# NFS resource definition for the cluster complete
#
done


#
# Create the Netscape_web resource.
# Information to create a Netscape_web resource is obtained from:
#       /var/cluster/cmgr-templates/cmgr-create-resource-Netscape_web
#

#
# If multiple resources of resource type Netscape_web have to be
# created, repeat the following filesystem definition template.
#
# Name of the Netscape WEB server.  The name of the resource has to be
# a unique identifier.
# Example value: ha80
#
define resource web-server of resource_type Netscape_web in cluster
failsafe-cluster

#
```

```
# The locations of the servers startup and stop scripts.
# This field is mandatory.
# Example value: /usr/ns-home/ha86
       set admin-scripts to /var/netscape/suitespot/https-control3
#
# the TCP port number with the server listens on.
# This field is mandatory.
# Example value: 80
       set port-number to 80
#
# The desired monitoring level, the user can specify either;
#       1 - checks for process existence
#       2 - issues an HTML query to the server.
# This field is mandatory.
# Example value: 2
       set monitor-level to 2
#
# The locations of the WEB servers initial HTML page
# This field is mandatory.
# Example value: /var/www/htdocs
       set default-page-location to /var/www/htdocs
#
# The WEB servers IP address, this must be a configured IP_address
# resource.
# This resource should be the same as the IP_address dependency, see
# below.
# This field is mandatory.
# Example value: 28.12.9.5
       set web-ipaddr to 150.166.41.179
done

#
# Add the resource dependency
#
modify resource web-server of resource_type Netscape_web in cluster
failsafe-cluster
#
# The Netscape_web resource type definition also contains a resource
# dependency on a IP_address resource.
# This field is mandatory.
# Example value: 28.12.9.5
       add dependency 150.166.41.179 of type IP_address
#
# Netscape_web resource definition for the cluster complete
#
```

```
done


#
# Create the resource groups.
# Information to create a resource group is obtained from:
#       /var/cluster/cmgr-templates/cmgr-create-resource_group
#

#
# Name of the resource group. Name of the resource group must be unique
# in the cluster.
#
define resource_group nfs-group in cluster failsafe-cluster
#
# Failover policy for the resource group. This field is mandatory.
# Failover policy should be already defined in the CDB.
#
        set failover_policy to sleepy-primary
#
# List of resources in the resource group.
# Repeat the following line for each resource to be added to the
# resource group.
        add resource 150.166.41.99 of resource_type IP_address
#
# Add more resources to the resource group here.
#
        add resource bagheera of resource_type volume

        add resource bhaloo of resource_type volume

        add resource /haathi of resource_type filesystem

        add resource /sherkhan of resource_type filesystem

        add resource /haathi/statmon of resource_type statd

        add resource /haathi of resource_type NFS

        add resource /sherkhan of resource_type NFS

#
# Resource group definition complete
#
```

```
done

#
# Name of the resource group. Name of the resource group must be unique
# in the cluster.
#
define resource_group web-group in cluster failsafe-cluster
#
# Failover policy for the resource group. This field is mandatory.
# Failover policy should be already defined in the CDB.
#
        set failover_policy to grumpy-primary
#
# List of resources in the resource group.
# Repeat the following line for each resource to be added to the
# resource group.
        add resource 150.166.41.179 of resource_type IP_address

#
# Add more resources to the resource group here.
#

 add resource web-server of resource_type Netscape_web

#
# Resource group definition complete
#
done


#
# Script complete. This should be last line of the script
#
quit
```

Chapter 6

IRIS FailSafe 2.0 System Operation

This chapter describes administrative tasks you perform to operate and monitor an IRIS FailSafe 2.0 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:

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 2.0

After you have configured your IRIS FailSafe 2.0 system and run diagnostic tests on its components, you can activate the high-availability 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.

Activating IRIS FailSafe 2.0 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 2.0 with the Cluster Manager CLI

To activate IRIS FailSafe 2.0 in a cluster, use the following command:

```
cmgr> start ha_services [on node A] [for cluster B]
```

System Status

While the IRIS FailSafe 2.0 system is running, you can monitor the status of the IRIS FailSafe 2.0 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 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 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 2.0. 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.

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 cluster's 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
- DOWN
- UNKNOWN

If a node's status is UNKOWN, the status of the resources and resource groups online in that node will not be available (or ONLINE_READY in the case of a resource group).

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
- INACTIVE

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 Cellular: 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 Cellular: 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 Cellular: 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 Cellular: 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 2.0 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.

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.

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" on page 168.

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, FaillSafe 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)" on page 170.

Caution: Detaching a resource group leaves the resources in the resource group running at the cluster node where it was online. After stopping HA services on that cluster 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 2.0 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.

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 2.0 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 2.0

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.

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-high availability 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 2.0 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.

Chapter 7

Testing IRIS FailSafe 2.0 Configuration

This chapter explains how to test the IRIS FailSafe 2.0 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 2.0 Administration Tools."

The sections in this chapter are as follows:

Overview of FailSafe Diagnostic Commands

Table 7-1 shows the tests you can perform with IRIS FailSafe diagnostic commands:

Table 7-1 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 2.0 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 invaild 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>
```

Chapter 8

IRIS FailSafe 2.0 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 high-availability 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 up log configurations, see "FailSafe System Log Configuration" on page 121 in Chapter 5, "IRIS FailSafe 2.0 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)
```

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
```

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.

Node 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 node membership. When failover occurs, the runtime failover domain can include only those nodes that are in the cluster membership.

Node Membership in Cluster

Nodes can enter into the cluster 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 cluster 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 cluster membership.

Node 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 cluster 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" on page 80 in Chapter 5, "IRIS FailSafe 2.0 Configuration."

Resetting Nodes

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.

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 cluster membership is formed, you should check the following areas for possible problems:

- Is the cluster 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” on page 153 of Chapter 6, “IRIS FailSafe 2.0 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 2.0 Configuration" and Chapter 6, "IRIS FailSafe 2.0 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" on page 200).
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

Follow this procedure if the status of a node is UNKNOWN in an active cluster:

1. Check to see if the control networks in the node are working (see "Control Network Failure Recovery" on page 200).
2. Check to see if the serial reset cables to reset the node are working (see "Serial Cable Failure Recovery" on page 200).
3. 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_**node name***) 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 group offline with the **force** option and then move the resource group online.

   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" on page 155 in Chapter 6, "IRIS FailSafe 2.0 System Operation" for explanations of resource group error codes.

1. Look at the *failsafe* and SRM logs (default directory: */var/cluster/ha/logs*, files: f*ailsafe*_**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.

   At this point, look at the *srmd* logs on each of these machines 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.

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" on page 168 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 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" on page 175 in Chapter 6, "IRIS FailSafe 2.0 System Operation."

IRIS FailSafe 2.0 Cluster Manager GUI and CLI Inconsistencies

If the FailSafe 2.0 Cluster Manager GUI is displaying information that is inconsistent with the FailSafe 2.0 *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.

Chapter 9

Upgrading and Maintaining Active Clusters

When a IRIS FailSafe 2.0 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 7, "Testing IRIS FailSafe 2.0 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 cluster 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
   ```

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 7, "Testing IRIS FailSafe 2.0 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" on page 207. 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.

Appendix A

Updating from IRIS FailSafe 1.2 to IRIS FailSafe 2.0

IRIS FailSafe 2.0 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 2.0 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 2.0.

This chapter provides a description of the procedures you perform to upgrade a system form IRIS FailSafe 1.2 to IRIS FailSafe 2.0. It includes the following sections:

Hardware Changes

There are no hardware changes that are required when you upgrade a system to FailSafe 2.0. A FailSafe 1.2 system will be a dual-hosted storage with reset ring two-node configuration in FailSafe 2.0.

With FailSafe 2.0, you can test the hardware configuration with FailSafe diagnostic commands. See Chapter 7, "Testing IRIS FailSafe 2.0 Configuration" for instructions on using FailSafe to test the connections. These diagnostics are not run automatically when you start FailSafe 2.0; you must run them manually.

You can also use the *admin ping* CLI command to test the serial reset line in FailSafe 2.0. 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 2.0 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 2.0 Administration Tools" for information on using CLI commands.

Software Changes

FailSafe 2.0 consists of a different set of files than FailSafe 1.2. FailSafe 1.2 and FailSafe 2.0 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 2.0, 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.

Configuration Changes

You must reconfigure your FailSafe 1.2 system by using the FailSafe 2.0 Cluster Manager GUI or the FailSafe 2.0 Cluster Manager CLI to configure the system as a FailSafe 2.0 system. For information on using these administration tools, see Chapter 4, "IRIS FailSafe 2.0 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 2.0 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 2.0 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 2.0 Cluster Manager GUI. These task sets lead you through these configuration steps.

For a configuration example that compares FailSafe 1.2 configuration to FailSafe 2.0 configuration, see “Upgrade Examples” on page 215.

Scripts

All FailSafe 1.2 scripts must be rewritten for FailSafe 2.0. The *IRIS FailSafe 2.0 Programmer's Guide* provides detailed information on FailSafe 2.0 scripts as well as detailed instructions for migrating FailSafe 1.2 scripts to their FailSafe 2.0 functional equivalent.

Operational Comparison

In FailSafe 1.2, the unit of failover is the node. In FailSafe 2.0, 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 2.0. In addition, all FailSafe scripts differ between the two releases.

Table A-1 summarizes the differences between the releases..

Table A-1 Differences Between IRIS FailSafe 1.2 and 2.0

IRIS FailSafe 1.2	IRIS FailSafe 2.0
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 2.0 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

Table A-1 (continued) Differences Between IRIS FailSafe 1.2 and 2.0

IRIS FailSafe 1.2	IRIS FailSafe 2.0
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
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 high-availability service

Upgrade Examples

In order to upgrade a FailSafe 1.2 system to a FailSafe 2.0 system, you must examine your *ha.conf* file to determine how to define the equivalent parameters in the FailSafe 2.0 configuration database.

The following sections show upgrade examples for the following tasks:

- Defining a Node
- Defining a Cluster
- 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 2.0 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 2.0 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 2.0:

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

For information on using these values to define a node in FailSafe 2.0, see "Defining Cluster Nodes" on page 71 in Chapter 5, "IRIS FailSafe 2.0 Configuration." Note that there are additional parameters you will need to specify when you define this node.

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 2.0 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 2.0 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.

For information on setting the node timeout and the heartbeat interval, as well as information on heartbeat networks in general, see "IRIS FailSafe HA Parameters" on page 80 in Chapter 5, "IRIS FailSafe 2.0 Configuration."

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 2.0 uses in its cluster definition: the email 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 2.0, you can use this as the email address to use for problem notification.

There are other things you must provide in addition to this parameter when you define a FailSafe 2.0 cluster, such as the email program to use for this notification and, of course, the nodes to include in the cluster. For information on defining a cluster, see "Defining a Cluster" on page 82 in Chapter 5, "IRIS FailSafe 2.0 Configuration."

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 2.0 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 2.0:

volume name:	apache-vol
user name of device file owner:	root
group name of device file:	sys
device file permissions:	600

For information on using these values to define a volume in FailSafe 2.0, see "Defining Resources" on page 86 in Chapter 5, "IRIS FailSafe 2.0 Configuration.".

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 2.0 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 2.0:

resource name (mount point):	/apache-vol
xlv volume:	apache-vol
mount options:	rw, noauto

For information on using these values to define a filesystem in FailSafe 2.0, see "Defining Resources" on page 86 in Chapter 5, "IRIS FailSafe 2.0 Configuration."

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 2.0 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 2.0:

Resource name:	54.3.252.7
broadcast address:	54.3.252.255
network mask:	0xffffff00

For information on using these values to define an IP address in FailSafe 2.0, see "Defining Resources" on page 86 in Chapter 5, "IRIS FailSafe 2.0 Configuration."

Additional FailSafe 2.0 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" on page 117 in Chapter 5, "IRIS FailSafe 2.0 Configuration."

After you have configured you system, you can start FailSafe services, as described in "Activating (Starting) IRIS FailSafe 2.0" on page 152 in Chapter 6, "IRIS FailSafe 2.0 System Operation."

Status

In FailSafe 1.2, you produced a display of the system status with the *ha_admin -a* command. In FailSafe 2.0, 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/haStatus* 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" on page 153 in Chapter 6, "IRIS FailSafe 2.0 System Operation."

Appendix B

IRIS FailSafe 2.0 Software

This appendix summarizes software to be installed on systems used for IRIS FailSafe 2.0.

Note: "Installing Required Software" on page 38 in Chapter 3 contains step-by-step instructions for installing the software.

This appendix consists of these sections:

Subsystems on the IRIS FailSafe 2.0 CD

The IRIS FailSafe 2.0 base CD requires about 25 MB.

Table B-1 lists IRIS FailSafe 2.0 subsystems on the IRIS FailSafe 2.0 CD.

Table B-1 IRIS FailSafe 2.0 CD

Purpose	System
Base system administration	*sysadm_base* *sysadm_base.idb* *sysadm_base.man* *sysadm_base.sw*
Cluster administration	*cluster_admin* *cluster_admin.idb* *cluster_admin.man* *cluster_admin.sw* *cluster_admin.sw32*
High-availability clustering	*cluster_ha* *cluster_ha.idb* *cluster_ha.man* *cluster_ha.sw*
Cluster control	*cluster_control* *cluster_control.idb* *cluster_control.man* *cluster_control.sw*
IRIS FailSafe 2.0	*failsafe2* *failsafe2.idb* *failsafe2man* *failsafe2.sw* *failsafe2.books* (InSight versions of customer manuals)
FailSafe system administration (IRIS FailSafe 2.0 Cluster Manager GUI)	*sysadm_failsafe2* *sysadm_failsafe2.idb* *sysadm_failsafe2.man* *sysadm_failsafe2.sw*

Table B-1 (continued) IRIS FailSafe 2.0 CD

Purpose	System
Java	*java_eoe* *java_eoe.idb* *java_eoe.man* *java_eoe.sw* *java_eoe.sw32*
Java Plug-in	*java_plugin* *java_plugin.idb* *java_plugin.man* *java_plugin.sw* *java_plugin.sw32*

Each IRIS FailSafe 2.0 option, such as the WebFORCE® and NFS options, include a separate CD with the software for that option.

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 2.0 Pool

Table B-2 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*
IRIS FailSafe 2.0 administration server	*sysadm_failsafe2.sw.server*	*sysadm_base.sw.server* *cluster_admin.sw.base* *cluster_ha.sw.cli* *cluster_control.sw.cli* *failsafe2.sw.cli*
Cluster administration	*cluster_admin.sw* *cluster_control.sw*	*sysadm_base.sw.dso* For IRIX 6.2, check for latest POSIX patch set for IRIX pthreads support (included in later versions of IRIX)
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 2.0 Cluster

Table B-3 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 cluster 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
High-availability clustering software	*cluster_ha.sw*	*cluster_admin.sw* *cluster_control.sw*
IRIS FailSafe 2.0 software	*failsafe2.sw*	*cluster_ha.sw*
Optional: IRIS FailSafe 2.0 NFS	*failsafe2_nfs.sw*	*failsafe2.sw* *nfs.sw.nfs* (IRIX; might already be present)
Optional: IRIS FailSafe 2.0 Web	*failsafe2_web.sw*	*failsafe2.sw* *ns_admin.sw.server*(Netscape; might already be present) *ns_fasttrack.sw.server* OR *ns_enterprise.sw.server* (Netscape; might already be present)

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.

Table B-4 Subsystems Required for IRIX Administrative Workstations

Product	Subsystems	Prerequisites
IRIS FailSafe 2.0 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 2.0 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.

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 2.0 Cluster Manager GUI is run from a browser, the browser is redirected to the Web site in Table B-5.

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: *probe*, *exclusive*, *start*, *stop*, *monitor*, and *restart*.

cluster

A collection of one or more *cluster nodes* coupled to each other by networks or other similar interconnections. A cluster is identified by a simple name; this name must be unique within the *pool*. A particular node may be a member of only one cluster.

cluster administrator

The person responsible for managing and maintaining an IRIS FailSafe cluster.

cluster configuration database

Contains configuration information about all resources, resource types, resource groups, failover policies, nodes, and clusters.

cluster node

A single IRIX image. Usually, a cluster 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.

control messages

Messages that cluster software sends between the cluster nodes to request operations on or distribute information about cluster nodes and resource groups. IRIS FailSafe sends control messages for the purpose of ensuring 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.

A node's control networks should not be set to accept control messages if the node is not a dedicated IRIS FailSafe node. Otherwise, end users who run non-IRIS FailSafe jobs on the machine can have their jobs killed unexpectedly when IRIS FailSafe resets the node.

control network

The network that connects nodes through their network interfaces (typically Ethernet) such that IRIS FailSafe can maintain a cluster's high availability by sending heartbeat messages and control messages through the network to the attached nodes. IRIS 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.

dependency list

See *resource dependency list* or *resource type dependency list*.

failover

The process of allocating a *resource group* to another *node* to another, according to a *failover policy*. A failover may be triggered by the failure of a resource, a change in the node 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 *AutoFailback* or *ControlledFailback*), 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 *runtime failover domain* by the *failover script*; the runtime failover domain is what is actually used to select the failover node. IRIS FailSafe stores the runtime failover domain and uses it as input to the next failover script invocation. The initial and runtime failover domains may be identical, depending upon the contents of the failover script. In general, IRIS FailSafe allocates a given resource group to the first node listed in the runtime failover domain that is also in the node membership; the point at which this allocation takes place is affected by the *failover attributes*.

failover policy

The method used by IRIS 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 *runtime failover domain* and returns it to the IRIS FailSafe process. The IRIS FailSafe process applies the failover attributes and then selects the first node in the returned failover domain that is also in the current node membership.

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 IRIS 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 *runtime failover domain* by the *failover script*; the runtime 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 runtime failover domain and potentially modifies it; the initial failover domain is never used again. Depending on the runtime conditions and contents of the failover script, the initial and runtime failover domains may be identical. See also *runtime 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 IRIS FailSafe 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: IRIS 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 IRIS FailSafe processes that use the same log configuration. A log group usually corresponds to one IRIS FailSafe daemon, such as *gcd*.

log level

A number controlling the number of log messages that IRIS FailSafe will write into an associated log group's log file. A log level is part of the log configuration for a log group.

node

See *cluster node*

node ID

A 16-bit positive integer that uniquely defines a cluster node. During node definition, IRIS 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 membership

The list of nodes in a cluster on which IRIS FailSafe can allocate resource groups.

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 IRIS 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, IRIS FailSafe stops the group (if needed) and stops monitoring the group. An offline resource group can be running on a node, yet not under IRIS FailSafe control. If the cluster administrator specifies the *detach only* option while taking the group offline, then IRIS 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 IRIS 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, IRIS 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 IRIS FailSafe will not start the group but will begin monitoring the group.

owner host

A system that can control an IRIS FailSafe node remotely, such as power-cycling the node). Serial cables must physically connect the two systems through the node's system controller port. At run time, the owner host must be defined as a node in the IRIS FailSafe 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 IRIS FailSafe node with the system controller port, so the node can be controlled remotely by the owner host.

pool

The entire set of *nodes* involved with a group of clusters. The group of clusters are usually close together and should always serve a common purpose. A replicated database is stored on each node in the pool.

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*, IRIS FailSafe tracks the response from a node's system controller as it makes reset requests to a cluster node. When these requests fail to reset the node successfully, IRIS FailSafe uses heuristics to try to estimate whether the machine has been powered down. If the heuristic algorithm returns with success, IRIS FailSafe assumes the remote machine has been reset successfully. When powerfail mode is turned *off*, the heuristics are not used and IRIS 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 be one or more processes 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 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 for IRIS FailSafe.

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 cluster.

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.

runtime failover domain

The ordered set of nodes on which the resource group can execute upon failures, as modified by the *failover script*. The runtime 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 non-negative integer. In a resource group, the start/stop orders of the resource types determine the order in which the resources will be started when IRIS FailSafe brings the group online and will be stopped when IRIS 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 IRIS FailSafe brings a resource group online, all volume resources in the group will be started before all filesystem resources in the group.

system controller port

A port sitting 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 IRIS 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 IRIS FailSafe to use in the process of computing node 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, IRIS 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 filesystem is located) and specify options for how to mount the filesystem (for example, as readable and writable).

Index

E

F

G

H

I

L

M

N

S

T

U

V

X