IRIS FailSafe™ Version 2
Administrator's Guide

007–3901–006

CONTRIBUTORS

Written by Jenn Byrnes, Susan Ellis, Lori Johnson, Steven Levine

Edited by Susan Wilkening

Illustrated by Chrystie Danzer, Dany Galgani

Production by Glen Traefald

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

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

New Features in This Guide

This revision contains the following new information:

- Support for the L1 system controller port for SGI Origin 300, SGI Origin 3200C, SGI Onyx 300, and SGI Onyx 3200C systems. See "Define a Node", page 116, and "Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C Console Support", page 225.
- Access to the Performance Co-Pilot (PCP) `hbvis`(1) and `rmvis`(1) tools from the FailSafe GUI under the **File** menu; see "GUI Overview", page 89.
- Information about FailSafe and CXFS metadata server relocation; see "CXFS Metadata Server Relocation", page 288.
- Changes to the `cluster_status`(1M) command; see "Monitoring System Status with `cluster_status`", page 234.
- Clarifications about the use of `offline_detach`; see "Resuming Two-Node Use", page 230.
- Additional FailSafe GUI troubleshooting information; see "GUI Will Not Run", page 285.
- Updated FailSafe GUI icons; see "Key to Icons and States", page 235.
- Information about the software layers, communication paths, and cluster database of a FailSafe system has been moved into Appendix A, "Software Overview", page 307.

Record of Revision

Version	Description
002	December 1999 Published in conjunction with FailSafe 2.0 rollup patch. Supports IRIX 6.5.2 and later.
003	November 2000 Supports the IRIS FailSafe 2.1 release
004	May 2001 Supports the IRIS FailSafe 2.1.1 release
005	October 2001 Supports the IRIS FailSafe 2.1.2 release
006	April 2002 Supports the IRIS FailSafe 2.1.3 release

Contents

About This Guide xxvii
Audience xxvii
Assumptions xxvii
Structure of This Guide xxvii
Related Documentation xxviii
Obtaining Publications xxx
Conventions xxx
Reader Comments xxxi

1. Overview 1
High Availability and IRIS FailSafe 1
Cluster Environment 3
Terminology 3
Cluster 3
Node 3
Pool 3
Cluster Database 4
Membership 4
Quorum 5
Private Network 7
Resource 8
Resource Type 8
Resource Name 9
Resource Group 9
Dependency 9

Failover 11
Failover Policy 12
Failover Domain 12
Failover Attribute 12
Failover Scripts 13
Action Scripts 13
Cluster Process Group 14
Plug-In 14
Hardware Components 15
Disk Connections 17
Supported Configurations 17
Additional Features 20
Dynamic Management 20
Fine-Grain Failover 21
Local Restarts 21
Administration 21
Highly Available Resources 21
Nodes 22
Network Interfaces and IP Addresses 22
Disks 24
Highly Available Applications 25
Failover and Recovery Processes 26
Overview of Configuring and Testing a New Cluster 27

2. Configuration Planning 29

Overview 29
Questions to be Answered 29
Example 31

Disk Configuration 33
Planning Disk Configuration 33
Configuration Parameters for Disks 39
Logical Volume Configuration 39
Planning XLV Logical Volumes 39
Example Logical Volume Configuration 41
Configuration Parameters for Logical Volumes 42
Filesystem Configuration 42
Planning Filesystems 42
Example Filesystem Configuration 44
HA IP Address Configuration 45
Planning Network Interface and HA IP Address Configuration 45
Example HA IP Address Configuration 48
Local Failover of HA IP Addresses 48
Coexecution of CXFS and IRIS FailSafe 48
Size of the Coexecution Cluster 49
Cluster Type 49
Node Types for CXFS Metadata Servers 50
CXFS Metadata Servers and Failover Domain 50
`CXFS` Resource Type for FailSafe 50
Separate CXFS and FailSafe GUIs 52
Conversion Between CXFS and FailSafe 52
Network Interfaces 53

3. Installation and System Preparation 55

Install Software 55
Configure System Files 59
Hostname Resolution: `/etc/sys_id`, `/etc/hosts`, `/etc/nsswitch.conf` 60

/etc/services 62
/etc/config/cad.options 62
/etc/config/fs2d.options 63
Example 1 65
Example 2 66
/etc/config/cmond.options 66
Set the corepluspid System Parameter 67
Set NVRAM Variables 67
Create XLV Logical Volumes and XFS Filesystems 67
Configure Network Interfaces 69
Configure the Serial Ports for a Ring Reset 73
Install Patches 74
Installing FailSafe 2.*x* and a FailSafe Patch at the Same Time 74
Installing a FailSafe Patch on an Existing FailSafe 2.*x* Cluster 75
Install Performance Co-Pilot (PCP) Software 79
Installing the Collector Host 79
Removing Performance Metrics from a Collector Host 82
Installing the Monitor Host 82
Test the System 83
Private Network Interface 84
Serial Reset Connection 84

4. Administration Tools 87

FailSafe Manager GUI 87
Starting the GUI 87
GUI Overview 89
Viewing the Cluster Components 90
Viewing Component Details 90

Performing Tasks . . . 91
Screens . . . 91
`cmgr` Command . . . 93
Getting Help . . . 94
Using Prompt Mode . . . 94
Completing Actions and Cancelling . . . 96
Command Line Editing within `cmgr` . . . 96
Long-Running Tasks . . . 97
Startup Script . . . 97
Entering Subcommands on the Command Line . . . 98
Using Script Files . . . 98
Template Scripts . . . 100
Invoking a Shell from within `cmgr` . . . 101

5. Configuration . . . 103

Preliminary Steps . . . 103
Verify that the Cluster `chkconfig` Flag is On . . . 104
Start the Cluster Daemons . . . 104
Verify that the Cluster Daemons are Running . . . 104
Determine the Hostname of the Node . . . 105
Name Restrictions . . . 106
Configuring Timeout Values and Monitoring Intervals . . . 106
Setting Configuration Defaults with `cmgr` . . . 107
Guided Configuration with the GUI . . . 108
Set Up a New Cluster . . . 109
Set Up a Highly Available Resource Group . . . 110
Set Up an Existing CXFS Cluster for FailSafe . . . 111

Fix or Upgrade Cluster Nodes . . . 112
Make Changes to Existing Cluster . . . 112
Optimize Node Usage . . . 112
Define Custom Resource . . . 113
Customize FailSafe Failure Detection . . . 113
Customize Resource Group Failover Behavior . . . 114
Customize Resource Failover Behavior . . . 114
Redistribute Resource Load in Cluster . . . 115
Node Tasks . . . 116
Define a Node . . . 116
Define a Node with the GUI . . . 116
Define a Node with `cmgr` . . . 119
Add/Remove Nodes in the Cluster . . . 125
Add/Remove Nodes in the Cluster with the GUI . . . 125
Modify a Node Definition . . . 126
Modify a Node Definition with the GUI . . . 126
Modify a Node with `cmgr` . . . 128
Example of Partitioning . . . 129
Convert a CXFS Node to FailSafe . . . 130
Convert a CXFS Node to FailSafe with the GUI . . . 131
Convert a Node to CXFS or FailSafe with `cmgr` . . . 131
Delete a Node . . . 132
Delete a Node with the GUI . . . 132
Delete a Node with `cmgr` . . . 133
Display a Node . . . 135
Display a Node with the GUI . . . 135
Display a Node with `cmgr` . . . 135
Cluster Tasks . . . 136

Define a Cluster . . . 137
Define a Cluster with the GUI . . . 137
Define a Cluster with cmgr . . . 138
Modify a Cluster Definition . . . 141
Modify a Cluster Definition with the GUI . . . 141
Modify a Cluster Definition with cmgr . . . 141
Convert a CXFS Cluster to FailSafe . . . 142
Convert a CXFS Cluster to FailSafe with the GUI . . . 142
Converting a CXFS Cluster to Failsafe with cmgr . . . 143
Delete a Cluster . . . 143
Delete a Cluster with the GUI . . . 143
Delete a Cluster with cmgr . . . 144
Display a Cluster . . . 145
Display a Cluster with the GUI . . . 145
Displaying a Cluster with cmgr . . . 145
Resource Type Tasks . . . 146
Define a Resource Type . . . 146
Define a Resource Type with the GUI . . . 146
Define a Resource Type with cmgr . . . 149
Redefine a Resource Type for a Specific Node . . . 155
Redefine a Resource Type for a Specific Node with the GUI . . . 155
Defining a Node-Specific Resource Type with cmgr . . . 158
Add/Remove Dependencies for a Resource Type . . . 159
Add/Remove Dependencies for a Resource Type with the GUI . . . 159
Add/Remove Dependencies for a Resource Type with cmgr . . . 161
Load a Resource Type . . . 162
Load a Resource Type with the GUI . . . 162

Load a Resource Type with cmgr . . . 162
Modify a Resource Type Definition . . . 162
Modify a Resource Type with the GUI . . . 162
Modify a Resource Type with cmgr . . . 165
Delete a Resource Type . . . 167
Delete a Resource Type with the GUI . . . 167
Delete a Resource Type with cmgr . . . 168
Display a Resource Type . . . 168
Displaying Resource Types with the GUI . . . 168
Display Resource Types with cmgr . . . 168
Resource Tasks . . . 168
Define a New Resource . . . 169
Define a New Resource with the GUI . . . 169
CXFS Attributes . . . 170
filesystem Attributes . . . 170
IP_address Attributes . . . 171
MAC_address Attributes . . . 171
volume Attributes . . . 172
Define a New Resource with cmgr . . . 172
Specify Resource Attributes with cmgr . . . 173
Redefine a Resource for a Specific Node . . . 175
Redefine a Resource for a Specific Node with the GUI . . . 175
Redefine a Resource for a Specific Node with cmgr . . . 176
Add/Remove Dependencies for a Resource Definition . . . 176
Add/Remove Dependencies for a Resource Definition with the GUI . . . 177
Add/Remove Dependencies for a Resource Definition with cmgr . . . 178
Modify a Resource Definition . . . 179

Modify a Resource Definition with the GUI . . . 179
Modify a Resource Definition with `cmgr` . . . 180
Delete a Resource . . . 180
Delete a Resource with the GUI . . . 180
Delete a Resource with `cmgr` . . . 181
Display a Resource . . . 181
Display a Resource with the GUI . . . 181
Display a Resource with `cmgr` . . . 181
Failover Policy Tasks . . . 182
Define a Failover Policy . . . 182
Define a Failover Policy with the GUI . . . 182
Define a Failover Policy with `cmgr` . . . 187
Modify a Failover Policy Definition . . . 188
Modify a Failover Policy Definition with the GUI . . . 188
Modify a Failover Policy Definition with `cmgr` . . . 191
Delete a Failover Policy . . . 191
Delete a Failover Policy with the GUI . . . 191
Delete a Failover Policy with `cmgr` . . . 192
Display a Failover Policy . . . 192
Display a Failover Policy with the GUI . . . 192
Display a Failover Policy with `cmgr` . . . 192
Resource Group Tasks . . . 193
Define a Resource Group . . . 193
Define a Resource Group with the GUI . . . 193
Defining a Resource Group with `cmgr` . . . 194
Modify a Resource Group Definition . . . 195
Modify a Resource Group Definition with the GUI . . . 195

Modify a Resource Group Definition with cmgr . . . 195
Delete a Resource Group . . . 195
Delete a Resource Group with the GUI . . . 196
Delete a Resource Group with cmgr . . . 196
Add/Remove Resources in Resource Group . . . 196
Move a Resource Group . . . 197
Move a Resource Group with the GUI . . . 197
Move a Resource Group with cmgr . . . 198
Display a Resource Group . . . 198
Display a Resource Group with the GUI . . . 198
Display a Resource Group with cmgr . . . 198
FailSafe HA Services Tasks . . . 199
Start FailSafe HA Services . . . 199
Start FailSafe HA Services with the GUI . . . 199
Start HA Services with cmgr . . . 200
Stop FailSafe HA Services . . . 200
Stop FailSafe HA Services with the GUI . . . 200
Stopping HA Services on One Node . . . 202
Stopping HA Services on All Nodes in a Cluster . . . 202
Stop FailSafe HA Services with cmgr . . . 203
Set FailSafe HA Parameters . . . 203
Set FailSafe HA Parameters with the GUI . . . 204
Set FailSafe HA Parameters with cmgr . . . 205
Set Log Configuration . . . 206
Set Log Configuration with the GUI . . . 206
Default Log File Names . . . 206
Display Log Group Definitions with the GUI . . . 209

Define Log Groups with cmgr . . . 209
Configure Log Groups with cmgr . . . 210
Modify Log Groups with cmgr . . . 211
Display Log Group Definitions . . . 212
Display Log Group Definitions with cmgr . . . 212

6. Configuration Examples . . . 213

Example: Define a Three-Node Cluster . . . 213
Example: Script to Define a Three-Node Cluster . . . 214
Example: Local Failover of HA IP Address . . . 220
Example: Modify a Cluster to Include a CXFS Filesystem . . . 221
Example: Export CXFS Filesystems . . . 222
Example: Create a Resource Group . . . 223

7. IRIS FailSafe System Operation . . . 225

Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C Console Support . . . 225
System Operation Considerations . . . 226
Two-Node Clusters: Single-Node Use . . . 227
Using a Single Node . . . 227
Resuming Two-Node Use . . . 230
System Status . . . 233
Monitoring System Status with cluster_status . . . 234
Monitoring System Status with the GUI . . . 235
Key to Icons and States . . . 235
Querying Cluster Status with cmgr . . . 237
Monitoring Resource and Reset Serial Line with cmgr . . . 238
Querying Resource Status with cmgr . . . 238

Performing a `ping` of a System Controller with `cmgr` . . . 238
Resource Group Status . . . 238
Resource Group State . . . 239
Resource Group Error State . . . 240
Resource Owner . . . 241
Monitoring Resource Group Status with GUI . . . 241
Querying Resource Group Status with `cmgr` . . . 241
Node Status . . . 242
Monitoring Node Status with `cluster_status` . . . 242
Monitoring Cluster Status with the GUI . . . 242
Querying Node Status with `cmgr` . . . 242
Performing a `ping` the System Controller with `cmgr` . . . 242
Viewing System Status with the `haStatus` Script . . . 243
Embedded Support Partner (ESP) Logging of FailSafe Events . . . 249
Resource Group Failover . . . 250
Bring a Resource Group Online . . . 250
Bring a Resource Group Online with the GUI . . . 250
Bring a Resource Group Online with `cmgr` . . . 251
Take a Resource Group Offline . . . 252
Take a Resource Group Offline with the GUI . . . 252
Take a Resource Group Offline with `cmgr` . . . 254
Move a Resource Group . . . 254
Move a Resource Group with the GUI . . . 254
Move a Resource Group with `cmgr` . . . 255
Suspend and Resume Monitoring of a Resource Group . . . 255
Suspend Monitoring a Resource Group with the GUI . . . 255
Resume Monitoring of a Resource Group with the GUI . . . 256

Putting a Resource Group into Maintenance Mode with cmgr 256
Resume Monitoring of a Resource Group with cmgr 256
Stopping FailSafe 257
Resetting Nodes 257
Reset a Node with the GUI 257
Reset a Node with cmgr 257
Backing Up and Restoring Configuration with cmgr 258
Log File Management 259
Rotating All Log Files 259

8. Testing the Configuration 261

Overview of FailSafe Diagnostic Commands 261
Performing Diagnostic Tasks with the GUI 262
Test Connectivity with the GUI 262
Test Resources with the GUI 262
Test Failover Policies with the GUI 263
Performing Diagnostic Tasks with cmgr 263
Test the Serial Connections with cmgr 263
Test Network Connectivity with cmgr 264
Test Resources with cmgr 265
Test Logical Volumes with cmgr 266
Test Filesystems with cmgr 267
Test Resource Groups with cmgr 268
Test Failover Policies with cmgr 269

9. System Recovery and Troubleshooting 271

Overview of System Recovery 271
Disabling Resource Groups for Maintenance 272

FailSafe Log Files 272
FailSafe Membership and Resets 274
FailSafe Membership and Tie-Breaker Node 274
No Membership Formed 275
Status Monitoring 276
Dynamic Control of FailSafe Services 276
Recovery Procedures 277
Single-Node Recovery 278
Cluster Error Recovery 278
Resource Group Recovery 279
Node Error Recovery 279
Resource Group Maintenance and Error Recovery 280
Clear Resource Error State 283
Control Network Failure Recovery 284
Serial Cable Failure Recovery 284
Cluster Database Sync Failure 285
Cluster Database Maintenance and Recovery 285
GUI Will Not Run 285
GUI and `cmgr` Inconsistencies 287
GUI Does Not Report Information 287
Using the `cdbreinit` Command 287
CXFS Metadata Server Relocation 288
Other Problems with CXFS Coexecution 288

10. Upgrading and Maintaining Active Clusters 289

Add a Node to an Active Cluster 289
Delete a Node from an Active Cluster 291
Change Control Networks in a Cluster 293

Upgrade OS Software in an Active Cluster 295
Upgrade FailSafe Software in an Active Cluster 296
Add New Resource Groups or Resources in an Active Cluster 297
Adding a New Hardware Device in an Active Cluster 298

11. Performance Co-Pilot for FailSafe 299
Using the Visualization Tools 299
PCP for FailSafe Performance Metrics 303
PCP Gray Display 304

Appendix A. Software Overview 307
Software Layers 307
Interface Agent Daemon (IFD) 311
Communication Paths 311
Communication Paths in a Coexecution Cluster 316
Execution of FailSafe Action and Failover Scripts 317
When a `start` Script Fails 321
When a `stop` Script Fails 321
Components 321

Appendix B. Updating from IRIS FailSafe 1.2 to IRIS FailSafe 2.1.*x* 323
Hardware Changes 323
Software Changes 324
Configuration Changes 324
Scripts 325
Operational Comparison 326
Upgrade Examples 327
Defining a Node 328

Defining a Cluster 329
Setting HA Parameters 330
Defining a Resource: XLV Volume 332
Defining a Resource: XFS Filesystem 332
Defining a Resource: IP Address 333
Additional FailSafe 2.1.*x* Tasks 334
Status 335

Appendix C. IRIS FailSafe 2.1.*x* Software 337

Subsystems on the CD 337
Subsystems for Servers and Workstations in the Pool 339
Additional Subsystems for Nodes in the FailSafe Cluster 340
Additional Subsystems for Administrative Workstations 340
Subsystems for IRIX Administrative Workstations 341
Subsystems for Non-IRIX Administrative Workstations 341

Appendix D. Metrics Exported by PCP for FailSafe 343

Glossary 351

Index 361

Figures

Figure 1-1 Pool and Cluster Concepts 4
Figure 1-2 FailSafe Membership 6
Figure 1-3 Tie-breaker and Membership 7
Figure 1-4 Resource Type Dependencies 11
Figure 1-5 Sample System Components 15
Figure 1-6 Configuration Types 18
Figure 1-7 Reset Types 19
Figure 1-8 Disk Storage Failover on a Two-Node System 25
Figure 2-1 Example Configuration with Four Resource Groups 32
Figure 2-2 Non-Shared Disk Configuration and Failover 35
Figure 2-3 Shared Disk Configuration for Active/Backup Use 37
Figure 2-4 Shared Disk Configuration For Dual-Active Use 38
Figure 2-5 Example Logical Volume Configuration 41
Figure 2-6 Filesystems and Logical Volumes 45
Figure 3-1 Example Interface Configuration 69
Figure 4-1 FailSafe Manager GUI 92
Figure 4-2 GUI Showing Details for a Resource 93
Figure 5-1 Dependencies 160
Figure 5-2 Example of Resource Dependency 177
Figure 5-3 Mutual Dependency of Resources Is Not Allowed 178
Figure 6-1 FailSafe Configuration Example 214
Figure 11-1 Heartbeat Response Statistics 300
Figure 11-2 Resource Monitoring Statistics 301

Figure A-1 Software Layers 309
Figure A-2 Administrative Communication within One Node 312
Figure A-3 Daemon Communication within One Node 313
Figure A-4 Communication between Nodes in the Pool 314
Figure A-5 Communication for a Node Not in the Cluster 315
Figure A-6 Administrative Communication within One Node under Coexecution . . . 316
Figure A-7 Daemon Communication within One Node under Coexecution 317
Figure A-8 Message Paths for Action Scripts and Failover Policy Scripts 320

Tables

Table 1-1 Example `Webgroup` Resource Group 9

Table 2-1 XLV Logical Volume Configuration Parameters 42

Table 2-2 Filesystem Configuration Parameters 44

Table 2-3 HA IP Address Configuration Parameters 48

Table 3-1 `fs2d.options` File Options 64

Table 3-2 PCP for FailSafe Collector Subsystems 80

Table 3-3 PCP for FailSafe Monitor Subsystems 82

Table 4-1 Template Scripts for `cmgr` 100

Table 5-1 System Controller Types 123

Table 5-2 Resource Type Attributes 174

Table 5-3 Failover Attributes 184

Table 5-4 Log Levels 207

Table 5-5 Default Log File Names 208

Table 6-1 Resources and Failover Policies for `RG1` and `RG2` 215

Table 7-1 Key to Icons 236

Table 7-2 Key to States 237

Table 8-1 FailSafe Diagnostic Test Summary 261

Table 9-1 Message Levels 273

Table A-1 Provided and Optional Plug-Ins 307

Table A-2 Contents of `/usr/cluster/bin` 310

Table A-3 Contents of the `/var/cluster/ha` directory 322

Table B-1 Differences Between IRIS FailSafe 1.2 and 2.1.*x* 326

Table C-1 IRIS FailSafe 2.1.*x* CD 338

Table C-2 Subsystems Required for Nodes in the Pool (Servers and GUI Client(s)) . . . 339
Table C-3 Additional Subsystems Required for Nodes in the Cluster 340
Table C-4 Subsystems Required for IRIX Administrative Workstations 341
Table D-1 PCP Metrics 343

About This Guide

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

This guide was prepared in conjunction with release 2.1.3 of the IRIS FailSafe product. It supports IRIX 6.5.16 and later.

Audience

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

Assumptions

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

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

Structure of This Guide

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

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

- Chapter 3, "Installation and System Preparation", describes several procedures that must be performed on nodes in a cluster to prepare them for FailSafe.
- Chapter 4, "Administration Tools", provides an overview of the FailSafe Manager GUI and the `cmgr`(1M) command.
- Chapter 5, "Configuration", explains how to configure a FailSafe system.
- Chapter 6, "Configuration Examples", shows an example of a FailSafe three-node configuration and some variations on that configuration.
- Chapter 7, "IRIS FailSafe System Operation", explains how to operate and monitor a FailSafe system.
- Chapter 8, "Testing the Configuration", describes how to test the configured FailSafe system.
- Chapter 9, "System Recovery and Troubleshooting", describes the log files used by FailSafe and recovery procedures.
- Chapter 10, "Upgrading and Maintaining Active Clusters", describes some procedures you may need to perform without shutting down a FailSafe cluster.
- Chapter 11, "Performance Co-Pilot for FailSafe", tells you how to use PCP to monitor the availability of a FailSafe cluster.
- Appendix B, "Updating from IRIS FailSafe 1.2 to IRIS FailSafe 2.1.*x*", describes the upgrade procedure.
- Appendix C, "IRIS FailSafe 2.1.*x* Software", summarizes the systems to install on each component of a cluster.
- Appendix D, "Metrics Exported by PCP for FailSafe", lists the metrics implemented by `pmdafsafe`(1).

Related Documentation

The following documentation will be useful in a FailSafe environment:

- *IRIS FailSafe Version 2 Programmer's Guide*
- *Performance Co-Pilot User's and Administrator's Guide*
- *CXFS Version 2 Software Installation and Administration Guide*

- *IRIS FailSafe 2.0 DMF Administrator's Guide*
- *IRIS FailSafe 2.0 INFORMIX Administrator's Guide*
- *IRIS FailSafe 2.0 Netscape Server Administrator's Guide*
- *IRIS FailSafe Version 2 NFS Administrator's Guide*
- *IRIS FailSafe 2.0 Oracle Administrator's Guide*
- *IRIS FailSafe Version 2 Samba Administrator's Guide*
- *IRIS FailSafe Version 2 TMF Administrator's Guide*

Embedded Support Partner User Guide

The IRIS FailSafe man pages are as follows:

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

- `haStatus`(1M)

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

Release Note	Product
`cluster_admin`	Cluster administration services
`cluster_control`	Node control services
`cluster_services`	Cluster services
`failsafe2`	IRIS FailSafe 2.1.*X*
`failsafe2_dmf`	IRIS FailSafe/DMF
`failsafe2_informix`	IRIS FailSafe INFORMIX
`failsafe2_nfs`	IRIS FailSafe NFS
`failsafe2_oracle`	IRIS FailSafe Oracle
`failsafe2_samba`	IRIS FailSafe Samba
`failsafe2_tmf`	IRIS FailSafe/TMF
`failsafe2_web`	IRIS FailSafe Netscape Web

Obtaining Publications

To obtain SGI documentation, go to the SGI Technical Publications Library at:

`http://techpubs.sgi.com`.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.

`manpage`(*x*)	Man page section identifiers appear in parentheses after man page names. (1) indicates a user command, (1M) indicates an administrator command.
variable	Italic typeface denotes variable entries and words or concepts being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.
[ ]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.

In this guide, the term *FailSafe* is used as an abbreviation for *IRIS FailSafe*.

Reader Comments

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

You can contact us in any of the following ways:

- Send e-mail to the following address:

  `techpubs@sgi.com`

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

  `http://techpubs.sgi.com`

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

- Send mail to the following address:

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

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

We value your comments and will respond to them promptly.

Overview

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

- "High Availability and IRIS FailSafe"
- "Cluster Environment", page 3
- "Additional Features", page 20
- "Administration", page 21
- "Highly Available Resources", page 21
- "Highly Available Applications", page 25
- "Failover and Recovery Processes", page 26
- "Overview of Configuring and Testing a New Cluster", page 27

Also see Appendix A, "Software Overview", page 307

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. This solution is very expensive and requires specialized hardware and software.
- Highly available systems. These systems survive single points of failure by using redundant off-the-shelf components and specialized software. They provide a lower degree of availability than the fault-tolerant systems, but at much lower cost. Typically these systems provide high availability only for client/server

applications, and base their redundancy on cluster architectures with shared resources.

The SGI IRIS FailSafe product provides a general facility for providing highly available services. FailSafe provides highly available services for a cluster that contains multiple nodes (*N*-node configuration).

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

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

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

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

Cluster Environment

This section discusses the following:

- "Terminology", page 3
- "Hardware Components", page 15
- "Disk Connections", page 17
- "Supported Configurations", page 17

Terminology

This section defines the terminology necessary to configure and monitor highly available services with FailSafe.

Cluster

The *cluster* is the set of systems (nodes) configured to work together as a single computing resource. The cluster is identified by a simple name; this name must be unique within the pool. (For example, you cannot use the same name for the cluster and for a node.)

Node

A *node* is an operating system (OS) image, usually an individual computer. The nodes are connected to a storage area network (SAN) that connects the storage systems to the nodes in the cluster. A node can belong to only one cluster.

This use of the term *node* does not have the same meaning as a node in an SGI Origin 3000 or SGI 2000 system.

Pool

The *pool* is the set of nodes from which a particular cluster may be formed. Only one cluster may be configured from a given pool, and it need not contain all of the available nodes. (Other pools may exist, but each is disjoint from the other. They share no node or cluster definitions.)

Figure 1-1 shows the concepts of pool and cluster.

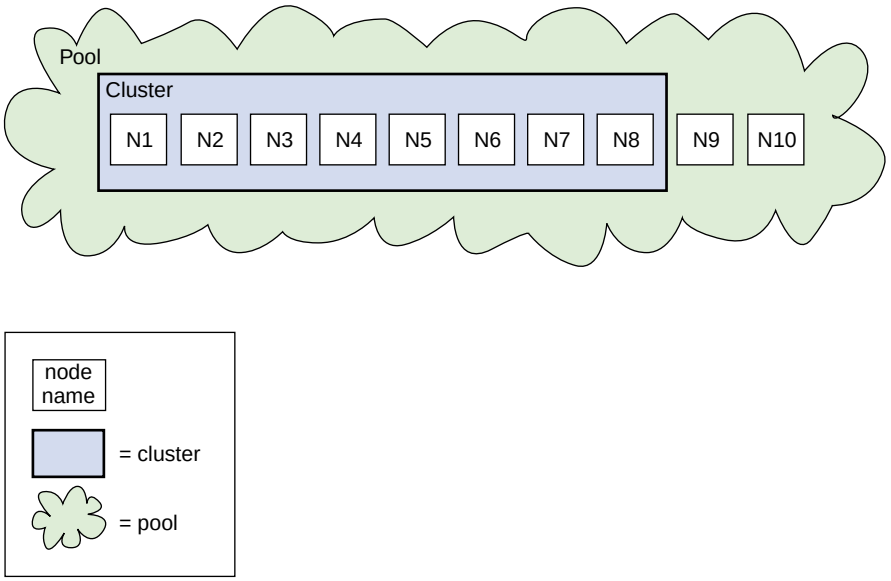

Figure 1-1 Pool and Cluster Concepts

Cluster Database

The *cluster database* contains configuration information about all nodes and the cluster. The `fs2d` daemon manages the distribution of the cluster database (CDB) across the nodes in the pool.

Membership

There are the following types of membership:

- *FailSafe membership* is the list of FailSafe nodes in the **cluster** on which FailSafe can make resource groups online.
- `fs2d` *database membership* (also known as *user-space membership*) is the group of nodes in the **pool** that are accessible to `fs2d`. The `fs2d` daemon manages the distribution of the cluster database across the nodes in the pool; nodes available to `fs2d` are able to receive cluster database updates, and are therefore part of the `fs2d` database membership; this may be a subset of the nodes defined in the pool.

- *Process membership* is the list of process instances in a cluster that form a cluster process group. There can be multiple process groups per node.

With CXFS coexecution, there is also *CXFS membership*. For more information about CXFS, see "Coexecution of CXFS and IRIS FailSafe", page 48, and the *CXFS Version 2 Software Installation and Administration Guide*.

Quorum

The *quorum* is the number of nodes required to form a cluster, which differs according to membership:

- For FailSafe membership: **>50%** (a majority) of the nodes in the cluster where highly available (HA) services were started must be in a known state (successfully reset or talking to each other using heartbeat and control networks) are required to form and maintain a cluster.
- For `fs2d` database membership, **50%** (half) of the **nodes in the pool** must be available to the `fs2d` daemon (and can therefore receive cluster database updates) to form and maintain a cluster.

Figure 1-2 shows an example of FailSafe and `fs2d` memberships. The figure describes the following:

- A pool consisting of six nodes, N1 through N6.
- A cluster that has been defined to have four nodes, N1 to N4.
- HA services have been started on four nodes, N1 to N4. (HA services can only be started on nodes that have been defined as part of the cluster; however, not all nodes within the cluster must have HA services started.)
- Three nodes are up (N1 through N3) and three nodes are down (N4 through N6).
- The `fs2d` membership consists of N1 through N3, 3 of 6 nodes in the pool (50%).
- The FailSafe membership also consists of nodes N1 through N3, 3 of 4 nodes where HA services were started.

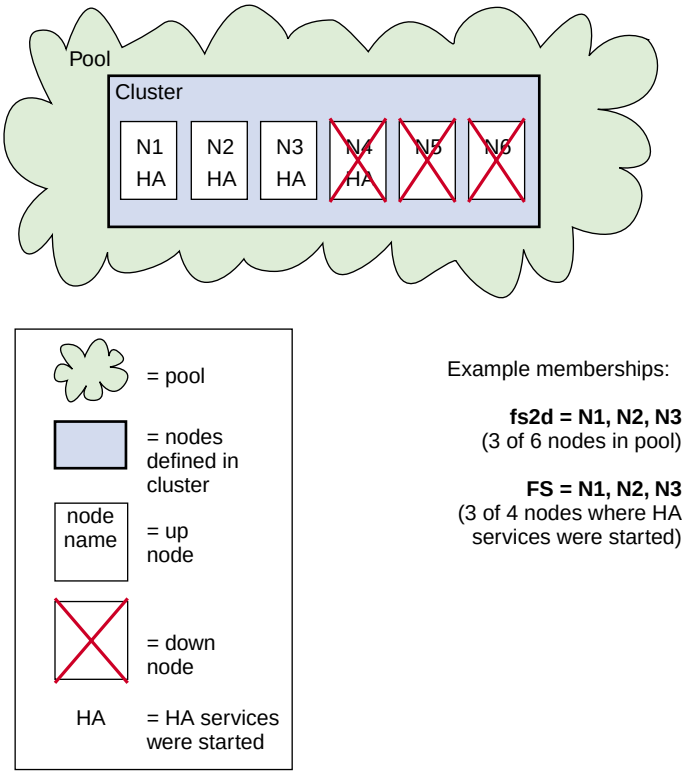

Figure 1-2 FailSafe Membership

If a network partition results in a tied membership, in which there are two sets of nodes each consisting of 50% of the cluster, a node from the set containing the tie-breaker node will attempt to reset a node in the other set in order to maintain a quorum. Figure 1-3 shows an example of this, including the following:

- Because nodes N1 and N2 are the set containing the tie breaker, they will both try to reset the nodes in the other set (N3 and N4).
- After the reset is successful and the membership of N1 and N2 is confirmed, resource groups that were in N3 and N4 will be moved to N1 and N2, assuming that is what is defined in their failover policies.

Note: The figure really shows one sequence of events; the different time designations are for illustration purposes.

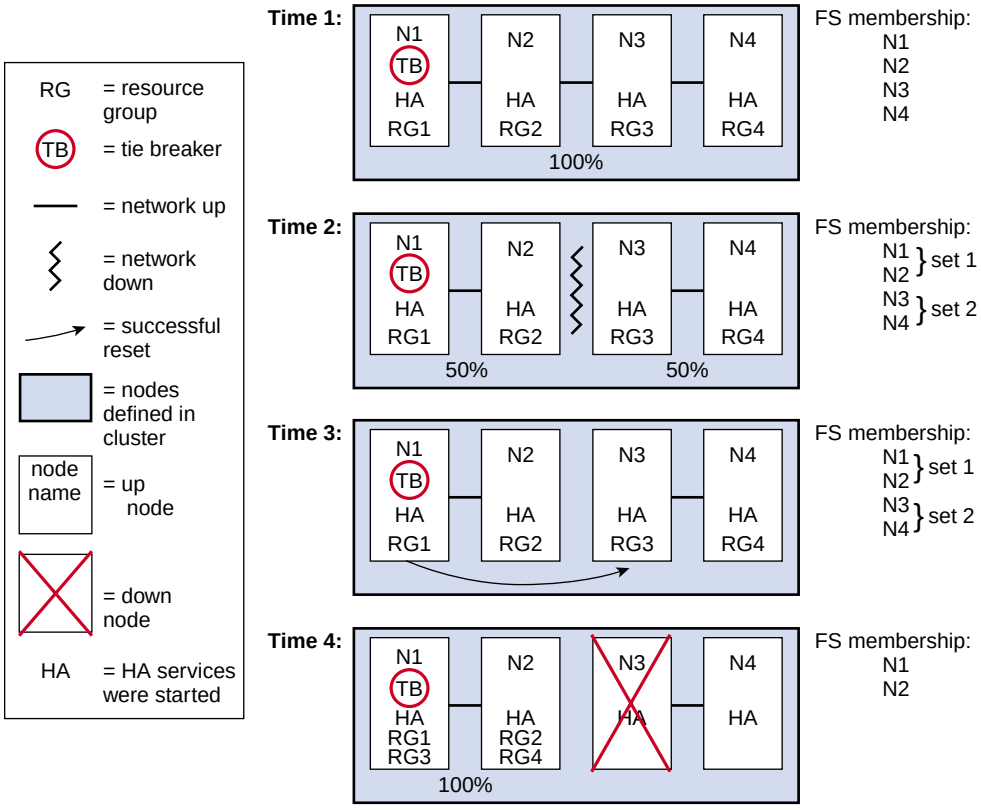

Figure 1-3 Tie-breaker and Membership

Private Network

A *private network* is one that is **dedicated** to cluster communication and is accessible by administrators but not by users.

The cluster software uses the private network to send the heartbeat/control messages necessary for the cluster configuration to function. If there are delays in receiving

heartbeat messages, the cluster software may determine that a node is not responding and remove it from FailSafe membership.

Using a private network limits the traffic on the public network and therefore will help avoid unnecessary resets or disconnects.

The messaging protocol does not prevent snooping (viewing) or spoofing (in which one machine on the network masquerades as another); therefore, a private network is safer than a public network.

Therefore, because the performance and security characteristics of a public network could cause problems in the cluster and because heartbeat is very timing-dependent (even small variations can cause problems), SGI recommends a dedicated private network to which all nodes are attached. This private network would then be used to send heartbeat/control messages.

In addition, SGI recommends that all nodes be on the same local network segment.

Note: If there are any network issues on the private network, fix them before trying to use FailSafe.

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.

Resource Type

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

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

The FailSafe software includes many predefined resource types. If these types fit the application you want to make highly available, you can reuse them. If none fit, you can create additional resource types by using the instructions in the *IRIS FailSafe Version 2 Programmer's Guide*.

Resource Name

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

Resource Group

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

Table 1-1 Example `Webgroup` Resource Group

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

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

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

Dependency

One resource can be dependent on one or more other resources; if so, it will not be able to start (that is, be made available for use) unless the dependent resources are also started. Dependent resources must be part of the same resource group and are

identified in a *resource dependency list*. Resource dependencies are verified when resources are added to a resource group, not when resources are defined.

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. A *resource type dependency list* details the resource types upon which a resource type depends.

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 `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`. This is shown in Figure 1-4.

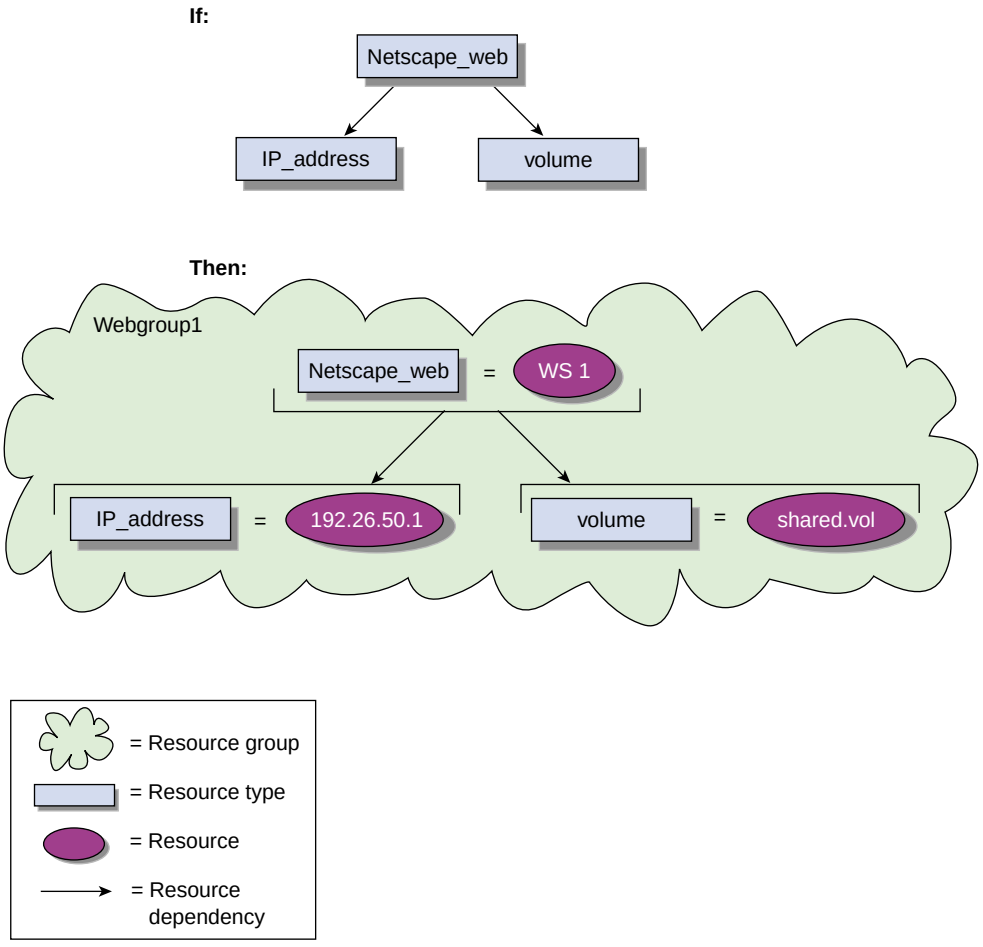

Figure 1-4 Resource Type Dependencies

Failover

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

Failover Policy

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

- Failover domain
- Failover attributes
- Failover script

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

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

Failover Domain

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

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

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

Failover Attribute

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

Failover Scripts

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

The following failover scripts are provided with the FailSafe release:

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

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

Action Scripts

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

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

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

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

Cluster Process Group

A *cluster process group* is a group of application instances in a distributed application that cooperate to provide a service. Each application instance can consist of one or more UNIX processes and spans only one node.

For example, distributed lock manager instances in each node would form a process group. By forming a process group, they can obtain process membership and reliable, ordered, atomic communication services.

Note: There is no relationship between UNIX process group and cluster process group.

Plug-In

A *plug-in* is the set of software required to make an application highly available, including a resource type and action scripts. There are plug-ins provided with the base FailSafe release, optional plug-ins available for purchase from SGI, and customized plug-ins you can write using the instructions in the *IRIS FailSafe Version 2 Programmer's Guide*. For a list of provided and optional plug-ins, see "Software Layers", page 307.

Hardware Components

Figure 1-5 shows an example of FailSafe hardware components, in this case for a two-node system.

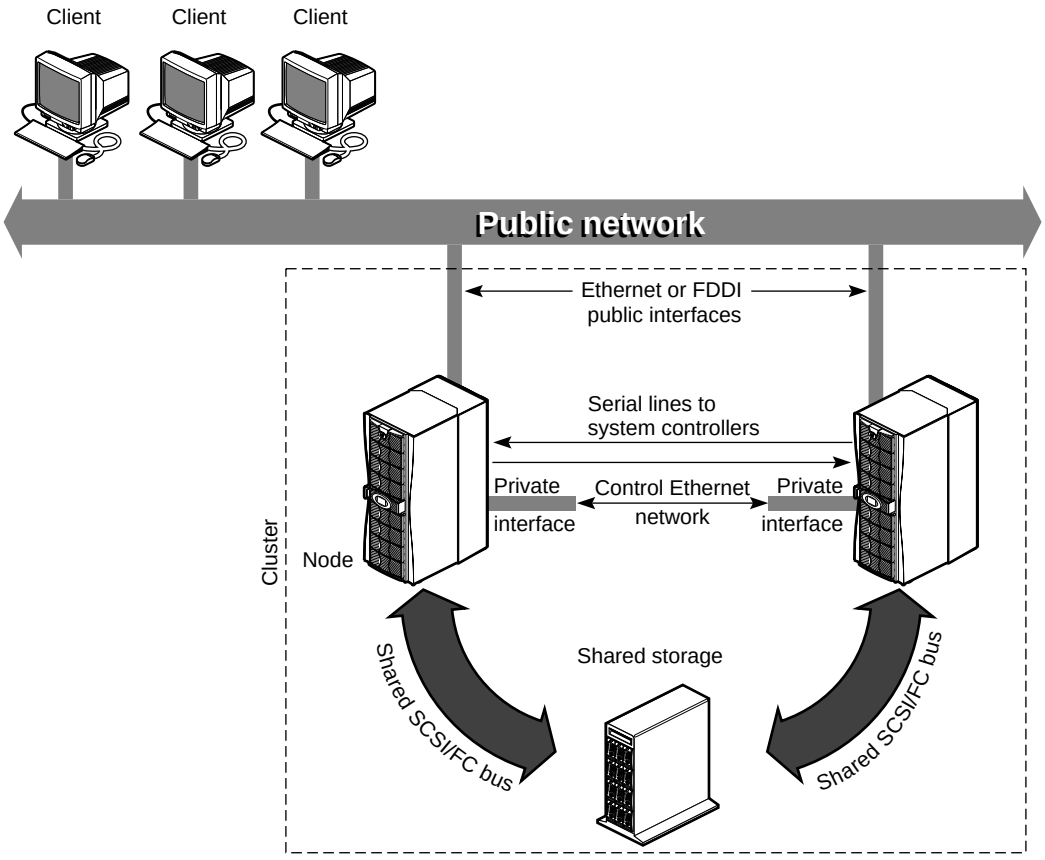

Figure 1-5 Sample System Components

The hardware components are as follows:

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

At least two Ethernet or FDDI interfaces on each node are required for the control network *heartbeat* connection, by which each node monitors the state of other nodes. The 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/Fibre Channel shared by the nodes in the cluster.

  The nodes in the FailSafe system share dual-hosted disk storage over a shared fast and wide SCSI bus or Fibre Channel. The storage connection is shared so that either node can take over the disks in case of failure. The hardware required for the disk storage can be one of the following:

  – Origin SCSI JBOD/RAID

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

  – TP9100

  – TP9400

  – TP9900

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

Note: The 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.

Disk Connections

A FailSafe system supports the following disk connections:

- RAID support
  - Single 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.

Supported Configurations

FailSafe supports the following highly available configurations:

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

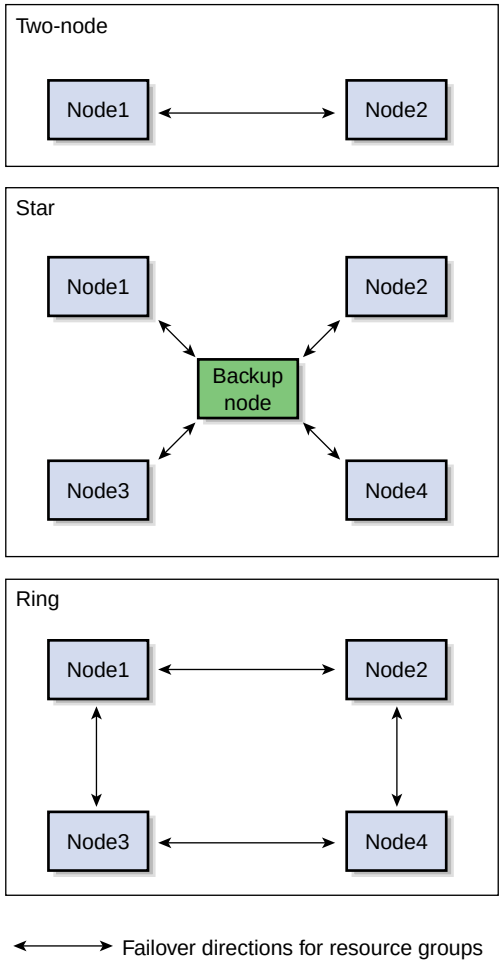

Figure 1-6 Configuration Types

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

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

- Server-to-server. Each server is directly connected to another for reset. May be unidirectional.

- Network. Each server can reset any other by sending a signal over the control network to an EL-16 multiplexer.
- IRISconsole. Each server can request that the IRISconsole perform resets.

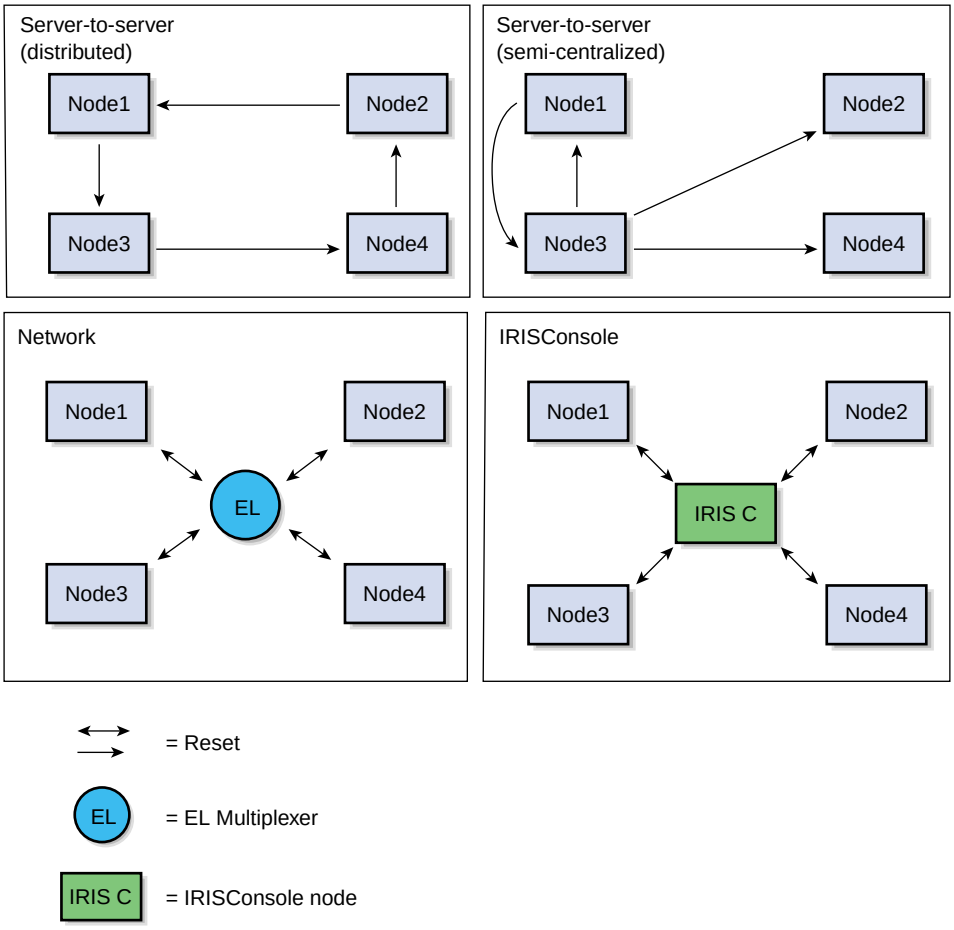

Figure 1-7 Reset Types

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

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

Additional Features

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

- Dynamic management
- Fine-grain failover
- Local restarts

These features are summarized in the following sections.

Dynamic Management

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

- **Monitor applications**. You can turn monitoring of an application on and off while FailSafe continues to run. This allows you to perform online application upgrades without bringing down the FailSafe system.
- **Managed resources**. You can add resources while the FailSafe system is online.
- **Upgrade FailSafe software**. You can upgrade FailSafe software on one node at a time without taking down the entire FailSafe cluster.

Fine-Grain Failover

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

Local Restarts

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

Administration

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

If you wish, you can perform administrative tasks directly by using the `cmgr`(1M) command, which provides a command-line interface for the administration tasks.

For more information, see Chapter 4, "Administration Tools", page 87, see Chapter 5, "Configuration", page 103, and Chapter 7, "IRIS FailSafe System Operation", page 225.

Highly Available Resources

This section discusses the highly available resources in a FailSafe system:

- Nodes
- Network interfaces and IP addresses
- Disks

Nodes

If a node crashes or hangs (for example, due to a parity error or bus error), the 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 FailSafe system handles or eliminates these points of failure.

Network Interfaces and IP Addresses

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

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

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

Note: That is, the hostname is bound to a different IP address that never moves.

FailSafe requires that each network interface in a cluster have an IP address that does not fail over. These IP addresses, called *fixed IP addresses*, are used to monitor network interfaces. The fixed IP address would be the same address you would use if you configured it as a normal system and put it on the network before FailSafe was even installed.

Each fixed IP address must be configured to a network interface at system boot up time. All other IP addresses in the cluster are configured as *highly available (HA) IP addresses*.

Highly available IP addresses are configured on a network interface. During failover and recovery processes they moved to another network interface in the other node by FailSafe. Highly available IP addresses are specified when you configure the FailSafe

system. 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. FailSafe uses medium access control (MAC) address impersonation (*re-MACing*) to support these networking implementations. Re-MACing moves the physical MAC address of a network interface to another interface. This is done by using the `macconfig` command. Re-MACing is done in addition to the standard `ifconfig` process that FailSafe uses to move IP addresses. This requires two network connections into the public network for each MAC address. For each MAC address being moved, a dedicated backup network interface is required. To do re-MACing in FailSafe, a resource of type `MAC_Address` is used.

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

Re-MACing is required when packets called *gratuitous ARP packets* are not passed through the network. These packets are generated automatically when an IP address is added to an interface (as in a failover process). They announce a new mapping of an IP address to a 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 HA IP Address Configuration", page 45, 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 FailSafe system supports storage based on SCSI or Fibre Channel.

XLV plexing must be used to mirror disks in a JBOD configuration. If highly available applications use filesystems, XFS filesystems or CXFS filesystems must be used. When CXFS filesystems are used, they must be on XVM volumes.

Note: No SCSI storage nor FIBRE JBOD is supported in a SAN configuration and therefore cannot use CXFS.

The storage components should not have a single point of failure. All data should be in RAID or mirrored (using XLV). It is recommended that there are at least two paths from storage to the servers for redundancy.

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

For all the above storage systems, if a disk or disk controller fails, either XLV or XVM will keep the service available through a redundant path as appropriate.

If no alternate paths are available to the storage subsystems, then FailSafe will initiate a failover process.

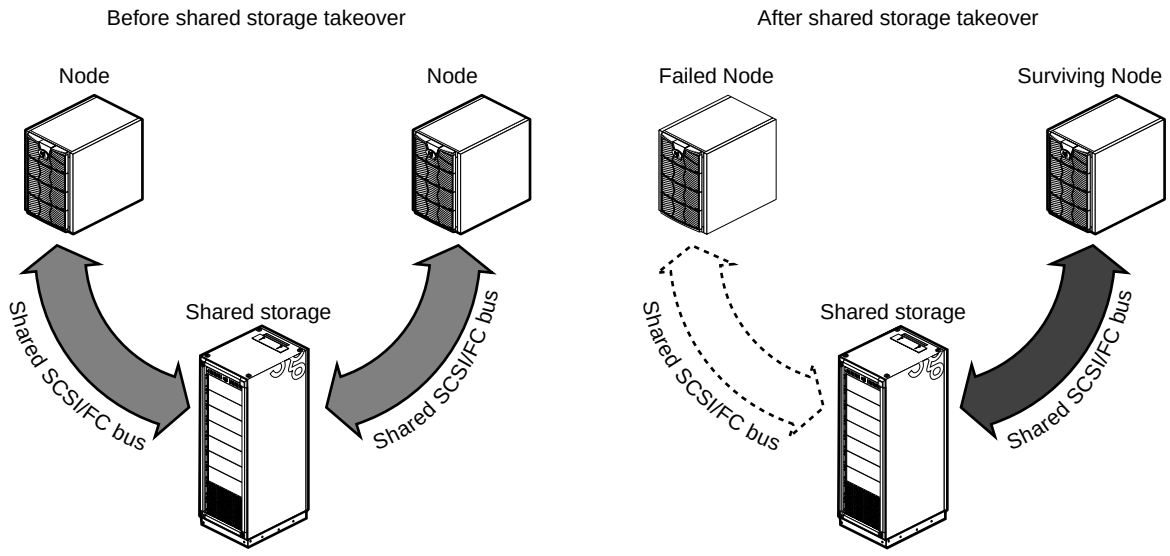

Figure 1-8 Disk Storage Failover on a Two-Node System

Highly Available Applications

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

All information about resources, including the primary node, components of the resource group, and failover policy is specified when you configure your FailSafe system with the GUI or with the `cmgr`(1M) command. Information on configuring the system is provided in Chapter 5, "Configuration", page 103. Monitoring scripts detect the failure of a resource.

The FailSafe software provides a framework for making applications highly available services. By writing scripts and configuring the system in accordance with those scripts, you can turn client/server applications into highly available applications. For information, see the *IRIS FailSafe Version 2 Programmer's Guide*.

Failover and Recovery Processes

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

Depending on which node detects the failure, the sequence of actions following the failure is different.

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

- Stops the highly available resource group running on the failed node
- Moves the highly available resource group to a different node, according to the defined failover policy for the resource group
- Asks the new node 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 the following operations:

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

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

- Power-cycles the failed node (to prevent corruption of data) by using the serial connection between the nodes
- Transfers ownership of the resource group from the failed node to the other nodes in the cluster, based on the resource group failover policy
- Starts offering the resource group services that were running on the failed node

When a failed node comes back up, whether the node automatically starts to provide highly available services again depends on the failover policy you define.

For more information, see "Define a Failover Policy with the GUI", page 182.

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

For further information on FailSafe execution during startup and failover, see "Execution of FailSafe Action and Failover Scripts", page 317.

Overview of Configuring and Testing a New Cluster

After the FailSafe cluster hardware has been installed, use the following general procedure to configure and test the FailSafe system:

1. Become familiar with FailSafe terms by reviewing this chapter.
2. Plan the configuration of highly available applications and services on the cluster using Chapter 2, "Configuration Planning".
3. Perform various administrative tasks, including the installation of prerequisite software, that are required by FailSafe, as described in Chapter 3, "Installation and System Preparation".
4. Define the configuration as explained in Chapter 5, "Configuration", page 103.
5. Test the system in three phases:
   - Test individual components prior to starting FailSafe software.
   - Test normal operation of the system.
   - Simulate failures to test the operation of the system after a failure occurs.

Configuration Planning

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

- "Overview"
- "Disk Configuration", page 33
- "Logical Volume Configuration", page 39
- "Filesystem Configuration", page 42
- "HA IP Address Configuration", page 45
- "Coexecution of CXFS and IRIS FailSafe", page 48

Overview

You must first decide how you want to use the cluster. You can then configure the disks and interfaces to meet the needs of the highly available (HA) services you want the cluster to provide.

Questions to be Answered

Questions you must answer during the planning process are as follows:

- How do you plan to use the nodes?

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

- Which of these uses will be provided as an HA service?

  SGI has developed FailSafe software options for some HA applications; see "Software Layers", page 307. To offer other applications as HA services, you must develop a set of application monitoring shell scripts as described in the *IRIS FailSafe Version 2 Programmer's Guide*. If you need assistance, contact SGI Professional Services, which offers custom FailSafe agent development and integration services.

- Which node will be the primary node for each HA service?

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

- For each HA 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.

- How will shared disks be configured?

  – As raw XLV logical volumes?

  – As XLV logical volumes with XFS filesystems on them?

  – As CXFS filesystems, which use XVM logical volumes? For information on using FailSafe and CXFS, see "Coexecution of CXFS and IRIS FailSafe", page 48.

  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 HA services?

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

- Which resources will be part of a resource group?

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

- What will be the failover domain of the resource group? (For more information about failover domains, see "Failover Domain", page 12.)

  The failover domain determines the list of nodes in the cluster where the resource group can reside. For example, a volume resource that is part of a resource group

can reside only in nodes from which the disks composing the volume can be accessed.

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

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

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

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

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

Example

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

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

There are four resource groups:

- `NFSgroup1`
- `NFSgroup2`
- `Webgroup1`
- `Webgroup2`

`NFSgroup1` and `NFSgroup2` are the NFS resource groups; `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 `ef1` interfaces in each node are connected to each other to form a private network.

Figure 2-1 depicts this configuration.

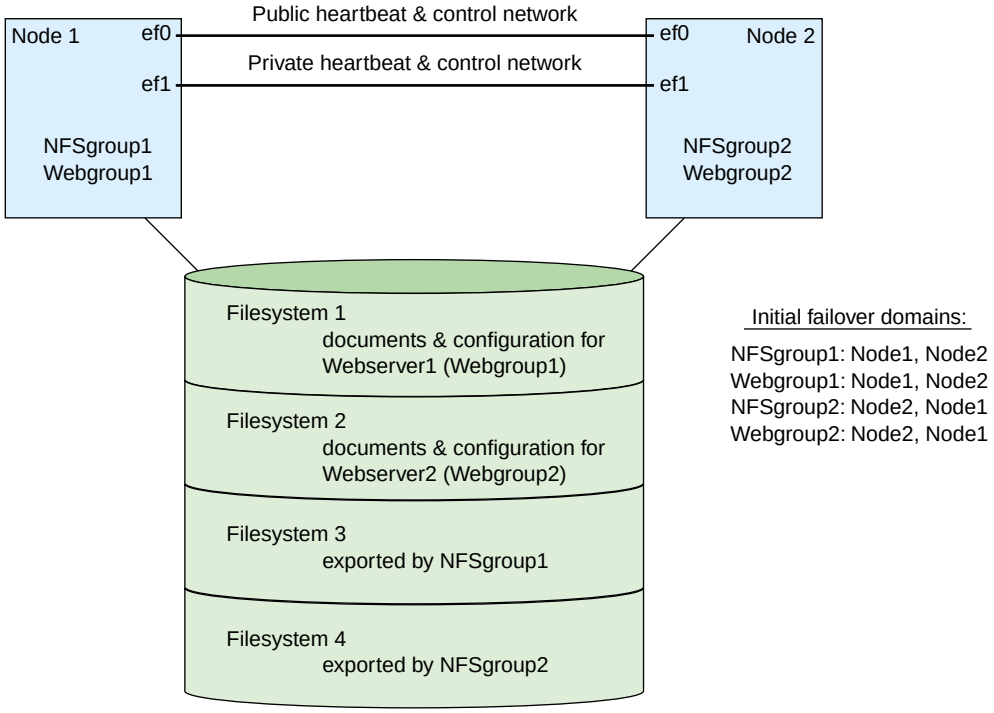

Figure 2-1 Example Configuration with Four Resource Groups

Disk Configuration

This section contains the following:

- "Planning Disk Configuration"
- "Configuration Parameters for Disks", page 39

Planning Disk Configuration

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

The nodes in a FailSafe cluster must follow these requirements:

- The system disk must be a non-shared disk.
- The FailSafe software must be on a non-shared disk.
- All system directories (`/tmp`, `/var`, `/usr`, `/bin`, `/dev`, ...)should be in non-shared disk.

Only HA application data and configuration data can be placed on a shared disk. Choosing to make a disk shared or non-shared depends on the needs of the HA services that use the disk. Each HA 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 FailSafe clusters before and after failover. A cluster can contain a combination of the following basic disk configurations:

- A non-shared disk on each node
- Multiple shared disks containing Web server and NFS file server documents

Note: 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.

Figure 2-2 shows two nodes in a cluster, each of which has a non-shared disk with two resource groups. When non-shared disks are used by HA applications, the data required by those applications must be duplicated on non-shared disks on both nodes. The clients should access the data in the shared disk using HA IP alias. When a failover occurs, HA IP aliases fail over.

Note: The hostname is bound to a different IP address that never moves.

The data that was originally available on the failed node is still available from the replacement node by using the HA IP alias to access it.

The configuration in Figure 2-2 contains two resource groups:

Resource Group	Resource Type	Resource
`Group1`	`IP_address`	`192.26.50.1`
`Group2`	`IP_address`	`192.26.50.2`

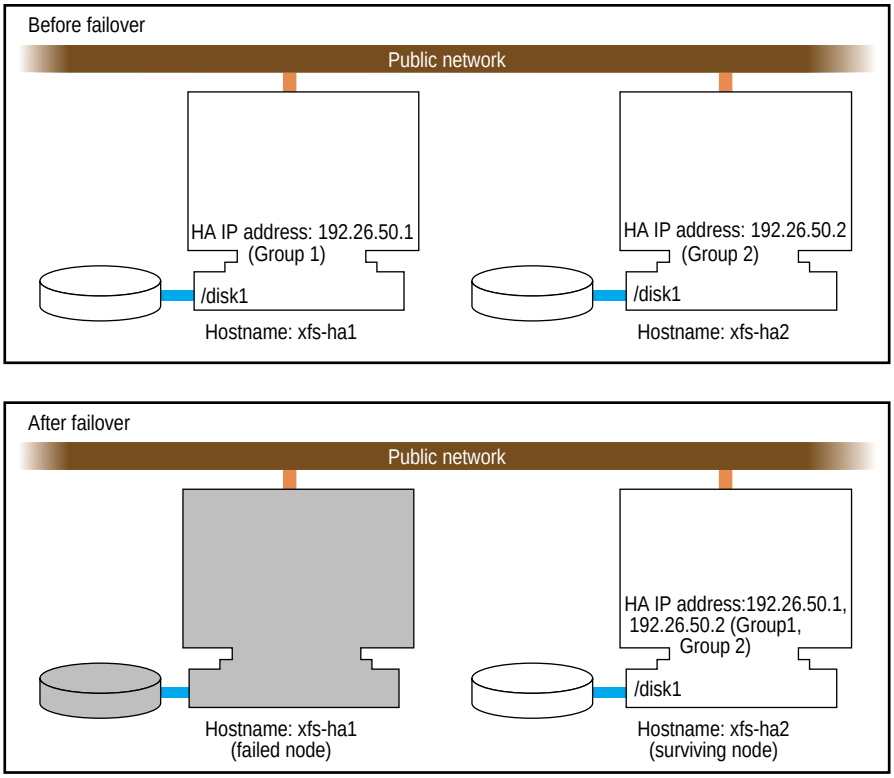

Figure 2-2 Non-Shared Disk Configuration and Failover

Figure 2-3 shows a two-node configuration with one resource group, `Group1`:

Resource Group	Resource Type	Resource	Failover Domain
`Group1`	`IP_address`	`192.26.50.1`	`(xfs-ha1, xfs-ha2)`
	`filesystem`	`/shared`	
	`volume`	`shared_vol`	

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 does not run any HA resource groups until a failover occurs.

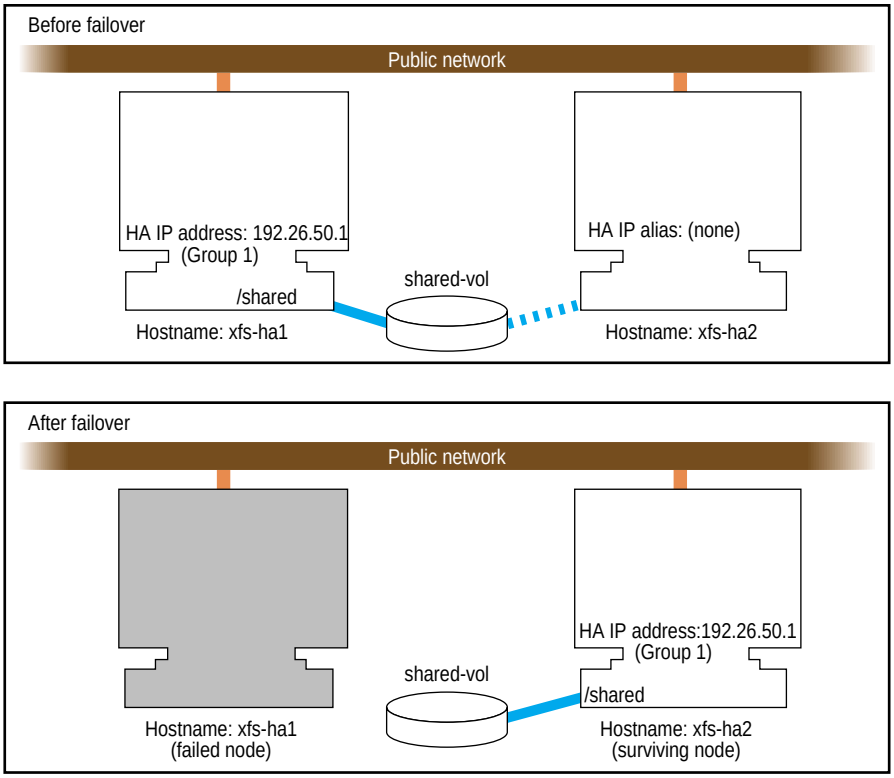

Figure 2-3 Shared Disk Configuration for Active/Backup Use

Figure 2-4 shows two shared disks in a two-node cluster with two resource groups, `Group1` and `Group2`:

Resource Group	Resource Type	Resource	Failover Domain
`Group1`	`IP_address`	`192.26.50.1`	`(xfs-ha1, xfs-ha2)`
	`filesystem`	`/shared1`	
	`volume`	`shared1_vol`	
`Group2`	`IP_address`	`192.26.50.2`	`(xfs-ha2, xfs-ha1)`

Resource Group	Resource Type	Resource	Failover Domain
	`filesystem`	`/shared2`	
	`volume`	`shared2_vol`	

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 fail over. The dotted lines show the connections to the backup nodes. After a failover, the surviving node has all the resource groups.

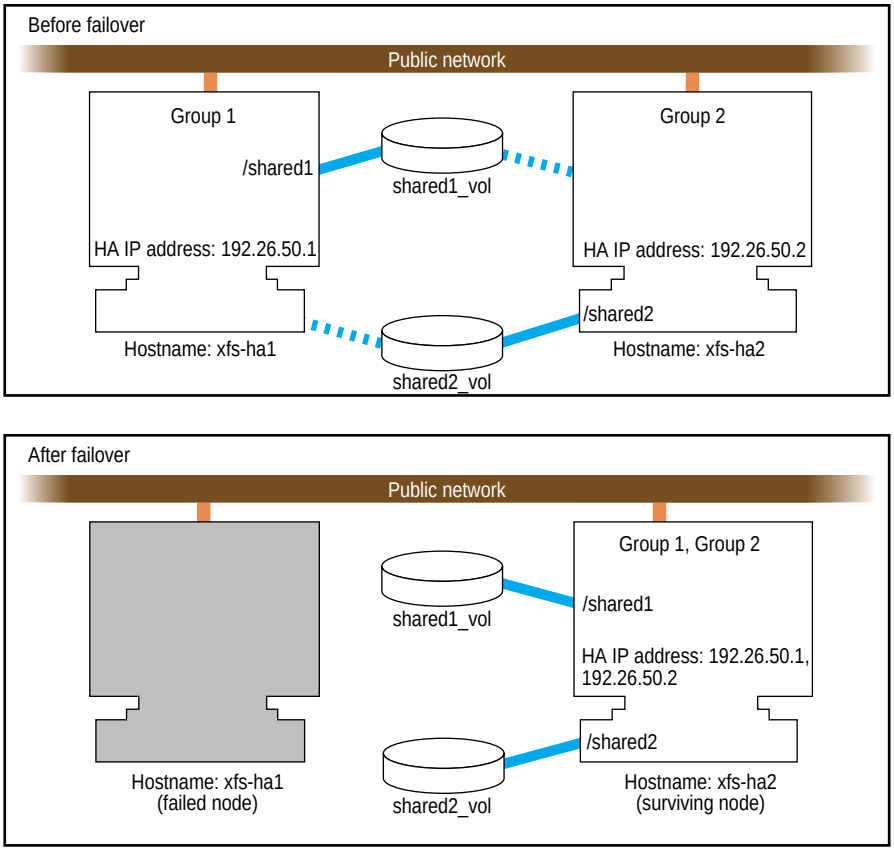

Figure 2-4 Shared Disk Configuration For Dual-Active Use

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 HA service.

Configuration Parameters for Disks

There are no configuration parameters associated with non-shared disks. They are not specified when you configure a FailSafe system. Only the XLV logical volumes on shared disks are specified at configuration. For more information, see "Configuration Parameters for Logical Volumes", page 42.

For information on using CXFS filesystems (which use XVM logical volumes) in a FailSafe configuration, see "Coexecution of CXFS and IRIS FailSafe", page 48.

Logical Volume Configuration

Note: This section describes logical volume configuration using XLV logical volumes. For information on coexecution of FailSafe and CXFS filesystems (which use XVM logical volumes), see "Coexecution of CXFS and IRIS FailSafe", page 48.

This section contains the following:

- "Planning XLV Logical Volumes"
- "Example Logical Volume Configuration", page 41
- "Configuration Parameters for Logical Volumes", page 42

Planning XLV Logical Volumes

All shared disks must contain XLV logical volumes. You can work with XLV logical volumes on shared disks as you would work with other disks. However, you must follow these rules:

- All data that is used by HA applications on shared disks must be stored in XLV logical volumes.

- If you create more than one XLV volume on a single physical disk, all of those volumes must 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 "Create XLV Logical Volumes and XFS Filesystems", page 67, for more information about XLV volume ownership.)
- Each disk in a Fibre Channel or SCSI Vault or RAID logical unit number (LUN) must be part of one resource group. Therefore, you must divide the Fibre Channel or SCSI Vault disks and RAID LUNs into one set for each resource group. If you create multiple volumes on a Fibre Channel or SCSI Vault disk or RAID LUN, all those volumes must be part of one resource group.
- Do not simultaneously access a shared XLV volume from more than one node. Doing so causes data corruption.

The FailSafe software relies on the XLV naming scheme to operate correctly. A fully qualified XLV volume name uses one of the following formats:

pathname`/`*volname*
pathname`/`*nodename*`.`*volname*

The components are these:

- *pathname* is `/dev/xlv`
- *nodename* by default is the same as the hostname of the node on which the volume was created
- *volname* is 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 the peer `ha2`, however, the same raw character volume appears as `/dev/rxlv/ha1.vol1`, 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.

Caution: If this rule is not followed, FailSafe does not operate correctly.

FailSafe 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 (that is, with no filesystem) for storing database data, the database system may require that the device names in `/dev/xlv` 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 defaults are `root`, `sys`, and `0600`, respectively).

Example Logical Volume Configuration

As an example of logical volume configuration, say that you have the following logical volumes on disks that we will call `Disk1` through `Disk5`:

- `/dev/xlv/VolA` (volume A) contains `Disk1` and a portion of `Disk2`
- `/dev/xlv/VolB` (volume B) contains the remainder of `Disk2` and `Disk3`
- `/dev/xlv/VolC` (volume C) that contains `Disk4` and `Disk5`

`VolA` and `VolB` must be part of the same resource group because they share a disk. `VolC` could be part of any resource group. Figure 2-5 describes this.

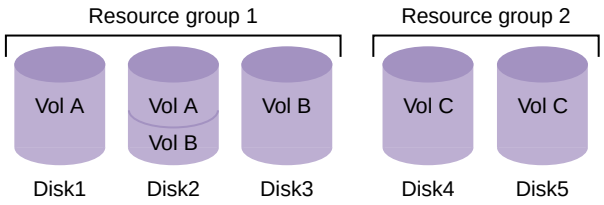

Figure 2-5 Example Logical Volume Configuration

Configuration Parameters for Logical Volumes

Configuration parameters for XLV logical volumes list the following:

- Owner of the device file (default value: `root`)
- Group of the device (default value: `sys`)
- Mode of the device (default value: `0600`)

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	Owner of the device name
devname-group	sys	sys	root	Group of the device name
devname-mode	0600	0600	0600	Mode of the device name

See the section "Create XLV Logical Volumes and XFS Filesystems", page 67, for information about creating XLV logical volumes.

Filesystem Configuration

This section describes filesystem configuration for FailSafe using XFS filesystems. For information on coexecution of FailSafe and CXFS filesystems, see "Coexecution of CXFS and IRIS FailSafe", page 48.

Planning Filesystems

FailSafe supports the failover of XFS filesystems on shared disks. Shared disks must be either Fibre Channel or SCSI JBOD or in RAID storage systems that are shared between nodes in the FailSafe cluster. Fibre Channel and SCSI JBOD storage systems must use XLV mirroring.

The following are special issues that you must be aware of when you are working with filesystems on shared disks in a 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 a cluster must be created on either mirrored disks (using the XLV plexing software) or on the Fibre Channel RAID storage system.
- Create the mount points for the filesystems on all nodes in the failover domain.
- When you set up the various IRIS FailSafe filesystems on each node, make sure that each filesystem uses a different mount point.
- Do not simultaneously mount filesystems on shared disks on more than one node. Doing so causes data corruption. Normally, FailSafe performs all mounts of filesystems on shared disks. If you manually mount a filesystem on a shared disk, make sure that it is not being used by another node.
- Do not place filesystems on shared disks in the `/etc/fstab` file. FailSafe mounts these filesystems only after making sure that another node does not have these filesystems mounted.

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

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

Caution: Do not cross-mount filesystems using NFS in a FailSafe cluster (that is, do not mount a locally mounted filesystem on a different node using NFS). This configuration is not reliable and will not work with FailSafe. Instead, you should use the CXFS (clustered XFS) plug-in, which provides this functionality. For more information, see *IRIS FailSafe Version 2 NFS Administrator's Guide*.

Use of NFS over TCP is not recommended: if the client loses the TCP connection and does not reconnect, it can cause the client to hang on a failover. You should use UDP rather than TCP. Note that TCP may be the default for your NFS clients, requiring you to reconfigure them to use UDP. One method to accomplish this is to create the `/etc/config/nfsd.options` file with the content `-p UDP`, which will prevent the server from accepting TCP mount requests.

Example Filesystem Configuration

Continuing with the example configuration from the section "Example Logical Volume Configuration", page 41, suppose you have the following XFS filesystems:

- `xfsA` on `VolA` is mounted at `/sharedA` with modes `rw` and `noauto`
- `xfsB` on `VolB` is mounted at `/sharedB` with modes `rw`, `noauto`, and `wsync`
- `xfsC` on `VolC` is mounted at `/sharedC` with modes `rw` and `noauto`

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

Table 2-2 Filesystem Configuration Parameters

Attribute	/sharedA	/sharedB	/sharedC	Comments
monitor-level	2	2	2	There are two levels of monitoring: 1 – checks /etc/mtab file 2 – checks if the filesystem is mounted using the stat(1M) command
volume-name	VolA	VolB	VolC	The label of the logical volume on which the filesystem was created
mode	rw, noauto	rw, noauto, wsync	rw, noauto	The modes of the filesystem (identical to the modes specified in /etc/fstab)

Figure 2-6 shows the following:

- Resource group 1 has two XFS filesystems (`xfsA` and `xfsB`) and two XLV volumes (`VolA` and `VolB`)
- Resource group 2 has one XFS filesystem (`xfsC`) and one XLV volume (`VolC`)

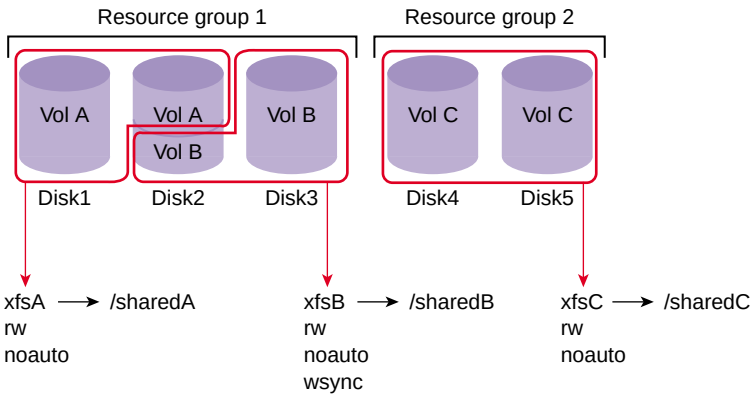

Figure 2-6 Filesystems and Logical Volumes

See "Create XLV Logical Volumes and XFS Filesystems", page 67, for information about creating XFS filesystems.

HA IP Address Configuration

This section contains the following:

- "Planning Network Interface and HA IP Address Configuration"
- "Example HA IP Address Configuration", page 48
- "Local Failover of HA IP Addresses", page 48

Planning Network Interface and HA IP Address Configuration

Use the following guidelines when planning interface configuration for the private control network between nodes:

- Each interface has one IP address.
- The HA 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 HA IP address in `/etc/hosts`.

- A naming convention that identifies these HA IP addresses 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`.

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

Use the following procedure to determine whether re-MACing is required (see the section "Network Interfaces and IP Addresses" for information about re-MACing). It requires the use of three nodes: `node1`, `node2`, and `node3`. `node1` and `node2` can be nodes of a FailSafe cluster, but they need not be. They must be on the same subnet. `node3` is a third node. If you must 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 HA 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`, contact the HA IP address used in step 1 using the `ping`(1M) command :

   ```
   # 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 the following command on `node1` to shut down the interface you configured in step 1:

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

4. On `node2`, enter the following command to move the HA IP address to `node2`:

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

5. On `node3`, contact the HA IP address:

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

   If the `ping`(1) command fails, gratuitous ARP packets are not being accepted and re-MACing is needed to fail over the HA IP address.

Example HA IP Address Configuration

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

Table 2-3 HA IP Address Configuration Parameters

Resource Attribute	Resource Name: `192.26.50.1`	Resource Name: `192.26.50.2`
Network mask	`0xffffff00`	`0xffffff00`
Broadcast address	`192.26.50.255`	`192.26.50.255`
Interface	`ef0`	`ef0`

Local Failover of HA IP Addresses

You can configure your system so that an HA IP address will fail over to a second interface within the same node, for example from `ef0` to `ef1`. A configuration example that shows the steps you must follow for this configuration is provided in "Example: Local Failover of HA IP Address", page 220.

Coexecution of CXFS and IRIS FailSafe

CXFS, the clustered XFS filesystem, allows groups of computers to coherently share large amounts of data while maintaining high performance. Users can use FailSafe in a CXFS cluster to provide highly available services (such as NFS or Web) running on a CXFS filesystem. This combination provides high-performance shared data access for highly available applications.

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

A subset of nodes in a coexecution cluster can be configured to be used as FailSafe nodes; a coexecution cluster can have up to eight nodes that run FailSafe.

This section contains the following:

- "Size of the Coexecution Cluster", page 49

- "Cluster Type", page 49
- "Node Types for CXFS Metadata Servers", page 50
- "CXFS Metadata Servers and Failover Domain", page 50
- "`CXFS` Resource Type for FailSafe", page 50
- "Separate CXFS and FailSafe GUIs", page 52
- "Conversion Between CXFS and FailSafe", page 52
- "Network Interfaces", page 53

See also "Communication Paths in a Coexecution Cluster", page 316.

Size of the Coexecution Cluster

All nodes in a CXFS cluster will run CXFS, and up to eight of those nodes can also run FailSafe. Even when you are running CXFS and FailSafe, there is still only one pool, one cluster, and one cluster configuration.

It is recommended that a production cluster can be configured with a minimum of three weighted nodes (CXFS weight) and a maximum of 16 nodes. (A cluster with reset cables and only two weighted nodes is supported, but there are inherent issues with this configuration; see the *CXFS Version 2 Software Installation and Administration Guide*.)

Cluster Type

The cluster can be one of three types:

- `FailSafe`. In this case, all nodes will also be of type `FailSafe`.
- `CXFS`. In this case, all nodes will be of type `CXFS`.
- `CXFS and FailSafe` (coexecution). In this case, all nodes will be a mix of type `CXFS` and type `CXFS and FailSafe`, using FailSafe for application-level high availability and CXFS.

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

Node Types for CXFS Metadata Servers

All potential metadata server nodes must be of one of the following types:

- `CXFS`
- `CXFS and FailSafe`

CXFS Metadata Servers and Failover Domain

The metadata server list must exactly match the failover domain list (the names and the order of names).

CXFS Resource Type for FailSafe

FailSafe provides the `CXFS` resource type, which can be used to fail over applications that use CXFS filesystems. CXFS resources must be added to the resource group that contain the resources that depend on a CXFS filesystem. The name of the `CXFS` resource is the CXFS filesystem mount point.

The `CXFS` resource type has the following characteristics:

- It does not start all resources that depend on the CXFS filesystem until the CXFS filesystem is mounted on the local node.
- The `start` and `stop` action scripts for the `CXFS` resource type do not mount and unmount CXFS filesystems, respectively. (The `start` script waits for the CXFS filesystem to become available; the `stop` script does nothing but its existence is required by FailSafe.) Users should use the CXFS GUI or `cmgr`(1M) command to mount and unmount CXFS filesystems.
- It monitors CXFS filesystem for failures.
- Optionally, for applications that must run on a CXFS metadata server, the `CXFS` resource type relocates the CXFS metadata server when there is an application failover. In this case, the application failover domain (AFD) for the resource group should consist of the CXFS metadata server and the metadata server backup nodes.

The CXFS filesystems that an NFS server exports should be mounted on all nodes in the failover domain using the CXFS GUI or the `cmgr`(1M) command.

For example, following are the commands used to create resources `NFS`, `CXFS` and `statd_unlimited` based on a CXFS filesystem mounted on `/FC/lun0_s6`. (This

example assumes that you have defined a cluster named `test-cluster` and that you have already created a failover policy named `cxfs-fp` and a resource group named `cxfs-group` based on this policy.)

```
cmgr> define resource /FC/lun0_s6 of resource_type CXFS in cluster test-cluster
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to create with set command:

Type Specific Attributes - 1: relocate-mds


No resource type dependencies to add

resource /FC/lun0_s6 ? set relocate-mds to false
resource /FC/lun0_s6 ? done


==========================================

cmgr> define resource /FC/lun0_s6 of resource_type NFS in cluster test-cluster
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to create with set command:

Type Specific Attributes - 1: export-info
Type Specific Attributes - 2: filesystem


No resource type dependencies to add

resource /FC/lun0_s6 ? set export-info to rw
resource /FC/lun0_s6 ? set filesystem to /FC/lun0_s6
resource /FC/lun0_s6 ? done


==========================================
cmgr> define resource /FC/lun0_s6/statmon of resource_type statd_unlimited in cluster test-cluster
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to create with set command:
```

```
Type Specific Attributes - 1: ExportPoint


Resource type dependencies to add:

Resource Dependency Type - 1: NFS


resource /FC/lun0_s6/statmon ? set ExportPoint to /FC/lun0_s6
resource /FC/lun0_s6/statmon ? add dependency /FC/lun0_s6 of type NFS
resource /FC/lun0_s6/statmon ? done

==========================================
cmgr> define resource_group cxfs-group in cluster test-cluster
Enter commands, when finished enter either "done" or "cancel"

resource_group cxfs-group ? set failover_policy to cxfs-fp
resource_group cxfs-group ? add resource /FC/lun0_s6 of resource_type NFS
resource_group cxfs-group ? add resource /FC/lun0_s6 of resource_type CXFS
resource_group cxfs-group ? add resource /FC/lun0_s6/statmon of resource_type statd_unlimited
resource_group cxfs-group ? done
```

Separate CXFS and FailSafe GUIs

There is one `cmgr`(1M) command but separate graphical user interfaces (GUIs) for CXFS and for FailSafe. You must manage CXFS configuration with the CXFS GUI and FailSafe configuration with the FailSafe GUI; you can manage both with `cmgr`.

Conversion Between CXFS and FailSafe

Using the CXFS GUI or `cmgr`(1M), you can convert an existing `FailSafe` cluster and nodes to type `CXFS` or to type `CXFS and FailSafe`. You can perform a parallel action using the FailSafe GUI. A converted node can be used by FailSafe to provide application-level high-availability and by CXFS to provide clustered filesystems.

However:

- You cannot change the type of a node if the respective high availability (HA) or CXFS services are active. You must first stop the services for the node.

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

Network Interfaces

For FailSafe, you must have at least two network interfaces. However, CXFS uses only one interface for **both** heartbeat and control messages.

When using FailSafe and CXFS on the same node, only the priority 1 network will be used for CXFS and it must be set to allow both heartbeat and control messages.

Note: CXFS will not fail over to the second network. If the priority 1 network fails, CXFS will fail but FailSafe services may move to the second network if the node is `CXFS and FailSafe`.

If CXFS resets the node due to the loss of the priority 1 network, it will cause FailSafe to remove the node from the FailSafe membership; this in turn will cause resource groups to fail over to other FailSafe nodes in the cluster.

Installation and System Preparation

Note: The procedures in this chapter assume that you have done the planning described in Chapter 2, "Configuration Planning", page 29.

The following steps are required for IRIS FailSafe installation and system preparation:

- "Install Software"
- "Configure System Files", page 59
- "Set the `corepluspid` System Parameter", page 67
- "Set NVRAM Variables", page 67
- "Create XLV Logical Volumes and XFS Filesystems", page 67
- "Configure Network Interfaces", page 69
- "Configure the Serial Ports for a Ring Reset", page 73
- "Install Patches", page 74
- "Install Performance Co-Pilot (PCP) Software", page 79
- "Test the System", page 83

Install Software

Installing the IRIS FailSafe base CD requires about 10 MB of free space.

To install the required software, do the following:

1. On each node in the pool, upgrade to a supported release of IRIX according to the *IRIX 6.5 Installation Instructions* and the FailSafe product release notes:

   # **relnotes failsafe2 [***chapter_number***]**

   To verify that a given node has been upgraded, use the following command to display the currently installed system:

   # **uname -aR**

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 node, install the version of the serial port server driver that is appropriate to the operating system. Use the CD that accompanies the serial port server. Reboot the system after installation.

   For more information, see the following documentation provided with the serial port server:

   - *EL Serial Port Server Installation Guide* (provided by Digi Corporation)
   - *EL Serial Port Server Installation Guide Errata*

4. On each node, install the following software, in this order:

   a. `sysadm_base.sw.dso`

   b. `sysadm_base.sw.server`

   c. `sysadm_cluster.sw.server`

   d. `cluster_admin.sw.base`

   e. `cluster_control.sw.base`

   f. `cluster_services.sw.base`

   g. `cluster_services.sw.cli`

   h. `cluster_control.sw.cli`

   i. `failsafe2.sw.cli`

j. `sysadm_failsafe2.sw.server`

k. `cluster_control.sw`

When `sysadm_base` is installed, `tcpmux` service added to the `/etc/inetd.conf` file.

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

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

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

a. `java_eoe.sw`, version 3.1.1

b. `sysadm_base.sw.client`

c. `sysadm_cluster.sw.client`

d. `sysadm_failsafe2.sw.client`

e. `sysadm_failsafe2.sw.web`

Caution: The GUI only operates with Java 1.1.8. This is the version of Java that is provided with the IRIX 6.5.*x* release.

The SGI Web site also contains Java 2. However, you **cannot** use this version of Java with the GUI. Using a Java version other than 1.1.8 will cause the GUI to fail.

---

5. On each node, install the following **additional** software, in the order given.

   a. `cluster_services.sw`.

   b. `failsafe2.sw`.

c. If necessary: `nfs.ws.nfs` (From IRIX; might already be present.) You should have purchased the optional FailSafe/NFS software to make NFS server highly available.

d. `failsafe2_nfs.sw`.

e. If necessary: `ns_admin.sw.server` (from Netscape; might already be present).

f. If necessary: `ns_fasttrack.sw.server` OR `ns_enterprise.sw.server` (from Netscape). You should have purchased the optional FailSafe/Web software to make netscape servers highly available.

g. `failsafe2_web.sw`.

6. If you want to run the administrative workstation (GUI client) from an IRIX desktop, install the following subsystems on the desktop:

   - `sysadm_failsafe2.sw.desktop`.
   - `sysadm_failsafe2.sw.client`.
   - `sysadm_base.sw.client`.
   - `sysadm_cluster.sw.client`.
   - `java_eoe.sw`, version 3.1.1.
   - If the administrative workstation is an IRIX machine that launches the GUI client from a Web browser that supports Java, install the `java_plugin` from the CXFS CD. (However, launching the GUI from a Web browser is not the recommended method on IRIX. Running the GUI client from an IRIX desktop is preferred.)

     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`). Do not install these three subsystems because the GUI does not use them.

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

7. On the appropriate nodes, install other optional software, such as storage management or network board software.

8. If the cluster is using plexed XLV logical volumes, do the following:

a. Install a disk plexing license on each node in the `/var/flex1m/license.dat` file. For more information on XLV logical volumes and on XFS plexing and filesystems, see Chapter 2, "Configuration Planning".

b. Verify that the license has been successfully installed on each node in the cluster by using the `xlv_mgr`(1M) command:

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

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

```
Plexing license: present
```

c. Quit `xlv_mgr`.

9. Install recommended patches for FailSafe.

For instructions on installing a FailSafe patch, see "Install Patches", page 74.

Set the `AutoLoad` variable to `Yes`; this can be done when you set host SCSI IDs, as explained in "Set NVRAM Variables", page 67.

Note: For reference, Appendix C, "IRIS FailSafe 2.1.*x* Software", page 337, summarizes systems to install on each component of a cluster or node.

Configure System Files

This section discusses the following:

- "Hostname Resolution: `/etc/sys_id`, `/etc/hosts`, `/etc/nsswitch.conf`"
- "`/etc/services`", page 62
- "`/etc/config/cad.options`", page 62
- "`/etc/config/fs2d.options`", page 63
- "`/etc/config/cmond.options`", page 66

Hostname Resolution: /etc/sys_id, /etc/hosts, /etc/nsswitch.conf

Caution: It is critical that you understand these rules before attempting to configure a FailSafe cluster.

The following hostname resolution rules and recommendations apply to FailSafe clusters:

- Hostnames cannot begin with an underscore (_) or include any whitespace characters.
- The value of the /etc/sys_id file must match the node's primary hostname in the /etc/hosts file (that is, the first field after the node's IP address in /etc/hosts) for all nodes in the cluster. This field can be either the hostname or the fully qualified domain name.

  The /etc/hosts file has the following format, where *primary_hostname* can be the simple hostname or the fully qualified domain name:

  IP_address primary_hostname aliases

  For example, suppose your /etc/hosts file contains the following:

  ```
  # The public interface:
  128.2.3.4  color-green.sgi.com color-green green

  # The private interface:
  192.0.1.1  color-green-private.sgi.com  color-green-private green-private
  ```

  The /etc/sys_id file could contain either the hostname color-green or the fully qualified domain name color-green.sgi.com. It cannot contain the alias green.

  In this case, you would enter the hostname color-green or the fully qualified domain name color-green.sgi.com for the **Server** field in the login screen and for the **Hostname** field in the **Define a new node** window.

- If you use the nsd(1M) name service daemon, you must configure your system so that local files are accessed before either the network information service (NIS) or the domain name service (DNS). That is, the hosts line in /etc/nsswitch.conf must list files first. For example:

  ```
  hosts:      files nis dns
  ```

(The order of `nis` and `dns` is not significant to FailSafe; `files` must be first.)

The `/etc/config/netif.options` file must have one of the interfaces be equal to the value of `/etc/sys_id` (`$HOSTNAME`).

For more information about the Unified Name Service (UNS) and the name service daemon, see the `nsd`(1M) man page.

- If you change the `/etc/nsswitch.conf` or `/etc/hosts` files, you must restart `nsd` by using the `nsadmin restart` command, which also flushes its cache.

  The reason you must restart `nsd`(1M) after making a change to these files is that the `nsd` name service daemon actually takes the contents of `/etc/hosts` and places the contents in its memory cache in a format that is faster to search. Thus, you must restart `nsd` in order for it to see that change and place the new `/etc/hosts` information into RAM cache. If `/etc/nsswitch.conf` is changed, `nsd` must re-read this file so that it knows what type of files (for example, `hosts` or `passwd`) to manage, what services it should call to get information, and in what order those services should be called.

  The IP addresses on a running node in the cluster and the IP address of the first node in the cluster cannot be changed while cluster services are active.

- You should be consistent when using fully qualified domain names in the `/etc/hosts` file. If you use fully qualified domain names in `/etc/sys_id` on a particular node, then all of the nodes in the cluster should use the fully qualified name of that node when defining the IP/hostname information for that host in their `/etc/hosts` file.

  The decision to use fully qualified domain names is usually a matter of how the clients (such as NFS) are going to resolve names for their client server programs, how their default resolution is done, and so on.

- If you change hostname resolution settings in the `/etc/nsswitch.conf` file after you have defined the first node (which creates the cluster database), you must recreate the database.

- When using coexecution with CXFS, never add an `/etc/hosts` entry that associates the value of `/etc/sys_id` with an IP address alias. You must use the primary address.

/etc/services

Edit the /etc/services file so that it contains entries for sgi-cad and sgi-crsd before you install 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: sgi-cad requires a TCP port for communication between FailSafe nodes.

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
```

Edit the /etc/services file so that it contains entries for sgi-cmsd and sgi-gcd on each node before starting highly available (HA) services on the node. The port numbers assigned for these processes must be the same in all nodes in the cluster.

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

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

/etc/config/cad.options

The /etc/config/cad.options file contains the list of parameters that the cad(1M) cluster administration daemon reads when the process is started. cad provides cluster information to the 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 characters 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
```

The contents of the `/etc/config/cad.options` file cannot be modified using the `cmgr`(1M) command or the GUI.

Note: If you make a change to the `cad.options` file at any time other than initial configuration, you must restart the `cad` processes in order for these changes to take effect. You can do this by rebooting the nodes or by entering the following command:

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

If you execute this command on a running cluster, it will remain up and running. However, the GUI will lose connection with the `cad`(1M) daemon; the GUI will prompt you to reconnect.

/etc/config/fs2d.options

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

Table 3-1 shows the options can that can be set in the `fs2d.options` file.

Table 3-1 `fs2d.options` File Options

Option	Description
`-logevents` *event name*	Log selected events. The following event names may be used: `all`, `internal`, `args`, `attach`, `chandle`, `node`, `tree`, `lock`, `datacon`, `trap`, `notify`, `access`, `storage`. The default is `all`.
`-logdest` *log destination*	Set log destination. The following 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 `logfile`.
`-logfile` *filename*	Set log filename. The default 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 is 10000000.
`-loglevel` *loglevel*	Set log level. The following log levels may be used: `always`, `critical`, `error`, `warning`, `info`, `moreinfo`, `freq`, `morefreq`, `trace`, `busy`. The default is `info`.
`-trace` *trace_class*	Trace selected events. The following trace classes may be used: `all`, `rpcs`, `updates`, `transactions`, `monitor`. If you specify this option, you must also specify `-tracefile` and/or `-tracelog`. 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 is `transactions`.
`-tracefile` *filename*	Set trace filename. There is no default.
`-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`, and a new file will be created.
`-[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. The default is `-tracelog.`
`-[no]parent_timer`	[Do not] exit when parent exits. The default is `-noparent_timer`.

Option	Description
`-[no]daemonize`	[Do not] run as a daemon. The default is `-daemonize`.
`-l`	Do not run as a daemon.
`-h`	Print usage message.
`-o help`	Print usage message.

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, are sent to the `/var/cluster/ha/log/fs2d_log` file. When the file size reaches 10 MB, 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.

Note: If you make a change to the `fs2d.options` file at any time other than the initial configuration time, you must restart the `fs2d` processes in order for those changes to take effect. You can do this by rebooting the nodes or by entering the following command:

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

If you execute this command on a running cluster, it should remain up and running. However, the GUI will lose connection with the `cad`(1M) daemon; the GUI will prompt you to reconnect.

Example 1

The following example shows an `/etc/config/fs2d.options` file that directs logging and tracing information as follows:

- All log events are sent to `/var/adm/SYSLOG`.
- Tracing information for RPCs, updates, and transactions are sent to `/var/cluster/ha/log/fs2d_ops1`.

  When the size this file exceeds 100,000,000 bytes, 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.

(Line breaks added here only for readability.)

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

Example 2

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 100,000,000 bytes is specified. `-tracelog` directs the tracing to the log file.

(Line breaks added here only for readability.)

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

/etc/config/cmond.options

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

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

`-L` *log_level*	Set log level to *log_level*. The legal values for *log_level* are `normal`, `critical`, `error`, `warning`, `info`, `frequent`, and `all`.
`-d`	Run in debug mode
`-l`	Lazy mode, where `cmond` does not validate its connection to the cluster database
`-t` *nap_interval*	The time interval in milliseconds after which `cmond` checks for liveliness of process groups it is monitoring
`-s`	Log messages to standard error.

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
```

Set the `corepluspid` System Parameter

Use the `systune`(1M) command to set the `corepluspid` flag to `1` on every node. If this flag is set, IRIX will suffix all core files with a process ID (PID). This prevents a core dump from being overwritten by another process core dump.

Set NVRAM Variables

During the hardware installation of FailSafe nodes, two non-volatile random-access memory (NVRAM) variables must be set:

- The boot parameter `AutoLoad` must be set to `yes`. FailSafe requires the nodes to be automatically booted when they are reset or when the node is powered on.
- The SCSI IDs of the nodes, 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 the following commands:

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

To set these variables, use the following 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 ensure 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.

Create XLV Logical Volumes and XFS Filesystems

You can create XLV logical volumes by following the instructions in the guide *IRIX Admin: Disks and Filesystems*.

Note: This section describes logical volume configuration using XLV logical volumes. For information on coexecution of FailSafe and CXFS filesystems (which use XVM logical volumes), see "Coexecution of CXFS and IRIS FailSafe", page 48. For information on creating CXFS filesystems, see the *CXFS Version 2 Software Installation and Administration Guide*. For information on creating XVM logical volumes, see the *XVM Volume Manager Administrator's Guide*.

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

- If the shared disks are not in a RAID storage system, you should create plexed XLV logical volumes.
- Each XLV logical volume must be owned by the same node that is the primary node for the resources that use the logical volume (see "Planning XLV Logical Volumes", page 39). To simplify the management of the owners of volumes on shared disks, use the following 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 (that is, with no filesystem) for storing database data, the database system may require that the device names (in `/dev/rxlv` and `/dev/xlv`) have specific owners, groups, and modes. If this is the case (see the documentation provided by the database vendor), use the `chown`(1) and `chmod`(1) commands to set the owner, group, and mode as required.
- No filesystem entries are made in `/etc/fstab` for XFS filesystems on shared disks; FailSafe software mounts the filesystems on shared disks. However, to simplify system administration, consider adding comments to `/etc/fstab` that list the XFS filesystems configured for FailSafe. Thus, a system administrator who sees mounted FailSafe filesystems in the output of the `df` command and looks for the filesystems in the `/etc/fstab` file will learn that they are filesystems managed by FailSafe.
- Be sure to create the mount point directory for each filesystem on all nodes.

Configure Network Interfaces

This section describes how to configure the network interfaces. The example shown in Figure 3-1 is used in the procedure.

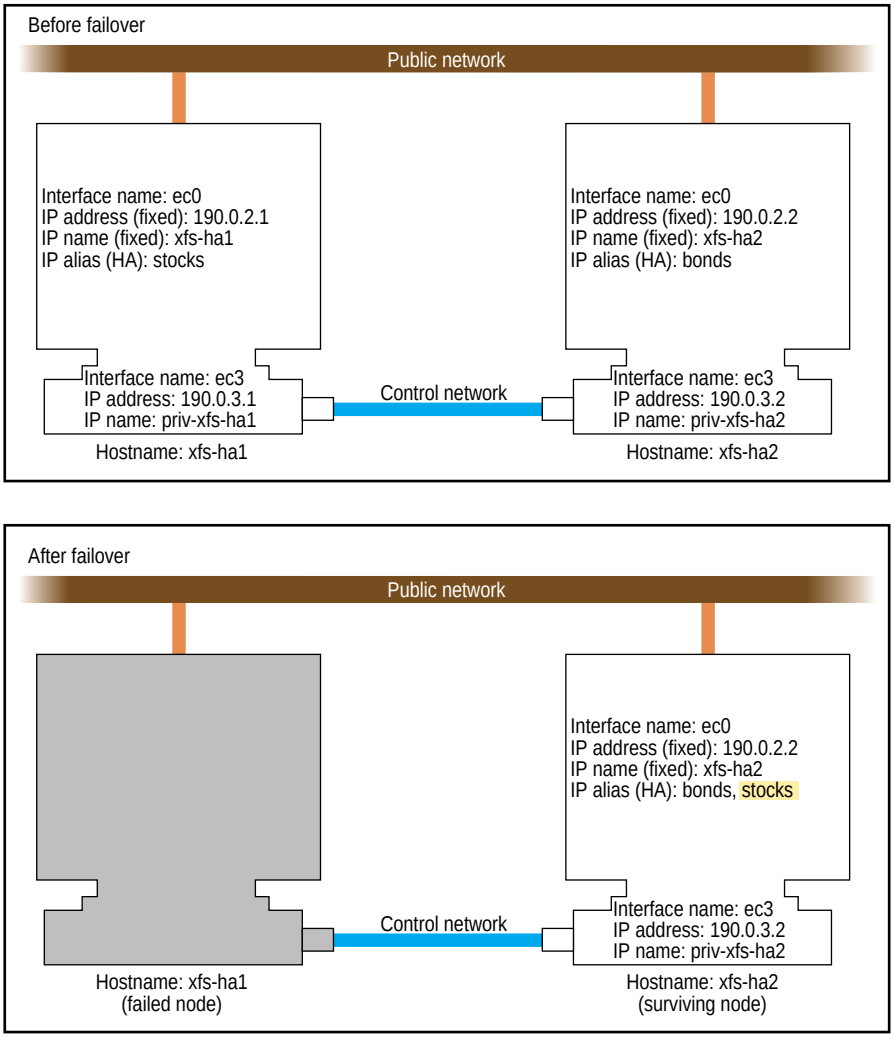

Figure 3-1 Example Interface Configuration

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 HA services are not added to the file `/etc/config/ipaliases.options`. Similarly, if all IP aliases are used only by HA 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 by entering the following 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* does not 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` . However, highly available (HA) 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 as follows:

   ```
   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 such as the following:

   ```
   if2name=
   if2addr=
   ```

   In the example, the control network name and IP address are as follows:

   ```
   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 fewer 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 `/etc/config/routed.options` file on each node so that the routes are not advertised over the control network. See the `routed`(1M) man page for a list of options.

   For example:

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

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

The options do the following:

- Turn off advertising of routes
- Cause host or point-to-point routes to not be advertised (provided there is a network route going the same direction)
- Set the normal interval with which router discovery advertisements are transmitted to 45 seconds (and their lifetime to 135 seconds)

9. Verify that IRIS FailSafe 2.*x* is turned off on each node, using the `chkconfig`(1M) command:

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

If `failsafe2` is set to `on` on a node, enter this command on that node:

```
# chkconfig failsafe2 off
```

If Failsafe 1.*x* is present, you must also ensure that it is not configured `on` for any node:

```
# 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 FailSafe e-mail notifications of cluster transitions to a user outside of the 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`, add the following to `/usr/lib/aliases` on `xfs-ha1`:

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

On `xfs-ha2`, add the following 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 — that is, `yp` has been set to `on` using `chkconfig`(1M) — or the BIND domain name server (DNS), switching to local name resolution is recommended. Modify the `/etc/nsswitch.conf` file so that it reads as follows:

    ```
    hosts:                  files nis dns
    ```

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

12. If you are using FDDI, finish configuring and verifying the new FDDI station, as explained in the FDDIXpress release notes and the *FDDIXPress Administration Guide*.

13. Reboot all nodes to put the new network configuration into effect.

Configure the Serial Ports for a Ring Reset

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. Perform the following 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/SGIconsole system, do not configure the reset port into the IRISconsole/SGIconsole, because it will conflict with the reset daemon that the FailSafe system is running. (The reset port is the tty port that is connected to the system controller port of another node in the cluster either directly or using serial multiplexor.)

FailSafe does not impact console connections obtained using the IRISconsole/SGIconsole product.

Install Patches

The procedures in this section describe how to install a FailSafe patch. The patch should be installed on all nodes.

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

Installing FailSafe 2.*x* and a FailSafe Patch at the Same Time

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

Do the following on each node:

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

   # **`/etc/init.d/cluster stop`**
4. Verify that the cluster processes (`cad`, `cmond`, `crsd`, and `fs2d`) have stopped:

   # **`ps -ef | egrep '(cad|cmond|crsd|fs2d)'`**
5. Start cluster processes on the node:

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

You are now ready to run the FailSafe Manager GUI or the `cmgr`(1M) command to set up a FailSafe cluster.

Installing a FailSafe Patch on an Existing FailSafe 2.*x* Cluster

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

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

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

1. If you have the FailSafe GUI client software installed on a machine that is not a node, first install the patch client subsystems on that machine. The GUI client software subsystems are as follows, where *xxxxxxx* is the patch number:

   - `patchSG`*xxxxxxx*`.sysadm_base_sw.client`
   - `patchSG`*xxxxxxx*`.sysadm_failsafe2_sw.client`
   - `patchSG`*xxxxxxx*`.sysadm_failsafe2_sw.desktop`

2. Choose a node on which to install the patch. Start up the FailSafe GUI or `cmgr`(1M) command on that node.

   For convenience, connect the GUI to a node that you are **not** upgrading.

   Note: If you connect to the node that you are upgrading, then in a later step (when you stop FailSafe HA services), FailSafe will no longer report accurate status to the GUI; in another later step (when you stop cluster services), the GUI will lose its connection.

   Use the following `cmgr` command to specify a default node (later commands in this procedure assume the cluster name has already been set):

   ```
   cmgr> set cluster clustername
   ```

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

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

   If you are using `cmgr`, execute the following command:

   ```
   cmgr> admin offline_detach resource_group groupname
   ```

4. Stop HA services on the node. (When FailSafe HA services stop, FailSafe will no longer be able to report current cluster and node state if the FailSafe Manager is

connected to that node. To monitor the cluster state during installation, connect the FailSafe Manager to the node that you are not upgrading.)

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

If you are using `cmgr`, execute the following command:

```
cmgr> stop ha_services on node nodename
```

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

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

If you are using `cmgr`, execute the following command:

```
cmgr> admin move resource_group groupname to node nodename
```

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

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

   When you are using the GUI, if the **connection lost** dialogue appears, click **No**. If you wish to continue using the GUI, restart the GUI, connecting to a node you are not patching.

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

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

7. Use `chkconfig`(1M) to turn off the `cluster` flag:

   ```
   # chkconfig cluster off
   ```

   Note: You cannot use the `failsafe2` flag to turn off the HA services on a node. You must use the GUI or `cmgr` commands to stop HA services; these commands can be run from any node in the pool. If necessary, you can use the force option. For more information, see "Stop FailSafe HA Services", page 200.

   ---

8. Install the patch on the node.

9. Use `chkconfig` to turn on the `cluster` flag:

   # **chkconfig cluster on**

10. Start cluster processes on the node:

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

11. Start HA services on the node.

    If you are using the GUI and you are running the GUI in a Web browser, do the following:

    a. Exit your browser

    b. Restart the Web server on the node you have just patched

    c. Restart the GUI, connecting to the patched node

    d. Run the **Start FailSafe HA Services** task, specifying the node that you just patched in the **One Node Only** field.

       If the GUI claims that FailSafe HA services are active on the cluster, then you are using an unpatched client; in this case, run the `cmgr` command instead, run the GUI on a patched client, or run the GUI in a Web browser from the patched node.

    If you are using `cmgr`, execute the following command:

    cmgr> **start ha_services on node** *nodename*

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

    If you are using the GUI, select **View: Groups Owned by Nodes** in the tree view. Confirm that the resource group icons indicates online status.

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

    If you are using `cmgr`, execute the following command:

    cmgr> **show status of resource_group** *groupname*

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

Install Performance Co-Pilot (PCP) Software

You can deploy Performance Co-Pilot (PCP) for FailSafe as a collector agent or as a monitor client:

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

Installing the Collector Host

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

- The `pcp_eoe.sw` subsystem from IRIX 6.5.11 or later
- IRIS FailSafe 2.1 or later
- PCP 2.1 or later

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

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

Table 3-2 PCP for FailSafe Collector Subsystems

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

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

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

   ```
   # inst
   ```

4. Specify the installation location:

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

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

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

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

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

   ```
   Inst> install default
   ```

6. Ensure that there are no conflicts:

   ```
   Inst> conflicts
   ```

7. Install the software:

   ```
   Inst> go
   ```

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

   # **cd /var/pcp/pmdas/fsafe**

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

   # **./Install**

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

    - `collector`, which collects performance statistics on this system
    - `monitor`, which allows this system to monitor local and/or remote systems
    - `both`, which allows collector and monitor configuration for this system

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

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

Removing Performance Metrics from a Collector Host

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

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

   # **cd /var/pcp/pmdas/fsafe**

2. Run the `Remove` utility:

   # **./Remove**

Installing the Monitor Host

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

- The `pcp_eoe.sw` subsystem of IRIX 6.5.11 or later, including the subsystem `pcp_eoe.sw.monitor`
- PCP 2.1 or later, including the subsystem `pcp.sw.monitor`

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

After this software is installed, install the subsystems of PCP for FailSafe listed in Table 3-3 on each collector host.

Table 3-3 PCP for FailSafe Monitor Subsystems

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

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

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

   ```
   # inst
   ```

4. Specify the installation location:
   - If you are installing from the local CD-ROM drive, enter the following:

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

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

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

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

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

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

   ```
   Inst> conflicts
   ```

7. Install the software:

   ```
   Inst> go
   ```

Test the System

This section discusses the following ways of testing the system:

- "Private Network Interface"
- "Serial Reset Connection", page 84

Private Network Interface

For each private network on each node in the pool, enter the following, where *nodeIPaddress* is the IP address of the node:

```
# /usr/etc/ping -c 3 nodeIPaddress
```

Typical `ping`(1M) output should appear, such as the following:

```
PING IPaddress (190.x.x.x: 56 data bytes
64 bytes from 190.x.x.x: icmp_seq=0 tt1=254 time=3 ms
64 bytes from 190.x.x.x: icmp_seq=1 tt1=254 time=2 ms
64 bytes from 190.x.x.x: icmp_seq=2 tt1=254 time=2 ms
```

If `ping` fails, follow these steps:

1. Verify that the network interface was configured up using `ifconfig`; for example:

```
# /usr/etc/ifconfig ec3
ec3: flags=c63<UP,BROADCAST,NOTRAILERS,RUNNING,FILTMULTI,MULTICAST>
inet 190.x.x.x netmask 0xffffff00 broadcast 190.x.x.x
```

   The `UP` in the first line of output indicates that the interface was configured up.

2. Verify that the cables are correctly seated.

Repeat this procedure on each node.

Serial Reset Connection

To test the serial reset connections, do the following:

1. Ensure that the nodes and the serial multiplexer are powered on.
2. Start the `cmgr`(1M) command on one of the nodes in the pool:

   ```
   # cmgr
   ```

3. Stop HA services on each node:

   ```
   stop ha_services for cluster clustername
   ```

   For example:

   ```
   cmgr> stop ha_services for cluster fs6-8
   ```

Wait until the node has successfully transitioned to inactive state and the FailSafe processes have exited. This process can take a few minutes.

4. Test the serial connections by entering one of the following:

   - To test the whole cluster, enter the following:

     `test serial in cluster` *clustername*

     For example:

```
cmgr> test serial in cluster fs6-8
Status: Testing serial lines ...
Status: Checking serial lines using crsd (cluster reset services) from node fs8
Success: Serial ping command OK.

Status: Checking serial lines using crsd (cluster reset services) from node fs6
Success: Serial ping command OK.

Status: Checking serial lines using crsd (cluster reset services) from node fs7
Success: Serial ping command OK.

Notice: overall exit status:success, tests failed:0, total tests executed:1
```

   - To test an individual node, entering the following:

     `test serial in cluster` *clustername* `node` *machinename*

     For example:

```
cmgr> test serial in cluster fs6-8 node fs7
Status: Testing serial lines ...
Status: Checking serial lines using crsd (cluster reset services) from node fs6
Success: Serial ping command OK.

Notice: overall exit status:success, tests failed:0, total tests executed:1
```

   - To test an individual node using just a `ping`, enter the following:

     `admin ping node` *nodename*

For example:

```
cmgr> admin ping node fs7

ping operation successful
```

5. If a command fails, make sure all the cables are seated properly and rerun the command.
6. Repeat the process on other nodes in the cluster.

Administration Tools

You can perform IRIS FailSafe administration tasks using either the FailSafe Manager graphical user interface (GUI) or the `cmgr`(1M) command. Although these tools use the same underlying software command line interface (CLI) to configure and monitor a FailSafe system, the GUI provides the additional features that are particularly important in a production system; see "GUI Overview", page 89.

FailSafe Manager GUI

The FailSafe Manager GUI lets you configure, administer, and monitor a FailSafe cluster and nodes.

Starting the GUI

When FailSafe daemons have been started, you must be sure to connect to a node that is running all of the FailSafe cluster daemons in order to obtain the correct cluster status (see "Verify that the Cluster Daemons are Running", page 104). When FailSafe cluster daemons have not yet been started in a cluster, you can connect to any node in the pool.

Note: The node from which you run the GUI affects your view of the cluster. You should wait for a change to appear in the tree view before making another change; the change is not guaranteed to be propagated across the cluster until it appears in the tree view. The entire cluster status information is sent each time a change is made to the cluster database; therefore, the larger the configuration, the longer it will take.

You should only make changes from one instance of the GUI running at any given time. (Changes made by a second GUI instance — a second invocation of `fstask` — may overwrite changes made by the first instance, because different GUI instances are updated independently at different times. In time, however, independent GUI instances will provide the same information.) However, multiple windows accessed via the **File** menu are all part of a single GUI instance; you can make changes from any of these windows.

To ensure that the required privileges are available for performing all of the tasks, you should log in to the GUI as `root`. However, some or all privileges can be

granted to any user by the system administrator using the Privilege Manager, part of the IRIX Interactive Desktop System Administration (`sysadmdesktop`) product. For more information, see the *Personal System Administration Guide*.

To start the GUI, use one of these methods:

- Enter the following command line:

  # **`/usr/sbin/fstask`**

  Note: If you invoke `fstask` before you have defined a cluster, you will get an error message. If you are in the process of defining a cluster, you can ignore it.

  The `fsdetail` command performs the identical function as `fstask`; both commands are kept for historical purposes.

- Choose **FailSafe Manager** from the toolchest.

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

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

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

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

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

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

GUI Overview

The **FailSafe Manager** GUI allows you to administer the entire cluster from a single point. It provides access to the tasks that help you set up and administer your cluster. *Guided Configuration* tasks 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 individual tasks by simply clicking their titles.

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 **File** menu lets you do the following:

- Invoke multiple windows for this instance of the GUI
- Display the `/var/adm/SYSLOG` system log file, and the `/var/sysadm/salog` system administration log file (which shows the commands accessed by the GUI)
- Launch the Performance Co-Pilot (PCP) tools to perform resource monitoring (`rmvis`(1)) and heartbeat monitoring (`hbvis`(1))
- Close the current window
- Exit the GUI completely

The **Edit** menu lets you expand and collapse the contents of the tree view. You can also choose to automatically expand the display to reflect new nodes added to the pool or cluster.

The **Tasks** menu contains the following:

- **Find Tasks**, which lets you use keywords to search for a specific task
- **Guided Configuration**, which contains the task sets to set up your cluster, define filesystems, modify an existing cluster, and check status
- **Nodes**, which contains tasks to define and manage the nodes
- **Cluster**, which contains tasks to define and manage the cluster
- **Resource Types**, which contains tasks to set up and configure highly available resource types such as `volume`
- **Resources**, which contains tasks to set up and configure individual resources

- **Failover Policies**, which contains tasks to determine how FailSafe should keep resource groups highly available
- **Resource Groups**, which contains tasks to define resource groups and manage them
- **FailSafe HA Services**, which allows you to start and stop highly available (HA) services, set the FailSafe tie-breaker node, and set the log configuration
- **Diagnostics**, which contains the tasks to test connectivity, resources, and failover policies

By default, the window is divided into two sections: the *tree view* and the *item view*. You can use the arrows in the middle of the window to shift the display. To find a component in the current tree, enter the name of the component and click the **Find** button.

To deselect an item in the tree view, click anywhere in the tree view except on the name of an item.

Viewing the Cluster Components

Choose what you want to view from the **View** selection: the nodes in the cluster, the nodes in the pool (all defined nodes), or filesystems in the cluster.

Viewing Component Details

To view the details on any component, click on its icon name in the tree view. The configuration and status details will appear in the item view to the right. To see the details about an item in the item view, select its name (which will appear in blue); details will appear in a new window. Terms with glossary definitions also appear in blue.

You can view details about any of the following components by selecting the icon:

- Cluster
- Nodes
- Resource types
- Resources

- Resource groups
- Failover policies

Performing Tasks

To perform an individual task, do the following:

1. Select the task name from the **Task** menu or click the right mouse button within the tree view. For example:

   Task
   > Guided Configuration
   > Set Up a New Cluster

   The task window appears.

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

2. Enter information in the appropriate fields and click **OK** to complete the task. (Some tasks consist of more than one page; in these cases, click **Next** to go to the next page, complete the information there, and then click **OK**.) In every task, the cluster configuration will not update until you click **OK**.

   A dialog box appears confirming the successful completion of the task.

3. Continue launching tasks as needed.

Screens

Figure 4-1 shows the **FailSafe Manager** window. Figure 4-2 shows details in the item view.

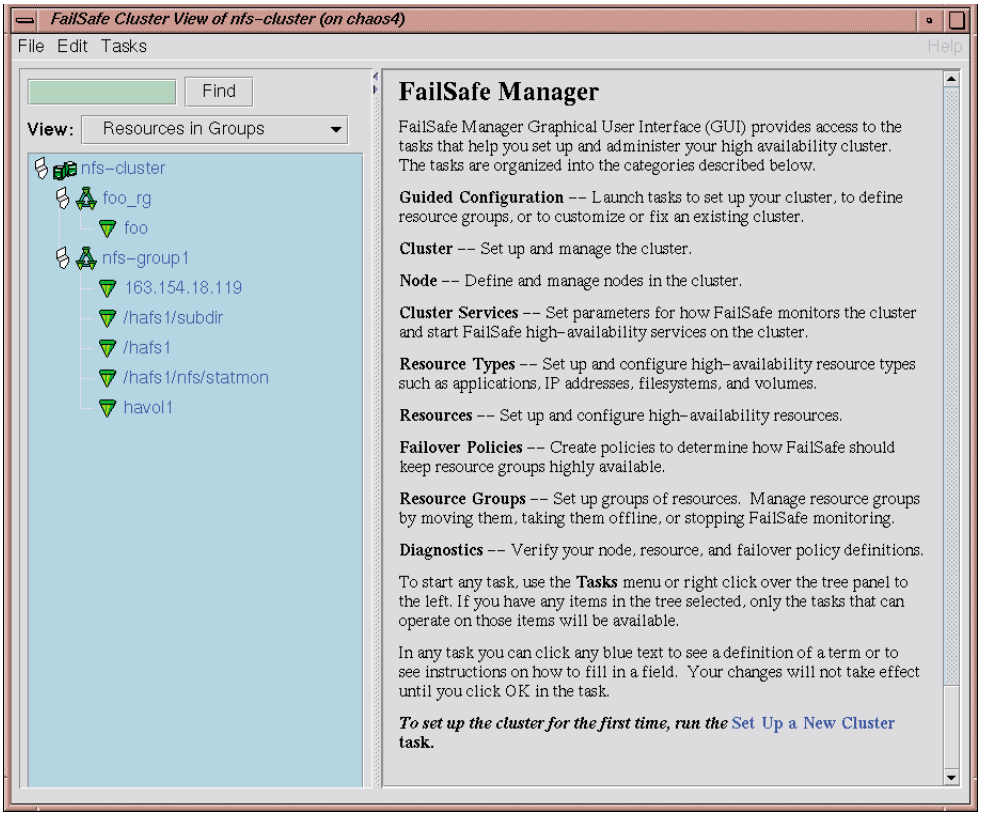

Figure 4-1 FailSafe Manager GUI

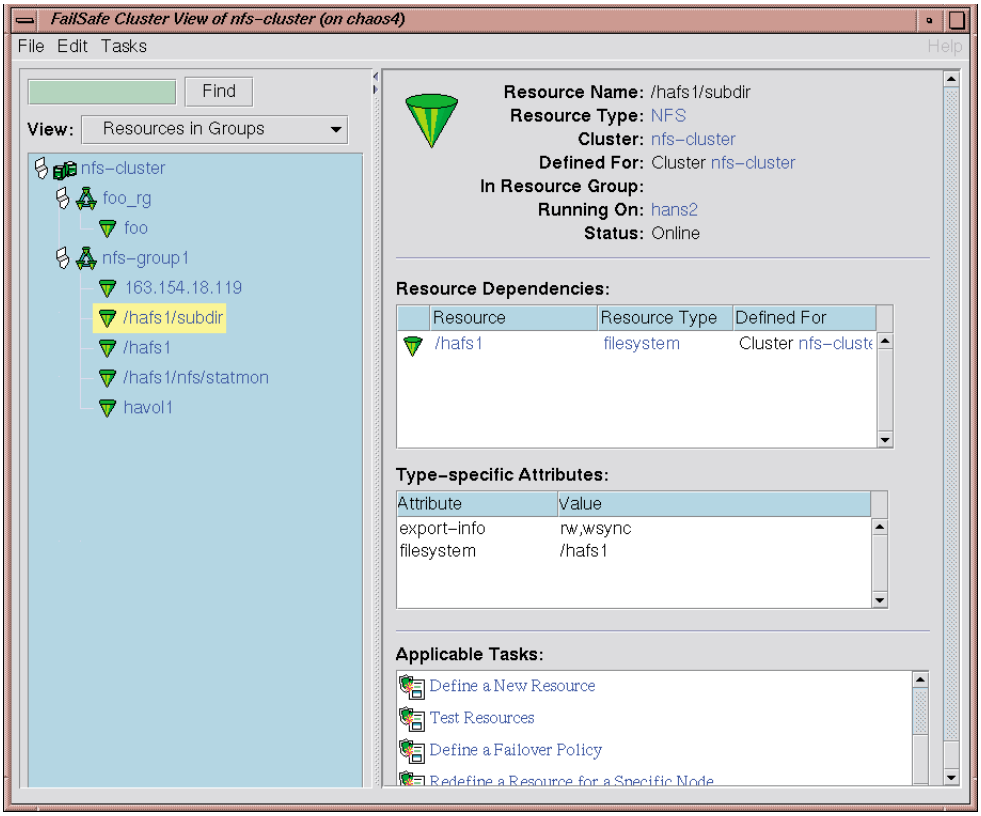

Figure 4-2 GUI Showing Details for a Resource

`cmgr` Command

The `cmgr`(1M) command is more limited in its functions. It enables you to configure and administer a 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. However, an experienced FailSafe administrator may find `cmgr` to be convenient when performing basic FailSafe configuration tasks, executing isolated single tasks in a production environment, or running scripts to automate some cluster administration tasks.

This section documents how to perform FailSafe administrative tasks by means of the `cmgr` command. You must be logged in as `root`.

The `cmgr` command uses the same underlying FailSafe commands as the GUI.

To use `cmgr`, enter either of the following:

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

For more assistance, you can use the `-p` option on the command line; see "Using Prompt Mode", page 94.

After you have entered this command, you will see the following:

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

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

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

Getting Help

After the command prompt displays, you can enter subcommands. At any time, you can enter `?` or `help` to bring up the `cmgr` help display.

Using Prompt Mode

The `cmgr`(1M) command provides an option which displays detailed prompts for the required inputs of that define and modify FailSafe components. You can run in prompt mode in either of the following ways:

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

  ```
  # cmgr -p
  ```

- Execute a `set prompting on` command while in normal interactive mode, 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 `cmgr` commands.

To get out of prompt mode, enter the following command:

```
cmgr> set prompting off
```

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

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

cm1a?
```

At the `cm1a?` prompt, enter the individual node definition commands in the following format (for full information on defining nodes, see "Define a Node with `cmgr`", page 119). For example:

```
cm1a? set hostname to hostname
```

A series of commands is required to define a node. If you are running `cmgr` in prompt mode, however, you are prompted for each required command, as shown in the following example:

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

Node Name [cm1a]? cm1a

Hostname[optional]? cm1a
Is this a FailSafe node <true|false> ? true
Is this a CXFS node <true|false> ? false
Node ID ? 1
Partition ID[optional] ? (0)
Reset type <powerCycle> ? (powerCycle)
Do you wish to define system controller info[y/n]:y
Sysctrl Type <msc|mmsc|l2>? (msc) msc
Sysctrl Password [optional]? ( )
Sysctrl Status <enabled|disabled>? enabled
Sysctrl Owner? cm2
Sysctrl Device? /dev/ttyd2
Sysctrl Owner Type <tty> [tty]?
Number of Network interfaces [2]? 2
NIC 1 - IP Address? cm1
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false>? true
NIC 1 - (use network for control messages) <true|false>? true
```

```
NIC 1 - Priority <1,2,...>? 1
NIC 2 - IP Address? cm2
NIC 2 Heartbeat HB (use network for heartbeats) <true|false>? true
NIC 2 - (use network for control messages) <true|false>? false
NIC 2 - Priority <1,2,...>? 2
```

Completing Actions and Cancelling

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

- `cancel`, which aborts the current mode and discards any changes you have made
- `done`, which commits the current definitions or modifications and returns to the `cmgr>` prompt

Command Line Editing within `cmgr`

The `cmgr` command supports the following `cmgr` command-line editing commands:

`h [`*n*`]` or `history [`*n*`]`	Displays command line history.The optional *n* can be used to set the number commands that will be remembered.
`!!`	Refers to the previous command. By itself, this substitution repeats the previous command.
`!`*n*	Refers to command line *n*.
`!-`*n*	Refers to the current command line minus *n*.
`!`*string*	Refers to the most recent command starting with *string*.
`exit`	Exits from the shell.
`Ctrl-W`	Deletes the previous word.
`Ctrl-D`	Deletes the current character.
`Ctrl-A`	Goes to the beginning of the line.
`Ctrl-E`	Goes to the end of the line.
`Ctrl-F`	Moves forward one character.
`Ctrl-B`	Moves backward one character.
`Ctrl-H`	Deletes the previous character.

`Ctrl-N`	Moves down in the history.
`Ctrl-K`	Erases to the end of the line from the cursor.
`Ctrl-L`	Clears the screen and redisplays the prompt.
`Ctrl-P`	Moves up in the history.
`Ctrl-U`	Erases to the beginning of line from the cursor.
`Ctrl-R`	Redraws the input line.
`Esc-f`	Moves forward one word.
`Esc-b`	Moves backward one word.
`Esc-d`	Deletes the next word.
`Esc-DEL`	Deletes the previous word.

Long-Running Tasks

The tasks to define the cluster and to stop HA services are long-running tasks that might take a few minutes to complete. The `cmgr` command will provide intermediate task status for such tasks. For example:

```
cmgr> stop ha_services in cluster nfs-cluster
Making resource groups offline
Stopping HA services on node node1
Stopping HA services on node node2
```

Startup Script

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

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

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

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

```
# setenv CMGR_START_FILE /cmgr_rc
```

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

Entering Subcommands on the Command Line

You can enter some `cmgr` subcommands directly from the command line using the following format:

`cmgr -c "`*subcommand*`"`

where *subcommand* can be any of the following with the appropriate operands:

- `admin`, which allows you to perform certain actions such as resetting a node
- `delete`, which deletes a cluster or a node
- `help`, which displays help information
- `show`, which displays information about the cluster or nodes
- `start`, which starts FailSafe HA services and sets the configuration so that HA services will be automatically restarted upon reboot
- `stop`, which stops FailSafe HA services and sets the configuration so that HA services are not restarted upon reboot
- `test`, which tests connectivity

For example, to display information about the cluster, enter the following:

```
# cmgr -c "show clusters"
1 Cluster(s) defined
      eagan
```

See Chapter 5, "Configuration", page 103, and the `cmgr`(1M) man page for more information.

Using Script Files

You can execute a series of `cmgr` commands by using the `-f` option and specifying an input file, as follows:

`cmgr -f` *input_file*

Or, you could include the following as the first line of the file and then execute it as a script:

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

Each line of the file must be a valid `cmgr` command line, comment line (starting with #), or a blank line. (You must include a `done` command line to finish a multilevel command and end the file with a `quit` command line.)

If any line of the input file fails, `cmgr` will exit. You can choose to ignore the failure and continue the process by using the `-i` option with the `-f` option, as follows:

`cmgr -if` *input_file*

Or include it in the first line for a script:

```
#!/usr/cluster/bin/cmgr -if
```

Note: If you include `-i` when using a `cmgr` command line as the first line of the script, you must use this exact syntax (that is, `-if`).

For example, suppose the file `/tmp/showme` contains the following:

```
fs6# more /tmp/showme
show clusters
show nodes in cluster fs6-8
quit
```

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

```
fs6# /usr/cluster/bin/cmgr -if /tmp/showme

1 Cluster(s) defined
        fs6-8


Cluster fs6-8 has following 3 machine(s)
        fs6
        fs7
        fs8
```

Or you could include the cmgr command line as the first line of the script, give it execute permission, and execute showme itself:

```
fs6# more /tmp/showme
#!/usr/cluster/bin/cmgr -if
#
show clusters
show nodes in cluster fs6-8
quit

fs6# /tmp/showme

1 Cluster(s) defined
        fs6-8


Cluster fs6-8 has following 3 machine(s)
        fs6
        fs7
        fs8
```

Template Scripts

Template files of 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 a list of cmgr commands to create a particular object, as well as comments describing each field. The template also provides default values for optional fields.

Table 4-1 shows the template scripts for cmgr that are found in the var/cluster/cmgr-templates directory.

Table 4-1 Template Scripts for cmgr

File name	Description
cmgr-create-cluster	Creation of a cluster
cmgr-create-failover_policy	Creation of failover policy

File name	Description
`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*	Script template for creation of resource of type *resource type*

To create a FailSafe configuration, you can concatenate multiple templates into one file and execute the resulting script. 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 `cmgr` script must have only one `quit` line.

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

- Three copies of the `cmgr-create-node` file
- One copy of each of the following files:
  - `cmgr-create-cluster`
  - `cmgr-create-failover_policy`
  - `cmgr-create-resource_group`
  - `cmgr-create-resource-volume`
  - `cmgr-create-resource-filesystem`
  - `cmgr-create-resource-IP_address`
  - `cmgr-create-resource-NFS`

Invoking a Shell from within `cmgr`

You can invoke a shell from within the `cmgr`. Enter the following command to invoke a shell:

```
cmgr> sh
```

To exit the shell and to return to the `cmgr` prompt, enter `exit` at the shell prompt.

Configuration

This chapter provides a summary of the steps required to configure a cluster using either the FailSafe Manager graphical user interface (GUI) or the `cmgr`(1M) command.

Note: For the initial installation, SGI **highly** recommends that you use the GUI guided configuration tasksets. See "Guided Configuration with the GUI", page 108.

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

The following sections describe the preliminary steps you should follow, information you must understand, the GUI guided configuration, and the various individual tasks using the GUI and `cmgr`.

Preliminary Steps

The cluster processes are started automatically when FailSafe and cluster subsystems from the IRIX CD are installed. Before starting cluster daemons, verify if the cluster daemons are running. Complete the following steps to ensure that you are ready to configure the initial cluster:

- "Verify that the Cluster `chkconfig` Flag is On", page 104
- "Start the Cluster Daemons", page 104
- "Verify that the Cluster Daemons are Running", page 104
- "Determine the Hostname of the Node", page 105

During the course of configuration, you will see various information-only messages in the log files.

Verify that the Cluster `chkconfig` Flag is On

Ensure that the output from `chkconfig`(1M) shows the following flag set to `on`:

```
# chkconfig
        Flag                State
        ====                =====
        cluster             on
```

If it is not, set it to `on`. For example:

```
# /etc/chkconfig cluster on
```

Start the Cluster Daemons

Enter the following to start the cluster daemons:

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

After you start highly available (HA) services, the following daemons are also started on a base FailSafe system (without optional plug-ins):

- `ha_fsd`
- `ha_cmsd`
- `ha_gcd`
- `ha_srmd`
- `ha_ifd`

Verify that the Cluster Daemons are Running

When you **first install** the software, the following cluster daemons should be running:

- `fs2d`
- `cmond`
- `cad`
- `crsd`

To determine which daemons are running, enter the following:

```
ps -ef | grep cluster
```

The following shows an example of the output when just the initial daemons are running; for readability, whitespace has been removed and the daemon names are highlighted:

```
# ps -ef | grep cluster
root 31431     1 0 12:51:36 ?     0:14 /usr/lib32/cluster/cbe/fs2d /var/cluster/cdb/cdb.db #
root 31456 31478 0 12:53:01 ?     0:03 /usr/cluster/bin/crsd -l
root 31475 31478 0 12:53:00 ?     0:08 /usr/cluster/bin/cad -l -lf /var/cluster/ha/log/cad_log --append_log
root 31478     1 0 12:53:00 ?     0:00 /usr/cluster/bin/cmond -L info -f /var/cluster/ha/log/cmond_log
root 31570 31408 0 14:01:52 pts/0 0:00 grep cluster
```

If you do not see these processes, go to the logs to see what the problem might be. If you must restart the daemons, enter the following:

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

Determine the Hostname of the Node

When you are initially configuring the cluster, you must use the IP address or the value of `/etc/sys_id` (which must match the primary name of the IP address for the node in `/etc/hosts`) when logging in to the GUI and when defining the nodes in the pool. The value of `/etc/sys_id` is displayed by the `hostname`(1) command. For example:

```
# hostname fs6
```

Also, if you use `nsd`(1M), you must configure your system so that local files are accessed before the network information service (NIS) or the domain name service (DNS). See "Hostname Resolution: `/etc/sys_id`, `/etc/hosts`, `/etc/nsswitch.conf`", page 60.

Caution: It is critical that these files are configured properly and that you enter the primary name for the nodes. See "Install Software", page 55.

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 GUI or `cmgr`.

Configuring Timeout Values and Monitoring Intervals

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

application_downtime = *failure_detection* + *time_to_handle_failure* + *failure_recovery_time*

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

- When a node goes down, there will be a node failure detection after the node timeout time, which is one of the parameters that you can modify. All failures that

translate into a node failure (such as heartbeat failure and OS failure) fall into this failure category. Node timeout has a default value of 15 seconds.

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

  For information on setting values for a resource type, see "Define a Resource Type with the GUI", page 146.

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

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

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

- Execute the failover policy script (approximately 5 seconds).
- Run the `stop` action script for all resources in the resource group. This is not required for node failure; the failing node will be reset.
- Run the `start` action script for all resources in the resource group.

Setting Configuration Defaults with `cmgr`

Certain `cmgr` commands require you to specify a cluster, node, or resource type. Before you configure the components of a FailSafe system, you can set defaults for these values that will be used if you do not specify an explicit value. The default values are in effect only for the current session of `cmgr`.

Use the following `cmgr` commands:

- Default cluster:

  `set cluster` *clustername*

  For example:

  ```
  cmgr> set cluster test-cluster
  ```

- Default node:

  `set node` *nodename*

  For example:

  ```
  cmgr> set node node1
  ```

- Default resource type:

  `set resource_type` *resource_type_name*

  For example:

  ```
  cmgr> set resource_type IP_address
  ```

To view the current default configuration values, use the following command:

```
show set defaults
```

Guided Configuration with the GUI

The GUI provides guided configuration task sets to help you configure your FailSafe cluster.

The node from which you run the GUI affects your view of the cluster. You should wait for a change to appear in the tree view before making another change; the change is not guaranteed to be propagated across the cluster until the icons appear in the tree view.

You should only make changes from one instance of the GUI running at any given time; changes made by a second GUI instance (a second invocation of `fstask`) may overwrite changes made by the first instance. However, multiple windows accessed

via the **File** menu are all part of a single GUI instance; you can make changes from any of these windows.

Set Up a New Cluster

Note: Within the tasks, you can click on any **blue** text to get more information about that concept or input field. In every task, the cluster configuration will not update until you click on **OK**.

The **Set Up a New Cluster** taskset in the **Guided Configuration** leads you through the steps required to create a new cluster. It encompasses tasks that are detailed elsewhere.

The GUI provides a convenient display of a cluster and its components. Verify your progress to avoid adding nodes too quickly.

Do the following:

1. Select the following:

   Tasks
   > Guided Configuration
   > Set up a New Cluster

2. Click on **Define a Node** to define the node to which you connected. The hostname that appears in `/etc/sys_id` is used for all node definitions; see "Determine the Hostname of the Node", page 105. See "Define a Node", page 116.

3. (*Optional*) **After** the first node icon appears in the tree view, click on step 2, **Define a Node**, to define the other nodes in the cluster. The hostname/IP-address pairings and priorities of the networks must be the same for each node in the cluster.

   Note: Do not define another node until the icon for this node appears in the tree view. If you add nodes too quickly (before the database can include the node), errors will occur.

   Repeat this step for each node. For large clusters, SGI recommends that you define only the first 3 nodes and then continue on to the next step; add the remaining nodes after you have a successful small cluster.

4. Click on **Define a Cluster** to create the cluster definition. See "Define a Cluster", page 137. Verify that the cluster appears in the tree view; choose **View: Nodes in Cluster**.
5. Click on **Add/Remove Nodes in Cluster** to add the nodes to the new cluster. See "Add/Remove Nodes in the Cluster with the GUI", page 125.

   Click on **Next** to move to the second screen of tasks.
6. (*Optional*) Click on **Test Connectivity** to verify that the nodes are physically connected. See "Test Connectivity with the GUI", page 262. (This test requires the proper configuration of the `/etc/.rhosts` file.)
7. Click on **Start HA Services**.
8. Click on **Close**. Clicking on **Close** exits the taskset; it does not undo the task.

Set Up a Highly Available Resource Group

Note: Within the tasks, you can click on any blue text to get more information about that concept or input field. In every task, the cluster configuration will not update until you click on **OK**.

The **Set Up a Highly Available Resource Group** taskset leads you through the steps required to define a resource group. It encompasses tasks that are detailed elsewhere.

Do the following:

1. Define a new resource. See "Define a New Resource", page 169.
2. Add any required resource dependencies. See"Add/Remove Dependencies for a Resource Definition", page 176.
3. Verify the resources and dependencies. See "Test Resources with the GUI", page 262.
4. Define a failover policy to specify where the resources can run. See "Define a Failover Policy", page 182.
5. Test the failover policies. See "Test Failover Policies with the GUI", page 263.
6. Define a resource group that uses the failover policy you defined earlier. See "Define a Resource Group", page 193.

7. Add or remove resources in resource group. See "Test Failover Policies with the GUI", page 263.
8. Set the resources in the resource group to start when HA services are started. See "Bring a Resource Group Online", page 250.
9. Start FailSafe HA services if they have not already been started. See "Start FailSafe HA Services with the GUI", page 199.

Repeat these steps for each resource group.

Set Up an Existing CXFS Cluster for FailSafe

This taskset appears on the GUI **if** you also have CXFS installed.

Note: Within the tasks, you can click on any blue text to get more information about that concept or input field. In every task, the cluster configuration will not update until you click on **OK**.

The **Set Up an Existing CXFS Cluster for FailSafe** taskset leads you through the steps required to convert existing CXFS nodes and the cluster to FailSafe. It encompasses tasks that are detailed elsewhere.

There is a single database for FailSafe and CXFS. If a given node applies to both products, ensure that any modifications you make are appropriate for both products.

Do the following:

1. Click on **Convert a CXFS Cluster to FailSafe**. This will change the cluster type to `CXFS and FailSafe`. See "Convert a CXFS Cluster to FailSafe with the GUI", page 142.
2. Stop CXFS services on the nodes to be converted using the CXFS GUI. See the *CXFS Version 2 Software Installation and Administration Guide*.
3. Click on **Convert a CXFS Node to FailSafe** to convert the local node (the node to which you are connected). A converted node can be of type `CXFS and FailSafe` or `FailSafe`. See "Convert a CXFS Node to FailSafe with the GUI", page 131.
4. Click on **Convert a CXFS Node to FailSafe** to convert another node. Repeat this step for each node you want to convert.

5. Click on **Start HA Services**.

Fix or Upgrade Cluster Nodes

You can use the following tasks to fix or upgrade nodes:

- Stop FailSafe HA services on the cluster node(s). See "Stop FailSafe HA Services", page 200.
- Perform the necessary maintenance on the node. Only if required, see "Reset a Node with the GUI", page 257.
- Monitor the state of the cluster components in the tree view. See "System Status", page 233.

Make Changes to Existing Cluster

You can make most changes while the HA services are active, such as changing the way the cluster administrator is notified of events; however, you must first stop cluster services before testing connectivity.

See the following:

- "Modify a Cluster Definition", page 141
- "Define a Node", page 116
- "Test Connectivity with the GUI", page 262
- "Add/Remove Nodes in the Cluster with the GUI", page 125
- "Set FailSafe HA Parameters", page 203

Optimize Node Usage

You can improve cluster performance by taking advantage of a particular node's hardware. For example, one node in the cluster may have a larger disk or a faster CPU.

Depending upon your situation, you may find the following tasks useful:

- Ensure that a resource group will always run on the more powerful node; list that node first in the failover domain and choose `Automatic` as the recover attribute. See "Modify a Failover Policy Definition with the GUI", page 188.
- "Move a Resource Group", page 197.
- Create a resource that has a custom definition on a specific node. See "Redefine a Resource for a Specific Node", page 175.
- Create a resource type that is defined only for the chosen node (as opposed to the entire cluster). See "Redefine a Resource Type for a Specific Node", page 155.

Define Custom Resource

You can use the following tasks to define a custom resource:

- "Define a Resource Type", page 146
- "Redefine a Resource for a Specific Node", page 175
- "Add/Remove Dependencies for a Resource Type", page 159
- "Define a New Resource", page 169
- "Add/Remove Dependencies for a Resource Definition", page 176
- "Test Resources with the GUI", page 262

Customize FailSafe Failure Detection

You can do the following to customize how FailSafe monitors and fails over resource groups:

- Change the node timeout or the heartbeat period (the time intervat at which FailSafe sends messages between nodes). See "Set FailSafe HA Parameters", page 203.
- Change the monitor action timeout and the restart action timeout used by a resource type. See "Modify a Resource Type Definition", page 162.

Customize Resource Group Failover Behavior

You can use various tasks to change failover behavior in the cluster or the resource group:

- To change how the cluster detects when a failover is necessary, see "Set FailSafe HA Parameters", page 203.
- To change the nodes and their ordering in the failover domain, see "Modify a Failover Policy Definition", page 188.
- To change monitoring settings for the resource types used in the resource group, see "Modify a Resource Type Definition", page 162.

You can also create a custom failover policy script:

1. Use the *IRIS FailSafe Version 2 Programmer's Guide* to write a custom failover script.
2. Place the scripts in the `/var/cluster/ha/policies` directory.
3. Restart the FailSafe Manager GUI.
4. Change the desired failover policy to use your new custom failover script. See "Modify a Failover Policy Definition", page 188.
5. Select **View: Groups owned by Nodes** in the GUI tree view.
6. Test the script by moving a resource group from one node to another, simulating failover. Watch the resource group behavior in the tree view to confirm that failover behavior works as expected. See "Move a Resource Group", page 197.

Customize Resource Failover Behavior

You can customize resource failover behavior by editing existing action scripts or creating new scripts. Do the following:

1. Make a copy of the action scripts you want to modify. Action scripts for each resource type are contained in the `/var/cluster/ha/resource_types` directory.
2. Edit the copies or create new scripts. See the *IRIS FailSafe Version 2 Programmer's Guide*.
3. Place the edited/new scripts in the appropriate subdirectory in `/var/cluster/ha/resource_types`.

4. Restart the FailSafe Manager GUI.
5. Make use of the new scripts in the resource type. See "Define a Resource Type", page 146, and "Modify a Resource Type Definition", page 162.
6. Define resources using the new resource type. See "Define a New Resource", page 169.
7. Verify that FailSafe can manage the new custom resources. See "Test Resources with the GUI", page 262.
8. Add the new resource. See "Add/Remove Nodes in the Cluster", page 125.

Redistribute Resource Load in Cluster

After setting up resource groups and observing how they fail over, you may want to distribute the resource groups differently to balance the load among the nodes in the cluster.

1. Determine the current load. For example, invoke the IRIX System Manager tool from the Toolchest, then launch the graphical system monitor window by selecting the **System Performance** category and then the **View System Resources** task to view various system load statistics. For more information, see the `gr_osview`(1) man page.
2. If you want to redistribute the resource groups among the nodes, see "Move a Resource Group", page 197.
3. If you want to create a new failover policy that uses nodes in a different order or uses different nodes, do the following:
   - Create a new failover policy to use nodes more efficiently. See "Define a Failover Policy", page 182.
   - Use the new failover policy for the resource group. See "Modify a Resource Group Definition", page 195.
   - Move the resource group to activate the new failover policy. (FailSafe will only start using a failover policy when the associated resource group is moved.) See "Move a Resource Group", page 197.

Node Tasks

A *node* is an operating system (OS) image, usually an individual computer. The nodes are connected to a storage area network (SAN) that connects the storage systems to the nodes in the cluster. A node can belong to only one cluster.

This use of the term *node* does not have the same meaning as a node in an SGI Origin 3000 or SGI 2000 system.

This section describes the following node configuration tasks:

- "Define a Node"
- "Add/Remove Nodes in the Cluster", page 125
- "Modify a Node Definition", page 126
- "Convert a CXFS Node to FailSafe", page 130
- "Delete a Node", page 132
- "Display a Node", page 135

Define a Node

This section describes how to define a node.

Define a Node with the GUI

The first node you define must be the node that you have logged into, in order to perform cluster administration. You **must** use the hostname that appears in `/etc/sys_id`.

Note: Within the tasks, you can click on any blue text to get more information about that concept or input field. In every task, the cluster configuration will not update until you click on **OK**.

To define a node, do the following:

1. Enter the following:
   - **Hostname**: Hostname of the node you are defining, such as `mynode.company.com` (this can be abbreviated to `mynode` if it is resolved on

all nodes). You **must** use the hostname that appears in `/etc/sys_id`. Use the `hostname`(1) command to display the hostname as it appears in the `/etc/sys_id` file. See "Hostname Resolution: `/etc/sys_id`, `/etc/hosts`, `/etc/nsswitch.conf`", page 60.

- **Logical Name**: The same as the hostname, or an abbreviation of the hostname (such as `lilly`), or an entirely different name (such as `nodeA`). Logical names cannot begin with an underscore (_) or include any whitespace characters, and can be at most 255 characters.

  Note: If you want to rename a node, you must delete it and then define a new node.

- **Networks for Incoming Cluster Messages**: Do the following:

  – **Network**: Enter the IP address or hostname of the private network. (The hostname must be resolved in the `/etc/hosts` file.) The priorities of the networks must be the same for each node in the cluster. For information about using the hostname, see "Hostname Resolution: `/etc/sys_id`, `/etc/hosts`, `/etc/nsswitch.conf`", page 60. For information about why a private network is required, see "Private Network", page 7.

  – **Messages to Accept**: Select the appropriate type. You can use the **None** setting if you want to temporarily define a network but do not want it to accept messages.

  – Click on **Add** to add the network to the list.

  If you later want to modify the network, click on the network in the list to select it, then click on **Modify**.

  If you want to delete a network from the list, click on the network in the list to select it, then click on **Delete**.

- **Node ID**: (*Optional*) An integer in the range 1 through 32767 that is unique among the nodes in the pool. If you do not specify a number, FailSafe will calculate an ID for you. The default ID is a 5-digit number based on the machine's serial number and other machine-specific information; it is not sequential. You must not change the node ID number after the node has been defined.

- **Partition ID**: *(Optional)* Uniquely defines a partition in a partitioned SGI Origin 3000 system. If your system is not partitioned, leave this field empty.

Note: Use the `mkpart`(1M) command to determine the partition ID value:

- The `-n` option lists the partition ID (which is 0 if the system is not partitioned).
- The `-l` option lists the bricks in the various partitions (use *rack#.slot#* format in the GUI)

  For example (output truncated here for readability):

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

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

```
1
001.10
```

Click **Next** to move to the next screen.

- You can choose whether or not to use the system controller port to reset the node. If you want FailSafe to be able to use the system controller to reset the node, you select the **Set Reset Parameters** checkbox and provide the following information:

  - This node:

    - **Port Type:** select **L1** (L1 system controller for Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C systems), **L2** (L2 system controller for Origin 3400, Origin 3800, Origin 300 with NUMAlink module, and Onyx 3000 series), **MSC** (module system controller for Origin 200, Onyx2 deskside, and SGI 2100, 2200 deskside systems), or **MMSC** (multimodule system controller for rackmount SGI 2400, SGI 2800 and Onyx2 systems). See also "Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C Console Support", page 225

    - **Port Password:** system controller password for privileged commands, **not** the node's `root` password or PROM password. On some machines, the system administrator may not have set this password. If you wish

to set or change the system controller port password, consult the hardware manual for your node.

- **Temporarily Disable Port:** if you want to provide reset information now but do not want to allow the reset capability at this time, check this box. If this box is checked, FailSafe cannot reset the node.

– Owner (node that sends reset command):

- **Logical Name:** name of the node that sends the remote reset command. Serial cables must physically connect the node being defined and the owner node through the system controller port. At run time, the node must be defined in the pool.

  You can select a logical name or enter the logical name of a node that is not yet defined. However, you must define the node before you run the node connectivity diagnostics task.

- **TTY Device:** name of the terminal port (TTY) on the owner node to which the system controller is connected, such as `/dev/ttyd2`. The other end of the cable connects to this node's system controller port, so the node can be controlled remotely by the other node.

If you do not want to use the reset function at all, click on the **Set System Controller Parameters** box to deselect (uncheck) it.

2. Click on **OK** to complete the task.

You can use the hostname or the IP address as input to the network interface field. However, using the hostname requires DNS on the nodes; therefore, you may want to use the actual IP address.

Note: Do not add a second node until the first node icon appears in the tree view. The entire cluster status information is sent each time a change is made to the cluster database; therefore, the larger the configuration, the longer it will take.

Define a Node with `cmgr`

To define a node, use the following commands:

```
define node logical_hostname
    set hostname to hostname
    set nodeid to nodeID
```

```
set partition_id to partitionID
set reset_type to powerCycle
set sysctrl_type to msc|mmsc|l2|l1_(based_on_node_hardware)
set sysctrl_password to password
set sysctrl_status to enabled|disabled
set sysctrl_owner to node_sending_reset_command
set sysctrl_device to /dev/ttyd2
set sysctrl_owner_type to tty_device
set is_failsafe to true|false
set is_cxfs to true|false
set weight to 0|1
add nic IP_address_or_hostname_(if_DNS)
        set heartbeat to true|false
        set ctrl_msgs to true|false
        set priority to integer
remove nic IP_address_or_hostname_(if_DNS)
```

Usage notes:

- `node` is the same as the hostname (such as `mynode.company.com`), or an abbreviation of the hostname (such as `mynode`), or an entirely different name (such as `nodeA`). Logical names cannot begin with an underscore (_) or include any whitespace characters, and can be at most 255 characters.
- `hostname` is the hostname as returned by the `hostname`(1) command on the node being defined. Other nodes in the pool must all be able to resolve this hostname correctly via `/etc/hosts` or a name resolution mechanism. The default for *hostname* is the value for *logical_hostname*; therefore, you must supply a value for this command if you use a value other than the hostname or an abbreviation of it for *logical_hostname*.
- `nodeid` is an integer in the range 1 through 32767 that is unique among the nodes in the pool. If you do not specify a number, FailSafe will calculate an ID for you. The default ID is a 5-digit number based on the machine's serial number and other machine-specific information; it is not sequential. You must not change the node ID number after the node has been defined.
- `partition_id` uniquely defines a partition in a partitioned SGI Origin 3000 system.

Note: Use the `mkpart`(1M) command to determine this value:

– The `-n` option lists the partition ID (which is 0 if the system is not partitioned).
– The `-l` option lists the bricks in the various partitions (use *rack#.slot#* format in `cmgr`).

For example (output truncated here for readability):

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

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

```
1
001.10
```

If your system is not partitioned, use a value of 0.

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

- `reset_type` has only one legal value: `powerCycle`.
- `sysctrl_type` is the system controller type, based on the node hardware, as shown in Table 5-1, page 123.
- `sysctrl_password` is the password for the system controller port, not the node's `root` password or PROM password. On some nodes, the system administrator may not have set this password. If you wish to set or change the system controller password, consult the hardware manual for your node.
- `sysctrl_status` is either `enabled` or `disabled`. This allows you to provide information about the system controller but temporarily disable by setting this value to `disabled` (meaning that FailSafe cannot reset the node). To allow FailSafe to reset the node, enter `disabled`.
- `sysctrl_owner` is the logical name of the node that can reset this node via the system controller port. A node may reset another node when it detects that the node is not responding to heartbeat messages or is not responding correctly to requests. A serial cable must physically connect one of the owner's serial ports to the system controller port of the node being defined. The owner must be a node

in the pool. (You can specify the name of a node that is not yet defined. However, the owner must be defined as a node before the node connectivity diagnostic test is run and before the cluster is activated.)

- `sysctrl_device` is the system controller device. `/dev/ttyd2` is the only legal value.
- `sysctrl_owner_type` is the name of the terminal port (TTY) on the owner node to which the system controller is connected, such as `/dev/ttyd2`. The other end of the cable connects to this node's system controller port, so the node can be controlled remotely by the other end.
- `is_failsafe` and `is_cxfs` specify the node type. If you are running just FailSafe on this node, set `is_cxfs` to `false` and `is_failsafe` to `true`. If you are running both CXFS and FailSafe on this node in coexecution cluster, set both values to `true`.
- `weight` is the node weight, which is used only for CXFS.
- `nic` is the IP address or hostname of the private network. (The hostname must be resolved in the `/etc/hosts` file.)

  There can be up to eight network interfaces, but only the first priority network is used (there is no failover). SGI requires that this network be private; see "Private Network", page 7.

  The priorities of the networks must be the same for each node in the cluster. For more information about using the hostname, see "Hostname Resolution: `/etc/sys_id`, `/etc/hosts`, `/etc/nsswitch.conf`", page 60. For information about why a private network is required, see "Private Network", page 7.

Table 5-1 System Controller Types

l1	l2	mmsc	msc
Origin 300 (see also "Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C Console Support", page 225)	Origin 3400	SGI 2400 rackmount	Origin 200
Origin 3200c	Origin 3800	SGI 2800 rackmount	Onyx2 deskside
Onyx 300	Origin 300 with NUMAlink module	Onyx2 rackmount	SGI 2100 deskside
Onyx 3200c	Onyx 3000 series		SGI 2200 deskside

Use the `add nic` command to define the network interfaces. When you enter this command, the following prompt appears:

`NIC -` *nic*`#?`

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

`set heartbeat to` *true*|*false*
`set ctrl_msgs to` *true*|*false*
`set priority to` *integer*

Use the following command from the node name prompt to remove a network controller:

`remove nic` *IP_address*

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

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

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

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

NIC - cm1 > set heartbeat to true
NIC - cm1 > set ctrl_msgs to true
NIC - cm1 > set priority to 0
NIC - cm1 > done
cm1a? done
```

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

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

Hostname[optional]? cm1a
Is this a FailSafe node <true|false> ? true
Is this a CXFS node <true|false> ? false
Node ID ? 1
Reset type <powerCycle> ? (powerCycle)
Do you wish to define system controller info[y/n]:y
Sysctrl Type <msc|mmsc|l2|l1>? (msc) msc
Sysctrl Password [optional]? ( )
Sysctrl Status <enabled|disabled>? enabled
Sysctrl Owner? cm2
Sysctrl Device? /dev/ttyd2
Sysctrl Owner Type <tty> [tty]?
Number of Network interfaces [2]? 2
NIC 1 - IP Address? cm1
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false>? true
```

```
NIC 1 - (use network for control messages) <true|false>? true
NIC 1 - Priority <1,2,...>? 1
NIC 2 - IP Address? cm2
NIC 2 Heartbeat HB (use network for heartbeats) <true|false>? true
NIC 2 - (use network for control messages) <true|false>? false
NIC 2 - Priority <1,2,...>? 2
```

Add/Remove Nodes in the Cluster

This section describes how to add or remove nodes.

Add/Remove Nodes in the Cluster with the GUI

After you have added nodes to the pool and defined the cluster, you can indicate which of those nodes to include in the cluster.

Note: Do not add or remove nodes node until the cluster icon appears in the tree view; select **View: Nodes in Cluster**.

Do the following:

1. Add or remove the desired nodes:
   - To add a node, select its logical name from the **Available Nodes** menu and click on **Add**. The node name will appear in the **Nodes to Go into Cluster** list.
   - To delete a node, click on its logical name in the **Nodes to Go into Cluster** list. (The logical name will be highlighted.) Then click on **Remove**.
2. Click on **OK** to complete the task.

Modify a Node Definition

This section describes how to modify a node definition.

Modify a Node Definition with the GUI

Note: If you want to rename a node, you must delete it and then define a new node.

To modify a node, do the following:

1. **Logical Name**: select the logical name of the node. After you do this, information for this node will be filled into the various fields.
2. Change the information in the appropriate field as follows:
   - **Networks for Incoming Cluster Messages:** the priorities of the networks must be the same for each node in the cluster.
     - **Network:** if you want to add a network for incoming cluster messages, enter the IP address or hostname into the **Network** text field and click on **Add**.
       - If you want to modify a network that is already in the list, click on the network in the list in order to select it. Then click on **Modify**. This moves the network out of the list and into the text entry area. You can then change it. To add it back into the list, click on **Add**.
       - If you want to delete a network, click on the network in the priority list in order to select it. Then click on **Delete**.
       - If you want to change the priority of a network, click on the network in the priority list in order to select it. Then click on the up and down arrows in order to move it to a different position in the list.
     - **Partition ID:** *(optional)* uniquely defines a partition in a partitioned SGI Origin 3000 system. If your system is not partitioned, leave this field empty.

Note: Use the `mkpart`(1M) command to determine the partition ID value:

- The `-n` option lists the partition ID (which is 0 if the system is not partitioned).
- The `-l` option lists the bricks in the various partitions (use *rack#.slot#* format in `cmgr`).

  For example (output truncated here for readability):

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

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

```
1
001.10
```

Click on **Next** to move to the next page

- You can choose whether or not to use the system controller port to reset the node. If you want FailSafe to be able to use the system controller to reset the node, you select the **Set Reset Parameters** checkbox and provide the following information:

  – This node:

    - **Port Type:** select **L1** (L1 system controller for Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C systems), **L2** (L2 system controller for Origin 3400, Origin 3800, Origin 300 with NUMAlink module, and Onyx 3000 series), **MSC** (module system controller for Origin 200, Onyx2 deskside, and SGI 2100, 2200 deskside systems), or **MMSC** (multimodule system controller for rackmount SGI 2400, SGI 2800 and Onyx2 systems). See also "Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C Console Support", page 225

    - **Port Password:** the password for the system controller port, **not** the node's `root` password or PROM password. On some machines, the system administrator may not have set this password. If you wish to

set or change the system controller port password, consult the hardware manual for your node.

- **Temporarily Disable Port:** if you want to provide reset information now but do not want to allow the reset capability at this time, check this box. If this box is checked, FailSafe cannot reset the node.

– Owner (node that sends reset command):

- **Logical Name:** name of the node that sends the remote reset command. Serial cables must physically connect the node being defined and the owner node through the system controller port. At run time, the node must be defined in the pool.

  You can select a logical name or enter the logical name of a node that is not yet defined. However, you must define the node before you run the node connectivity diagnostics task.

- **TTY Device:** name of the terminal port (TTY) on the owner node to which the system controller is connected, such as `/dev/ttyd2`. The other end of the cable connects to this node's system controller port, so the node can be controlled remotely by the other node.

If you do not want to use the reset function at all, click on the **Set System Controller Parameters** box to deselect (uncheck) it.

3. Click on **OK** to complete the task.

Modify a Node with `cmgr`

To modify an existing node, use the following commands:

```
modify node logical_hostname
    set hostname to hostname
    set nodeid to nodeID
    set partition_id to partitionID
    set reset_type to powerCycle
    set sysctrl_type to msc|mmsc|l2|l1_(based_on_node_hardware)
    set sysctrl_password to password
    set sysctrl_status to enabled|disabled
    set sysctrl_owner to node_sending_reset_command
    set sysctrl_device to /dev/ttyd2
    set sysctrl_owner_type to tty_device
    set is_failsafe to true|false
```

```
set is_cxfs to true|false
set weight to 0|1
add nic IP_address_or_hostname_(if_DNS)
        set heartbeat to true|false
        set ctrl_msgs to true|false
        set priority to integer
remove nic IP_address_or_hostname_(if_DNS)
```

The commands are the same as those used to define a node. You can change any of the information you specified when defining a node except the node ID. For details about the commands, see "Define a Node with cmgr", page 119.

Caution: Do not change the node ID number after the node has been defined.

Example of Partitioning

The following shows an example of partitioning an SGI Origin 3000 system:

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

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

n_preston ? set partition_id to 1
n_preston ? done

Successfully modified node n_preston
```

To perform this function with prompting, enter the following:

```
# cmgr -p
Welcome to SGI Cluster Manager Command-Line Interface

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

Hostname[optional] ? (preston.engr.sgi.com)
Is this a FailSafe node <true|false> ? (true)
Is this a CXFS node <true|false> ? (true)
Node ID[optional] ? (606)
```

```
Partition ID[optional] ? (0) 1
Reset type <powerCycle> ? (powerCycle)
Do you wish to modify system controller info[y/n]:n
Number of Network Interfaces ? (2)
NIC 1 - IP Address ? (preston)
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false> ? (true)
NIC 1 - (use network for control messages) <true|false> ? (true)
NIC 1 - Priority <1,2,...> ? (1)
NIC 2 - IP Address ? (192.168.168.1)
NIC 2 - Heartbeat HB (use network for heartbeats) <true|false> ? (true)
NIC 2 - (use network for control messages) <true|false> ? (true)
NIC 2 - Priority <1,2,...> ? (2)
Node Weight ? (1)

Successfully modified node n_preston

cmgr> show node n_preston
Logical Machine Name: n_preston
Hostname: preston.engr.sgi.com
Node Is FailSafe: true
Node Is CXFS: true
Nodeid: 606
Partition id: 1
Reset type: powerCycle
ControlNet Ipaddr: preston
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 1
ControlNet Ipaddr: 192.168.168.1
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 2
Node Weight: 1
```

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

Convert a CXFS Node to FailSafe

This section tells you how to convert a FailSafe node to also apply to CXFS.

Convert a CXFS Node to FailSafe with the GUI

This task appears on the GUI if you also have CXFS installed.

To convert an existing CXFS node (of type `CXFS`) to type `CXFS and FailSafe` or type `CXFS`, do the following:

1. Stop CXFS services on the node to be converted using the CXFS GUI. See the *CXFS Version 2 Software Installation and Administration Guide*.
2. Convert the node:
   - **Logical Name:** select the logical name of the node.
   - **Keep CXFS Settings:**
     - To convert to type `CXFS and FailSafe`, click the checkbox
     - To convert to type `FailSafe`, leave the checkbox blank
   - Click on **OK** to complete the task.

Note: If you want to rename a node, you must delete it and then define a new node.

To change other parameters, see "Modify a Node Definition with the GUI", page 126. Ensure that modifications you make are appropriate for both FailSafe and CXFS.

Convert a Node to CXFS or FailSafe with `cmgr`

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

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

For example, in normal mode:

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

cxfs6 ? set is_FailSafe to true
cxfs6 ? done
```

```
Successfully modified node cxfs6
```

For example, in prompting mode:

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

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

Successfully modified node cxfs6
```

Delete a Node

This section tells you how to delete a node.

Delete a Node with the GUI

You must remove a node from a cluster before you can delete the node from the pool. For information, see "Modify a Cluster Definition", page 141.

To delete a node, do the following:

1. **Node to Delete:** select the logical name of the node to be deleted.
2. Click on **OK** to complete the task.

Delete a Node with `cmgr`

To delete a node, use the following command:

`delete node` *hostname*

You can delete a node only if the node is not currently part of a cluster. If a cluster currently contains the node, you must first modify that cluster to remove the node from it.

For example, suppose you had a cluster named `cxfs6-8` with the following configuration:

```
cmgr> show cluster cxfs6-8
Cluster Name: cxfs6-8
Cluster Is FailSafe: true
Cluster Is CXFS: true
Cluster ID: 20
Cluster HA mode: normal
Cluster CX mode: normal
FileSystem Device Name: /dev/cxvm/dks2d72s0
FileSystem Mount Point: /mnts/fs1
FileSystem Mount Options:
FileSystem Status: enabled
FileSystem Force flag: false
        FileSystem Server Node: cxfs6
        FileSystem Server Rank: 0
        FileSystem Server Node: cxfs7
        FileSystem Server Rank: 1
FileSystem Device Name: /dev/cxvm/dks2d73s0
FileSystem Mount Point: /mnts/fs2
FileSystem Mount Options: enabled
FileSystem Status: enabled
FileSystem Force flag: false
        FileSystem Server Node: cxfs8
        FileSystem Server Rank: 0


Cluster cxfs6-8 has following 3 machine(s)
        cxfs6
        cxfs7
        cxfs8
```

To delete node cxfs8, you would do the following in prompting mode (assuming that CXFS services and FailSafe HA services have been stopped on the node):

```
cmgr> modify cluster cxfs6-8
Enter commands, when finished enter either "done" or "cancel"

Is this a FailSafe cluster <true|false> ? (true)
Is this a CXFS cluster <true|false> ? (true)
Cluster Notify Cmd [optional] ?
Cluster Notify Address [optional] ?
Cluster HA mode <normal|experimental>[optional] ? (normal)
Cluster ID ? (20)
Number of Cluster FileSystems ? (0)

Current nodes in cluster cxfs6-8:
Node - 1: cxfs6
Node - 2: cxfs7
Node - 3: cxfs8

Add nodes to or remove nodes from cluster cxfs6-8
Enter "done" when completed or "cancel" to abort

cxfs6-8 ? remove node cxfs8
cxfs6-8 ? done
Successfully modified cluster cxfs6-8

cmgr> show cluster cxfs6-8
Cluster Name: cxfs6-8
Cluster Is FailSafe: true
Cluster Is CXFS: true
Cluster ID: 20
Cluster HA mode: normal


Cluster cxfs6-8 has following 2 machine(s)
        cxfs6
        cxfs7
```

To delete cxfs8 from the pool, enter the following:

```
cmgr> delete node cxfs8
```

```
IMPORTANT: NODE cannot be deleted if it is a member of a cluster.
The LOCAL node can not be deleted if some other nodes are still defined.


Deleted machine (cxfs6).
```

Display a Node

This section tells you how to display a node.

Display a Node with the GUI

After you define nodes, you can display the following:

1. Nodes that have been defined (**Nodes in Pool**)
2. Nodes that are members of a specific cluster (**Nodes in Cluster**)
3. Attributes of a node

To view filesystems, select **View: Filesystems**.

Click on any name or icon in the tree view to see detailed status and configuration information.

Display a Node with `cmgr`

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

`show node` *nodename*

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

```
cmgr> show node cm1
Logical Machine Name: cm1
Hostname: cm1
Node Is FailSafe: true
Node is CXFS: false
Nodeid: 1
Reset type: powerCycle
System Controller: msc
System Controller status: enabled
```

```
System Controller owner: cm2
System Controller owner device: /dev/ttyd2
System Controller owner type: tty
ControlNet Ipaddr: cm1
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 0
```

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

```
show nodes in pool
```

For example:

```
cmgr> show nodes in pool

3 Machine(s) defined
        cxfs8
        cxfs6
        cxfs7
```

If you have specified a default cluster, you do not need to specify a cluster when you use this command. For example:

```
cmgr> set cluster cxfs6-8
cmgr> show nodes

Cluster cxfs6-8 has following 3 machine(s)
        cxfs6
        cxfs7
        cxfs8
```

Cluster Tasks

The *cluster* is the set of nodes in the pool that have been defined as a cluster. The cluster is identified by a simple name; this name must be unique within the pool. (For example, you cannot use the same name for the cluster and for a node.)

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

This section describes the following cluster configuration tasks:

- "Define a Cluster", page 137
- "Modify a Cluster Definition", page 141
- "Convert a CXFS Cluster to FailSafe", page 142
- "Delete a Cluster", page 143
- "Display a Cluster", page 145

Define a Cluster

This section tells you how to define a cluster.

Define a Cluster with the GUI

A *cluster* is a collection of nodes coupled to each other by a private network. A cluster is identified by a simple name. A given node may be a member of only one cluster.

To define a cluster, do the following:

1. Enter the following:
   - **Cluster Name:** the logical name of the cluster. The name can have a maximum length of 255 characters.
   - **Cluster Mode:** usually, you should set the cluster to the default `Normal` mode.

     Setting the mode to `Experimental` turns off resetting so that you can debug the cluster without causing node resets. You should only use `Experimental` mode when debugging.
   - **Notify Administrator** (of cluster and node status changes):
     - **By e-mail:** this choice requires that you specify the e-mail program (`/usr/sbin/Mail` by default) and the e-mail addresses of those to be identified. To specify multiple addresses, separate them with commas. FailSafe will send e-mail to the addresses whenever the status changes for a node or cluster. If you do not specify an address, notification will not be sent.
     - **By other command:** this choice requires that you specify the command to be run whenever the status changes for a node or cluster.

- **Never:** this choice specifies that notification is not sent.

2. Click on **OK** to complete the task. This is a long-running task that might take a few minutes to complete.

Define a Cluster with `cmgr`

When you define a cluster with `cmgr`, you define a cluster and add nodes to the cluster with the same command. For general information, see "Define a Cluster", page 137.

Use the following commands to define a cluster:

```
define cluster clustername
    set is_failsafe to true|false
    set is_cxfs to true|false
    set notify_cmd to notify_command
    set notify_addr to email_address
    set ha_mode to normal|experimental
    set cx_mode to normal|experimental
    add node node1name
    add node node2name
    ...
```

Usage notes:

- `cluster` is the logical name of the cluster. Logical names cannot begin with an underscore (_) or include any whitespace characters, and can be at most 255 characters.
- `is_failsafe` and `is_cxfs` specify the cluster type. If you are running just FailSafe, set `is_cxfs` to `false` and `is_failsafe` to `true`. If you are running a coexecution cluster, set both values to `true`.
- `notify_cmd` is the command to be run whenever the status changes for a node or cluster.
- `notify_addr` is the email address to be notified of cluster and node status changes. To specify multiple addresses, separate them with commas. FailSafe will send e-mail to the addresses whenever the status changes for a node or cluster. If you do not specify an address, notification will not be sent If you use the `notify_addr` command, you must specify the e-mail program (such as `/usr/sbin/Mail`) as the *notify_command*.

- `set ha_mode` and `set cx_mode` should normally be set to `normal`. Setting the mode to `experimental` turns off resetting so that you can debug the cluster without causing node resets. You should only use `experimental` mode when debugging. The `set cx_mode` command applies only to CXFS, and the `set ha_mode` command applies only to IRIS FailSafe.

This is a long-running task that might take a few minutes to complete. Failsafe also adds the resource types that are installed in the node to the new cluster; this process takes time.

The following shows the commands with prompting:

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

Is this a FailSafe cluster <true|false> ? true|false
Is this a CXFS cluster  <true|false> ? true|false
Cluster Notify Cmd [optional] ?
Cluster Notify Address [optional] ?
Cluster HA mode <normal|experimental> [optional] ? normal
No nodes in cluster clustername

Add nodes to or remove nodes from cluster clustername
Enter "done" when completed or "cancel" to abort

clustername ? add node node1name
clustername ? add node node2name
...
clustername ? done
Creating resource type MAC_address
Creating resource type IP_address
Creating resource type filesystem
Creating resource type volume
Successfully defined cluster clustername
```

You should set the cluster to the default `normal` mode. Setting the mode to `Experimental` turns off resetting so that you can debug the cluster without causing node resets. You should only use `Experimental` mode when debugging. However, you should never use `experimental` mode on a production cluster and should only use it if directed to by SGI customer support. SGI does not support the use of `experimental` by customers.

For example:

```
cmgr> define cluster fs6-8
Enter commands, you may enter "done" or "cancel" at any time to exit

Is this a FailSafe cluster <true|false> ? true
Is this a CXFS cluster  <true|false> ? false
Cluster Notify Cmd [optional] ?
Cluster Notify Address [optional] ?
Cluster HA mode <normal|experimental> [optional] ?

No nodes in cluster fs6-8

Add nodes to or remove nodes from cluster fs6-8
Enter "done" when completed or "cancel" to abort

fs6-8 ? add node fs6
fs6-8 ? add node fs7
fs6-8 ? add node fs8
fs6-8 ? done
Creating resource type MAC_address
Creating resource type IP_address
Creating resource type filesystem
Creating resource type volume
Successfully defined cluster fd6-8
```

To do this without prompting, enter the following:

```
cmgr> define cluster fs6-8
Enter commands, you may enter "done" or "cancel" at any time to exit

cluster fs6-8? set is_failsafe to true
cluster fs6-8? add node fs6
cluster fs6-8? add node fs7
cluster fs6-8? add node fs8
cluster fs6-8? done
Creating resource type MAC_address
Creating resource type IP_address
Creating resource type filesystem
Creating resource type volume
Successfully defined cluster fs6-8
```

Modify a Cluster Definition

This section tells you how to modify a cluster definition.

Modify a Cluster Definition with the GUI

To change how the cluster administrator is notified of changes in the cluster's state, do the following:

1. **Cluster Name:** select the name of the cluster.
2. **Cluster Mode:** usually, you should set the cluster to the default **Normal** mode. See "Define a Cluster", page 137, for information about **Experimental** mode.
3. **Notify Administrator** (of cluster and node status changes):
   - **By e-mail:** this choice requires that you specify the e-mail program (`/usr/sbin/Mail` by default) and the e-mail addresses of those to be identified. To specify multiple addresses, separate them with commas. FailSafe will send e-mail to the addresses whenever the status changes for a node or cluster. If you do not specify an address, notification will not be sent.
   - **By other command:** this choice requires that you specify the command to be run whenever the status changes for a node or cluster.
   - **Never:** this choice specifies that notification is not sent.
4. Click on **OK**.

To modify the nodes that make up a cluster, see "Add/Remove Nodes in the Cluster with the GUI", page 125.

Note: If you want to rename a cluster, you must delete it and then define a new cluster.

Modify a Cluster Definition with `cmgr`

The commands are as follows:

```
modify cluster clustername
    set is_failsafe to true|false
    set is_cxfs to true|false
    set notify_cnd to command
```

```
set notify_addr to email_address
set ha_mode to normal|experimental
set cx_mode to normal|experimental
add node node1name
add node node2name
...
remove node node1name
remove node node2name...
```

For example, to add node `newnode` to the cluster `testcluster`, enter the following:

```
cmgr> modify cluster mycluster
cluster testcluster? add node newnode
cluster testcluster? done
cmgr>
```

Convert a CXFS Cluster to FailSafe

This section tells you how to convert a CXFS cluster so that it also applies to FailSafe.

Convert a CXFS Cluster to FailSafe with the GUI

This task appears on the GUI if you also have CXFS installed.

To convert the information from an existing CXFS cluster (that is, of type `CXFS`) to create a cluster that also applies to FailSafe (that is, of type `CXFS and FailSafe`), do the following:

1. **Cluster Name:** select the name of the cluster.
2. Click on **OK** to complete the task.

The cluster will apply to both FailSafe and CXFS. To modify the nodes that make up a cluster, see "Add/Remove Nodes in the Cluster", page 125.

Note: If you want to rename a cluster, you must delete it and then define a new cluster.

Converting a CXFS Cluster to Failsafe with cmgr

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

`modify cluster` *clustername*
`set is_failsafe to` *true|false*
`set is_cxfs to` *true|false*
`set clusterid to` *clusterID*

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

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

TEST ?set is_failsafe to true
```

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

Delete a Cluster

This section tells you how to delete a cluster.

Delete a Cluster with the GUI

You cannot delete a cluster that contains nodes; you must move those nodes out of the cluster first. For information, see "Modify a Cluster Definition", page 141.

To delete a cluster, do the following:

1. **Cluster to Delete:** select the cluster name.
2. Click on **OK** to complete the task.

Delete a Cluster with cmgr

To delete a cluster, use the following command:

`delete cluster` *clustername*

However, you cannot delete a cluster that contains nodes; you must first stop HA services on the nodes and then redefine the cluster so that it no longer contains the nodes.

Example in normal mode:

```
cmgr> modify cluster fs6-8
Enter commands, when finished enter either "done" or "cancel"

fs6-8 ? remove node fs6
fs6-8 ? remove node fs7
fs6-8 ? remove node fs8
fs6-8 ? done
Successfully modified cluster fs6-8

cmgr> delete cluster fs6-8

cmgr> show clusters

cmgr>
```

Example using prompting:

```
cmgr> modify cluster fs6-8
Enter commands, you may enter "done" or "cancel" at any time to exit

Cluster Notify Cmd [optional] ?
Cluster Notify Address [optional] ?
Cluster HA mode <normal|experimental>[optional] ? (normal)

Current nodes in cluster  fs6-8:
Node - 1: fs6
Node - 2: fs7
Node - 3: fs8

Add nodes to or remove nodes from cluster fs6-8
Enter "done" when completed or "cancel" to abort
```

```
fs6-8 ? remove node fs6
fs6-8 ? remove node fs7
fs6-8 ? remove node fs8
fs6-8 ? done
Successfully modified cluster fs6-8

cmgr> delete cluster fs6-8

cmgr> show clusters

cmgr>
```

Display a Cluster

This section tells you how to display a cluster.

Display a Cluster with the GUI

The GUI provides a convenient display of a cluster and its components. From the **View** selection, you can choose elements within the cluster to examine. To view details of the cluster, click on the cluster name or icon.

The status details will appear in the item view on the right.

Displaying a Cluster with `cmgr`

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

```
show cluster clustername
show clusters
```

For example:

```
cmgr> show clusters

1 Cluster(s) defined
        nfs-cluster

cmgr> show cluster nfs-cluster
```

```
Cluster Name: nfs-cluster
Cluster Is FailSafe: true
Cluster Is CXFS: false
Cluster HA mode: normal

Cluster nfs-cluster has following 2 machine(s)
        hans2
        hans1
```

You can also use the `cluster_status` command.

Resource Type Tasks

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.

This section describes the following resource type tasks:

- "Define a Resource Type"
- "Redefine a Resource Type for a Specific Node", page 155
- "Add/Remove Dependencies for a Resource Type", page 159
- "Load a Resource Type", page 162
- "Modify a Resource Type Definition", page 162
- "Delete a Resource Type", page 167
- "Display a Resource Type", page 168

Define a Resource Type

This section describes how to define a resource type.

Define a Resource Type with the GUI

The FailSafe software includes many predefined resource types. Resource types in the cluster are created for the FailSafe plug-ins installed in the node using the `/usr/cluster/bin/cdb-create-resource-type` script. Resource types that

were not created when the cluster was configured can be added later using the `resource type install` command, as described in "Load a Resource Type with the GUI", page 162.

If these predefined resource types fit the application you want to make into an HA service, you can reuse them. If none fits, you can define additional resource types. Complete information on defining resource types is provided in the *IRIS FailSafe Version 2 Programmer's Guide*. This manual provides a summary of that information.

To define a new resource type, do the following:

1. **Resource Type:** specify the name of the new resource type, with a maximum length of 255 characters.

   Click **Next** to move to the next screen.

2. Specify settings for required actions (time values are in milliseconds):

   - **Start/Stop Order:** order of performing the action scripts for resources of this type in relation to resources of other types:
     - Resources are started in the increasing order of this value.
     - Resources are stopped in the decreasing order of this value.

     See the *IRIS FailSafe Version 2 Programmer's Guide* for a full description of the order ranges available.
   - **Start Timeout:** the maximum duration for starting a resource of this type.
   - **Stop Timeout:** the maximum duration for stopping a resource of this type.
   - **Exclusive Timeout:** the maximum duration for verifying that a resource of this type is not already running.
   - **Monitoring Timeout:** the maximum duration for monitoring a resource of this type.
   - **Monitor Interval:** the amount of time between successive executions of the `monitor` action script; this is only valid for the `monitor` action script.
   - **Monitoring Start Time:** the amount of time between starting a resource and beginning monitoring of that resource.

     Click **Next** to move to the next screen.

3. Specify settings for optional actions as needed:

- **Restart Enabled:** check the box to enable restarting of the resource. You should enable restart if you want a resource of this type to automatically be restarted on the current node after a monitoring failure. Enabling restart can decrease application downtime.

  For example, suppose FailSafe detects that a resource's monitor action has failed:

  – If restart is disabled, FailSafe will immediately attempt to move the whole group to another node in the failover domain. The application will be down until the entire group is failed over.

  – If restart is enabled, FailSafe will attempt to restart the resource on the current node where the rest of the resource group is running. If this succeeds, the resource group will be made available as soon as the resource restarts; if this fails, only then will FailSafe attempt to move the whole group to another node in the failover domain.

  The local restart flag enables local failover:

  – If local restart is enabled and the resource monitor script fails, SRMD executes the restart script for the resource.

  – If the restart script is successful, SRMD continues to monitor the resource.

  – If the restart script fails or the restart count is exhausted, SRMD sends a resource group monitoring error to FSD. FSD itself is not involved in local failover.

  When a resource is restarted, all other resources in the resource group are not restarted. It is not possible to do a local restart of a resource using the GUI or `cmgr`.

  If you find that you need to reset the restart counter for a resource type, you can put the resource group in maintenance mode and remove it from maintenance mode. This process will restart counters for all resources in the resource group. For information on putting a resource group in maintenance mode, see "Suspend and Resume Monitoring of a Resource Group", page 255.

- **Restart Timeout:** the maximum amount of time to wait before restarting the resource after a monitoring failure occurs.

- **Restart Count:** the maximum number of attempts that FailSafe will make to restart a resource of this type on the current node. Enter an integer greater than zero.

- **Probe Enabled:** check if you want FailSafe to verify that a resource of this type is configured on a node.
- **Probe Timeout:** the maximum amount of time for FailSafe to attempt to verify that a resource of this type is configured on a node.

4. Change settings for type-specific attributes: specify any attributes specific to the resource type. You must provide the following for each attribute:

   - **Attribute key:** name of the attribute
   - **Data Type:** select either **String** or **Integer**
   - **Default Value:** optionally, provide a default value

   For example, NFS requires the following attributes:

   - `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
     ```

   Click **Add** to add the attribute, and repeat as necessary for other attributes.

5. Click on **OK** to complete the task.

Define a Resource Type with `cmgr`

Use the following commands:

```
define resource_type RT_name on node nodename [in cluster clustername]

define resource_type RT_name [in cluster clustername]
   set order to start/stop_order_number
```

```
set restart_mode to 0|1
set restart_count to number_of_attempts
add action action_script_name
        set exec_time to exclusive_timeout
        set monitor_interval to monitor_interval
        set monitor_time to monitor_time
add type_attribute type-specific_attribute_name
        set data_type to string|integer
        set default_value to default
add dependency RT_name
remove action action_script_name
remove type_attribute type-specific_attribute_name
remove dependency dependency_name
```

Usage notes:

- `resource_type` is the name of the resource type to be defined, with a maximum length of 255 characters.
- `order` is the order of performing the action scripts for resources of this type in relation to resources of other types:
  - Resources are started in the increasing order of this value
  - Resources are stopped in the decreasing order of this value

  See the *IRIS FailSafe Version 2 Programmer's Guide* for a full description of the order ranges available.
- `restart_mode` is as follows:
  - 0 = Do not restart on monitoring failures (disable restart)
  - 1 = Restart a fixed number of times (enable restart)

  You should enable restart if you want a resource of this type to automatically be restarted on the current node after a monitoring failure. Enabling restart can decrease application downtime.
- `restart_count` is the maximum number of attempts that FailSafe will make to restart a resource of this type on the current node. Enter an integer greater than zero.

- `action` is the name of the action script (`exclusive`, `start`, `stop`, `monitor`, or `restart`). For more information, see "Action Scripts", page 13. The following time values are in milliseconds:
  - `exec_time` is the maximum time for executing the action script
  - `monitor_interval` is the amount of time between successive executions of the `monitor` action script (this is valid only for the `monitor` action script)
  - `monitor_time` is the amount of time between starting a resource and beginning monitoring of that resource
- `type_attribute` is a type-specific attribute
  - `data_type` is either `string` or `integer`
  - `default_value` is the default value for the attribute
- `dependency` adds a dependency upon the specified resource type (*RT_name*)

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.

For an example in normal mode, see the `create resource type` template for the `cmgr` command in the following file:

```
/var/cluster/cmgr-templates/cmgr-create-resource_type
```

Note: The following example in prompting mode only shows the prompts and answers for two action scripts (`start` and `stop`) for a new resource type named `newresourcetype`.

```
cmgr> define resource_type newresourcetype

(Enter "cancel" at any time to abort)

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

DEFINE RESOURCE TYPE OPTIONS
```

```
        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:1

No current resource type actions

Action name ? start
Executable timeout (in milliseconds) ? 40000

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:1

Current resource type actions:
        start

Action name stop
Executable timeout?  (in milliseconds) 40000

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
```

```
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:3

No current type specific attributes

Type Specific Attribute ? integer-att
Datatype ? integer
Default value[optional] ? 33

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:3

Current type specific attributes:
        Type Specific Attribute - 1: integer-att

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

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
```

```
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:5

No current resource type dependencies

Dependency name ? filesystem

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:7

Current resource type actions:
        Action - 1: start
        Action - 2: stop

Current type specific attributes:
        Type Specific Attribute - 1: integer-att
        Type Specific Attribute - 2: string-att

No current resource type dependencies

Resource dependencies to be added:
        Resource dependency - 1: filesystem

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
```

```
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:9
Successfully defined resource_type newresourcetype

cmgr> show resource_types

template
MAC_address
newresourcetype
IP_address
filesystem
volume

cmgr> exit
#
```

Redefine a Resource Type for a Specific Node

This section describes how to define a resource type that applies to a specific node.

Redefine a Resource Type for a Specific Node with the GUI

This task lets you take an existing clusterwide resource type and redefine it for use on the local node.

A resource type that is redefined for a specific node overrides a clusterwide definition with the same name; this allows an individual node to override global settings from a clusterwide resource type definition. You can use this feature if you want to have different script timeouts for a node or you want to restart a resource on only one node in the cluster.

For example, the `IP_address` resource has local restart enabled by default. If you would like to have an `IP_address` type without local restart for a particular node, you can make a copy of the `IP_address` clusterwide resource type with all of the parameters the same except for restart mode, which you set to 0.

Do the following:

1. **Cluster Node:** the name of the local node is filled in for you.
2. **Clusterwide Resource Type:** select the name of the resource type you want to redefine for the local node.

   Click **Next** to move to the next screen.
3. Change settings for required actions as needed (time values are in milliseconds):
   - **Start/Stop Order:** order of performing the action scripts for resources of this type in relation to resources of other types:
     - Resources are started in the increasing order of this value
     - Resources are stopped in the decreasing order of this value

     See the *IRIS FailSafe Version 2 Programmer's Guide* for a full description of the order ranges available.
   - **Start Timeout:** the maximum duration for starting a resource of this type.
   - **Stop Timeout:** the maximum duration for stopping a resource of this type.
   - **Exclusive Timeout:** the maximum duration for verifying that a resource of this type is not already running.
   - **Monitoring Timeout:** the maximum duration for monitoring a resource of this type.
   - **Monitor Interval:** the amount of time between successive executions of the `monitor` action script; this is only valid for the `monitor` action script.
   - **Monitoring Start Time:** the amount of time between starting a resource and beginning monitoring of that resource.

     Click **Next** to move to the next screen.
4. Change settings for optional actions as needed:
   - **Restart Enabled:** check the box to enable restarting of the resource. You should enable restart if you want a resource of this type to automatically be restarted on the current node after a monitoring failure. Enabling restart can decrease application downtime.

For example, suppose FailSafe detects that a resource's monitor action has failed:

- If restart is disabled, FailSafe will immediately attempt to move the whole group to another node in the failover domain. The application will be down until the entire group is failed over.
- If restart is enabled, FailSafe will attempt to restart the resource on the current node where the rest of the resource group is running. If this succeeds, the resource group will be made available as soon as the resource restarts; if this fails, only then will FailSafe attempt to move the whole group to another node in the failover domain.

The local restart flag enables local failover:

- If local restart is enabled and the resource monitor script fails, SRMD executes the restart script for the resource.
- If the restart script is successful, SRMD continues to monitor the resource.
- If the restart script fails or the restart count is exhausted, SRMD sends a resource group monitoring error to FSD. FSD itself is not involved in local failover.

When a resource is restarted, all other resources in the resource group are not restarted. It is not possible to do a local restart of a resource using the GUI or `cmgr`.

If you find that you need to reset the restart counter for a resource type, you can put the resource group in maintenance mode and remove it from maintenance mode. This process will restart counters for all resources in the resource group. For information on putting a resource group in maintenance mode, see "Suspend and Resume Monitoring of a Resource Group", page 255.

- **Restart Timeout:** the maximum amount of time to wait before restarting the resource after a monitoring failure occurs.
- **Restart Count:** the maximum number of attempts that FailSafe will make to restart a resource of this type on the current node. Enter an integer greater than zero.
- **Probe Enabled:** check if you want FailSafe to verify that a resource of this type is configured on a node.

- **Probe Timeout:** the maximum amount of time for FailSafe to attempt to verify that a resource of this type is configured on a node.

5. Change settings for type-specific attributes; specify any attributes specific to the resource type. You must provide the following for each attribute:

   - **Attribute key:** specify the name of the attribute
   - **Data Type:** select either **String** or **Integer**
   - **Default Value:** optionally, provide a default value

   For example, NFS requires the following attributes:

   - `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
     ```

   Click **Add** to add the attribute, and repeat as necessary for other attributes.

6. Click on **OK** to complete the task.

Defining a Node-Specific Resource Type with `cmgr`

With `cmgr`, you redefine a node-specific resource type similar to defining a clusterwide resource type, except that you specify a node on the command line. Use the following command to define a node-specific resource type:

`define resource_type` *RT_name* `on node` *nodename* `[in cluster` *clustername*`]`

Add/Remove Dependencies for a Resource Type

This section describes how to add dependencies to a resource type.

Add/Remove Dependencies for a Resource Type with the GUI

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`. Figure 5-1 shows these dependencies.

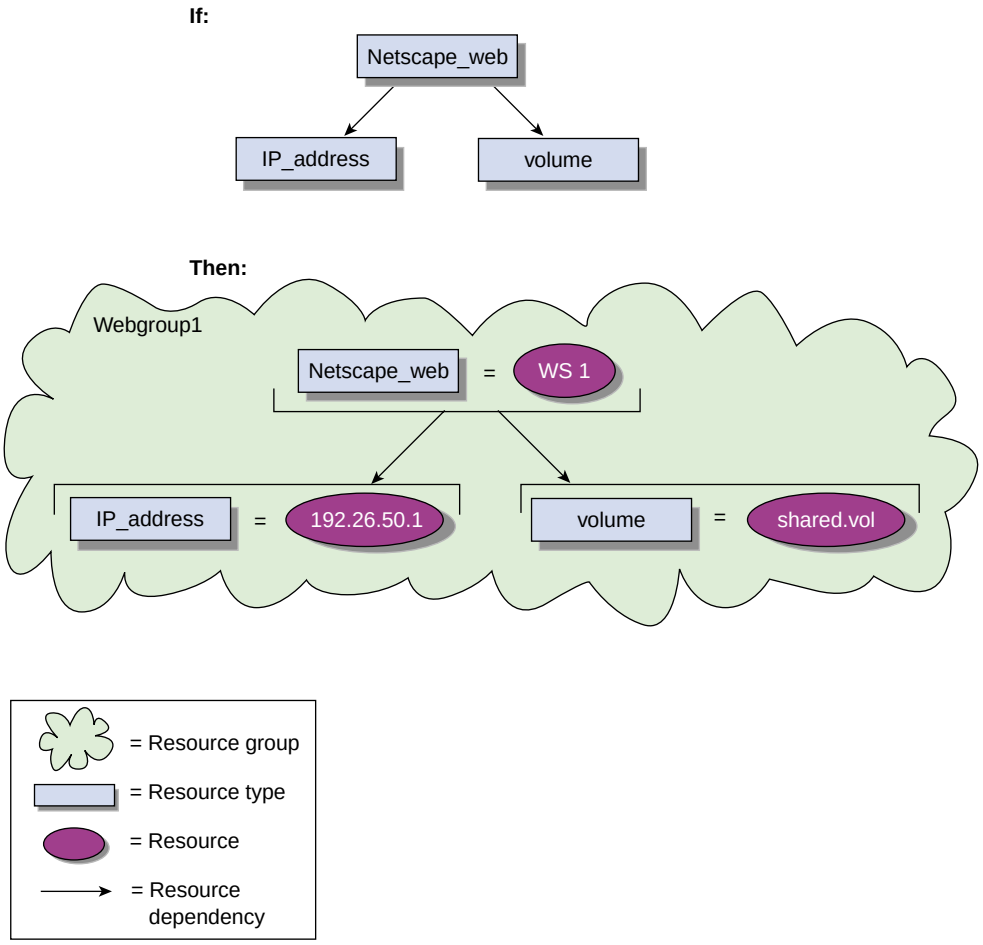

Figure 5-1 Dependencies

Enter the following information:

1. **Resource Type:** select the resource type.
2. **Dependency Type:** select the dependency type. Click **Add** to add the dependency to the list, click **Delete** to remove the dependency from the list.
3. Click **OK** to complete the task.

Add/Remove Dependencies for a Resource Type with cmgr

When using `cmgr`, you add or remove dependencies when you define or modify the resource type.

For example, suppose the `NFS` resource type in `nfs-cluster` has a resource type dependency on `filesystem` resource type. To change the `NFS` resource type to have a dependency on the `IP_address` resource type instead (and not on `filesystem`), do the following:

```
cmgr> show resource_type NFS in cluster nfs-cluster

Name: NFS
Predefined: true
....

Resource type dependencies
        filesystem

cmgr> modify resource_type NFS in cluster nfs-cluster
Enter commands, when finished enter either "done" or "cancel"

resource_type NFS ? remove dependency filesystem
resource_type NFS ? add dependency IP_address
resource_type NFS ? done
Successfully modified resource_type NFS

cmgr> show resource_type NFS in cluster nfs-cluster

Name: NFS
Predefined: true
....

Resource type dependencies
        IP_address
```

Load a Resource Type

This section describes how to install (load) a resource type.

Load a Resource Type with the GUI

When you define a cluster, FailSafe installs a set of resource type definitions that you can use; these definitions include default values. If you need to install additional, standard SGI 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 loading cannot already exist on the cluster.

Load a Resource Type with `cmgr`

Use the following command to install a resource type on a cluster:

`install resource_type` *RT_name* `[in cluster` *clustername*`]`

Modify a Resource Type Definition

This section describes how to modify a resource type.

Modify a Resource Type with the GUI

The process of modifying a resource type is similar to the process of defining a resource type.

Enter the following (time values are in milliseconds):

1. **Resource Type:** select the name of the resource type to be modified.

   Click **Next** to move to the next screen. The current settings for each field will be filled in for you.

2. **Start/Stop Order:** order of performing the action scripts for resources of this type in relation to resources of other types:

   - Resources are started in the increasing order of this value.
   - Resources are stopped in the decreasing order of this value.

See the *IRIS FailSafe Version 2 Programmer's Guide* for a full description of the order ranges available.

3. **Start Timeout:** the maximum duration for starting a resource of this type.
4. **Stop Timeout:** the maximum duration for stopping a resource of this type.
5. **Exclusive Timeout:** the maximum duration for verifying that a resource of this type is not already running.
6. **Monitoring Timeout:** the maximum duration for monitoring a resource of this type.
7. **Monitor Interval:** the amount of time between successive executions of the `monitor` action script; this is valid only for the `monitor` action script.
8. **Monitoring Start Time:** the amount of time between starting a resource and beginning monitoring of that resource.

   Click **Next** to move to the next screen.

9. **Restart Enabled:** check the box to enable restarting of the resource. You should enable restart if you want a resource of this type to automatically be restarted on the current node after a monitoring failure. Enabling restart can decrease application downtime.

   For example, suppose FailSafe detects that a resource's monitor action has failed:

   - If restart is disabled, FailSafe will immediately attempt to move the whole group to another node in the failover domain. The application will be down until the entire group is failed over.
   - If restart is enabled, FailSafe will attempt to restart the resource on the current node where the rest of the resource group is running. If this succeeds, the resource group will be made available as soon as the resource restarts; if this fails, only then will FailSafe attempt to move the whole group to another node in the failover domain.

   The local restart flag enables local failover:

   - If local restart is enabled and the resource monitor script fails, SRMD executes the restart script for the resource.
   - If the restart script is successful, SRMD continues to monitor the resource.

- If the restart script fails or the restart count is exhausted, SRMD sends a resource group monitoring error to FSD. FSD itself is not involved in local failover.

When a resource is restarted, all other resources in the resource group are not restarted. It is not possible to do a local restart of a resource using the GUI or `cmgr`.

If you find that you need to reset the restart counter for a resource type, you can put the resource group in maintenance mode and remove it from maintenance mode. This process will restart counters for all resources in the resource group. For information on putting a resource group in maintenance mode, see "Suspend and Resume Monitoring of a Resource Group", page 255.

10. **Restart Timeout:** the maximum amount of time to wait before restarting the resource after a monitoring failure occurs.
11. **Restart Count:** the maximum number of attempts that FailSafe will make to restart a resource of this type on the current node. Enter an integer greater than zero.
12. **Probe Enabled:** check if you want FailSafe to verify that a resource of this type is configured on a node.
13. **Probe Timeout:** the maximum amount of time for FailSafe to attempt to verify that a resource of this type is configured on a node.
14. **Type-Specific Attributes:** specify new attributes that are specific to the resource type, or modify an existing attribute by selecting its name. You must provide the following for each attribute:
    - **Attribute key:** specify the name of the attribute
    - **Data Type:** select either **String** or **Integer**
    - **Default Value:** *(optional)* provide a default value for the attribute

---

Note: You cannot modify the type-specific attributes if there are any existing resources of this type.

Click **Add** to add the attribute or **Modify** to modify the attribute, and repeat as necessary for other attributes. Click **OK** to complete the definition.

Modify a Resource Type with cmgr

Use the following commands to modify a resource type:

```
modify resource_type RT_name [in cluster clustername]
   set order to start/stop_order_number
   set restart_mode to 0|1
   set restart_count to number_of_attempts
   add action action_script_name
           set exec_time to exclusive_timeout
           set monitor_interval to monitor_interval
           set monitor_time to monitor_time
   modify action action_script_name
           set exec_time to exclusive_timeout
           set monitor_interval to monitor_interval
           set monitor_time to monitor_time
   add type_attribute type-specific_attribute_name
           set data_type to string|integer
           set default_value to default
   add dependency RT_name
   remove action action_script_name
   remove type_attribute type-specific_attribute_name
   remove dependency dependency_name
```

You modify a resource type using the same commands you use to define a resource type. See "Define a New Resource", page 169.

You can display the current values of the resource type timeouts, allowing you to modify any of the action timeouts.

The following example shows how to increase the `statd` resource type monitor executable timeout from 40 seconds to 60 seconds.

```
#cmgr> modify resource_type statd in cluster test-cluster
Enter commands, when finished enter either "done" or "cancel"

resource_type statd ? modify action monitor
Enter action parameters, when finished enter "done" or "cancel"

Current action monitor parameters:
        exec_time : 40000ms
        monitor_interval : 20000ms
        monitor_time : 50000ms
```

```
Action - monitor ? set exec_time to 60000
Action - monitor ? done
resource_type statd ? done
Successfully modified resource_type statd
```

The following examples show how to modify the resource type timeouts in prompt mode.

```
#cmgr> modify resource_type statd in cluster test-cluster


(Enter "cancel" at any time to abort)

Node[optional] ?
Order ? (411)
Restart Mode ? (0)

MODIFY RESOURCE TYPE OPTIONS

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:0
Current resource type actions:
        stop
        exclusive
        start
        restart
        monitor

Action name ? monitor
Executable timeout (in milliseconds) ? (40000ms) 60000
Monitoring Interval (in milliseconds) ? (20000ms)
```

```
Start Monitoring Time (in milliseconds) ? (50000ms)

        0) Modify Action Script.
        1) Add Action Script.
        2) Remove Action Script.
        3) Add Type Specific Attribute.
        4) Remove Type Specific Attribute.
        5) Add Dependency.
        6) Remove Dependency.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:9
Successfully modified resource_type statd
```

Delete a Resource Type

This section describes how to delete a resource type.

Delete a Resource Type with the GUI

To delete a resource type with the GUI, enter the following:

1. **Resource Type to Delete:** select the name of the resource type that you want to delete.

   Note: If you select a resource type that has been redefined for the local node, that special definition of the resource type will be deleted and the clusterwide resource type will used instead.

   You cannot delete a clusterwide resource type if there are any resources of that type.

   ---

2. Click on **OK** to complete the task.

Delete a Resource Type with cmgr

Use the following command to delete a resource type:

```
delete resource_type RT_name [in cluster clustername]
```

Display a Resource Type

This section describes how to display resource types.

Displaying Resource Types with the GUI

Select **View: Resource Types**. You can then click on any of the resource type icons in the tree view to examine the parameters of the resource type.

Display Resource Types with cmgr

Use the following commands to view resource types:

1. To view the parameters of a defined resource type (*RT_name*):

   ```
   show resource_type RT_name [in cluster clustername]
   ```

2. To view all of the defined resource types:

   ```
   show resource_types [in cluster clustername]
   ```

3. To view all of the defined resource types that have been installed:

   ```
   show resource_types installed
   ```

Resource Tasks

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.

This section describes the following resource tasks:

- "Define a New Resource", page 169
- "Redefine a Resource for a Specific Node", page 175

- "Add/Remove Dependencies for a Resource Definition", page 176
- "Modify a Resource Definition", page 179
- "Delete a Resource", page 180
- "Display a Resource", page 181

Define a New Resource

This section describes how to define a new resource.

Define a New Resource with the GUI

Resources are identified by a *resource type* and a *resource name*. A resource name identifies a specific instance of a resource type. All of the resources in a given resource type can be handled in the same way for the purposes of *failover*.

By default, resources apply clusterwide. However, you can use the **Redefine a Resource for a Specific Node** task to apply the resource to just the local node; see "Redefine a Resource for a Specific Node", page 175.

Provide appropriate values for the following:

1. **Resource Type:** the name of the resource type.

   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 clusterwide resource type definition of the same name; this allows an individual node to override global settings from a clusterwide resource type definition.

   The type of resource to define. The FailSafe system includes pre-defined resource types, listed in the GUI; also see "Software Layers", page 307. You can define your own resource type as well.

   The FailSafe software includes many predefined resource types. If these types fit the application you want to make into a highly available service, you can reuse them. If none fit, you can define additional resource types.

2. **Resource:** name of the resource to define, with a maximum length of 255 characters, that does not begin with an underscore. A resource is a single physical or logical entity that provides a service to clients or other resources. Examples include a single disk volume, a particular network highly available (HA) IP

address, or a specific application such as a web server. Particular resource types may have other naming requirements; see the sections below.

You can define up to 100 resources in a FailSafe configuration.

Click **Next** to move to the next screen.

3. **Type-specific attributes:** enter the attributes that apply to this resource. The following sections describe attributes for each resource type provided in the base FailSafe release; other attributes are available with FailSafe plug-in releases and are described in the documentation supplied with those releases. You can specify attributes for new resource types you create.
4. Click on **OK** to complete the task.

CXFS Attributes

The `CXFS` resource is the mount point of the CXFS filesystem, such as `/shared_CXFS`. The **relocate-mds** field specifies whether the metadata server of the CXFS filesystem should be relocated (`true`) or not (`false`).

`filesystem` 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 that was created on an XLV volume `xlv_vol` and mounted on the `/shared1` directory will have the resource name `/shared1`.

When you define a filesystem, you must specify the following parameters:

- **volume-name:** 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.
- **mount-options:**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).

- **monitoring-level:** 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.

`IP_address` Attributes

The `IP_address` resources are the IP addresses used by clients to access the HA services within the resource group. These HA IP addresses are moved from one node to another along with the other resources in the resource group when a failure is detected.

You specify the resource name of an `IP_address` resource in dot (".") 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 HA IP address you define as a FailSafe resource must not be the same as the IP address of a node hostname or the IP address of a node's control network.

When you define an `IP_address` resource, you can optionally specify the following parameters. If you specify any of these parameters, you must specify all of them.

- **NetworkMask:** the network mask of the HA IP address.
- **interfaces:** a comma-separated list of interfaces on which the HA IP address can be configured. This ordered list is a superset of all the interfaces on all nodes where this HA IP address might be allocated. You can specify multiple interfaces to configure local restart of the HA IP address, if those interfaces are on the same node.

  The order of the list of interfaces determines the priority order for determining which HA IP address will be used for local restarts of the node.
- **BroadcastAddress:** the broadcast address for the HA IP address

`MAC_address` Attributes

The MAC address is the link level 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.

You must specify the following attribute: **interface-name:** the interface that has to be reMAC-ed.

Only Ethernet interfaces are capable of undergoing the reMAC process.

`volume` Attributes

The `volume` resource type 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:

- **devname-group:** the group name of the XLV device file. The `sys` group is the default group name for XLV device files.
- **devname-owner:** the user name (login name) of the owner of the XLV device file. `root` is the default owner for XLV device files.
- **devname-mode:** the device file permissions, specified in octal notation. 600 mode is the default value for XLV device file permissions.

Define a New Resource with `cmgr`

Use the following command to define a clusterwide resource:

```
define resource resourcename [of resource_type RT_name] [in cluster clustername]
    set key to attribute_value
    add dependency dependencyname of type RT_name
    remove dependency dependencyname of type RT_name
```

Usage notes:

- `set` *key* specifies the name of the attribute, and *attribute_value* sets its value
- `add dependency` adds a dependency of the specified resource type (*RT_name*)
- `remove dependency` deletes a dependency of the specified resource type

When you use this command to define a resource, you define a clusterwide resource that is not specific to a node.

The legal values for `set` *key* `to` *attribute_value* will depend on the type of resource you are defining, as described in "Define a New Resource", page 169. For detailed information on how to determine the format for defining resource attributes, see "Specify Resource Attributes with `cmgr`", page 173.

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

For example:

```
cmgr> define resource /hafs1/nfs/statmon of resource_type statd_unlimited in cluster nfs-cluster
resource /hafs1/nfs/statmon? set ExportPoint to /hafs1/subdir
resource /hafs1/nfs/statmon? done
```

The following section of a `cmgr` script defines a resource of resource type `statd_unlimited`:

```
define resource /hafs1/nfs/statmon of resource_type statd_unlimited in cluster nfs-cluster
        set ExportPoint to /hafs1/subdir
done
```

Specify Resource Attributes with `cmgr`

To see the format in which you can specify the user-specific attributes that you must set for a particular resource type, you can enter the following command to see the full definition of that resource type:

`show resource_type` *RT_name* `[in cluster` *clustername*`]`

For example, to see the attributes you define for a resource of resource type `volumes`, enter the following command:

```
cmgr> show resource_type volume in cluster test-cluster
```

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:

- `devname-group` specifies the group ID of the XLV device file
- `devname_owner` specifies the owner of the XLV device file
- `devname_mode` specifies the device file permissions

For example, to set the group ID to `sys` for a resource name `A`, enter the following command:

```
resource A? set devname-group to sys
```

Table 5-2 summarizes the attributes you specify for the predefined FailSafe resource types with the `set` *key* `to` *attribute_value* command.

Table 5-2 Resource Type Attributes

Resource Type	Attribute	Description
`CXFS`	`relocate-mds`	Specifies if the metadata server of the CXFS filesystem should be relocated or not. (The name of a CXFS resource is the mount point of the CXFS filesystem. For example, `/shared_CXFS`.)
`filesystem`	`volume-name`	Specifies the name of the `xlv` volume associated with the filesystem.

Resource Type	Attribute	Description
	`mount-options`	Specifies the mount options to be used for mounting the filesystem.
`IP_address`	`NetworkMask`	Specifies the subnet mask of the IP address.
	`interfaces`	Specifies a comma-separated list of interfaces on which the IP address can be configured.
	`BroadcastAddress`	Specifies the broadcast address for the IP address.
`MAC_address`	`interface-name`	Specifies the name of the interface that has to be re-MACed.
`volume`	`devname-group`	Specifies the group ID of the `xlv` device file.
	`devname_owner`	Specifies the owner of the `xlv` device.
	`devname_mode`	Specifies device file permissions.

Redefine a Resource for a Specific Node

This section describes redefining a resource for a specific node.

Redefine a Resource for a Specific Node with the GUI

You can redefine an existing resource for a specific node from that node (the local node) only. Only existing clusterwide resources can be redefined.

You may want to use this feature when you configure heterogeneous clusters for an `IP_address` resource. For example, the resource `192.26.50.2` of type `IP_address` can be configured on a Gigabit Ethernet interface `eg0` on a server and on a 100baseT interface `ef0` on another server. The clusterwide resource definition for `192.26.50.2` will have the `interfaces` field set to `ef0` and the node-specific definition for the first node will have `eg0` as the `interfaces` field.

Provide appropriate values for the following:

1. **Cluster Node:** the name of the node on which the GUI is currently running, which is provided for you. You can only redefine a resource for this node. To redefine a resource for a different node, you must invoke the GUI on that node.

Caution: You should only make changes from one instance of the GUI at any given time in the cluster. Changes made by a second GUI instance — a second invocation of `fstask` — may overwrite changes made by the first instance, because different GUI instances are updated independently at different times. In time, however, independent GUI instances will provide the same information.) However, multiple windows accessed via the **File** menu are all part of a single GUI instance; you can make changes from any of these windows.

2. **Resource Type:** select the resource type.
3. **Clusterwide Resource:** name of the resource that you want to redefined for this node.

   Click **Next** to move to the next screen.
4. **Type-specific attributes:** change the information for each attribute as needed. For information about each attribute, see "Define a New Resource with the GUI", page 169.
5. Click on **OK** to complete the task.

Redefine a Resource for a Specific Node with `cmgr`

You can use `cmgr` 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 command to define a node-specific resource:

`define resource` *resourcename* `of resource_type` *RT_name* `on node` *nodename* `[in cluster` *clustername*`]`

If you have specified a default cluster, you do not need to specify a cluster in this command because `cmgr` will use the default.

Add/Remove Dependencies for a Resource Definition

This section describes how to add and remove dependencies for a resource.

Add/Remove Dependencies for a Resource Definition with the GUI

A 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 other resources. For example, a Web server may depend on a both an HA IP address and a filesystem. In turn, a filesystem may depend on a volume. This is shown in Figure 5-2.

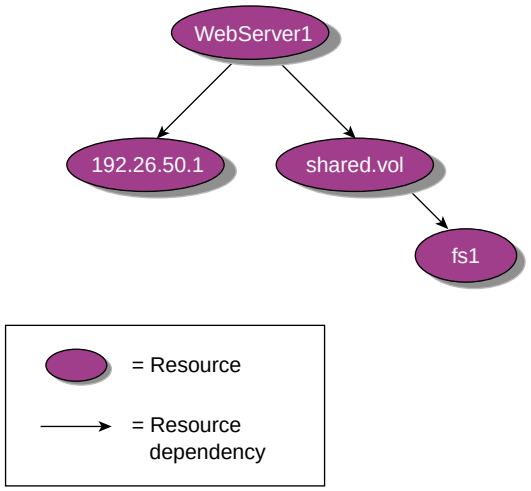

Figure 5-2 Example of Resource Dependency

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. This is shown in Figure 5-3.

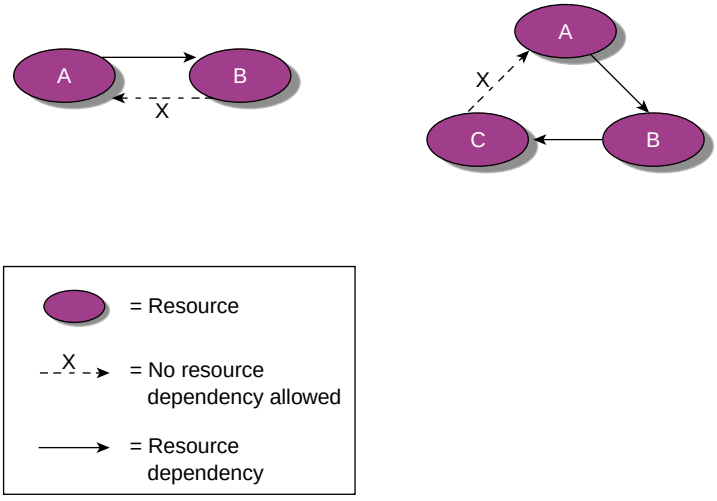

Figure 5-3 Mutual Dependency of Resources Is Not Allowed

Provide appropriate values for the following:

1. **Resource Type:** select the name of the resource type.
2. **Resource:** select the resource name.
3. **Dependency Type:** select the resource type to be added to or deleted from the dependency list.
4. **Dependency Name:** select the resource name to be added to or deleted from the dependency list. Click **Add** to add the displayed type and name to the list.
5. Click on **OK** to complete the task.

Add/Remove Dependencies for a Resource Definition with `cmgr`

To add or remove dependencies for a resource definition, use the `modify resource` command. For example:

```
cmgr> modify resource /hafs1/expdir of resource_type NFS in cluster nfs-cluster
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to modify with set command:
```

```
Type Specific Attribute - 1: export-info
Type Specific Attribute - 2: filesystem


Resource type dependencies to add or remove:

Resource dependency - 1: /hafs1         type: filesystem


resource /hafs1/expdir ? add dependency 100.102.10.101 of type IP_address
resource /hafs1/expdir ? done
Successfully modified resource /hafs1/expdir
```

Modify a Resource Definition

This section describes how to modify a resource definition.

Modify a Resource Definition with the GUI

You can modify only the type-specific attributes for a resource. You cannot rename a resource once it has been defined; to rename a resource, you must delete it and define the new resource.

Provide appropriate values for the following:

1. **Resource Type:** select the name of the resource type.
2. **Resource:** select the name of resource to be modified.

   Click **Next** to move to the next screen.
3. **Type-specific attributes:** modify the attributes as needed. For information about attributes, see "Define a New Resource with the GUI", page 169.
4. Click on **OK** to complete the task.

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.

Modify a Resource Definition with cmgr

Use the following commands to modify a resource:

```
modify resource resourcename [of resource_type RT_name] on node nodename [in cluster clustername]

modify resource resourcename [of resource_type RT_name] [in cluster clustername]
    set key to attribute_value
    add dependency dependencyname of type typename
    remove dependency dependencyname of type typename
```

You modify a resource using the same commands you use to define a resource. See "Define a New Resource", page 169.

Delete a Resource

This section describes how to delete a resource.

Delete a Resource with the GUI

A resource may not be deleted if it is part of a resource group. See "Modify a Resource Group Definition", page 195.

To delete a resource, provide the following:

1. **Resource Type:** select the resource type
2. **Resource to Delete:** select the name of the resource to be deleted
3. Click on **OK** to complete the task

If you select a resource that has been redefined for the node to which the GUI is connected, the delete operation will delete the redefined resource definition and also put into effect the clusterwide resource definition.

If you select a clusterwide resource definition, the delete operation will delete this definition and make it unavailable for use in a resource group. Deleting a clusterwide resource definition will fail if the resource is part of any resource group.

Delete a Resource with cmgr

Use the following command to delete a resource definition:

```
delete resource resourcename of resource_type RT_type [in cluster clustername]
```

Display a Resource

You can display the following:

- Attributes of a particular defined resource
- All of the defined resources in a specified resource group
- All the defined resources of a specified resource type

Caution: Only `root` can make changes to the cluster database. However, any user can use the GUI to view database information; therefore, you should not include any sensitive information in the cluster database.

Display a Resource with the GUI

The GUI provides a convenient display of resources through the tree view. Select **View: Resources in Groups** to see all defined resources. The status of these resources will be shown in the icon (grey indicates offline). Alternately, you can select **View: Resources by Type** or **View: Resources owned by Nodes**.

Display a Resource with cmgr

Use the following commands to display a resource:

- To view the parameters of a single resource:

  ```
  show resource resourcename of resource_type RT_name
  ```

- To view all of the defined resources in a resource group:

  ```
  show resources in resource_group RG_name [in cluster clustername]
  ```

- To view all of the defined resources of a particular resource type in a specified cluster:

  ```
  show resources of resource_type RT_name [in cluster clustername]
  ```

Failover Policy Tasks

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

- Failover domain
- Failover attributes
- Failover script

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

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

This section describes the following failover policy tasks:

- "Define a Failover Policy"
- "Modify a Failover Policy Definition", page 188
- "Delete a Failover Policy", page 191
- "Display a Failover Policy", page 192

Define a Failover Policy

This section describes how to define a failover policy.

Define a Failover Policy with the GUI

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, provide the following information:

1. **Failover Policy:** enter the name of the failover policy, with a maximum length of 63 characters, which must be unique within the pool.
2. **Script:** select the name of an existing failover script. generates the run-time failover domain and returns it to the FailSafe process. The FailSafe process applies the failover attributes and then selects the first node in the returned failover domain that is also in the current FailSafe membership:

- `ordered` is provided with the FailSafe release. The `ordered` script never changes the initial failover domain; when using this script, the initial and run-time failover domains are equivalent.
- `round-robin` is also provided with the FailSafe release. The `round-robin` script selects the resource group owner in a round-robin (circular) fashion. This policy can be used for resource groups that can be run in any node in the cluster.

Failover scripts are stored in the `/var/clusters/ha/policies` directory. If the `ordered` script does not meet your needs, you can define a new failover script and place it in the `/var/clusters/ha/policies` directory. When you are using the FailSafe GUI, the GUI automatically detects your script and presents it to you as a choice for you to use. You can configure the cluster database to use your new failover script for the required resource groups. For information on defining failover scripts, see the *IRIS FailSafe Version 2 Programmer's Guide*.

3. **Failback:** choose the name of the failover attribute.

   A *failover attribute* is a value that is passed to the failover script and used by FailSafe for the purpose of modifying the run-time failover domain used for a specific resource group.

   You can specify the following classes of failover attributes:

   - Required attributes: either `Auto_Failback` or `Controlled_Failback` (mutually exclusive)
   - Optional attributes:
     - `Auto_Recovery` or `InPlace_Recovery` (mutually exclusive)
     - `Critical_RG`
     - `Node_Failures_Only`

   Note: The starting conditions for the attributes differs by class:

   - For required attributes, a node joins the FailSafe membership when the cluster is already providing HA services.
   - For optional attributes, HA services are started and the resource group is running in only one node in the cluster.

   Table 5-3 describes each attribute.

Table 5-3 Failover Attributes

Class	Name	Description
Required	`Auto_Failback`	Specifies that the resource group is made online based on the failover policy when the node joins the cluster. This attribute is best used when some type of load balancing is required. You must specify either this attribute or the `Controlled_Failback` attribute.
	`Controlled_Failback`	Specifies that the resource group remains on the same node when a node joins the cluster. This attribute is best used when client/server applications have expensive recovery mechanisms, such as databases or any application that uses `tcp` to communicate. You must specify either this attribute or the `Auto_Failback` attribute.
Optional	`Auto_Recovery`	Specifies that the resource group is made online based on the failover policy even when an exclusivity check shows that the resource group is running on a node. This attribute is optional and is mutually exclusive with the `InPlace_Recovery` attribute. If you specify neither of these attributes, FailSafe will use this attribute by default if you have specified the `Auto_Failback` attribute.
	`InPlace_Recovery`	Specifies that the resource group is made online on the same node where the resource group is running. This attribute is optional and is mutually exclusive with the `Auto_Recovery` attribute. If you specify neither of these attributes, FailSafe will use this attribute by default if you have specified the `Controlled_Failback` attribute.

Class	Name	Description
	`Critical_RG`	Allows monitor failure recovery to succeed even when there are resource group release failures. When resource monitoring fails, FailSafe attempts to move the resource group to another node in the application failover domain. If FailSafe fails to release the resources in the resource group, FailSafe puts the Resource group into `srmd executable error` status. If the `Critical_RG` failover attribute is specified in the failover policy of the resource group, FailSafe will reset the node where the release operation failed and move the resource group to another node based on the failover policy.
	`Node_Failures_Only`	Allows failover only when there are node failures. This attribute does not have an impact on resource restarts in the local node. The failover does not occur when there is a resource monitoring failure in the resource group. This attribute is useful for customers who are using a hierarchical storage management system such as DMF; in this situation, a customer may want to have resource monitoring failures reported without automatic recovery, allowing operators to perform the recovery action manually if necessary.

See the *IRIS FailSafe Version 2 Programmer's Guide* for a full discussion of example failover policies.

4. **Recovery:** choose the recovery attribute:
   - **Let FailSafe Choose** means that FailSafe will determine the best attribute for the circumstances.
   - **Automatic** means that the group will be brought online on the initial node in the failover domain.
   - **In Place** means that the group will be brought online on the node where the group is already partially allocated.
5. **Critical Resource Group:** check to toggle selection. Selecting this attribute allows monitor failure recovery to succeed even when there are resource group release failures.

   When resource monitoring fails, FailSafe attempts to move the resource group to another node in the failover domain:

- If FailSafe fails to release the resources, it puts the resource group into `srmd executable error` state.
- If you select the **Critical Resource Group** state, FailSafe will reset the node where the release operation failed and move the resource group to another node based on the failover policy.

6. **Node Failures Only:** this attribute controls failover on resource monitoring failures. If you select this attribute, the resource group recovery (that is, failover to another node in the failover domain) is performed only when there are node failures.
7. **Other Attributes:** enter in additional attributes to be used for failover. These optional attributes are determined by the user-defined failover scripts that you can write and place into the `/var/cluster/ha/policies` directory.
8. **Ordered Nodes in 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 defined for the cluster; however, the failover domain does not have to include every node in the cluster. The failover domain can be also used to statically load-balance the resource groups in a cluster.

   Examples:

   - In a four-node cluster, a set of two nodes that have access to a particular XLV volume may be the failover domain of the resource group containing that XLV volume.
   - In a cluster of nodes named `venus`, `mercury`, and `pluto`, you could configure the following initial failover domains for resource groups `RG1` and `RG2`:
     - `RG1`: `mercury`, `venus`, `pluto`
     - `RG2: pluto`, `mercury`

   The administrator defines the *initial failover domain* when configuring a failover policy. The initial failover domain is used when a cluster is first booted. The ordered list specified by the initial failover domain is transformed into a *run-time failover domain* by the failover script. With each failure, the failover script takes the current run-time failover domain and potentially modifies it (for the `ordered` failover script, the order will not change); the initial failover domain is never used again. Depending on the run-time conditions, such as load and contents of the failover script, the initial and run-time failover domains may be identical.

For example, suppose that the cluster contains three nodes named `N1`, `N2`, and `N3`; that node failure is not the reason for failover; and that the initial failover domain is as follows:

```
N1 N2 N3
```

The run-time failover domain will vary based on the failover script:

- If `ordered`:

  ```
  N1 N2 N3
  ```

- If `round-robin`:

  ```
  N2 N3 N1
  ```

- If a customized failover script, the order could be any permutation, based on the contents of the script:

  ```
  N1 N2 N3                N1 N3 N2
  N2 N1 N3                N2 N3 N1
  N3 N1 N2                N3 N2 N1
  ```

  FailSafe stores the run-time failover domain and uses it as input to the next failover script invocation.

9. Click on **OK** to complete the task.

Complete information on failover policies and failover scripts, with an emphasis on writing your own failover policies and scripts, is provided in the *IRIS FailSafe Version 2 Programmer's Guide*.

Define a Failover Policy with `cmgr`

For details about failover policies, see "Define a Failover Policy with the GUI", page 182.

Use the following to define a failover policy:

```
define failover_policy policyname
    set attribute to attributename
    set script to scriptname
    set domain to nodename
```

The following prompt appears:

`failover_policy` *policyname*?

When you define a failover policy, you can set as many attributes and domains as your setup requires, executing the `add attribute` and `add domain` commands with different values. You can also specify multiple domains in one command of the following format:

`set domain to` *node1 node2 node3 ...*

The components of a failover policy are described in detail in the *IRIS FailSafe Version 2 Programmer's Guide* and in summary in "Define a Failover Policy with the GUI", page 182.

For example, suppose you have a failover policy named `fp_ord` with attributes `Auto_Failback`, `Auto_Recovery` and `Critical_RG` and a failover domain of `node2 node1`. The primary node is `node2` and the backup node is `node1`. Following is an example of defining the failover policy in normal mode:

```
cmgr> define failover_policy fp_ord
Enter commands, when finished enter either "done" or "cancel"

failover_policy fp_ord? set attribute to Auto_Failback
failover_policy fp_ord? set attribute to Auto_Recovery
failover_policy fp_ord? set attribute to Critical_RG
failover_policy fp_ord? set script to ordered
failover_policy fp_ord? set domain to node2 node1
failover_policy fp_ord? done
```

Modify a Failover Policy Definition

This section describes how to modify a failover policy.

Modify a Failover Policy Definition with the GUI

The process of deleting a failover policy is similar to defining a new policy. See "Define a Failover Policy with the GUI", page 182.

Do the following:

1. **Failover Policy:** select the name of the failover policy.

2. **Script:** use the menu to select the name of an existing failover script:

   - `ordered` is provided with the FailSafe release. The `ordered` script never changes the initial domain; when using this script, the initial and run-time domains are equivalent.
   - `round-robin` is also provided with the FailSafe release. The `round-robin` script selects the resource group owner in a round-robin (circular) fashion. This policy can be used for resource groups that can be run in any node in the cluster.

   Failover scripts are stored in the `/var/clusters/ha/policies` directory. If the `ordered` script does not meet your needs, you can define a new failover script and place it in the `/var/clusters/ha/policies` directory. When you are using the FailSafe GUI, the GUI automatically detects your script and presents it to you as a choice for you to use. You can configure the cluster database to use your new failover script for the required resource groups. For information on defining failover scripts, see the *IRIS FailSafe Version 2 Programmer's Guide*.

3. **Failback:** choose the name of the failover attribute. You can specify the following classes of failover attributes:

   - Required attributes: either `Auto_Failback` or `Controlled_Failback` (mutually exclusive)
   - Optional attributes:
     - `Auto_Recovery` or `InPlace_Recovery` (mutually exclusive)
     - `Critical_RG`
     - `Node_Failures_Only`

   Note: The starting conditions for the attributes differs by class:

   - For required attributes, a node joins the FailSafe membership when the cluster is already providing HA services.
   - For optional attributes, HA services are started and the resource group is running in only one node in the cluster.

   Table 5-3, page 184 describes each attribute.

   See the *IRIS FailSafe Version 2 Programmer's Guide* for a full discussions of example failover policies.

4. **Recovery:** choose the recovery attribute, or let Failsafe choose the best attribute for the circumstances:

   - `Automatic` means that the group will be brought online on the initial node in the failover domain.
   - `InPlace` means that the group will be brought online on the node where the group is already partially allocated.

5. **Critical Resource Group:** check to toggle selection. Selecting this attribute allows monitor failure recovery to succeed even when there are resource group release failures.

   When resource monitoring fails, FailSafe attempts to move the resource group to another node in the failover domain:

   - If FailSafe fails to release the resources, it puts the resource group into `srmd executable error` state.
   - If you select the **Critical Resource Group** state, FailSafe will reset the node where the release operation fails and move the resource group to another node based on the failover policy.

6. **Node Failures Only:** this attribute controls failover on resource monitoring failures. If you select this attribute, the resource group recovery (that is, failover to another node in the failover domain) is performed only when there are node failures.

7. **Other Attributes:** enter in additional attributes to be used for failover. These optional attributes are determined by the user-defined failover scripts that you can write and place into the `/var/cluster/ha/policies` directory.

8. **Ordered Nodes in 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 defined for the cluster; however, the failover domain does not have to include every node in the cluster. The failover domain also can be used to statically load-balance the resource groups in a cluster.

   The administrator defines the *initial failover domain* when configuring a failover policy. The initial failover domain is used when a cluster is first booted. The ordered list specified by the initial failover domain is transformed into a *run-time failover domain* by the failover script. With each failure, the failover script takes the current run-time failover domain and potentially modifies it (for the `ordered` failover script, the order will not change); the initial failover domain is never used

again. Depending on the run-time conditions, such as load and contents of the failover script, the initial and run-time failover domains may be identical.

FailSafe stores the run-time failover domain and uses it as input to the next failover script invocation.

9. Click on **OK** to complete the task.

Complete information on failover policies and failover scripts, with an emphasis on writing your own failover policies and scripts, is provided in the *IRIS FailSafe Version 2 Programmer's Guide*.

Modify a Failover Policy Definition with `cmgr`

Use the following command to modify a failover policy:

`modify failover_policy` *policyname*

You modify a failover policy using the same commands you use to define a failover policy. See "Define a Failover Policy with `cmgr`", page 187.

Delete a Failover Policy

This section describes how to delete a failover policy.

Delete a Failover Policy with the GUI

This task lets you delete a failover policy. Deleting a failover policy does not delete the cluster nodes in the policy's failover domain.

Note: You cannot delete a failover policy that is currently being used by a resource group. You must first use the **Modify Resource Group** task to select a different failover policy for the resource group.

Do the following:

1. **Failover Policy to Delete:** select a policy
2. Click on **OK** to complete the task

Delete a Failover Policy with cmgr

Use the following command to delete a failover policy:

```
delete failover_policy policyname
```

Display a Failover Policy

You can use FailSafe to display any of the following:

- The components of a specified failover policy
- All of the failover policies
- All of the failover policy attributes
- All of the failover policy scripts

Display a Failover Policy with the GUI

Select **View: Failover Policies** to see all defined failover policies in the tree view. Select the name of a specific policy in the tree view in order to see details about it in the item view.

Display a Failover Policy with cmgr

Use the following commands to display a failover policy:

- To view all of the failover policies:

  ```
  show failover policies
  ```

- To view the parameters of a specific failover policy:

  ```
  show failover_policy policyname
  ```

- To view all of the failover policy attributes:

  ```
  show failover_policy attributes
  ```

- To view all of the failover policy scripts:

  ```
  show failover_policy scripts
  ```

Resource Group Tasks

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.

This section describes the following resource group tasks:

- "Define a Resource Group"
- "Modify a Resource Group Definition", page 195
- "Delete a Resource Group", page 195
- "Add/Remove Resources in Resource Group", page 196
- "Move a Resource Group", page 197
- "Display a Resource Group", page 198

Define a Resource Group

This section describes how to define a resource group.

Define a Resource Group with the GUI

Resources are configured together into *resource groups*. 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 FailSafe.

For example, a resource group could contain all of the resources that are required for the operation of a Web node, such as the web node itself, the HA 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.

Do the following:

1. **Failover Policy:** select the name of the failover policy. This policy will determine which node will take over the services of the resource group upon failure.
2. **Resource Group Name:** enter the name of the resource group, with a maximum length of 63 characters.

3. Click on **OK** to complete the task.

To add resources to the group, see "Add/Remove Resources in Resource Group", page 196.

Note: 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 cmgr

Use the following command to define a resource group:

```
define resource_group RG_name [in cluster clustername]
    set failover_policy to policyname
    add resource resourcename of resource_type RT_name
    remove resource resourcename of resource_type RT_name
```

Usage notes:

- `failover_policy` specifies the failover policy name
- `resource` specifies the resource name
- `resource_type` specifies the resource type

For example:

```
cmgr> define resource_group group1 in cluster filesystem-cluster
Enter commands, when finished enter either "done" or "cancel"

resource_group group1? failover_policy to fp_ord
resource_group group1? add resource 10.154.99.99 of resource_type IP_address
resource_group group1? add resource havol of resource_type volume
resource_group group1? add resource /hafs of resource_type filesystem
resource_group group1? done
```

For a full example of resource group creation using `cmgr` see "Example: Create a Resource Group", page 223.

Modify a Resource Group Definition

This section describes how to modify a resource group.

Modify a Resource Group Definition with the GUI

This task lets you change a resource group by changing its failover policy.

Do the following:

1. **Resource Group:** select a resource group
2. **Failover Policy:** select a failover policy
3. Click **OK** to complete the task

To change the contents of the resource group, see "Add/Remove Resources in Resource Group", page 196.

Modify a Resource Group Definition with `cmgr`

Use the following commands to modify a resource group:

```
modify resource_group RG_name [in cluster clustername]
    set failover_policy to policyname
    add resource resourcename of resource_type RT_name
    remove resource resourcename of resource_type RT_name
```

For example:

```
cmgr> modify resource_group WS1 in cluster test-cluster
```

You modify a resource group using the same commands you use to define a resource group. See "Defining a Resource Group with `cmgr`", page 194.

Delete a Resource Group

This section describes how to delete a resource group.

Delete a Resource Group with the GUI

This task lets you delete an offline resource group. Deleting the group does not delete the individual resources that are members of the group.

Note: You cannot delete a resource group that is online.

Do the following:

1. **Group to Delete:** select the name of the resource group you want to delete. Only offline resource groups are listed.
2. Click on **OK** to complete the task.

Delete a Resource Group with `cmgr`

Use the following command to delete a resource group:

`delete resource_group` *RG_name*`[in cluster` *clustername*`]`

For example:

```
cmgr> delete resource_group WS1 in cluster test-cluster
```

Add/Remove Resources in Resource Group

This task lets you change a resource group by adding or removing resources.

Note: You cannot have a resource group online that does not contain any resources; therefore, 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.

Resources must be added and deleted in atomic units; this means that resources that are interdependent must be added and deleted together.

Do the following:

1. **Resource Group:** select a resource group. A list of existing resources in the group appears.

2. To add a resource to the group:
   - **Resource Type:** select a resource type
   - **Resource Name:** select a resource name
   - Click **Add**
3. To modify a resource in the group:
   - Select its name from the display window
   - Click **Modify**
4. To delete a resource from the group:
   - Select its name from the display window
   - Click **Delete**
5. Click **OK** to complete the task.

Move a Resource Group

This section describes how to move a resource group.

Move a Resource Group with the GUI

This task lets you move a resource group (and all of its resources) from the node on which it is currently running to another node in the group's failover domain.

Enter the following:

1. **Group to Move:** select the name of the resource group to be moved.

   Click **Next** to move to the next screen.
2. **Failover Domain Node:** select the node to which you want to move the resource group. The node must be in the failover domain for this resource group.

   This step is optional; if you do not select a node, FailSafe will move the resource group to the next available node in the failover domain.
3. Click **OK** to complete the task.

Move a Resource Group with cmgr

To move a resource group, use the following command:

```
admin move resource_group FG_name [in cluster clustername] [to node nodename]
```

For example, to move resource group `nfs-group1` running on node `primary` to node `backup` in the cluster `nfs-cluster`, do the following:

```
cmgr> admin move resource_group nfs-group1 in cluster nfs-cluster to node backup
```

If the user does not specify the node, the resource group's failover policy is used to determine the destination node for the resource group.

Display a Resource Group

This section describes how to display a resource group.

Display a Resource Group with the GUI

You can display the parameters of a defined resource group, and you can display all of the resource groups defined for a cluster.

Display a Resource Group with cmgr

Use the following commands to display a resource group

- To view a specific resource group:

  ```
  show resource_group RG_name [in cluster clustername]
  ```

  For example:

  ```
  cmgr> show resource_group small-rg in cluster test-cluster
  Resource Group: small-rg
          Cluster: test-cluster
          Failover Policy: test_fp

  Resources:
          100.101.10.101  (type: IP_address)
          /hafs  (type: filesystem)
          havol  (type: volume)
  ```

- To view all of the resource groups:

  `show resource_groups [in cluster` *clustername*`]`

  For example:

```
cmgr> show resource_groups in cluster test-cluster

Resource Groups:
        bar-rg
        foo-rg
        small-rg
```

FailSafe HA Services Tasks

After you have configured your FailSafe system and run diagnostic tests on its components, you can activate FailSafe by starting the highly available (HA) services. You can start HA services on all of the nodes in a cluster or on a specified node only.

This section describes the following tasks:

- "Start FailSafe HA Services"
- "Stop FailSafe HA Services", page 200
- "Set FailSafe HA Parameters", page 203
- "Set Log Configuration", page 206

Start FailSafe HA Services

This section describes how to start FailSafe HA services.

Start FailSafe HA Services with the GUI

You can start FailSafe HA services on all of the nodes in a cluster or on a specified node only:

1. **Cluster Name:** the name of the cluster is specified for you.
2. **One Node Only:** if you want HA services to be started on one node only, choose its name. If you leave this field blank, HA services will be started on every node in the cluster.

Caution: When you start HA services on a subset of nodes, you should make sure that resource groups are running on only the started nodes. For example, if a cluster contains nodes `N1`, `N2`, and `N3` and HA services are started on nodes `N1` and `N2` but not on node `N3`, you should make sure that resource groups are not running on node `N3`. FailSafe will not perform exclusivity checks on nodes where HA services are not started.

When you start HA services, the following actions are performed:

- All nodes in the cluster (or the selected node only) are enabled
- FailSafe returns success to the user after modifying the cluster database
- The local `cmond` gets notification from the `fs2d` daemon
- The local `cmond` starts all HA processes (`cmsd`, `gcd`, `srmd`, `fsd`) and `ifd`
- `cmond` sets the `failsafe2` `chkconfig` flag to `on`

Start HA Services with `cmgr`

Use the following command to start HA services:

`start ha_services [on node` *nodename*`] [for cluster` *clustername*`]`

For example:

- To start HA services across the cluster:

  `cmgr>` **`start ha_services for cluster test-cluster`**

- To start HA services just on node `N1`:

  `cmgr>` **`start ha_services on node N1 for cluster test-cluster`**

Stop FailSafe HA Services

This section describes how to stop FailSafe HA services.

Stop FailSafe HA Services with the GUI

You can stop HA services on all of the nodes in a cluster or on one specified node.

Note: This is a long-running task that might take a few minutes to complete.

Stopping a node or a cluster is a complex operation that involves several steps and can take several minutes. Aborting a stop operation can leave the nodes and the resources in an unintended state.

When stopping 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 that are in transition will cause any stop 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 stopped a node or a cluster, it 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; in the latter case, resource groups are moved offline, as described in the following sections.

When you stop HA services, the FailSafe daemons perform the following actions:

- A shutdown request is sent to the `fsd` daemon
- `fsd` releases all resource groups and puts them in `ONLINE-READY` state
- All nodes in the cluster in the cluster database are disabled (one node at a time and the local node last)
- FailSafe waits until the node is removed from the FailSafe membership before disabling the node
- The shutdown is successful only when all nodes are not part of the FailSafe membership
- `cmond` receives notification from the cluster database when nodes are disabled
- The local `cmond` sends a `SIGTERM` to all HA processes and `ifd`
- All HA processes clean up and exit with "don't restart" code
- All other `cmsd` daemons remove the disabled node from the FailSafe membership

If HA services are stopped on one node, that node's online resource groups will be moved, according to the failover policy, to a node where HA services are active. If

HA services are stopped on the cluster, all online resource groups will be taken offline, making them no longer highly available.

See the caution in "Start FailSafe HA Services with the GUI", page 199.

Stopping HA Services on One Node

To stop HA services on one node, enter the following:

- **Force:** click the checkbox to forcibly stop the services even if there are errors that would normally prevent them from being stopped.

  The operation of stopping 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 an HA 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 stop HA services on that node.

  In this circumstance, you can specify the **Force** option to shut down the node even if resource groups cannot be moved or released. This will normally leave resource groups allocated in a non-highly-available state on that same node. Using the `force` option might result in the node getting reset. In order to guarantee that the resource groups remain allocated on the last node in a cluster, all online resource groups should be detached.

  If you wish to move resource groups offline that are owned by the node being shut down, you must do so prior to stopping the node.

- **Cluster Name:** the name of the cluster is specified for you.
- **One Node Only:** select the node name.
- Click **OK** to complete the task.

Stopping HA Services on All Nodes in a Cluster

Stopping HA services across the 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.

To stop HA services on all nodes, enter the following:

- **Force:** click the checkbox to force the stop even if there are errors
- **Cluster Name:** the name of the cluster is specified for you
- ***One Node Only***: leave this field blank
- Click **OK** to complete the task

Stop FailSafe HA Services with `cmgr`

To stop FailSafe HA services, use the following command:

`stop ha_services [on node` *nodename*`] [for cluster` *clustername*`] [force]`

The `force` option will cause the stop to occur even if there are errors.

This is a long-running task might take a few minutes to complete. The `cmgr` command will provide intermediate task status for such tasks. For example:

```
cmgr> stop ha_services in cluster nfs-cluster
Making resource groups offline
Stopping HA services on node node1
Stopping HA services on node node2
```

Set FailSafe HA Parameters

This section tells you how to set FailSafe HA parameters.

Set FailSafe HA Parameters with the GUI

This task lets you change how FailSafe monitors the cluster and detects the need for node resets and group failovers:

1. **Cluster Name:** name of the cluster; this value is provided for you.
2. **Heartbeat Interval:** 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. It has a default value of 1000 milliseconds.

   The higher the number of heartbeats (smaller heartbeat interval), the greater the potential for slowing down the network. Conversely, the fewer the number of heartbeats (larger heartbeat interval), the greater the potential for reducing availability of resources.
3. **Node Timeout:** if no heartbeat is received from a node within the node timeout period, the node is considered to be dead and is not considered part of the FailSafe membership.

   Enter a value in milliseconds. The node timeout must be at least 5 seconds. In addition, the node timeout must be at least 10 times the heartbeat interval for proper FailSafe operation; otherwise, false failovers may be triggered. It has a default value of 15000 milliseconds.

   Node timeout is a clusterwide parameter.
4. **Node Wait Time:** the interval, in milliseconds, during which a node waits for other nodes to join the cluster before declaring a new FailSafe membership. If the value is not set for the cluster, FailSafe calculates this value by multiplying the **Node Timeout** value by the number of nodes.
5. **Powerfail Mode:** check the box to turn it on. The powerfail mode indicates whether a special power failure algorithm should be run when no response is received from a system controller after a reset request. Powerfail is a node-specific parameter, and should be defined for the node that performs the reset operation.
6. **Tie-Breaker Node:** select a node name. The failsafe tie-breaker node is the node used to compute the FailSafe membership in situations where 50% of the nodes in a cluster can talk to each other. If you do not specify a tie-breaker node, the node with the lowest node ID number is used.

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 stopped, leaving an even number of nodes to determine membership.

In a 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.

Set FailSafe HA Parameters with `cmgr`

You can modify the FailSafe parameters with the following command:

```
modify ha_parameters [on node nodename] [in cluster clustername]
    set node_timeout to timeout_value
    set heartbeat to heartbeat_interval
    set run_pwrfail to true|false
    set node_wait to node_wait_time
    set tie_breaker to tie_breaker_nodename
```

Usage notes:

- `node_timeout` is the node time-out period. If no heartbeat is received from a node within the node timeout period, the node is considered to be dead and is not considered part of the FailSafe membership.

  Enter a value in milliseconds. The node timeout must be at least 5 seconds. In addition, the node timeout must be at least 10 times the heartbeat interval for proper FailSafe operation; otherwise, false failovers may be triggered. It has a default value of 60000 milliseconds.

  `node_timeout` is a clusterwide parameter.

- `heartbeat` is the heartbeat 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. It has a default value of 1000 milliseconds.

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

- `run_pwrfail` indicates whether a special power failure algorithm should be run (`true`) when no response is received from a system controller after a reset request.

Powerfail is a node-specific parameter, and should be defined for the node that performs the reset operation.

- `node_wait` is the interval, in milliseconds, during which a node waits for other nodes to join the cluster before declaring a new FailSafe membership. If the value is not set for the cluster, FailSafe calculates this value by multiplying the **Node Timeout** value by the number of nodes.
- `tie_breaker` is the name of the node to act as the FailSafe tie breaker.

  Setting `tie_breaker` to `""` (no space between quotation marks) unsets the `tie_breaker` value. Unsetting the `tie_breaker` is equivalent to not setting the value in the first place. In this case, FailSafe will use the node with the lowest node ID as the tie breaker node.

Set Log Configuration

This section describes how to set log configuration.

Set Log Configuration with the GUI

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. Changes apply as follows:

- To all nodes in the pool for the `cli` and `crsd` log groups
- To 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.

Default Log File Names

FailSafe logs both normal operations and critical errors to the `/var/adm/SYSLOG` file, as well as to individual log files for each log group.

To set the log configuration, enter the appropriate values for the following fields:

1. **Log Group:** a log group is a set of processes that log to the same log file according to the same logging configuration. All FailSafe daemons make one log group each. FailSafe maintains the following log groups:

`cli`	Commands log
`crsd`	Cluster reset services (`crsd`) log
`diags`	Diagnostics log
`ha_agent`	HA monitoring agents (`ha_ifmx2`) log
`ha_cmsd`	FailSafe membership daemon (`ha_cmsd`) log
`ha_fsd`	FailSafe daemon (`ha_fsd`) log
`ha_gcd`	Group communication daemon (`ha_gcd`) log
`ha_ifd`	network interface monitoring daemon (`ha_ifd`) log
`ha_script`	Action and Failover policy scripts log
`ha_srmd`	System resource manager (`ha_srmd`) log

2. **Log Level:** the log level, specified as character strings with the GUI and numerically (1 to 19) with `cmgr`. 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 levels. Table 5-4 shows the logging levels as you specify them with the GUI and `cmgr`.

Table 5-4 Log Levels

GUI level	`cmgr` 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 (10 -19 in `cmgr`) 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	

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`.

3. **Log File:** a file that contains FailSafe notifications for a particular log group. Log file names beginning with a slash are absolute, while names not beginning with a slash are relative to the `/var/cluster/ha/log` directory.

   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*.

   shows the default log file names.

   For information on using log groups in system recovery, see Chapter 9, "System Recovery and Troubleshooting".

4. Click **OK** to complete the task.

Table 5-5 Default Log File Names

File Name	Description
`/var/cluster/ha/log/cmsd_`*nodename*	Log file for the FailSafe membership services daemon in node `nodename`
`/var/cluster/ha/log/gcd_`*nodename*	Log file for group communication daemon in node *nodename*
`/var/cluster/ha/log/srmd_`*nodename*	Log file for system resource manager daemon in node *nodename*
`/var/cluster/ha/log/failsafe_`*nodename*	Log file for the 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`

File Name	Description
`/var/cluster/ha/log/script_`*nodename*	Log file for scripts in node `nodename`
`/var/cluster/ha/log/cli_`*nodename*	Log file for internal administrative commands in node `nodename` invoked by the GUI and `cmgr`

Display Log Group Definitions with the GUI

To display log group definitions with the GUI, run choose the **Log Group** 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.

Define Log Groups with `cmgr`

Use the following command to define a log group:

```
define log_group groupname [on node nodename] [in cluster clustername]
    set log_level to level
    add log_file logfile_name
    remove log_file logfile_name
```

Usage notes:

- Specify the node name if you wish to customize the log group configuration for a specific node only. For details about legal values, see "Set Log Configuration with the GUI", page 206.
- `log_level` can have one of the following values:
  - `0` gives no logging
  - `1` logs notifications of critical errors and normal operation (these messages are also logged to the `SYSLOG` file)
  - `2` logs `Minimal` notifications plus warnings
  - `5` through `7` log increasingly more detailed notifications
  - `10` through `19` log increasingly more debug information, including data structures
- `log_file` is the file that contains FailSafe notifications for a particular log group. Log file names beginning with a slash are absolute, while names not beginning with a slash are relative to the `/var/cluster/ha/log` directory.

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*.

For a list of default log names, see Table 5-5, page 208.

Configure Log Groups with `cmgr`

You can configure a log group with the following command:

`define log_group` *log_group* `on node` *hostname* `[in cluster` *clustername*`]`

The *log_group* variable can be one of the following:

```
cli
crsd
diags
ha_agent
ha_cmsd
ha_fsd
ha_gcd
ha_ifd
ha_script
ha_srmd
```

Caution: Do not change the names of the log files. If you change the names, errors can occur.

For example, to define log group `cli` on node `fs6` with a log level of `5`:

```
cmgr> define log_group cli on node fs6 in cluster fs6-8

(Enter "cancel" at any time to abort)

Log Level ? (11) 5

CREATE LOG FILE OPTIONS

        1) Add Log File.
        2) Remove Log File.
        3) Show Current Log Files.
```

```
        4) Cancel. (Aborts command)
        5) Done. (Exits and runs command)

Enter option:5
Successfully defined log group cli
```

Modify Log Groups with cmgr

Use the following command to modify a log group:

`modify log_group` *log_group_name* `on node` *hostname* `[in cluster` *clustername*`]`

You modify a log group using the same commands you use to define a log group. See "Define Log Groups with cmgr", page 209.

For example, to change the log level of `cli` to be `10`, enter the following:

```
cmgr> modify log_group cli on node fs6 in cluster fs6-8

(Enter "cancel" at any time to abort)

Log Level ? (2) 10

MODIFY LOG FILE OPTIONS

        1) Add Log File.
        2) Remove Log File.
        3) Show Current Log Files.
        4) Cancel. (Aborts command)
        5) Done. (Exits and runs command)

Enter option:5
Successfully modified log group cli
```

For example, to set the log level for the `ha_script` log group to 11, enter the following:

```
cmgr> modify log_group ha_script

log_group ha_script ? set log_level to 11
log_group ha_script ? done
Successfully modified log group ha_script
```

Display Log Group Definitions

This section describes how to display log group definitions.

Display Log Group Definitions with `cmgr`

Use the following command to display log group definitions:

```
show log_groups
```

This command shows all of the log groups currently defined, with the log group name, the logging levels and the log files.

For information on viewing the contents of the log file, see Chapter 9, "System Recovery and Troubleshooting".

Configuration Examples

This chapter provides an example of an IRIS FailSafe configuration that uses a three-node cluster and some variations of that configuration. In addition, this chapter provides instructions for exporting CXFS filesystems in a FailSafe configuration. It contains the following sections:

- "Example: Define a Three-Node Cluster"
- "Example: Script to Define a Three-Node Cluster", page 214
- "Example: Local Failover of HA IP Address", page 220
- "Example: Modify a Cluster to Include a CXFS Filesystem", page 221
- "Example: Export CXFS Filesystems", page 222
- "Example: Create a Resource Group", page 223

Example: Define a Three-Node Cluster

The following illustration shows a three-node FailSafe cluster. This configuration consists of a pool containing nodes `N1`, `N2`, `N3`, and `N4`. Nodes `N1`, `N2`, and `N3` make up the FailSafe cluster. The nodes in this cluster share disks, and are connected to an El-8 serial port multiplexer, which is also connected to the private control network.

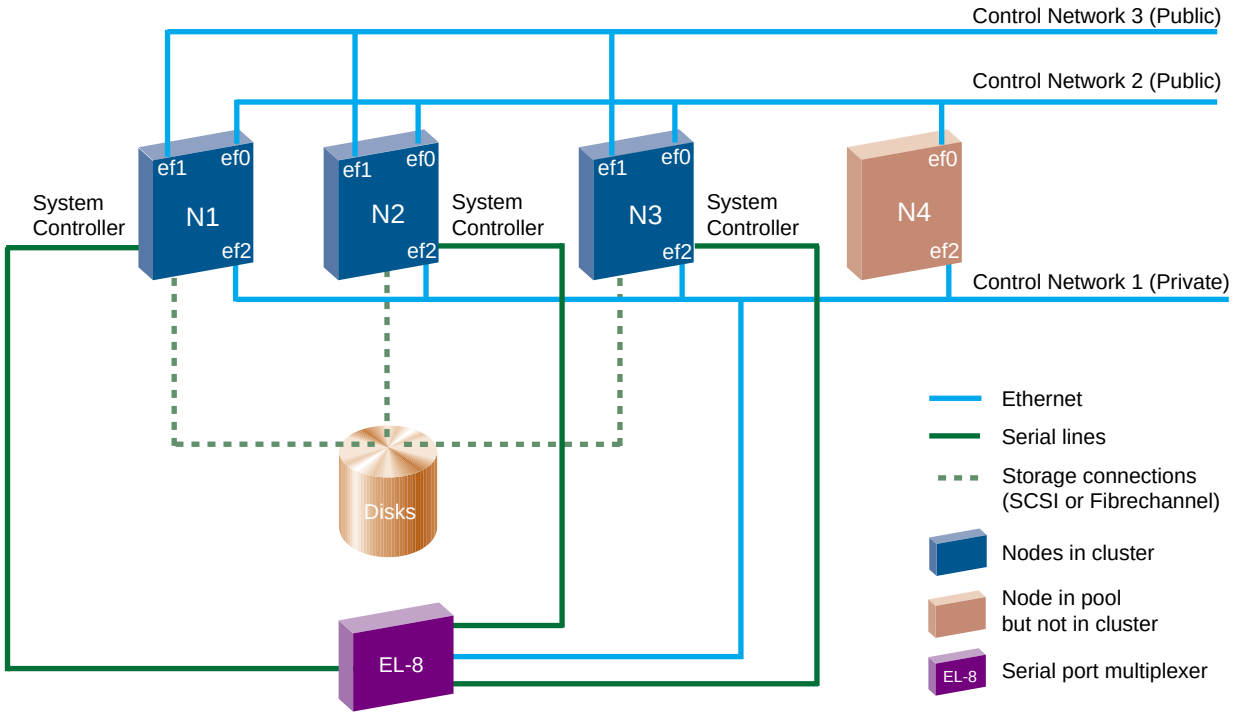

Figure 6-1 FailSafe Configuration Example

Examples of FailSafe configurations that use this setup are provided in the following sections.

Example: Script to Define a Three-Node Cluster

This section provides an example `cmgr` script that defines a FailSafe three-node cluster as shown in Figure 6-1. For general information on `cmgr` scripts see "Using Script Files", page 98. For information on the template files that you can use to create your own configuration script, see "Template Scripts", page 100.

This cluster has two resource groups, `RG1` and `RG2`, as shown in Table 6-1.

Table 6-1 Resources and Failover Policies for RG1 and RG2

Resources and Failover Policy	RG1	RG2
Resources		
IP_address	192.26.50.1	192.26.50.2
filesystem	/ha1	/ha2
volume	ha1_vol	ha2_vol
NFS	/ha1/export	/ha2/export
Failover policy		
Name	fp1	fp2
Script	ordered	round-robin
Attributes	Auto_Failback, Auto_Recovery	Controlled_Failback, InPlace_Recovery
Failover domain	N1, N2, N3	N2, N3

The cmgr script to define this configuration is as follows:

```
#!/usr/cluster/bin/cmgr -f
define node N1
        set hostname to N1
        set is_failsafe to true
        set sysctrl_type to msc
        set sysctrl_status to enabled
        set sysctrl_password to none
        set sysctrl_owner to N4
        set sysctrl_device to /dev/ttydn001
        set sysctrl_owner_type to tty
        add nic ef2-N1
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
        add nic ef0-N1
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 2
        done
```

```
        add nic ef1-N1
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 3
        done
done


define node N2
        set hostname to N2
        set is_failsafe to true
        set sysctrl_type to msc
        set sysctrl_status to enabled
        set sysctrl_password to none
        set sysctrl_owner to N4
        set sysctrl_device to /dev/ttydn002
        set sysctrl_owner_type to tty
        add nic ef2-N2
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
        add nic ef0-N2
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 2
        done
        add nic ef1-N2
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 3
        done
done

define node N3
        set hostname to N3
        set is_failsafe to true
        set sysctrl_type to msc
        set sysctrl_status to enabled
        set sysctrl_password to none
        set sysctrl_owner to N4
```

```
        set sysctrl_device to /dev/ttydn003
        set sysctrl_owner_type to tty
        add nic ef2-N3
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
        add nic ef0-N3
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 2
        done
        add nic ef1-N3
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 3
        done
done


define node N4
        set hostname to N4
        set is_failsafe to true
        add nic ef2-N4
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
        add nic ef0-N4
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 2
        done
done
define cluster TEST
        set is_failsafe to true
        set notify_cmd to /usr/bin/mail
        set notify_addr to failsafe_sysadm@company.com
        add node N1
        add node N2
        add node N3
```

```
done

define failover_policy fp1
        set attribute to Auto_Failback
        set attribute to Auto_Recovery
        set script to ordered
        set domain to N1 N2 N3
done


define failover_policy fp2
        set attribute to Controlled_Failback
        set attribute to InPlace_Recovery
        set script to round-robin
        set domain to N2 N3
done

define resource 192.26.50.1 of resource_type IP_address in cluster TEST
        set NetworkMask to 0xffffff00
        set interfaces to ef0,ef1
        set BroadcastAddress to 192.26.50.255
done

define resource ha1_vol of resource_type volume in cluster TEST
        set devname-owner to root
        set devname-group to sys
        set devname-mode to 666
done

define resource /ha1 of resource_type filesystem in cluster TEST
        set volume-name to ha1_vol
        set mount-options to rw,noauto
        set monitor-level to 2
done

modify resource /ha1 of resource_type filesystem in cluster TEST
        add dependency ha1_vol of type volume
done

define resource /ha1/export of resource_type NFS in cluster TEST
        set export-info to rw,wsync
```

```
        set filesystem to /ha1
done

modify resource /ha1/export of resource_type NFS in cluster TEST
        add dependency /ha1 of type filesystem
done
define resource_group RG1 in cluster TEST
        set failover_policy to fp1
        add resource 192.26.50.1 of resource_type IP_address
        add resource ha1_vol of resource_type volume
        add resource /ha1 of resource_type filesystem
        add resource /ha1/export of resource_type NFS
done


define resource 192.26.50.2 of resource_type IP_address in cluster TEST
        set NetworkMask to 0xffffff00
        set interfaces to ef0
        set BroadcastAddress to 192.26.50.255
done

define resource ha2_vol of resource_type volume in cluster TEST
        set devname-owner to root
        set devname-group to sys
        set devname-mode to 666
done

define resource /ha2 of resource_type filesystem in cluster TEST
        set volume-name to ha2_vol
        set mount-options to rw,noauto
        set monitor-level to 2
done

modify resource /ha2 of resource_type filesystem in cluster TEST
        add dependency ha2_vol of type volume
done

define resource /ha2/export of resource_type NFS in cluster TEST
        set export-info to rw,wsync
        set filesystem to /ha2
done
```

```
modify resource /ha2/export of resource_type NFS in cluster TEST
        add dependency /ha2 of type filesystem
done

define resource_group RG2 in cluster TEST
        set failover_policy to fp2
        add resource 192.26.50.2 of resource_type IP_address
        add resource ha2_vol of resource_type volume
        add resource /ha2 of resource_type filesystem
        add resource /ha2/export of resource_type NFS
done


quit
```

Example: Local Failover of HA IP Address

You can configure a FailSafe system to fail over a highly available (HA) IP address to a second interface within the same host. To do this, you specify multiple interfaces for resources of `IP_address` resource type. You can also specify different interfaces for supporting a heterogeneous cluster. For information on specifying HA IP address resources, see "`IP_address` Attributes", page 171.

The following example configures local failover of an HA IP address. It uses the configuration illustrated in Figure 6-1.

1. Define an HA IP address resource with two interfaces:

```
define resource 192.26.50.1 of resource_type IP_address in cluster TEST
          set NetworkMask to 0xffffff00
          set interfaces to ef0,ef1
          set BroadcastAddress to 192.26.50.255
done
```

HA IP address `192.26.50.1` will be locally failed over from interface `ef0` to interface `ef1` when there is an `ef0` interface failure.

In nodes `N1`, `N2`, and `N3`, either `ef0` or `ef1` should configure up automatically, when the node boots up. Both `ef0` and `ef1` are physically connected to the same

subnet `192.26.50`. Only one network interface connected to the same network should be configured up in a node.

2. Modify the `/etc/conf/netif.options` file to configure the `ef0` and `ef1` interfaces:

```
if1name-ef0
if1addr=192.26.50.10

if2name=ef1
if2addr=192.26.50.11
```

3. The `etc/init.d/network` script should configure the network interface `ef1` down in all nodes `N1`, `N2`, and `N3`. Add the following line to the file:

```
ifconfig ef1 down
```

Example: Modify a Cluster to Include a CXFS Filesystem

The following procedural example modifies the sample FailSafe configuration illustrated in Figure 6-1 so that it includes highly available NFS services on a CXFS filesystem. However, the `CXFS` resource type does not mount a CXFS filesystem. You should mount CXFS filesystem using the CXFS GUI as described in *CXFS Version 2 Software Installation and Administration Guide*. The `CXFS` resource type monitors the CXFS filesystem for mount failures.

Note: FailSafe assumes that CXFS filesystems are highly available because they do not require a FailSafe failover in order to be made available on another node in the cluster. Therefore, FailSafe does not directly start or stop CXFS filesystems nor does it stop, start, or monitor XVM volumes. XVM volumes should not be added to the FailSafe resource groups.

To modify the FailSafe configuration to include a CXFS filesystem, perform the following steps:

1. Convert the cluster `TEST` for CXFS use. For information on converting FailSafe clusters to CXFS, see the *CXFS Version 2 Software Installation and Administration Guide*.

2. Convert the nodes `N1` and `N2` for CXFS use. For information on converting FailSafe nodes to CXFS, see the *CXFS Version 2 Software Installation and Administration Guide*. Start CXFS services on the nodes.
3. Create a new resource type `NFS1`. This is the same as resource type `NFS` but without a filesystem dependency. To create this resource type you can perform the following steps:
   a. Using `cmgr`, execute the following:

      ```
      cmgr> show resource_type NFS in cluster TEST
      ```

      The parameters of resource type `NFS` will be displayed.
   b. Define resource type `NFS1` using the same configuration information that was displayed for resource type NFS, but do not copy the filesystem dependency.
4. Define a new failover policy, `FP3`, with the following attributes:
   - Failover domain: `N1`, `N2`
   - Script: `ordered`
   - Attribute: `InPlace_Recovery`
5. Create a resource named `/cxfs` of resource type `CXFS`. `/cxfs` is the mount point of the CXFS filesystem. You can decide to relocate the metadata server of the CXFS filesystem `/cxfs` when the resource group moves to another node.
6. Create a resource group named `rg3` with failover policy `fp3`, resource `ip3` of resource type `IP_address`, and resource `/cxfs` of resource type `CXFS`.
7. Mount `/cxfs` on nodes `N1` and `N2`. For information on defining and mounting a CXFS filesystem with an XVM volume, see *CXFS Version 2 Software Installation and Administration Guide*.
8. Bring resource group `RG3` online in cluster `TEST`.

Example: Export CXFS Filesystems

Perform the following steps to export CXFS filesystems in a FailSafe configuration:

1. Ensure that the latest patches of the FailSafe/NFS 2.1 release are installed on all FailSafe nodes in the cluster.

2. Perform all of steps mentioned in "Example: Modify a Cluster to Include a CXFS Filesystem", page 221.
3. If you are planning to export the `/cxfs/share` directory, create an NFS resource named `/cxfs/share`.
4. Add the NFS resource to the resource group `rg3` in addition to the HA IP address resource and the CXFS resource, using the following commands:

```
define resource_group rg3 in cluster TEST
set failover_policy to fp3
add resource 99.92.99.99 of resource_type IP_address
add resource /cxfs of resource_type CXFS
add resource /cxfs/share of resource_type NFS
done
```

Note: You cannot use this procedure to export the same CXFS filesystem or subdirectory from multiple nodes in the cluster.

Example: Create a Resource Group

Use the following procedure to create a resource group using `cmgr`:

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 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 must 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 `cmgr`(1M) command.

IRIS FailSafe System Operation

This chapter describes administrative tasks you perform to operate and monitor an IRIS FailSafe system. It describes how to perform tasks using the FailSafe Manager GUI and the `cmgr`(1M) command. The major sections in this chapter are as follows:

- "Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C Console Support"
- "System Operation Considerations"
- "Two-Node Clusters: Single-Node Use", page 227
- "System Status", page 233
- "Embedded Support Partner (ESP) Logging of FailSafe Events", page 249
- "Resource Group Failover", page 250
- "Stopping FailSafe", page 257
- "Resetting Nodes", page 257
- "Backing Up and Restoring Configuration with `cmgr`", page 258
- "Log File Management", page 259

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

Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C Console Support

On Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C systems, there is only one serial/USB port that provides both L1 and console support for the machine. In a FailSafe configuration, this port (the DB9 connector) is used for system reset. It is connected to a serial port in another node or to the Ethernet multiplexer.

To get access to console input and output, the console must be redirected to another serial port in the machine. Use the following procedure:

1. Edit the `/etc/inittab` file to use an alternate serial port.

2. Either issue an `init q` command or reboot.

For example, suppose you had the following in the `/etc/inittab` file (line breaks added for readability):

```
# on-board ports or on Challenge/Onyx MP machines, first IO4 board ports
t1:23:respawn:/sbin/suattr -C CAP_FOWNER,CAP_DEVICE_MGT,CAP_DAC_WRITE+ip
-c "exec /sbin/getty ttyd1 console"    # alt console
t2:23:off:/sbin/suattr -C CAP_FOWNER,CAP_DEVICE_MGT,CAP_DAC_WRITE+ip
-c "exec /sbin/getty -N ttyd2 co_9600"     # port 2
```

You could change it to the following:

```
# on-board ports or on Challenge/Onyx MP machines, first IO4 board ports
t1:23:off:/sbin/suattr -C CAP_FOWNER,CAP_DEVICE_MGT,CAP_DAC_WRITE+ip
-c "exec /sbin/getty ttyd1 co_9600"        # port 1
t2:23:respawn:/sbin/suattr -C CAP_FOWNER,CAP_DEVICE_MGT,CAP_DAC_WRITE+ip
-c "exec /sbin/getty -N ttyd2 console" # alt console
```

Caution: Redirecting the console by using the above method works only when IRIX is running. To access console when IRIX is not running (miniroot), you must physically reconnect the machine: unplug the reset cable from the console/L1 port and then connect the console cable.

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.

Two-Node Clusters: Single-Node Use

If you have a two-node cluster, you should create an emergency failover policy (step 1 below) for each node in preparation for a time when it may need to run by itself. This situation can occur if the other must stay down for maintenance or if it fails and cannot be brought up.

Caution: Without these emergency failover policies and the appropriate set of procedures, the surviving node will be in what is called the *lonely state*, meaning that it will never form a cluster by itself.

Using a Single Node

The following procedure describes the steps required to use just one node in the cluster.

1. Create an emergency failover policy for each node. Each policy should look like the following example when the `cmgr` command is issued, where *Active_node* is the name of the node using the policy (in the examples, `nodeA`) and *Down_node* is the name of the nonfunctioning node (in the examples, `nodeB`):

   ```
   cmgr> show failover_policy emergency-Active_node

   Failover Policy: emergency-Active_node
   Version: 1
   Script: ordered
   Attributes: Controlled_Failback InPlace_Recovery
   Initial AFD: Active_node
   ```

   For example, suppose you have two nodes, `nodeA` and `nodeB`. You would have two emergency failover policies:

   ```
   cmgr> show failover_policy emergency-nodeA

   Failover Policy: emergency-nodeA
   Version: 1
   Script: ordered
   Attributes: Controlled_Failback InPlace_Recovery
   Initial AFD: nodeA
   ```

```
cmgr> show failover_policy emergency-nodeB

Failover Policy: emergency-nodeB
Version: 1
Script: ordered
Attributes: Controlled_Failback InPlace_Recovery
Initial AFD: nodeB
```

For more information, see "Define a Failover Policy", page 182.

At this point, the procedure assumes that the cluster has one node that has tried to come up but is now in the lonely state. The other node is down. The procedure is for recovering from this point.

2. Modify each resource group to use the appropriate single-node emergency failover policy, using the following `cmgr` commands or the GUI:

   `modify resource_group` *RG_name* `in cluster` *clustername*
   `set failover_policy to emergency-`*Active_node*

   For example, on `nodeA`:

```
cmgr> set cluster test-cluster
cmgr> modify resource_group group1
Enter commands, when finished enter either "done" or "cancel"

resource_group group1 ? set failover_policy to emergency-nodeA
resource_group group1 ? done
Successfully modified resource group group1
```

3. Change the state of all of resource groups to `offline`. (The last known state of these groups was `online` before the machines went down; however, the resource groups are not actually online at this point because the cluster was booted with the active node having entered the lonely state, since the other node is not functional. This step just tells the database to label the state of the resource groups appropriately in preparation for later steps.)

   Use the following command:

   `admin offline resource_group` *RG_name* `in cluster` *clustername*

For example:

```
cmgr> set cluster test-cluster
cmgr> show resource_groups in test-cluster

Resource Groups:
        group1
        group2

cmgr> admin offline resource_group group1
cmgr> admin offline resource_group group2
```

4. Force the stop of HA services for the down node:

   `stop ha_services on` *Down_node* `for cluster` *clustername* `force`

   Note: This is a long running task that and might take a few minutes to complete. `cmgr` will provide intermediate task status.

   For example:

   ```
   cmgr> stop ha_services on nodeB for cluster test-cluster force
   ```

5. Mark the resource groups as `online` in the database. When HA services are started in future steps, the services will come online using the emergency failover policies.

   `admin online resource_group` *RG_name* `in cluster` *clustername*

   For example:

```
cmgr> set cluster test-cluster
cmgr> admin online resource_group group1
FailSafe daemon (ha_fsd) is not running on this local node or it is not ready to accept admin commands.
Resource Group (group1) is online-ready.

Failed to admin:
        online

admin command failed

cmgr> show status of resource_group group1 in cluster test-cluster
```

```
State: Online Ready
Error: No error
Check resource group group1 status in an active node if HA services are active in cluster
```

6. Start HA services on the active node:

   `start ha_services on` *Active_node* `for` *clustername*

   HA services are now active in this single node cluster. From this step through the rest of the recovery, services are active and there should be no downtime experienced.

   For example:

   ```
   cmgr> start ha_services on nodeA for cluster test-cluster
   ```

7. Remove and reinitialize the database on the down node, which is now booted in multiuser mode:

   ```
   # cd /var/cluster/cdb
   # rm -rf /var/cluster/cdb/cdb*
   # /usr/cluster/bin/cdbreinit /var/cluster/cdb/cdb.db
   ```

   Wait for a few minutes and use the `tail`(1) command to watch the `SYSLOG` file for messages indicating that this node knows its identity and has joined the cluster. For example (line breaks added here for readability):

```
# tail -f /var/adm/SYSLOG
...
Dec 14 18:23:16 6D:Down_node cmond[1074]:  Notification can not be processed, local machine and
  cluster name is not known.
Dec 14 18:23:16 6D:Down_node cmond[1074]:  Local machine belongs to cluster clustername.
Dec 14 18:23:16 6D:Down_node cmond[1074]:  Local machine name is Down_node"
```

Resuming Two-Node Use

To resume using the down node, do the following:

1. Boot the down node into single user mode.
2. Within single-user mode, use the `chkconfig`(1M) command to set all cluster services to `off`:

   ```
   # chkconfig | grep cluster
   ```

3. Boot the *Down_node* to multiuser mode.

4. Start HA services for the down node:

   ```
   cmgr> start ha_services on node Down_node for cluster clustername
   ```

5. Use the `tail`(1) command to watch the `SYSLOG` file for messages that indicate the formerly down node is now in membership and that services are in the `UP` state. Do not continue until you see messages such as the following:

```
# tail -f /var/adm/SYSLOG
...
miredb node Active_node [81] : UP  incarnation 7   age 2:0
miredb node Down_node [82] : UP  incarnation 1   age 1:0
```

6. Modify the resource groups to restore the original failover policies they were using before the failure:

   ```
   modify resource_group RG_name in cluster cluster_name
   set failover_policy to Original-Failover-Policy
   ```

   Note: This only restores the configuration for the static environment. The runtime environment will still be using the single-node policy at this time.

   For example, if the normal failover policy was `normal-fp`:

   ```
   cmgr> set cluster test-cluster
   cmgr> modify resource_group group1
   Enter commands, when finished enter either "done" or "cancel"

   resource_group group1 ? set failover_policy to normal-fp
   resource_group group1 ? done
   Successfully modified resource group group1

   cmgr modify resource_group group2
   Enter commands, when finished enter either "done" or "cancel"

   resource_group group2 ? set failover_policy to normal-fp
   resource_group group2 ? done
   Successfully modified resource group group2
   ```

7. Perform an `offline_detach` command on the resource groups in the cluster. 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. The resources will remain in service.

Note: Because FailSafe is no longer monitoring the group after the `offline_detach` (or `offline_detach_force`) is executed, you should not allow FailSafe to recover on any node other than where it was running at the time the `offline_detach` was performed.

This also means that no other nodes should be allowed to rejoin the FailSafe membership, especially if `Auto_Recovery` is set in the resource group's failover policy. This restriction is required because the failover policy scripts are run whenever there is a change in membership; rerunning the scripts could cause your previously offline detached resource group to start on a node other than the node where the `offline_detach` was performed.

FailSafe policy scripts are run only on nodes where FailSafe is running (nodes where HA services have been started). For example, suppose you have a four-node FailSafe cluster (with nodes A, B, C and D), where nodes A, B, C are in `UP` state and node D is `DOWN` state. If resource group `RG` is made `offline` using `offline_detach` or `offline_detach_force` command on Node B and HA services are shutdown on Node B, Node D should not rejoin the cluster before resources in `RG` are stopped manually on Node B. If Node D rejoins the cluster, the resource group `RG` will be made `online` on Nodes A, C or D.

Use the following command:

`admin offline_detach resource_group` *resource_group* `[in cluster` *clustername*`]`

For example:

```
cmgr> admin offline_detach resource_group group1 in cluster test-cluster
```

Show the status of the resource groups to be sure that they now show as `offline`.

Note: The resources are still in service even though this command output shows them as `offline`.

`show status of resource_group` *resource_group* `in cluster` *clustername*

For example:

```
cmgr> show status of resource_group group1 in cluster test-cluster
```

8. Make the resource groups online in the cluster:

   `admin online resource_group` *RG_name* `in cluster` *cluster_name*

   For example:

   ```
   cmgr> admin online resource_group group1 in cluster test-cluster
   cmgr> admin online resource_group group2 in cluster test-cluster
   ```

9. Move the resources back to their original nodes. Because our original policies included the `InPlace_Recovery` attribute, all of the resources have remained on the node that has been active throughout this process.

`admin move resource_group` *RG_name* `in cluster` *cluster_name* `to node` *Primary_owner*

For example:

```
cmgr> admin move resource_group group1 in cluster test-cluster to node nodeB
```

System Status

While the FailSafe system is running, you can monitor the status of the FailSafe components to determine the state of the component. FailSafe allows you to view the system status in the following ways:

- You can keep continuous watch on the state of a cluster using the `cluster_status` command, or the GUI.
- You can query the status of an individual resource group, node, or cluster using either the GUI or `cmgr`.
- You can use the `haStatus` script provided with the `cmgr` command to see the status of all clusters, nodes, resources, and resource groups in the configuration.

The following sections describe the procedures for performing these tasks.

Monitoring System Status with `cluster_status`

You can use the `cluster_status` command to monitor the cluster using a `curses`(3X) interface. For example, the following shows a two-node cluster configured for FailSafe only and `cluster_status` help text displayed:

```
# /var/cluster/cmgr-scripts/cluster_status
* Cluster=nfs-cluster  FailSafe=ACTIVE  CXFS=Not Configured         08:45:12
   Nodes =   hans2    hans1
FailSafe =      UP       UP
    CXFS =
       ResourceGroup           Owner           State           Error

       bartest-group                          Offline        No error
       footest-group                          Offline        No error
             bar_rg2           hans2           Online        No error
          nfs-group1           hans2           Online        No error
              foo_rg           hans2           Online        No error


              +-------+ cluster_status Help +--------+
              |  on s  - Toggle Sound on event       |
              |  on r  - Toggle Resource Group View  |
              |  on c  - Toggle CXFS View            |
              |     j  - Scroll up the selection     |
              |     k  - Scroll down the selection   |
              |   TAB  - Toggle RG or CXFS selection |
              | ENTER  - View detail on selection    |
              |     h  - Toggle help screen          |
              |     q  - Quit cluster_status         |
              +--- Press 'h' to remove help window --+
```

The above shows that a sound will be activated when a node or the cluster changes status. You can override the `s` setting by invoking `cluster_status` with the `-m` (mute) option. You can also use the arrow keys to scroll the selection.

Note: The `cluster_status` command can display no more than 128 CXFS filesystems.

Monitoring System Status with the GUI

The easiest way to keep a continuous watch on the state of a cluster is to use the GUI tree view.

System components that are experiencing problems 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 icon for the cluster turns red and blinks, as well as the resource group or node icon.

The cluster status can be one of the following:

- **ACTIVE**, which means the cluster is up and running and there is a valid FailSafe membership.
- **INACTIVE**, which means the start FailSafe HA services task has not been run and there is no FailSafe membership.
- **ERROR**, which means that some nodes are in a **DOWN** state; that is, the cluster **should** be running, but it is not.
- **UNKNOWN**,which means that the state cannot be determined because FailSafe HA services are not running on the node performing the query.

If you minimize the GUI window, the minimized-icon shows the current state of the cluster. When the cluster goes into error state, the icon shows a red cluster.

Key to Icons and States

The following tables show keys to the icons and states used in the FailSafe Manager GUI.

The full legend for component states in the FailSafe Cluster View is as follows:

Table 7-1 Key to Icons

Icon	Entity
	Node
	Cluster
	Resource
	Resource group
	Resource type
	Failover policy
	Expanded tree
	Collapsed tree
	

Table 7-2 Key to States

Icon	State
	Inactive or unknown (HA services may not be active)
	Online-ready state for a resource group
	Healthy and active or online
	(blinking) Transitioning to healthy and active/online or transitioning to offline
	Maintenance mode, in which the resource is not monitored by FailSafe
	(blinking red) Problems with the component

Querying Cluster Status with `cmgr`

To query node and cluster status, use the following command:

`show status of cluster` *clustername*

Monitoring Resource and Reset Serial Line with cmgr

You can use cmgr to query the status of a resource or to contact the system controller at a node, as described in the following subsections.

Querying Resource Status with cmgr

To query a resource status, use the following command:

```
show status of resource resource_name of resource_type RT_name [in cluster clustername]
```

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.

Performing a ping of a System Controller with cmgr

To perform a ping(1M) operation on a system controller by providing the device name, use the following command:

```
admin ping dev_name devicename of dev_typedevice_type with sysctrl_type system_controller_type
```

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 FailSafe. It is fully allocated if there is no error; otherwise, some resources may not be allocated or some resources may be in error state.
`ONLINE-PENDING`	FailSafe is running on the local nodes and the resource group is in the process of being allocated. This is a transient state.
`OFFLINE`	The resource group is not running or the resource group has been detached, regardless of whether FailSafe is running. When FailSafe starts up, it will not allocate this resource group.
`OFFLINE-PENDING`	FailSafe is running on the local nodes and the resource group is in the process of being released (becoming offline). This is a transient state.
`ONLINE-READY`	FailSafe is not running on the local node. When FailSafe starts up, it will attempt to bring this resource group online. No FailSafe process is running on the current node if this state is returned.
`ONLINE-MAINTENANCE`	The resource group is allocated in a node in the cluster but it is not being monitored by 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 FailSafe should not act on that resource for a period of time.
`INTERNAL ERROR`	An internal FailSafe error has occurred and FailSafe does not know the state of the resource group. Error

	recovery is required. This could result from a memory error, bugs in a program, or communication problems.
DISCOVERY (EXCLUSIVITY)	The resource group is in the process of going online if FailSafe can correctly determine whether any resource in the resource group is already allocated on all nodes in the resource group's failover 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 SGI 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 failover domain was not in the membership. FailSafe cannot bring the resource group online until that node is removed from the failover 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 SGI.

`NO AVAILABLE NODES`	A monitoring error has occurred on the last valid node in the FailSafe membership.

Resource Owner

The resource owner is the logical node name of the node that currently owns the resource.

Monitoring Resource Group Status with GUI

You can use the tree view to monitor the status of the resources in a FailSafe configuration:

- Select **View: Resources in Groups** to see the resources organized by the groups to which they belong.
- Select **View: 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 `cmgr`

To query a resource group status, use the following `cmgr` command:

`show status of resource_group` *RG_name* `[in cluster` *clustername*`]`

If you have specified a default cluster, you do not need to specify a cluster when you use this command and it will show the status of the indicated resource group in the default cluster.

Node Status

To query the status of a node, you provide the logical node name of the node. The node status can be one of the following:

`UP`	This node is part of the FailSafe membership.
`DOWN`	This node is not part of the FailSafe membership (no heartbeats) and this node has been reset. This is a transient state.
`UNKNOWN`	This node is not part of the FailSafe membership (no heartbeats) and this nod has not been reset (reset attempt has failed).
`INACTIVE`	HA services have not been started on this node.

When you start HA services, node states transition from `INACTIVE` to `UP`. It may happen that a node state may transition from `INACTIVE` to `UNKNOWN` to `UP`.

Monitoring Node Status with `cluster_status`

You can use the `cluster_status` command to monitor the status of the nodes in the cluster.

Monitoring Cluster Status with the GUI

You can use the GUI tree view to monitor the status of the clusters in a FailSafe configuration. Select **View: Groups owned by Nodes** to monitor the health of the default cluster, its resource groups, and the group's resources.

Querying Node Status with `cmgr`

To query node status, use the following command:

```
show status of node nodename
```

Performing a `ping` the System Controller with `cmgr`

When FailSafe is running, you can determine whether the system controller on a node is responding with the following command:

```
admin ping node nodename
```

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:

`admin ping standalone node` *nodename*

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.

Viewing System Status with the haStatus Script

The `haStatus` script provides status and configuration information about clusters, nodes, resources, and resource groups in the configuration. This script is installed in the `/var/cluster/cmgr-scripts` directory. You can modify this script to suit your needs. See the `haStatus`(1M) man page for further information about this script.

The following examples show the output of the different options of the `haStatus` script.

```
# haStatus -help
Usage: haStatus [-a|-i] [-c clustername]
where,
 -a prints detailed cluster configuration information and cluster
status.
 -i prints detailed cluster configuration information only.
 -c can be used to specify a cluster for which status is to be printed.
 ''clustername'' is the name of the cluster for which status is to be
printed.
# haStatus
Tue Nov 30 14:12:09 PST 1999
Cluster test-cluster:
        Cluster state is ACTIVE.
Node hans2:
        State of machine is UP.
Node hans1:
        State of machine is UP.
Resource_group nfs-group1:
        State: Online
```

```
        Error: No error
        Owner: hans1
        Failover Policy: fp_h1_h2_ord_auto_auto
        Resources:
                /hafs1  (type: NFS)
                /hafs1/nfs/statmon      (type: statd)
                150.166.41.95   (type: IP_address)
                /hafs1  (type: filesystem)
                havol1  (type: volume)
# haStatus -i
Tue Nov 30 14:13:52 PST 1999
Cluster test-cluster:
Node hans2:
        Logical Machine Name: hans2
        Hostname: hans2.engr.sgi.com
        Is FailSafe: true
        Is CXFS: false
        Nodeid: 32418
        Reset type: powerCycle
        System Controller: msc
        System Controller status: enabled
        System Controller owner: hans1
        System Controller owner device: /dev/ttyd2
        System Controller owner type: tty
        ControlNet Ipaddr: 192.26.50.15
        ControlNet HB: true
        ControlNet Control: true
        ControlNet Priority: 1
        ControlNet Ipaddr: 150.166.41.61
        ControlNet HB: true
        ControlNet Control: false
        ControlNet Priority: 2
Node hans1:
        Logical Machine Name: hans1
        Hostname: hans1.engr.sgi.com
        Is FailSafe: true
        Is CXFS: false
        Nodeid: 32645
        Reset type: powerCycle
        System Controller: msc
        System Controller status: enabled
```

```
        System Controller owner: hans2
        System Controller owner device: /dev/ttyd2
        System Controller owner type: tty
        ControlNet Ipaddr: 192.26.50.14
        ControlNet HB: true
        ControlNet Control: true
        ControlNet Priority: 1
        ControlNet Ipaddr: 150.166.41.60
        ControlNet HB: true
        ControlNet Control: false
        ControlNet Priority: 2
Resource_group nfs-group1:
        Failover Policy: fp_h1_h2_ord_auto_auto
                Version: 1
                Script: ordered
                Attributes: Auto_Failback Auto_Recovery
                Initial AFD: hans1 hans2
        Resources:
                /hafs1  (type: NFS)
                /hafs1/nfs/statmon      (type: statd)
                150.166.41.95   (type: IP_address)
                /hafs1  (type: filesystem)
                havol1  (type: volume)
Resource /hafs1 (type NFS):
        export-info: rw,wsync
        filesystem: /hafs1
        Resource dependencies
        statd /hafs1/nfs/statmon
        filesystem /hafs1
Resource /hafs1/nfs/statmon (type statd):
        InterfaceAddress: 150.166.41.95
        Resource dependencies
        IP_address 150.166.41.95
        filesystem /hafs1
Resource 150.166.41.95 (type IP_address):
        NetworkMask: 0xffffff00
        interfaces: ef1
        BroadcastAddress: 150.166.41.255
        No resource dependencies
Resource /hafs1 (type filesystem):
        volume-name: havol1
```

```
        mount-options: rw,noauto
        monitor-level: 2
        Resource dependencies
        volume havol1
Resource havol1 (type volume):
        devname-group: sys
        devname-owner: root
        devname-mode: 666
        No resource dependencies
Failover_policy fp_h1_h2_ord_auto_auto:
        Version: 1
        Script: ordered
        Attributes: Auto_Failback Auto_Recovery
        Initial AFD: hans1 hans2
# haStatus -a
Tue Nov 30 14:45:30 PST 1999
Cluster test-cluster:
        Cluster state is ACTIVE.
Node hans2:
        State of machine is UP.
        Logical Machine Name: hans2
        Hostname: hans2.engr.sgi.com
        Is FailSafe: true
        Is CXFS: false
        Nodeid: 32418
        Reset type: powerCycle
        System Controller: msc
        System Controller status: enabled
        System Controller owner: hans1
        System Controller owner device: /dev/ttyd2
        System Controller owner type: tty
        ControlNet Ipaddr: 192.26.50.15
        ControlNet HB: true
        ControlNet Control: true
        ControlNet Priority: 1
        ControlNet Ipaddr: 150.166.41.61
        ControlNet HB: true
        ControlNet Control: false
        ControlNet Priority: 2
Node hans1:
        State of machine is UP.
```

```
        Logical Machine Name: hans1
        Hostname: hans1.engr.sgi.com
        Is FailSafe: true
        Is CXFS: false
        Nodeid: 32645
        Reset type: powerCycle
        System Controller: msc
        System Controller status: enabled
        System Controller owner: hans2
        System Controller owner device: /dev/ttyd2
        System Controller owner type: tty
        ControlNet Ipaddr: 192.26.50.14
        ControlNet HB: true
        ControlNet Control: true
        ControlNet Priority: 1
        ControlNet Ipaddr: 150.166.41.60
        ControlNet HB: true
        ControlNet Control: false
        ControlNet Priority: 2
Resource_group nfs-group1:
        State: Online
        Error: No error
        Owner: hans1
        Failover Policy: fp_h1_h2_ord_auto_auto
                Version: 1
                Script: ordered
                Attributes: Auto_Failback Auto_Recovery
                Initial AFD: hans1 hans2
        Resources:
                /hafs1  (type: NFS)
                /hafs1/nfs/statmon      (type: statd)
                150.166.41.95   (type: IP_address)
                /hafs1  (type: filesystem)
                havol1  (type: volume)
Resource /hafs1 (type NFS):
        State: Online
        Error: None
        Owner: hans1
        Flags: Resource is monitored locally
        export-info: rw,wsync
        filesystem: /hafs1
```

```
        Resource dependencies
        statd /hafs1/nfs/statmon
        filesystem /hafs1
Resource /hafs1/nfs/statmon (type statd):
        State: Online
        Error: None
        Owner: hans1
        Flags: Resource is monitored locally
        InterfaceAddress: 150.166.41.95
        Resource dependencies
        IP_address 150.166.41.95
        filesystem /hafs1
Resource 150.166.41.95 (type IP_address):
        State: Online
        Error: None
        Owner: hans1
        Flags: Resource is monitored locally
        NetworkMask: 0xffffff00
        interfaces: ef1
        BroadcastAddress: 150.166.41.255
        No resource dependencies
Resource /hafs1 (type filesystem):
        State: Online
        Error: None
        Owner: hans1
        Flags: Resource is monitored locally
        volume-name: havol1
        mount-options: rw,noauto
        monitor-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)
```

Embedded Support Partner (ESP) Logging of FailSafe Events

The Embedded Support Partner (ESP) consists of a set of daemons that perform various monitoring activities. You can choose to configure ESP so that it will log FailSafe events (the FailSafe ESP event profile is not configured in ESP by default).

FailSafe uses an event class ID of 77 and a description of `IRIS FailSafe2`.

If you want to use ESP for FailSafe, enter the following command to add the `failsafe2` event profile to ESP:

```
# espconfig -add eventprofile failsafe2
```

FailSafe will then log ESP events for the following:

- Daemon configuration error
- Failover policy configuration error
- Resource group allocation (`start`) failure
- Resource group failures:
  - Allocation (`start`) failure

  - Release (`stop`) failure
  - Monitoring failure
  - Exclusivity failure
  - Failover policy failure
- Resource group status:
  - `online`
  - `offline`
  - `maintenance_on`
  - `maintenance_off`
- FailSafe shutdown (HA services stopped)
- FailSafe started (HA services started)

You can use the `espreport`(1M) or `launchESPartner`(1) commands to see the logged ESP events. See the `esp`(5) man page and the *Embedded Support Partner User Guide* for more information about ESP.

Resource Group Failover

While a FailSafe system is running, you can move a resource group online to a particular node, or you can take a resource group offline. In addition, you can move a resource group from one node in a cluster to another node in a cluster. The following subsections describe these tasks.

Bring a Resource Group Online

This section describes how to bring a resource group online.

Bring a Resource Group Online with the GUI

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.

You cannot bring a resource group online in the following circumstances:

- If the resource group has no members
- If the resource group is currently running in the cluster

To bring a resource group fully online, HA services must be active. When HA services are active, an attempt is made to allocate the resource group in the cluster. However, you can also execute a command to bring the resource group online when HA services are not active. When HA services are not active, the resource group is marked to be brought online when HA services become active; the resource group is then in an `ONLINE-READY` state. Failsafe tries to bring a resource group in an `ONLINE-READY` state online when HA services are started.

You can disable resource groups from coming online when HA services are started by using the GUI or `cmgr` to take the resource group offline, as described in "Take a Resource Group Offline", page 252.

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 "Take a Resource Group Offline", page 252.

Do the following:

1. **Group to Bring Online:** use the pull-down list to select the name of the resource group you want to bring online. The menu displays only resource groups that are not currently online.
2. Click on **OK** to complete the task.

Bring a Resource Group Online with `cmgr`

To bring a resource group online, use the following command:

`admin online resource_group` *RG_name* `[in cluster` *clustername*`]`

If you have specified a default cluster, you do not need to specify a cluster when you use this command.

For example:

```
cmgr> set cluster test-cluster
cmgr> admin online resource_group group1
FailSafe daemon (ha_fsd) is not running on this local node or it is not ready to accept admin commands.
Resource Group (group1) is online-ready.

Failed to admin:
        online

admin command failed

cmgr> show status of resource_group group1 in cluster test-cluster

State: Online Ready
Error: No error
Check resource group group1 status in an active node if HA services are active in cluster
```

Take a Resource Group Offline

This section tells you how to take a resource group offline.

Take a Resource Group Offline with the GUI

When you take a resource group offline, FailSafe takes each resource in the resource group offline in a predefined order. If any single resource gives an error during this process, the process stops, leaving all remaining resources allocated.

You can take a FailSafe resource group offline in any of the following 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 group. 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.

- Detach the resource group and force the error state to be cleared. 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. In addition, all error conditions of the resource group will be reset. 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 `cmgr`, as described in "Suspend and Resume Monitoring of a Resource Group", page 255.

If the `fsd` daemon is not running or is not ready to accept client requests, executing this task disables the resource group in the cluster database only. The resource group remains online and the command fails.

Enter the following:

1. **Detach Only:** check this box to stop monitoring the resource group. The resource group will not be stopped, but FailSafe will not have any control over the group.
2. **Detach Force:** check this box to stop monitoring the resource group. The resource group will not be stopped, but FailSafe will not have any control over the group. In addition, Failsafe will clear all errors.

Caution: The **Detach Only** and **Detach Force** settings leave the resource group's resources running on the node where the group was online. After stopping HA services on that node, do not bring the resource group online on another node in the cluster; doing so can cause data integrity problems. Instead, make sure that no resources are running on a node before stopping HA services on that node.

3. **Force Offline:** check this box to stop all resources in the group and clear all errors.
4. **Force Offline:** check this box to stop all resources in the group and clear all errors.
5. **Group to Take Offline:** select the name of the resource group you want to take offline. The menu displays only resource groups that are currently online.
6. Click on **OK** to complete the task.

Take a Resource Group Offline with cmgr

To take a resource group offline, use the following command:

```
admin offline resource_group RG_name [in cluster clustername]
```

To take a resource group offline with the force option in effect, forcing FailSafe to complete the action even if there are errors, use the following command:

```
admin offline_force resource_group RG_name [in cluster clustername]
```

To detach a resource group, use the following command:

```
admin offline_detach resource_group RG_name [in cluster clustername]
```

To detach the resource group and force the error state to be cleared:

```
admin offline_detach_force resource_group RG_name [in cluster clustername]
```

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. In addition, all error conditions of the resource group will be reset. This operation should rarely fail.

Move a Resource Group

This section tells you how to move a resource group.

Move a Resource Group with the GUI

While FailSafe is active, you can move a resource group to another node in the same cluster.

Note: When you move a resource group in an active system, you may find the unexpected behavior that the command appears to have succeeded, but the resource group remains online on the same node in the cluster. This can occur if the resource group fails to start on the node to which you are moving it. In this case, FailSafe will fail over the resource group to the next node in the application failover domain, which may be the node on which the resource group was originally running. Since FailSafe kept the resource group online, the command succeeds.

Do the following:

1. **Group to Move:** select the name of the resource group to be moved. Only resource groups that are currently online are displayed in the menu.
2. **Failover Domain Node:** *(optional)* select the name of the node to which you want to move the resource group. If you do not specify a node, FailSafe will move the resource group to the next available node in the failover domain.
3. Click on **OK** to complete the task.

Move a Resource Group with `cmgr`

To move a resource group to another node, use the following command:

```
admin move resource_group RG_name [in cluster clustername] [to node nodename]
```

Suspend and Resume Monitoring of a Resource Group

This section describes how to stop monitoring of a resource group in order to put it into maintenance mode.

Suspend Monitoring a Resource Group with the GUI

You can temporarily stop FailSafe from monitoring a specific resource group, which puts the resource group in maintenance mode. The resource group remains on the same node in the cluster but is no longer monitored by FailSafe for resource failures.

You can put a resource group into maintenance mode if you do not want FailSafe to monitor the group for a period of time. You may want to do this for upgrade or testing purposes, or if there is any reason that 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.

Do the following:

1. **Group to Stop Monitoring:** select the name of the group you want to stop monitoring. Only those resource groups that are currently online and monitored are displayed in the menu.
2. Click **OK** to complete the task.

Resume Monitoring of a Resource Group with the GUI

This task lets you resume monitoring for a resource group that FailSafe is not monitoring currently. (All resource groups that are in online state without error are monitored by default.)

Once monitoring is resumed, if the resource group or one of its resources fails, FailSafe will restart each failed component based on the failover policy (assuming that the restart action is enabled).

Perform the following steps:

1. **Group to Start Monitoring:** select the name of the group you want to stop monitoring. Only those resource groups that are currently online and not monitored are displayed in the menu.
2. Click **OK** to complete the task.

Putting a Resource Group into Maintenance Mode with `cmgr`

To put a resource group into maintenance mode, use the following command:

`admin maintenance_on resource_group` *RG_name* `[in cluster` *clustername*`]`

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 `cmgr`

To move a resource group back online from maintenance mode, use the following command:

`admin maintenance_off resource_group` *RG_name* `[in cluster` *clustername*`]`

Stopping FailSafe

You can stop the execution of FailSafe on all the nodes in a cluster or on a specified node only. See "Stop FailSafe HA Services", page 200.

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 that is used depends on how you have configured your system.

After 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.

Reset a Node with the GUI

You can use the GUI 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 the change and remove the node from the active cluster. When the node reboots, it will rejoin the FailSafe membership.

To reset a node, do the following:

1. **Node to Reset:** use the pull-down menu to select the node to be reset.
2. Click on **OK** to complete the task.

Reset a Node with `cmgr`

When FailSafe is running, you can reboot a node with the following command:

`admin reset node` *nodename*

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:

`admin reset standalone node` *nodename*

This command does not go through the FailSafe daemons.

Backing Up and Restoring Configuration with cmgr

The cmgr command provides scripts that you can use to backup and restore your configuration: cdbBackup and cdbRestore. These scripts are installed in the /usr/cluster/bin directory. You can modify these scripts to suit your needs.

The cdbBackup 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 cdbBackup and cdbRestore scripts, you should follow the following procedures:

- Run the cdbBackup and cdbRestore scripts only when no administrative commands are running. This could result in an inconsistent backup.
- You must back up 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 cluster database. Check the SYSLOG file for information to help determine when cluster database activity is occurring. As a rule of thumb, you should be able to perform a cdbDump if at least 15 minutes have passed since the last node joined the cluster or the last administration command was run.

Log File Management

You should rotate the log files at least weekly so that your disk will not become full.

The following sections provide example scripts. You may want to consider placing an entry in the `root crontab`(1) to run such scripts periodically.

For information about log levels, see "Set Log Configuration", page 206.

Rotating All Log Files

You can use a script such as the following to copy all files to a new location.

```
#!/bin/sh

DATE=‘/sbin/date +’%U-%a’‘
LOG_DIR="/var/cluster/ha/log"
HOST=‘/usr/bsd/hostname -s‘
LOG_FILES="cad_log cmond_log fs2d_log"
LOG_HFILES="cli cmsd crsd failsafe gcd ifd script srmd clconfd"

LOG_ARCH=$LOG_DIR"/Old-Log"

if [ ! -d $LOG_ARCH ] ; then
   mkdir $LOG_ARCH
fi

for file in $LOG_FILES
do

  rm -f ${LOG_ARCH}/${file}-${DATE}
  cp ${LOG_DIR}/${file} ${LOG_ARCH}/${file}-${DATE}
  echo "Log Rotation at ‘date‘" > ${LOG_DIR}/${file}
done

for file in $LOG_HFILES
do

  rm -f ${LOG_ARCH}/${file}_${HOST}-${DATE}
  cp ${LOG_DIR}/${file}_${HOST} ${LOG_ARCH}/${file}_${HOST}-${DATE}
  echo "Log Rotation at ‘date‘" > ${LOG_DIR}/${file}_${HOST}
done
```

The script can be executed as a `cron`(1) job to regularly clean up log files. This script rotates log files when HA services are active in the FailSafe cluster. Default log levels do not create large log files.

Testing the Configuration

This chapter explains how to test the IRIS FailSafe system configuration using the FailSafe Manager GUI and the `cmgr`(1M) command. For general information on using these tools, see Chapter 4, "Administration Tools", page 87.

The sections in this chapter are as follows:

- "Overview of FailSafe Diagnostic Commands"
- "Performing Diagnostic Tasks with the GUI", page 262
- "Performing Diagnostic Tasks with `cmgr`", page 263

Overview of FailSafe Diagnostic Commands

Table 8-1 shows the tests you can perform with FailSafe diagnostic commands:

Table 8-1 FailSafe Diagnostic Test Summary

Diagnostic Test	Description
Resource	Checks that: • Resource type parameters are set • Parameters are syntactically correct • Parameters exist
Resource group	Tests all resources defined in the resource group
Failover policy	Checks that: • Failover policy exists • Failover domain contains a valid list of hosts
Network connectivity	Checks that: • The control interfaces are on the same network • 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 that could prevent the resource group from starting successfully.

Performing Diagnostic Tasks with the GUI

This section describes how to perform diagnostic tasks with the GUI.

Test Connectivity with the GUI

This task requires `root rsh`(1) access between nodes. To test connectivity, do the following from the **FailSafe Manager**:

Note: The **Test Node Connectivity** screen requires `rsh`(1) access between hosts. The `/.rhosts` file must contain the hosts and local host between which you want to test connectivity.

- Choose whether to test by network or serial connectivity by clicking on the appropriate radio button.
- Choose a node to be tested from the pull-down list and add it to the test list by clicking on **Add**.

  To delete a node from the list of nodes to be tested, click on the logical name to select it and then click on **Delete**.
- To start the tests, click on **Start Tests**. To stop the tests, click on **Stop Tests**.
- To run another test, click on **Clear Output** to clear the status screen and start over with step 3.
- To exit from the window, click on **Close**.

Test Resources with the GUI

The **Test Resources** task lets you 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.

Test Failover Policies with the GUI

The **Test Failover Policy** task lets you test whether a failover policy is defined correctly. This test checks the failover policy by validating the policy script and failover attributes, and whether the application failover domain consists of valid nodes from the cluster.

Performing Diagnostic Tasks with `cmgr`

The following subsections described how to perform diagnostic tasks on your system using the `cmgr` command.

Test the Serial Connections with `cmgr`

You can use the `cmgr` command to test the serial connections between the FailSafe nodes. This test performs a `ping`(1M) on each specified node through the serial line and produces an error message if the `ping` is not successful.

Note: Do not execute this command while FailSafe is running.

Use the following command to test the serial connections for the machines in a cluster:

`test serial in cluster` *C_name* `[on node` *node1* `node` *node2* `...]`

For example, to test multiple nodes:

```
cmgr> test serial in cluster test-cluster on node blue node green
```

The serial 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.

For example:

```
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` 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>
```

Test Network Connectivity with `cmgr`

You can use the `cmgr` command 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.

Use the following command to test the network connectivity for the machines in a cluster:

`test connectivity in cluster` *C_name* `[on node` *node1* `node` *node2 ...*`]`

The following shows an example of the `test connectivity` 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.

Test Resources with `cmgr`

You can use the `cmgr` command to test any configured resource by resource name or by resource type.

Use the following to test a resource by name:

```
test resource resourcename of resource_type RT_name in cluster C_name [on node node1 node node2 ...]
```

For example:

```
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
```

Use the following to test a resource by resource type:

```
test resource_type RT_name in cluster C_name [on node node1 node node2 ...]
```

For example:

```
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 `cmgr` to test volume and filesystem resources in `destructive` mode. This provides a more thorough test of filesystems and volumes. `cmgr` tests will not run in destructive mode if FailSafe is running.

Use the following to test resources in destructive mode:

```
test resource resourcename of resource_type RT_name in cluster C_name [on node node1 node node2 ...] destructive
```

The following sections describe the diagnostic tests available for resources.

Test Logical Volumes with `cmgr`

You can use the `cmgr` command to test the logical volumes in a cluster. This test checks if the specified volume is configured correctly.

Use the following command to test a logical volume:

```
test resource resourcename of resource_type volume on cluster C_name[on node node1 node node2 ...]
```

For example:

```
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
```

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
```

Test Filesystems with cmgr

You can use `cmgr` to test the filesystems configured in a cluster. This test checks if the specified filesystem is configured correctly and if the volume the filesystem will reside on is configured correctly.

Use the following command to test a filesystem:

```
test resource resourcename of resource_type filesystems on cluster C_name [on node node1 node node2 ...]
```

The following example displays the filesystems that have been defined in a cluster and tests one of them:

```
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
```

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
```

Test Resource Groups with cmgr

You can use cmgr 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.

Use the following to test resource groups:

```
test resource_group RG_name in cluster C_name [on node node1 node node2 ...]
```

The following displays the resource groups that have been defined in a cluster and test one of them:

```
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
```

Test Failover Policies with cmgr

You can use cmgr 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.

Use the following to test a failover policy:

```
test failover_policy FP_name in cluster C_name [on node node1 node node2 ...]
```

The following example uses a show command to display the failover policies that have been defined in a cluster and tests one of them:

```
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
```

System Recovery and Troubleshooting

This chapter provides information on FailSafe system recovery, and includes sections on the following topics:

- "Overview of System Recovery"
- "Disabling Resource Groups for Maintenance", page 272
- "FailSafe Log Files", page 272
- "FailSafe Membership and Resets", page 274
- "Status Monitoring", page 276
- "Dynamic Control of FailSafe Services", page 276
- "Recovery Procedures", page 277
- "CXFS Metadata Server Relocation", page 288
- "Other Problems with CXFS Coexecution", page 288

Overview of System Recovery

When a FailSafe system experiences problems, you can use some of the FailSafe features and commands to determine where the problem is located.

FailSafe provides the following tools to evaluate and recover from system failure:

- Log files
- Commands to monitor status of system components
- Commands to start, stop, and fail over highly available services

Keep in mind that the FailSafe logs may not detect system problems that do not translate into FailSafe problems. For example, if a CPU goes bad, or hardware maintenance is required, FailSafe may not be able to detect and log these failures.

In general, when evaluating system problems of any nature on a FailSafe configuration, you should determine whether you need to shut down a node to address those problems.

When you shut down a node, perform the following steps:

1. Stop FailSafe HA services on that node
2. Shut down the node to perform needed maintenance and repair
3. Start up the node
4. Start FailSafe HA 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.

Disabling Resource Groups for Maintenance

If you must disable resources, such as when you want to perform maintenance on a node, use the following procedure:

1. Offline the resource groups by using the `offline_detach` option or `offline_detach_force` option (if the resource group is in error). For more information, see "Resource Group Recovery", page 279 , and "Resource Group Maintenance and Error Recovery", page 280.
2. Perform the needed maintenance.
3. Reboot the node.
4. Online the resource group.

FailSafe Log Files

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. Table 9-1 shows the levels of messages.

For information on setting logging for `cad`, `cmond`, and `fs2d`, see "Configure System Files", page 59. For information on setting up log configurations, see "Set Log Configuration", page 206 in Chapter 5, "Configuration", page 103.

Table 9-1 Message Levels

Message Level	Description
Normal	Normal messages report on the successful completion of a task. An example of a normal message is as follows (`<N` notation indicates a normal message): `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 (`<W` notation indications a warning. `<E` indicates an error.): `Wed Sep 2 13:45:47.199 <W crsd crs 9908:0 crs_config.c:634> CI_ERR_NOTFOUND, safer - no such node`
SYSLOG	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 `cmgr`). The following message is logged at `debug0` (see `D0` in the message) or log level `10`: `Thu Sep 27 14:43:24.233 <D0 ha_fsd fsd 57540:0 fs_failsafe.c:1471> Determine oldest state: coordinator: perf22/0x10001`

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 observe the activity of the various daemons immediately before error occurred, you may be able to determine what situation existed that caused the failure.

Note: Many megabytes of disk space can be consumed on the server when debug levels are used in a log configuration.

FailSafe Membership and Resets

In looking over the actions of a FailSafe system on failure to determine what has gone wrong and how processes have transferred, it is important to consider the concept of FailSafe membership. When failover occurs, the runtime failover domain can include only those nodes that are in the FailSafe membership.

FailSafe Membership and Tie-Breaker Node

Nodes can enter into the FailSafe membership only when they are not disabled and they are in a known state. This ensures that data integrity is maintained because only nodes within the FailSafe membership can access the shared storage. If nodes that are outside the membership and are not controlled by FailSafe were able to access the shared storage, two nodes might try to access the same data at the same time; this situation would result in data corruption. For this reason, disabled nodes do not participate in the membership computation.

Note: No attempt is made to reset nodes that are configured disabled before confirming the FailSafe membership.

FailSafe membership in a cluster is based on a quorum majority. For a cluster to be enabled, more than 50% of the nodes in the cluster must be in a known state, able to talk to each other, using heartbeat control networks. This quorum determines which nodes are part of the FailSafe membership that is formed.

If there are an even number of nodes in the cluster, it is possible that there will be no majority quorum; there could be two sets of nodes, each consisting of 50% of the total number of node, unable to communicate with the other set of nodes. In this case, FailSafe uses the node that has been configured as the tie-breaker node when you configured your FailSafe parameters. If no tie-breaker node was configured, FailSafe uses the node with the lowest ID number where HA services have been started.

The nodes in a quorum attempt to reset the nodes that are not in the quorum. Nodes that can be reset are declared `DOWN` in the membership, nodes that could not be reset are declared `UNKNOWN`. Nodes in the quorum are `UP`.

If a new majority quorum is computed, a new membership is declared whether any node could be reset or not.

If at least one node in the current quorum has a current membership, the nodes will proceed to declare a new membership if they can reset at least one node.

If all nodes in the new tied quorum are coming up for the first time, they will try to reset and proceed with a new membership only if the quorum includes the tie-breaker node.

If a tied subset of nodes in the cluster had no previous membership, then the subset of nodes in the cluster with the tie-breaker node attempts to reset nodes in the other subset of nodes in the cluster. If at least one node reset succeeds, a new membership is confirmed.

If a tied subset of nodes in the cluster had previous membership, the nodes in one subset of nodes in the cluster attempt to reset nodes in the other subset of nodes in the cluster. If at least one node reset succeeds, a new membership is confirmed. The subset of nodes in the cluster with the tie-breaker node resets immediately; the other subset of nodes in the cluster attempts to reset after some time.

Resets are done through system controllers connected to tty ports through serial lines. Periodic serial line monitoring never stops. If the estimated serial line monitoring failure interval and the estimated heartbeat loss interval overlap, the cause is likely a power failure at the node being reset.

No Membership Formed

When no FailSafe membership is formed, you should check the following areas for possible problems:

- Is the `ha_cmsd` FailSafe membership daemon running? Is the `fs2d` database daemon running?
- Can the nodes communicate with each other? Are the control networks configured as heartbeat networks?
- Can the control network addresses be reached by a `ping`(1M) command issued 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 `crsd` node control daemon 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 the location of system problems.

You can monitor the status of the FailSafe components continuously through their visual representation in the GUI tree view. Using the `cmgr` command, you can display the status of the individual components by using the `show` command.

For information on status monitoring and on the meaning of the states of the FailSafe components, see "System Status", page 233 of Chapter 7, "IRIS FailSafe System Operation".

Dynamic Control of FailSafe Services

FailSafe allows you to perform a variety of administrative tasks that can help you troubleshoot a system with problems without bringing down the entire system. These tasks include the following:

- You can add or delete nodes from a cluster without affecting the FailSafe services and the applications running in the cluster.
- You can add or delete a resource group without affecting other online resource groups.
- You can add or delete resources from a resource group while it is still online.
- You can change FailSafe parameters such as the heartbeat interval and the node timeout and have those values take immediate affect while the services are up and running.
- You can start and stop FailSafe services on specified nodes.
- You can move a resource group online, or take it offline.
- You can stop the monitoring of a resource group by putting the resource group into maintenance mode. This is not an expensive operation, as it does not stop

and start the resource group, it just puts the resource group in a state where it is not available to FailSafe.

- You can reset individual nodes.

For information on how to perform these tasks, see Chapter 5, "Configuration", page 103, and Chapter 7, "IRIS FailSafe System Operation".

Recovery Procedures

The following sections describe various recovery procedures you can perform when different failsafe components fail. Procedures for the following situations are provided:

- "Single-Node Recovery", page 278
- "Cluster Error Recovery", page 278
- "Resource Group Recovery", page 279
- "Node Error Recovery", page 279
- "Resource Group Maintenance and Error Recovery", page 280
- "Clear Resource Error State", page 283
- "Control Network Failure Recovery", page 284
- "Serial Cable Failure Recovery", page 284
- "Cluster Database Sync Failure", page 285
- "Cluster Database Maintenance and Recovery", page 285
- "GUI Will Not Run", page 285
- "GUI and `cmgr` Inconsistencies", page 287
- "GUI Does Not Report Information", page 287
- "Using the `cdbreinit` Command", page 287

Single-Node Recovery

When one of the nodes in a two-node cluster is intended to stay down for maintenance or cannot be brought up, a set of procedures must be followed so that the database on the surviving node knows that that node is down and therefore should not to be considered in the failover domain. Without these procedures, the surviving node will be in what is called the *lonely state*, meaning that it will never form a cluster by itself.

See the procedure in "Two-Node Clusters: Single-Node Use", page 227.

Cluster Error Recovery

Use the following procedure if status of the cluster is `UNKNOWN` in all nodes in the cluster:

1. Check to see if there are control networks that have failed (see "Control Network Failure Recovery", page 284).
2. Determine if there are sufficient nodes in the cluster that can communicate with each other using control networks in order to form a quorum. (At least 50% of the nodes in the cluster must be able to communicate with each other.) If there is an insufficient number of nodes, stop HA services on the nodes that cannot communicate (using the `force` option); this will change the number of nodes used in the quorum calculation.
3. If there are no hardware configuration problems, do the following:
   - Detach all resource groups that are online in the cluster (if any)
   - Stop HA services in the cluster
   - Restart HA services in the cluster

   See "Resource Group Recovery", page 279

For example, the following `cmgr` 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 command:

```
cmgr> stop ha_services for cluster web-cluster force
```

To start HA services in the cluster `web-cluster`, use the following command:

```
cmgr> start ha_services for cluster web-cluster
```

Resource Group Recovery

The fact that a resource group is in an error state does not mean that all resources in the resource group have failed. However, to get the resources back into an `online` state, you must first set them to the `offline` state. You can do without actually taking the resources offline by using the following `cmgr` command:

`admin offline_detach_force` *resource_group* `[in cluster` *clustername*`]`

For example:

```
cmgr> admin offline_detach_force RG1 in cluster test-cluster
```

Caution: You should use the `InPlace_Recovery` failover policy attribute when using this command. This attribute specifies that the resources will stay on the same node where they were running at the time when the `offline_detach_force` command was run.

Node Error Recovery

When a node is not able to talk to the majority of nodes in the cluster, the `SYSLOG` will display a message that the CMSD is in a lonely state. Another problem you may see is that a node is getting reset or going to an unknown state.

Use the following procedure to resolve node errors:

1. Check to see if the control networks in the node are working (see "Control Network Failure Recovery", page 284).
2. Check to see if the serial reset cables to reset the node are working (see "Serial Cable Failure Recovery", page 284).
3. Verify that the `sgi-cmsd` port is the same in all nodes in the cluster.
4. Check the node configuration; it should be consistent and correct.
5. Check `syslog` and `cmsd` logs for errors. If a node is not joining the cluster, check the logs of the nodes that are part of the cluster.

6. If there are no hardware configuration problems, stop HA services in the node and restart HA services.

   For example, to stop HA services in the node `web-node3` in the cluster `web-cluster`, ignoring errors (`force` option), use the following command:

```
cmgr> stop ha_services in node web-node3 for cluster web-cluster force
```

   For example, to start HA services in the node `web-node3` in the cluster `web-cluster`, use the following 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 must turn maintenance mode off when performing application maintenance.

Caution: If there is a 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.

For example:

1. To put a resource group `web-rg` in maintenance mode, use the following `cmgr` 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 command:

```
cmgr> admin maintenance_off resource_group web-rg in cluster web-cluster
```

   The resource group state changes back to `ONLINE`.

Perform the following procedure when a resource group is in an ONLINE state and has an SRMD EXECUTABLE ERROR:

1. Look at the SRM logs (default location: /var/cluster/ha/logs/srmd_*nodename*) to determine the cause of failure and the resource that has failed. Search for the ERROR string in the SRMD log file:

   ```
   Wed Nov 3 04:20:10.135
   <E ha_srmd srm 12127:1 sa_process_tasks.c:627>
   CI_FAILURE, ERROR: Action (start) for resource (192.0.2.45) of type
   (IP_address) failed with status (failed)
   ```

2. Check the script logs on that same node for IP_address start script errors.

3. Fix the cause of failure. This might require changes to resource configuration or changes to resource type stop/start/failover action timeouts.

4. After fixing the problem, move the resource group offline with the force option and then move the resource group online in the cluster.

   For example, the following 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 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.

Use the following procedure when a resource group is not online but is in an error state. Most of these errors occur as a result of the exclusivity process. This process, run when a resource group is brought online, determines if any resources are already allocated somewhere in the failure domain of a resource group. Note that exclusivity scripts return that a resource is allocated on a node if the script fails in any way. In other words, unless the script can determine that a resource is not present, it returns a value indicating that the resource is allocated.

Some possible error states include: SPLIT RESOURCE GROUP (EXCLUSIVITY), NODE NOT AVAILABLE (EXCLUSIVITY), NO AVAILABLE NODES in failure domain. See "Resource Group Status", page 238, for explanations of resource group error codes.

1. Look at the `failsafe` and SRMD logs (default directory: `/var/cluster/ha/logs`, files: `failsafe`_*nodename*, `srmd`_*nodename*) to determine the cause of the failure and the resource that failed.

   For example, suppose that the task of moving a resource group online results in a resource group with error state `SPLIT RESOURCE GROUP (EXCLUSIVITY)`. This means that parts of a resource group are allocated on at least two different nodes. One of the failsafe logs will have the description of which nodes are believed to have the resource group partially allocated:

`[Resource Group:`*RG_name*`]:Exclusivity failed -- RUNNING on` *node1* `and` *node2*

`[Resource Group:`*RG_name*`]:Exclusivity failed -- PARTIALLY RUNNING on` *node1* `and PARTIALLY RUNNING on` *node2*

   At this point, look at the `srmd` logs on each of these nodes for exclusive script errors to see what resources are believed to be allocated. In some cases, a misconfigured resource will show up as a resource that is allocated. This is especially true for `Netscape_web` resources.

2. Fix the cause of the failure. This might require changes to resource configuration or changes to resource type start/stop/exclusivity timeouts.

3. After fixing the problem, move the resource group offline with the `force` option and then move the resource group online.

Perform the following checks when a resource group shows a `no more nodes in AFD` error:

1. All nodes in the failover domain are not in the membership. Check CMSD logs for errors.

2. Check the SRMC/script logs on all nodes in the failover domain for start/monitor script errors.

There are a few double failures that can occur in the cluster that will cause resource groups to remain in a non-highly-available state. At times a resource group might be 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, do the following:

1. If the resource group is offline, try to move it online.

2. If the resource group is stuck on a node, detach the resource group and then bring it back 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 "Take a Resource Group Offline", page 252, for information on detaching resource groups and forcing resource groups offline.

Clear Resource Error State

Use this procedure when a resource that is not part of a resource group is in an `ONLINE` state with an error. This can happen when the addition or removal of resources from a resource group fails.

Do the following:

1. Look at the SRM logs to determine the cause of failure and the resource that has failed. The default location is:

   `/var/cluster/ha/logs/srmd_`*nodename*

2. Fix the problem that caused the failure. This might require changes to resource configuration or changes to resource type stop/start/failover action timeouts.
3. Clear the error state with the GUI or the `cmgr` command:
   - Use the **Clear Resource Error State** GUI task. Provide the following information:
     - **Resource Type:** select the type of the resource
     - **Resource in Error State:** select the name of the resource that should be cleared from the error state

     Click **OK** to complete the task.
   - Use the `cmgr admin offline_force` command to move the resource offline. For example, to remove the error state of resource `web-srvr` of type `Netscape_Web`, making it available to be added to a resource group, enter the following:

```
cmgr> admin offline_force resource web-srvr of resource_type Netscape_Web in cluster web-cluster
```

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*.

Check the node configuration to see whether serial cable connection is correctly configured.

The following `cmgr` command displays node configuration for `web-node3`

```
cmgr> show node web-node3
```

Use the `admin ping` command to verify the serial cables. The following command reports serial cables problems in node `web-node3`:

```
cmgr> admin ping node web-node3
```

Cluster Database Sync Failure

If the cluster database synchronization fails, use the following procedure:

1. Check for the following message in the `/var/adm/SYSLOG` file on the target node:

   ```
   Starting to receive CDB sync series from machine <node1_node_ID>
   ...
   Finished receiving CDB sync series from machine <node1_node_ID>
   ```

2. Check for control network or `portmapper/rpcbind` problems.
3. Check the node definition in the cluster database.
4. Check the `SYSLOG` and `fs2d` logs on the source node.

Cluster Database Maintenance and Recovery

When the entire cluster database must be reinitialized, execute the following command:

```
# /usr/cluster/bin/cdbreinit /var/cluster/cdb/cdb.db
```

This command restarts all cluster processes. The contents of the cluster database will be automatically synchronized with other nodes if other nodes in the pool are available.

Otherwise, the cluster database must be restored from backup at this point. For instructions on backing up and restoring the cluster database, see "Backing Up and Restoring Configuration with `cmgr`", page 258.

GUI Will Not Run

If the GUI will not run, check the following:

- Are the cluster daemons running?

  When you first install the software, the following daemons should be running:

  - `fs2d`
  - `cmond`

- cad
- crsd

To determine which daemons are running, enter the following:

```
# ps -ef | grep cluster
```

The following shows an example of the output when just the initial daemons are running; for readability, whitespace has been removed and the daemon names are highlighted:

```
fs6 # ps -ef | grep cluster
root 31431     1 0 12:51:36 ?     0:14 /usr/lib32/cluster/cbe/fs2d /var/cluster/cdb/cdb.db #
root 31456 31478 0 12:53:01 ?     0:03 /usr/cluster/bin/crsd -l
root 31475 31478 0 12:53:00 ?     0:08 /usr/cluster/bin/cad -l -lf /var/cluster/ha/log/cad_log --append_log
root 31478     1 0 12:53:00 ?     0:00 /usr/cluster/bin/cmond -L info -f /var/cluster/ha/log/cmond_log
root 31570 31408 0 14:01:52 pts/0 0:00 grep cluster
```

If you do not see these processes, go to the logs to see what the problem might be. If you must restart the daemons, enter the following:

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

- Are the `tcpmux` and `tcpmux/sgi_sysadm` services enabled in the `/etc/inetd.conf` file?

  The following line is added to the `/etc/inetd.conf` file when `sysamd_base` is installed:

```
tcpmux/sgi_sysadm stream tcp nowait root   ?/usr/sysadm/bin/sysadmd sysadmd
```

  If the `tcpmux` line is commented out, you must uncomment it and then run the following:

```
# kill -HUP inetd
```

- Are the `inetd` or `tcp` wrappers interfering? This may be indicated by `connection refused` or `login failed` messages.
- Are you connecting to an IRIX node? The `fstask`(1M) command can only be executed on an IRIX node. The GUI may be run from a node running an operating system other than IRIX via the Web if you connect the GUI to an IRIX node.

GUI and cmgr Inconsistencies

If the GUI is displaying information that is inconsistent with the FailSafe cmgr command, restart cad process on the node to which GUI is connected to by executing the following command:

```
# killall cad
```

The cluster administration daemon is restarted automatically by the cmond process.

GUI Does Not Report Information

If the GUI is not reporting configuration information and status, perform the following steps:

1. Check the information using the cmgr(1M) command. If cmgr is reporting correct information, there is a GUI update problem.
2. If there is a GUI update problem, kill the cad daemon on that node. Wait for a couple of minutes to see whether cad gets correct information. Check the cad logs on that node for errors.
3. Check the CLI logs on that node for errors.
4. If the status information is incorrect, check the cmsd or fsd logs on that node.

Using the cdbreinit Command

When the cluster databases are not in synchronization on all the nodes in the cluster, you can run the cdbreinit command to recover. The cdbreinit command should be run on the node which is not in sync.

Perform the following steps.

Note: Perform each step on all the nodes before proceeding to the next step in the recovery procedure.

---

1. Stop FailSafe services in the cluster using the GUI or cmgr.

2. Stop cluster processes on all nodes in the pool:

   ```
   # /etc/init.d/cluster stop
   # killall fs2d
   ```

3. Run `cdbreinit` on the node where the cluster database is not in sync.

4. Start cluster processes on all nodes in the pool:

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

5. Wait a couple of minutes for the cluster database to sync. There will be cluster database sync long messages in the `SYSLOG` on the node.

6. Start FailSafe services in the cluster.

CXFS Metadata Server Relocation

FailSafe uses a `umount`(1M) command with the `-k` option to move a resource in the case of a CXFS metadata server relocation if the `relocate-mds` attribute in the CXFS resource definition is set to `true`. The `umount -k` command will kill all server process using the CXFS filesystem.

Other Problems with CXFS Coexecution

For information solving problems involving coexecution with CXFS, see the troubleshooting chapter of the *CXFS Version 2 Software Installation and Administration Guide*.

Upgrading and Maintaining Active Clusters

When an IRIS FailSafe system is running, you may need to perform various administration procedures without shutting down the entire cluster. This chapter provides instructions for performing upgrade and maintenance procedures on active clusters. It includes the following procedures:

- "Add a Node to an Active Cluster"
- "Delete a Node from an Active Cluster", page 291
- "Change Control Networks in a Cluster", page 293
- "Upgrade OS Software in an Active Cluster", page 295
- "Upgrade FailSafe Software in an Active Cluster", page 296
- "Add New Resource Groups or Resources in an Active Cluster", page 297
- "Adding a New Hardware Device in an Active Cluster", page 298

Add a Node to an Active Cluster

Use the following procedure to add a node to an active cluster. This procedure assumes that `cluster_admin`, `cluster_control`, `cluster_ha`, and `failsafe2` products are already installed in this node.

1. Check control network connections from the node to the rest of the cluster using `ping`(1M) command. Note the list of control network IP addresses.
2. Check the serial connections to reset this node. Note the name of the node that can reset this node.
3. Run node diagnostics. For information on FailSafe diagnostic commands, see Chapter 8, "Testing the Configuration", page 261.
4. Make sure that the `sgi-cad`, `sgi-crsd`, `sgi-cmsd`, and `sgi-gcd` entries are present in the `/etc/services` file. The port numbers for these processes should match the port numbers in other nodes in the cluster.

Example entries:

```
sgi-cad         7200/tcp      # SGI cluster admin daemon
sgi-crsd        7500/udp      # SGI cluster reset services daemon
sgi-cmsd        7000/udp      # SGI FailSafe membership Daemon
sgi-gcd         8000/udp      # SGI group communication Daemon
```

5. Check if the 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 the GUI, the `cmgr` command, or the `cmgr` template (`/var/cluster/cmgr-templates/cmgr-create-node`), to define the node.

   Note: This node must be defined from one of nodes that is already in the cluster.

7. Use the `cmgr` command or the GUI to add the node to the cluster.

   For example: the following `cmgr` 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. Start highly available (HA) services on this node using the `cmgr` command to the GUI.

   For example, the following `cmgr` 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. Add this node to the failure domain of the relevant failover policy. In order to do this, the entire failover policy must be redefined, including the additional node in the failure domain.

Delete a Node from an Active Cluster

Use the following procedure to delete a node from an active cluster. This procedure assumes that the node status is `UP`.

1. If resource groups are online on the node, use the `cmgr` command or the GUI 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 `cmgr` command or the GUI 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 redefined, deleting the affected node from the failure domain.

3. Stop HA services on the node.

   For example, to stop HA services on the node `web-node3`, use the following `cmgr` 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. If there are resources that cannot 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 cluster database.

4. Delete the node from the cluster.

   For example, to delete node `web-node3` from `web-cluster` configuration, use the following `cmgr` 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 cluster database.

   The following `cmgr` command deletes the `web-node3` node definition from the cluster database:

   ```
   cmgr> delete node web-node3
   ```

6. Stop all cluster processes and delete the cluster database.

   The following commands stop cluster processes on the node and delete the cluster 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
   ```

Change Control Networks in a Cluster

Use the following procedure to change the control networks in a currently active cluster. This procedure is valid for a two-node cluster consisting of nodes `node1` and `node2`. In this procedure, you must complete each step before proceeding to the next step.

Note: Do not perform any other administration operations during this procedure.

---

1. From any node, stop HA services on the cluster. Make sure all HA processes have exited on both nodes.
2. From `node2`, stop the cluster processes on `node2`:

   ```
   # /etc/init.d/cluster stop
   # killall fs2d
   ```

   Make sure the `fs2d` process have been killed on `node2`.

3. From `node1`, modify the `node1` and `node2` definition. Use the GUI or the following `cmgr` commands:

   ```
   cmgr> modify node node1
   Enter commands, when finished enter either "done" or "cancel"
   node1?> remove nic old_nic _address
   node1> add nic new_nic_address
   NIC - new_nic_address set heartbeat to ...
   ```

```
NIC - new_nic_address set ctrl_msgs to ...
NIC - new_nic_address set priority to ...
NIC - new_nic_address done
node1? done
```

Repeat the same procedure to modify node2.

4. From node1, check if the node1 and node2 definitions are correct. Using cmgr on node1, execute the following commands to view the node definitions:

```
cmgr> show node node1
cmgr> show node node2
```

5. On both node1 and node2, modify the network interface IP addresses in /etc/config/netif.options and execute ifconfig to configure the new IP addresses on node1 and node2. Verify that the IP addresses match the node definitions in the cluster database.

6. From node1, stop the cluster process on node1:

```
# /etc/init.d/cluster stop
# killall fs2d
```

Make sure the fs2d process have been killed on node1.

7. From node2, execute the following command to start cluster process on node2:

```
# /usr/cluster/bin/cdbreinit /var/cluster/cdb/cdb.db
```

Answer **y** to the prompt.

8. From node1, start cluster processes on node1:

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

The following messages should appear in the SYSLOG on node2:

```
Starting to receive CDB sync series from machine <node1 node id>
...
Finished receiving CDB sync series from machine <node1 node id>
```

Wait for approximately 60 seconds for the synchronization to complete.

9. From any node, start HA services in the cluster.

Upgrade 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 `cmgr` command or the GUI 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 `cmgr` 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 `cmgr` command or the GUI. 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. For example, the following `cmgr` 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.

Upgrade 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 `cmgr` command or the GUI 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.

   For example, the following `cmgr` 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 `cmgr` command or the GUI. 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. For example, the following `cmgr` 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.

Add 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), perform 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 highly available (HA) 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. You can use the following `cmgr` template to create the failover policy:

   ```
   /var/cluster/cmgr-templates/cmgr-create-failover_policy
   ```

4. Configure the resources in cluster database. There are `cmgr` 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 cluster database. Create the resource dependencies for these resources.

5. Run resource diagnostics. For information on the diagnostic commands, see Chapter 8, "Testing the Configuration", page 261.

6. Create resource group and add resources to the resource group. You can use the following `cmgr` template to create resource group and add resources to resource group:

   ```
   /var/cluster/cmgr-templates/cmgr-create-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

You will add new hardware devices to an active cluster 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 "Upgrade OS Software in an Active Cluster", page 295. 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 cluster database, you must modify your resource configuration and your node configuration, where appropriate.

Performance Co-Pilot for FailSafe

This chapter tells you how to use Performance Co-Pilot (PCP) for FailSafe to monitor the availability of an IRIX FailSafe cluster. For information about installing PCP for FailSafe, see "Install Performance Co-Pilot (PCP) Software", page 79.

PCP provides the following:

- An agent for exporting FailSafe heartbeat and resource monitoring statistics to the PCP framework
- 3-D visualization tools for displaying these statistics in an intuitive presentation

The visualization of statistics provides valuable information about the availability of nodes and resources monitored by FailSafe. For example, it can highlight a reduction in monitoring response times that may indicate problems in availability of services provided by the cluster.

Because PCP for FailSafe is an extension to the PCP framework, you can use other PCP tools to analyze or present FailSafe monitoring statistics, and record PCP for FailSafe metrics as archives for deferred analysis. You can also use PCP to gather statistics about CPU and memory utilization, network and disk activity, and other performance metrics for each node in the cluster.

Using the Visualization Tools

To view statistics about the FailSafe cluster, use the `rmvis`(1) and `hbvis`(1) commands.

The `hbvis`(1) command constructs a display showing the distribution of heartbeat response times for every node in the cluster. Figure 11-1 shows an example display.

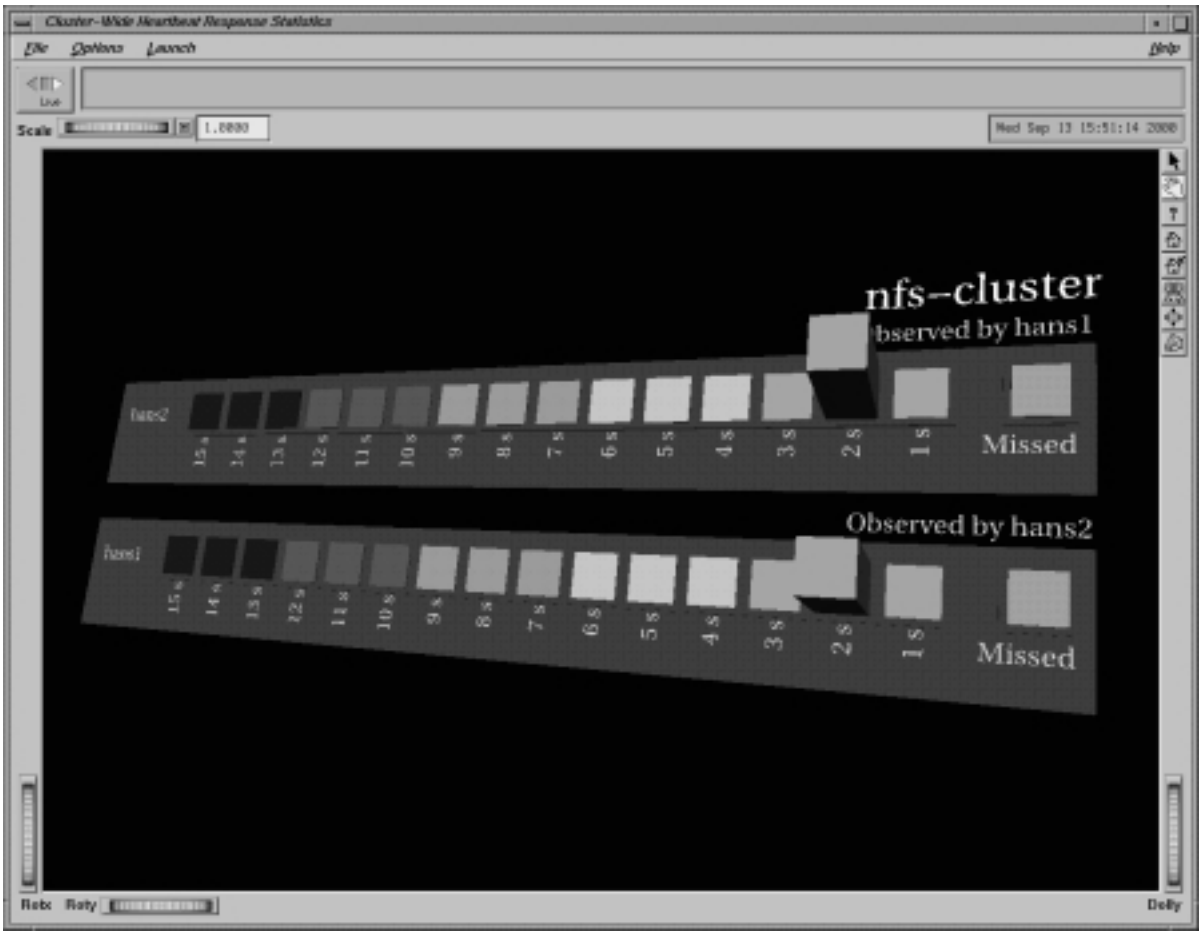

Figure 11-1 Heartbeat Response Statistics

Key features of the display include the frequency of heartbeat responses that arrive at particular intervals within the timeout period and the frequency of heartbeat responses that have been missed (determined not to have arrived). The bar representing the frequency of missed heartbeat responses changes color to indicate the urgency of problems with availability of a node.

The `rmvis`(1) command constructs a display of the resource monitoring response times for resources monitored on every node of the cluster. Figure 11-2 shows an example display.

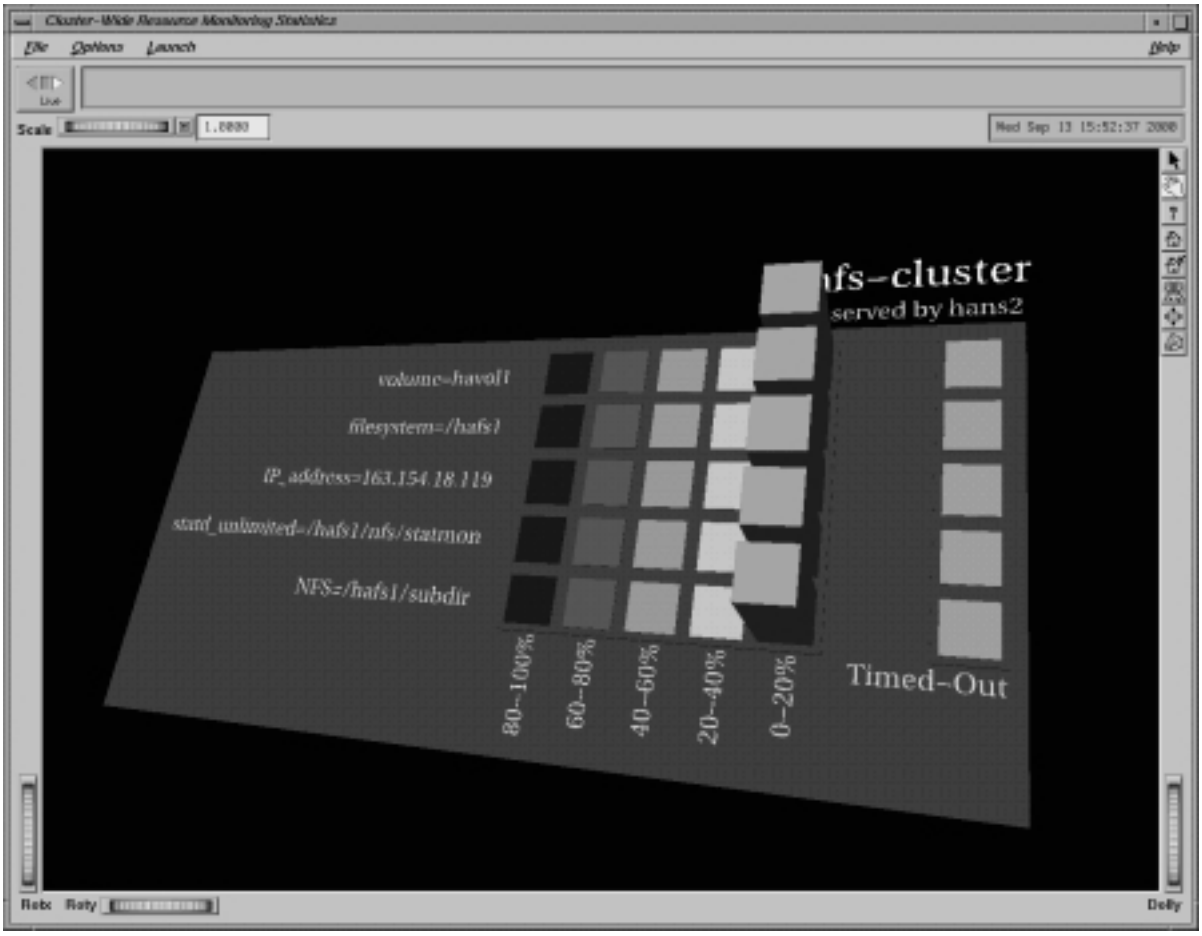

Figure 11-2 Resource Monitoring Statistics

The display is similar in concept to that of `hbvis`(1), showing the frequency of resource monitoring responses that arrive within the timeout period, and the frequency of responses that have timed out. The bar representing the frequency of

resource responses that have timed out also changes color to indicate the urgency of problems with the availability of particular resources.

If a node has failed or a resource has failed over, its statistics will disappear from the display.

To run a visualization tool on the monitor host, use the `-h` option to specify an available collector host in the cluster (*host*):

```
% hbvis -h host
```

or

```
% rmvis -h host
```

The collector host specified can be **any** collector host that is a member of the cluster for which you wish to view statistics.

You can also access these tools from the following FailSafe GUI menus:

File
> Launch Resource Monitoring

File
> Launch Heartbeat Monitoring

There are various options available to alter the display provided by `hbvis`(1) and `rmvis`(1) when launched from the command line:

`-H` *hostfile*	Provides a file that lists the nodes that are to appear in the visualization. This is useful in limiting the number of nodes in the display, because it takes more time to construct the display for clusters with more nodes.
`-t` *interval*	Assigns the sampling time of the visualization. There may be circumstances where extending the period of the sampling time may provide better application responsiveness, particularly for clusters with many nodes. Because FailSafe maintains the statistics, `hbvis`(1) and `rmvis`(1) will always show the latest statistics available for the sampling time selected. For details about the *interval* option, see the `pmview`(1) and `PCPIntro`(1) man pages.

`-r`	Selects the FailSafe metrics that present a sampling of statistics taken from the time of the last statistical reset. This enables `hbvis`(1) and `rmvis`(1) to improve the sensitivity of the visualization when abrupt changes appear in the FailSafe monitoring statistics. Without the `-r` option, the statistics presented are from a sampling of FailSafe metrics collected from the time `ha_cmsd`(1m) and/or `ha_srmd`(1m) was last restarted.
`-R`	Starts a new statistical sampling.
`-v`	(`hbvis`(1) only) Provides a visualization of heartbeat statistics for each node in the cluster, from the point of view of the selected collector host only. (The collector host is selected using the `-h` option). There is a graphical representation of heartbeat statistics for each node in the cluster as observed by the selected collector host.
`-w`	(`hbvis`(1) only) Provides a visualization of the aggregate of heartbeat statistics for all nodes in the cluster, from the point of view of the selected collector host only. (The collector host is selected using the `-h` option). There is a only one graphical representation of heartbeat statistics for the entire cluster as observed by the selected collector host.

For a complete description of options, see the `hbvis`(1) and `rmvis`(1) man pages.

The `hbvis`(1) and `rmvis`(1) commands use the command `pmview`(1) to display the 3-D visualization of FailSafe performance metrics. For a description of the various menu commands and controls in the visualization window, consult the man pages for `pmview`(1).

PCP for FailSafe Performance Metrics

PCP tools such as `pmlogger`(1), `pmchart`(1), and `pminfo`(1) can use the metrics exported by PCP for FailSafe.

Appendix D, "Metrics Exported by PCP for FailSafe", page 343, provides a description of PCP for FailSafe metrics. You can also display a description of metrics by using the following command:

```
% pminfo -tT -h host
```

(If you are logged in to a collector host, you can leave out the `-h` option).

PCP Gray Display

A gray display (that is, no colored rectangle bars appear on the node's gray baseplane) when using `hbvis`(1) or `rmvis`(1) may indicate one of the following:

- The node is down.

  If you wish to see only the nodes that are up, create a file containing a list of nodes that are to be displayed and pass it as an option to `hbvis`(1)/`rmvis`(1) using the `-H` option (or the environment variable `PCP_FSAFE_NODES`) so that a new picture of the cluster can be generated. Please refer to the `hbvis`(1)/`rmvis`(1) man pages for more details on the `-H` option.

- The collector daemons have been killed on that node.

  To solve this problem, restart `pmdafsafe`(1) in one of the following ways:

  – If `pmcd`(1) is still running, send `pmcd`(1) the `SIGHUP` signal by entering the following:

    ```
    # killall -HUP pmcd
    ```

  – If `pmcd`(1) is not running, restart PCP by entering the following:

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

- The timeout and sampling settings are too short.

  To change the sampling time, use the time controls available in the `pmview`(1) window. By default, this is two seconds; you may need to lengthen the sampling period if you are getting an unsatisfactory display.

  Alternatively, there may be timeout issues between `pmdafsafe`(1) and `pmcd`(1), or between `pmcd`(1) and `pmview`(1). Refer to the man pages for `pmcd`(1) and `PCPIntro`(1) for information on how to change the timeout settings for the various PCP tools.

- The resource has failed over (for `rmvis`(1)).

  In this case, restart `rmvis`(1) so that a new picture of the cluster can be generated.

Software Overview

This section contains the following:

- "Software Layers"
- "Interface Agent Daemon (IFD)", page 311
- "Communication Paths", page 311
- "Communication Paths in a Coexecution Cluster", page 316
- "Execution of FailSafe Action and Failover Scripts", page 317
- "When a `start` Script Fails", page 321
- "When a `stop` Script Fails", page 321
- "Components", page 321

Software Layers

A FailSafe system has the following software layers:

- Plug-ins, which create highly available services. The following table shows the provided and optional FailSafe plug-ins and their associated resource types.

Table A-1 Provided and Optional Plug-Ins

Provided Plug-In	Resource Type	Optional Plug-In	Resource Type
CXFS file system	`CXFS`	FailSafe/DMF	`DMF`
IP addresses	`IP_address`	FailSafe/NFS	`NFS` and `statd_unlimited`
MAC addresses	`MAC_address`	FailSafe/Informix	`INFORMIX_DB`
XFS file systems	`filesystem`	FailSafe/Oracle	`Oracle_DB`
XLV logical volumes	`volume`	FailSafe/Samba	`Samba`

Provided Plug-In	Resource Type	Optional Plug-In	Resource Type
		FailSafe/TMF	`TMF`
		FailSafe/Web (Netscape)	`Netscape_web`

See the release notes for information about the specific releases of these products that are supported.

If the application you want is not available, you can hire the SGI Professional Services group to develop the required software, or you can use the *IRIS FailSafe Version 2 Programmer's Guide* to write the software yourself.

- FailSafe base, which includes the ability to define resource groups and failover policies.
- Cluster services, which lets you define clusters, resources, and resource types (this consists of the `cluster_services` installation package)
- Cluster software infrastructure, which lets you do the following:
  - Perform node logging
  - Administer the cluster
  - Define nodes

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

Figure A-1 shows a graphic representation of these layers. The cluster services and cluster software infrastructure layers are shared with CXFS. Table A-2, page 310, describes the contents of the `/usr/cluster/bin` directory. For more information about CXFS, see the *CXFS Version 2 Software Installation and Administration Guide*.

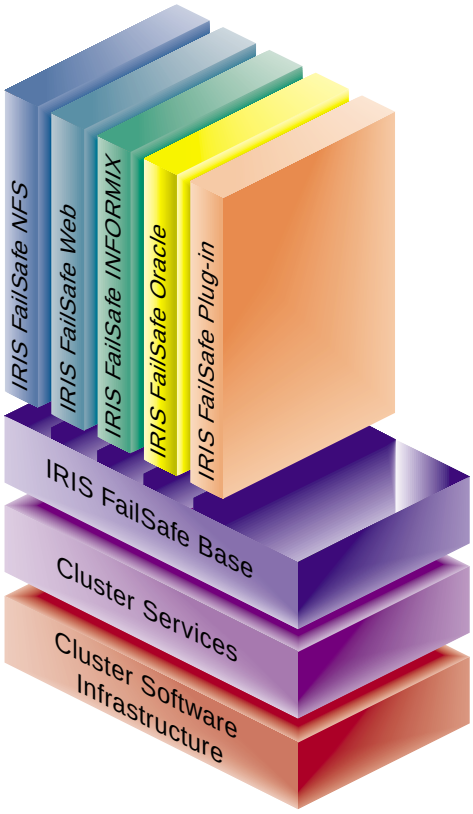

Figure A-1 Software Layers

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

Layer	Subsystem	Process	Description
Plug-ins	`failsafe_informix` `failsafe2_oracle`	`ha_ifmx2`	IRIS FailSafe database agents. Each database agent monitors all instances of one type of database.
IRIS FailSafe Base	`failsafe2`	`ha_fsd`	IRIS FailSafe daemon. Provides basic component of the IRIS FailSafe software.
Cluster services (high-availability processes)	`cluster_services`	`ha_cmsd`	The FailSafe membership daemon. Provides the list of nodes, called *FailSafe membership*, available to the cluster.
		`ha_gcd`	Group membership daemon. Provides group membership and reliable communication services in the presence of failures to IRIS FailSafe processes.
		`ha_srmd`	System resource manager daemon. Manages resources, resource groups, and resource types. Executes action scripts for resources.
		`ha_ifd`	Interface agent daemon. Monitors the local node's network interfaces. This daemon is described in detail in "Interface Agent Daemon (IFD)", page 311.
Cluster software infrastructure (cluster administrative processes)	`cluster_admin`	`cad`	Cluster administration daemon. Provides administration services.
	`cluster_control`	`crsd`	Node control daemon. Monitors the serial connection to other nodes. Has the ability to reset other nodes.

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

Interface Agent Daemon (IFD)

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

The IFD checks the number of input packets for each interace. If the number of input packets does not increase for a 10–second period, the IFD contacts the broadcast address of the interface by using the `ping`(1M) command. If the input packet count does not increase in the next 10–second period, the network interface and all IP addresses on the interface are marked as bad.

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

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

IFD logging can be controlled with the GUI and the `cmgr` command.

Communication Paths

The following figures show communication paths in FailSafe.

Note: The following figures do not represent the `cmond` cluster manager daemon. The purpose of this daemon is to keep the other daemons running.

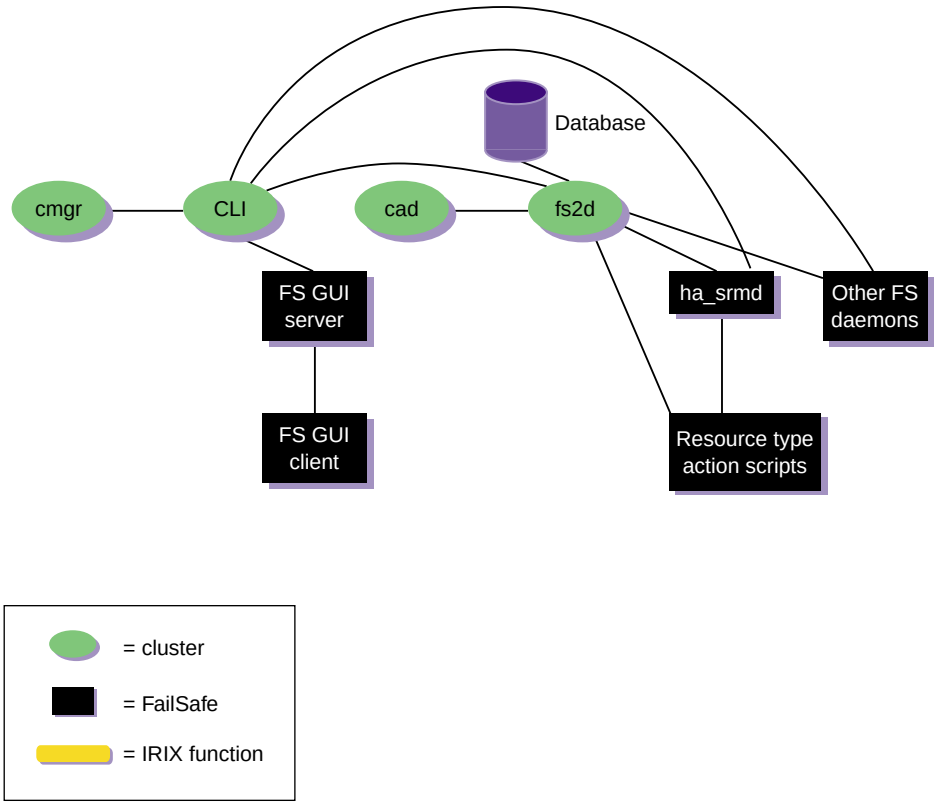

Figure A-2 Administrative Communication within One Node

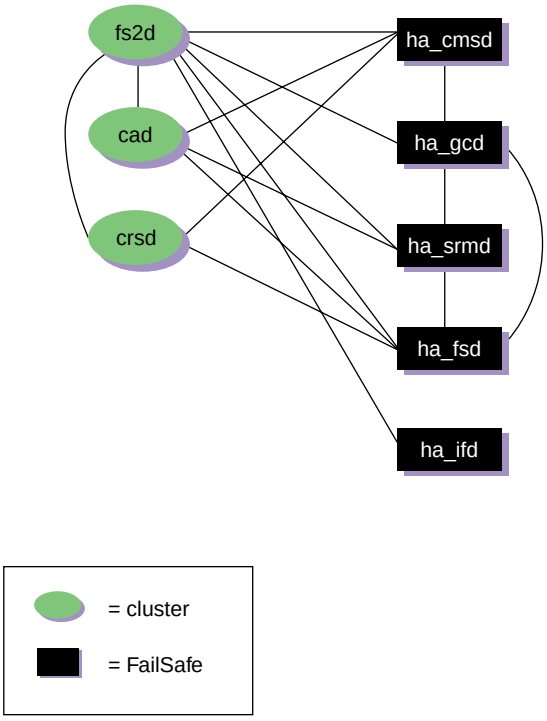

Figure A-3 Daemon Communication within One Node

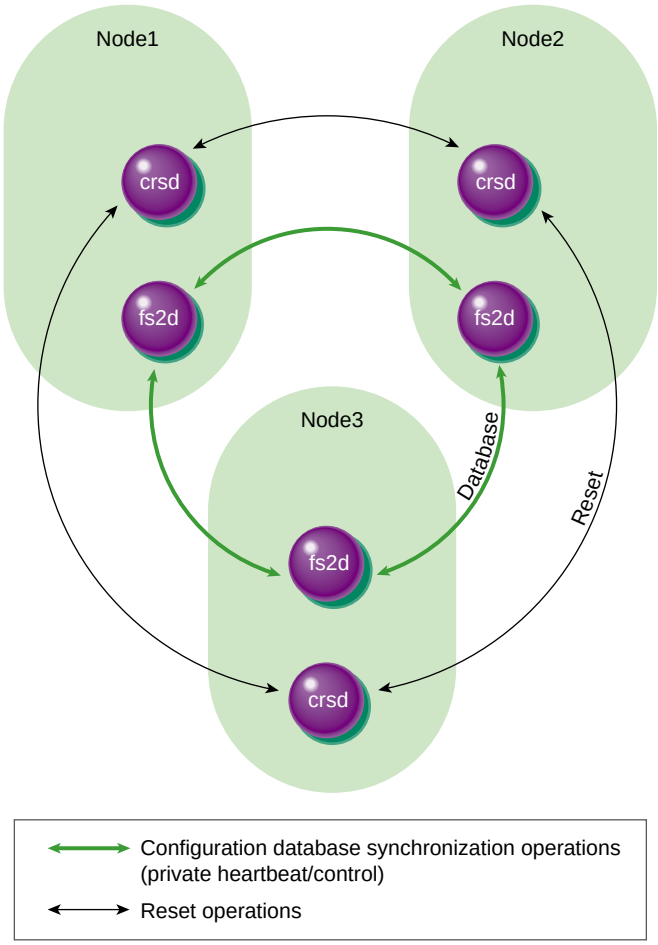

Figure A-4 Communication between Nodes in the Pool

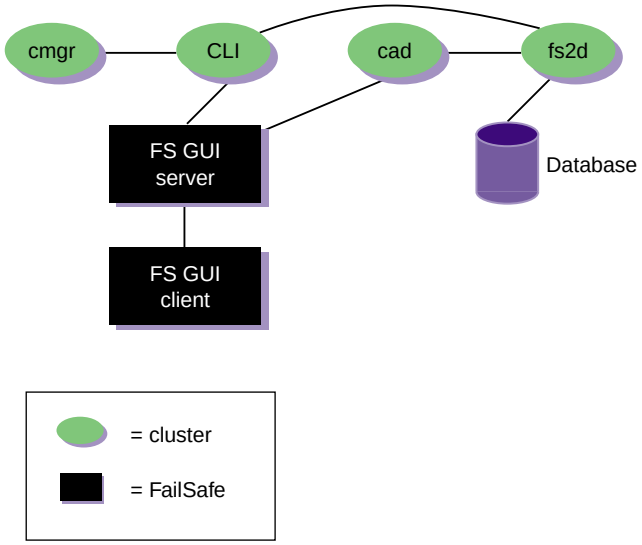

Figure A-5 Communication for a Node Not in the Cluster

Communication Paths in a Coexecution Cluster

The following figures show the communication paths within one node in a coexecution cluster.

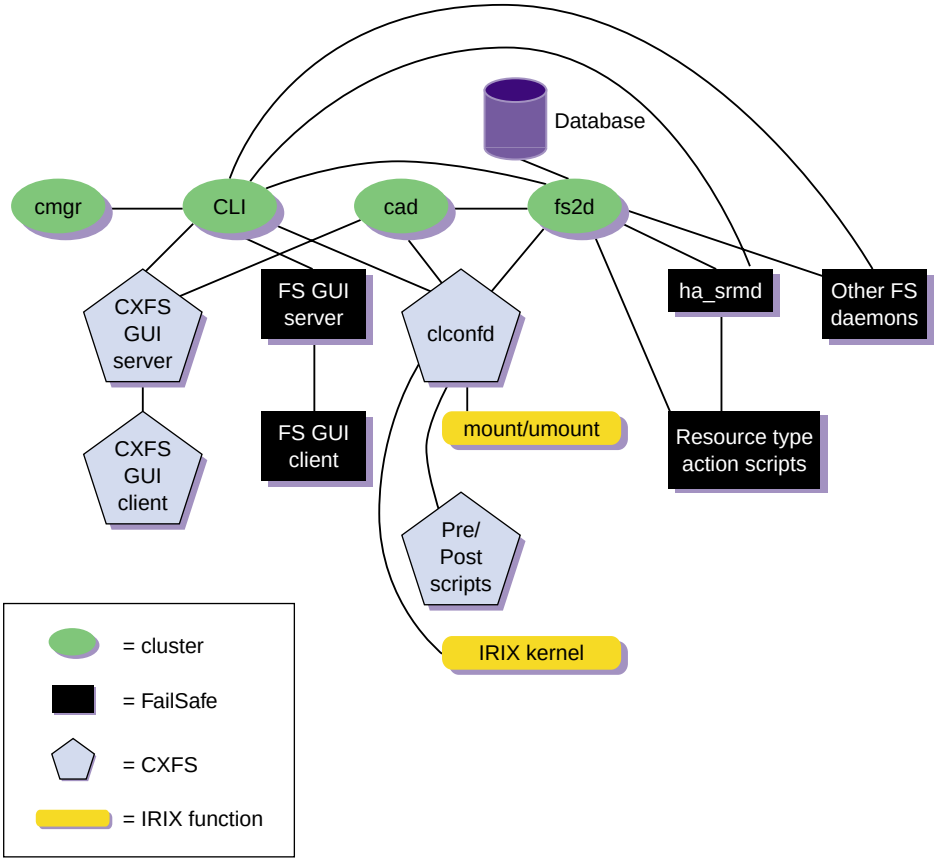

Figure A-6 Administrative Communication within One Node under Coexecution

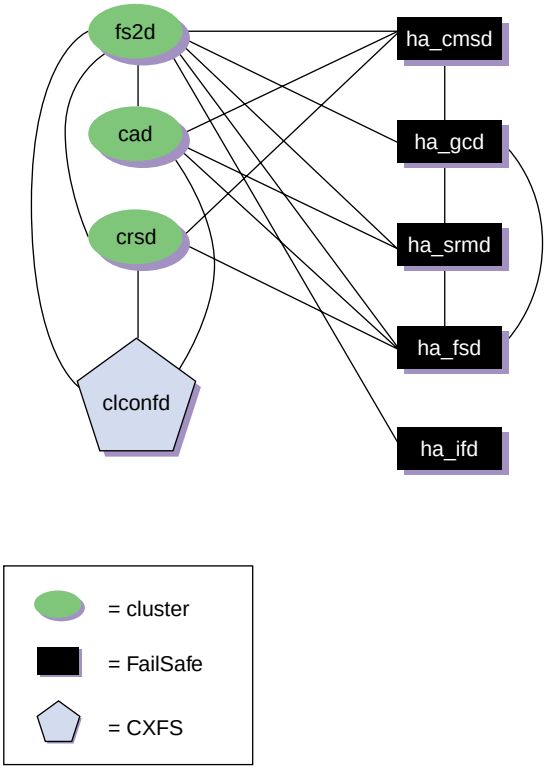

Figure A-7 Daemon Communication within One Node under Coexecution

Execution of FailSafe Action and Failover Scripts

The order of execution is as follows:

1. FailSafe starts up by using the `start ha_services` command in `cmgr` or as part of the node bootup procedure. It then reads the resource group information from the cluster database.
2. FailSafe tells the system resource manager (SRM) to run `exclusive` scripts for all resource groups that are in the `Online ready` state.

3. SRM returns one of the following states for each resource group:
   - `running`
   - `partially running`
   - `not running`
4. If a resource group has a state of `not running` in a node where HA services have been started, the following occurs:
   a. FailSafe runs the failover policy script associated with the resource group. The failover policy scripts takes the list of nodes that are capable of running the resource group (the *failover domain*) as a parameter.
   b. The failover policy script returns an ordered list of nodes in descending order of priority (the *run-time failover domain*) where the resource group can be placed.
   c. FailSafe sends a request to SRM to move the resource group to the first node in the run-time failover domain.
   d. SRM executes the `start` action script for all resources in the resource group:
      - If the `start` script fails, the resource group is marked `online` on that node with following error:

        ```
        srmd executable error
        ```

      - If the `start` script is successful, SRM automatically starts monitoring those resources. After the specified start monitoring time passes, SRM executes the `monitor` action script for the resource in the resource group.
5. If the state of the resource group has a status of `running` or `partially running` on only one node in the cluster, FailSafe runs the associated failover policy script:
   - If the highest priority node is the same node where the resource group is `partially running` or `running`, the resource group is made online on the same node. In the `partially running` case, FailSafe tells SRM to execute `start` scripts for all resources in the resource group.
   - If the highest priority node is another node in the cluster, FailSafe tells SRM to execute `stop` action scripts for resources in the resource group on other nodes. FailSafe then makes the resource group online in the highest priority node in the cluster.

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

Figure A-8 shows the message paths for action scripts and failover policy scripts.

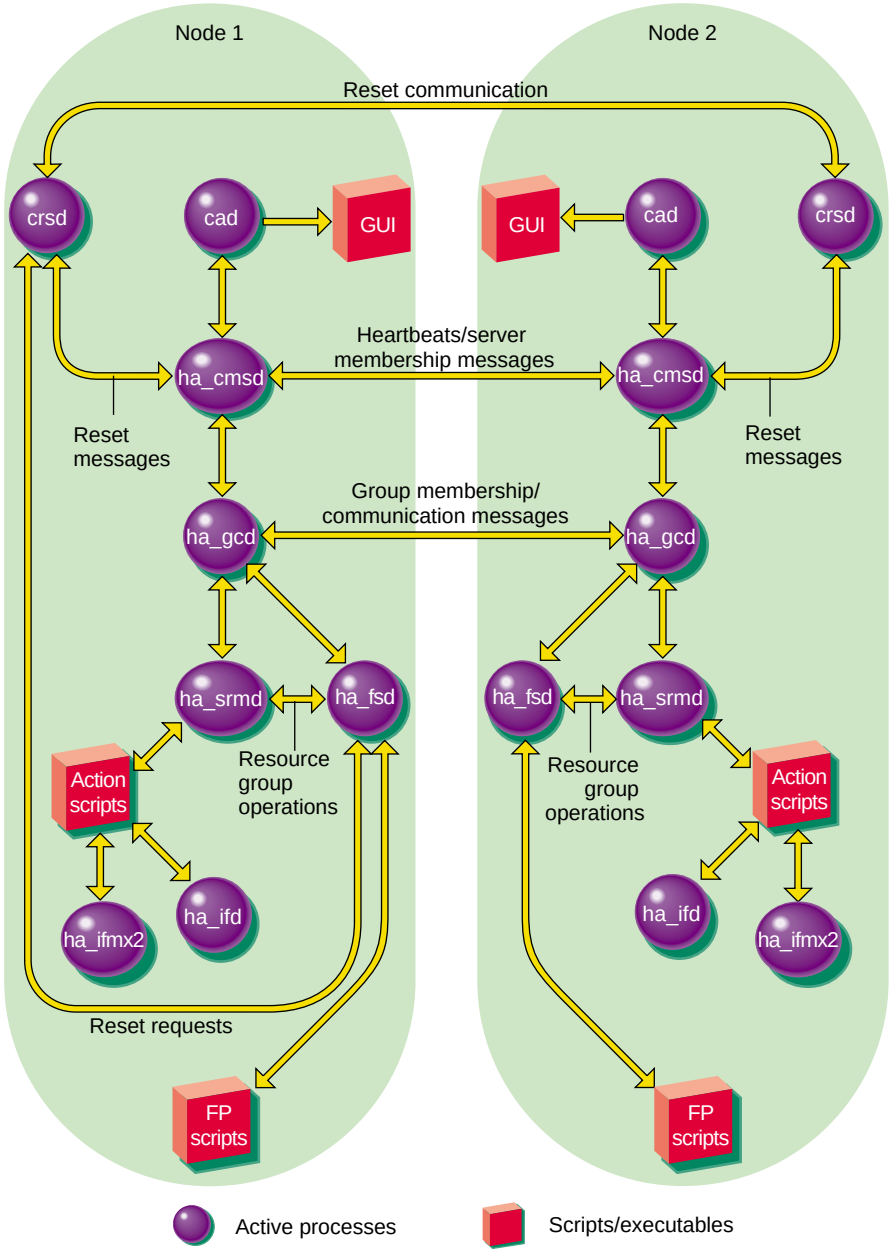

Figure A-8 Message Paths for Action Scripts and Failover Policy Scripts

When a `start` Script Fails

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

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

When a `stop` Script Fails

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

1. SRM notifies FailSafe of the `stop` action script failure as a resource group failure.
2. FailSafe marks the resource group with the following error:

   ```
   srmd executable error
   ```

3. The system administrator must use the `offline force` command to clear the error state after stopping the resource group in the node.

Components

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

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

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

The following table shows the contents of the `/var/cluster/ha` directory.

Table A-3 Contents of the `/var/cluster/ha` directory

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

Updating from IRIS FailSafe 1.2 to IRIS FailSafe 2.1.*x*

IRIS FailSafe 2.1.*x* is not a new release of the IRIS FailSafe 1.2 product but, instead, is a new set of files and scripts that provides many additional possibilities for the size and complexity of a highly available system. If you wish to migrate a FailSafe 1.2 system to a FailSafe 2.1.*x* system to take advantage of these features, you must upgrade your system configuration. There is no upgrade installation option to automatically upgrade FailSafe 1.2 to FailSafe 2.1.*x*.

This appendix provides a description of the procedures you perform to upgrade a system from FailSafe 1.2 to FailSafe 2.1.*x*. It includes the following sections:

- "Hardware Changes"
- "Software Changes", page 324
- "Configuration Changes", page 324
- "Scripts", page 325
- "Operational Comparison", page 326
- "Upgrade Examples", page 327
- "Additional FailSafe 2.1.*x* Tasks", page 334
- "Status", page 335

Hardware Changes

There are no hardware changes that are required when you upgrade a system to FailSafe 2.1.*x*. A FailSafe 1.2 system will be a dual-hosted storage with reset ring two-node configuration in FailSafe 2.1.*x*.

With FailSafe 2.1.*x*, you can test the hardware configuration with FailSafe diagnostic commands. See Chapter 8, "Testing the Configuration", page 261, for instructions on using FailSafe to test the connections. These diagnostics are not run automatically when you start FailSafe 2.1.*x*; you must run them manually.

You can also use the `admin ping` command to test the serial reset line in FailSafe 2.1.*x*. This command replaces the `ha_spng` command you used with FailSafe 1.2.

FailSafe 1.2 command to test serial reset lines:

```
# /usr/etc/ha_spng -i 1 -d msc -f /dev/ttyd2
# echo $status
```

FailSafe 2.1.*x* `cmgr` command to test serial reset lines:

```
cmgr> admin ping dev_name /dev/ttyd2 of dev_typetty with sysctrl_type msc
```

See Chapter 4, "Administration Tools", page 87, for information on using `cmgr` commands.

Software Changes

FailSafe 2.1.*x* consists of a different set of files than FailSafe 1.2. FailSafe 1.2 and FailSafe 2.1.*x* software 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 2.1.*x*, configuration information is contained in a cluster database at `/var/cluster/cdb/cdb.db` that is kept in all nodes in the pool. You create the cluster database using the `cmgr` command or the GUI.

The FailSafe 2.1.*x* cluster database is automatically copied to all nodes in the pool. The FailSafe 2.1.*x* configuration is kept in all nodes in the pool.

Configuration Changes

You must reconfigure your FailSafe 1.2 system by using the FailSafe 2.1.*x* GUI or the FailSafe 2.1.*x* `cmgr` command to configure the system as a FailSafe 2.1.*x* system. For information on using these administration tools, see Chapter 4, "Administration Tools", page 87.

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 2.1.*x* system, you perform the following steps:

1. Add nodes to the pool
2. Create cluster
3. Add nodes to the cluster
4. Set HA parameters (FailSafe 2.1.*x* can be started at this point, if desired)
5. Create resources
6. Create failover policy
7. Create resource groups
8. Add resources to resource groups
9. Put resource groups online

These steps are captured in the guided configuration task sets in the GUI. These task sets lead you through these configuration steps.

For a configuration example that compares FailSafe 1.2 configuration to FailSafe 2.1.*x* configuration, see "Upgrade Examples", page 327.

Scripts

All FailSafe 1.2 scripts must be rewritten for FailSafe 2.1.*x*. The *IRIS FailSafe Version 2 Programmer's Guide* provides detailed information on FailSafe 2.1.*x* scripts as well as detailed instructions for migrating FailSafe 1.2 scripts to their FailSafe 2.1.*x* functional equivalent.

Operational Comparison

In FailSafe 1.2, the unit of failover is the node. In FailSafe 2.1.*x*, the unit of failover is the resource group. Because of this, the concepts of node failover, node failback, and even node state do longer apply to FailSafe 2.1.*x*. In addition, all FailSafe scripts differ between the two releases.

The following table summarizes the differences between the releases.

Table B-1 Differences Between IRIS FailSafe 1.2 and 2.1.*x*

FailSafe 1.2	FailSafe 2.1.*x*
`ha.conf` configuration file.	Cluster 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 2.1.*x* database, but the format is completely different. You will configure the database using the Cluster Manager graphical user interface or the `cmgr` command.
Node states (standby, normal, degraded, booting or up).	Resource group states (online, offline, pending, maintenance, error).

Scripts:
`giveaway, giveback`
`takeover, takeback`
`check`
(no equivalent)
	Scripts:
	`stop`
	`start`
	`monitor`
	`exclusive`, `probe`, `restart`
	Failover script
	Failover attributes

All common functions and variables are kept in the `/var/ha/actions/common.vars` file.	All common functions and variables are kept in the `/var/cluster/ha/common_scripts/scriptlib` file.
Configuration information is read using the `ha_cfginfo` command.	Configuration information is read using the `ha_get_info()` and `ha_get_field()` shell functions.
Software links specify application ordering.	Software links are not used for ordering.
Scripts use `/sbin/sh`.	Scripts use `/sbin/ksh`.
Scripts require configuration checksum verification.	There is no configuration checksum verification in the scripts.

FailSafe 1.2	FailSafe 2.1.*x*
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 highly available service.

Upgrade Examples

In order to upgrade a FailSafe 1.2 system to a FailSafe 2.1.*x* system, you must examine your `ha.conf` file to determine how to define the equivalent parameters in the FailSafe 2.1.*x* cluster database.

The following sections show upgrade examples for the following tasks:

- Defining a Node
- Defining a Cluster
- Setting HA Parameters
- Defining a Resource: XLV Volume
- Defining a Resource: XFS Filesystem
- Defining a Resource: IP Address

For upgrade examples of the following tasks, see the *IRIS FailSafe Version 2 Programmer's Guide*, where customized resources and scripts are described.

- Defining a Resource Type
- Defining a Failover Policy
- Writing FailSafe Scripts

Defining a Node

The following example shows node definition in the FailSafe 1.2 `ha.conf` file. Parameters that you must use when configuring a FailSafe 2.1.*x* system are indicated in bold.

```
Node node1
{
interface node1-fxd
{
name = rns0
ip-address = 54.3.252.6
netmask = 255.255.255.0
broadcast-addr = 54.3.252.6
}
heartbeat
{
hb-private-ipname = 192.0.2.3
hb-public-ipname = 54.3.252.6
hb-probe-time = 6
hb-timeout = 6
hb-lost-count = 4
}
reset-tty = /dev/ttyd2

sys-ctlr-type = MSC
}
```

In this configuration example, you will use the following values when you define the same node in FailSafe 2.1.*x*:

- Node name: `node1`
- Primary network interface: `node1`
- Type of system controller: `msc`
- System control device name: `/dev/ttyd2`
- Control networks: `192.0.2.3`, `54.3.252.6`

Use the following `cmgr` command to use these values to define a node in FailSafe 2.1.*x*. Note that there are additional parameters you must specify when you define this node.

```
cmgr> define node node1
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional]? node1
Is this a FailSafe node <true|false> ? true
Is this a CXFS node <true|false> ? false
Node ID ? 10
Reset type <powerCycle> ? (powerCycle)
Do you wish to define system controller info[y/n]:y
Sysctrl Type <msc|mmsc>? (msc) msc
Sysctrl Password [optional]? ( )
Sysctrl Status <enabled|disabled>? enabled
Sysctrl Owner? node2
Sysctrl Device? /dev/ttyd2
Sysctrl Owner Type <tty> [tty]?
Number of Network interfaces [2]? 2
NIC 1 - IP Address? 192.0.2.3
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false>? true
NIC 1 - (use network for control messages) <true|false>? true
NIC 1 - Priority <1,2,...>? 1
...
```

As this `ha.conf` node definition shows, in FailSafe 1.2 you defined parameters to set the values that determined how often to send monitoring messages and how long of a time period without a response would indicate a failure when you defined a node. For information on setting monitoring values in FailSafe 2.1.*x*, see "Setting HA Parameters", page 330.

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 2.1.*x* uses in its cluster definition: the e-mail address to use to notify the system administrator when problems occur in the cluster.

The `ha.conf` file includes the following:

```
system configuration
{
mail-dest-addr = root@localhost
...
}
```

When you define a cluster in FailSafe 2.1.*x*, you can use this as the e-mail address to use for problem notification.

There are other things you must provide in addition to this parameter when you define a FailSafe 2.1.*x* cluster, such as the e-mail program to use for this notification and, of course, the nodes to include in the cluster. Use the following `cmgr` command to define a cluster:

```
cmgr> define cluster apache-cluster
Enter commands, you may enter "done" or "cancel" at any time to exit

cluster apache-cluster? set notify_addr to root@localhost
cluster A? done
```

Use the following `cmgr` command to add nodes to the cluster:

```
cmgr> modify cluster apache-cluster
Enter commands, you may enter "done" or "cancel" at any time to exit

cluster apache-cluster? add node node1
cluster A? done
```

Setting HA Parameters

The following example shows the sections of a FailSafe 1.2 `ha.conf` file that are used to set monitoring and timeout values. Parameters that you must use when configuring a FailSafe 2.1.*x* system are indicated in bold.

```
system-configuration
{
      pwrfail = true
      ...
}


Node node1
{
...
heartbeat
{
       hb-private-ipname = 192.0.2.3
       hb-public-ipname = 54.3.252.6
       hb-probe-time = 6
```

```
        hb-timeout = 6
        hb-lost-count = 4
}
...
}
```

As this `ha.conf` node-definition shows, in FailSafe 1.2 you defined `hb-probe-time`, `hb-timeout`, and `hb-lost-count` parameters to set the values that determined how often to send monitoring messages and how long of a time period without a response would indicate a failure. FailSafe 2.1.*x* uses a different method for monitoring the nodes in a cluster than FailSafe 1.2 uses, sending out continuous messages to the other nodes in a cluster and, in turn, maintaining continuous monitoring of the messages the other nodes are sending.

Because of the different monitoring methods between the two systems, there is no one-to-one correspondence between the values you set in the `ha.conf` file and the timeout and heartbeat intervals you set in FailSafe 2.1.*x* when you set FailSafe HA parameters. However, if you wish to maintain approximately the same time interval before which your system determines that failure has occurred, you can use the following formula to determine the value to which you should set your node timeout interval:

node_timeout = (*probetime* + *timeout*) * *lostcount*

This formula should account for the same total node-to-node communication time.

All FailSafe 2.1.*x* timeouts are in milliseconds, and can be changed when FailSafe 2.1.*x* is running. Timeouts can be specified for the cluster for a specific node in the cluster.

There is no long-timeout value in FailSafe 2.1.*x*. The long-timeout value equivalent is set with the resource type start and stop action monitor timeouts. The resource type start, monitor, and stop action timeouts can be changed using the GUI or `cmgr`.

Use the following `cmgr` command to modify the HA parameters for `node1` in FailSafe 2.1.*x*:

```
cmgr> modify ha_parameters on node node1 in cluster apache-cluster
Enter commands, when finished enter either "done" or "cancel"

node1 ? set node_timeout to 24000
node1 ? set heartbeat to 6000
node1 ? set run_pwrfail to true
node1 ? done
```

Defining a Resource: XLV Volume

The following example shows a volume definition in the FailSafe 1.2 `ha.conf` file. Parameters that you must use when configuring the same volume as a volume resource in a FailSafe 2.1.*x* system are indicated in bold.

```
volume apache-vol
{
   server-node = node1
   backup-node = node2
   devname = apache-vol
   devname-owner = root
   devname-group = sys
   devname-mode = 600
}
```

In this configuration example, you will use the following values when you define the same volume in FailSafe 2.1.*x*:

- Volume name: `apache-vol`
- User name of device file owner: `root`
- Group name of device file: `sys`
- Device file permissions: `600`

To create an XLV volume resource, use the following `cmgr` commands:

```
cmgr> define resource apache-vol of resource_type volume in cluster apache-cluster
Enter commands, when finished enter either "done" or "cancel"

resource apache-vol? set devname-owner to root
resource apache-vol? set devname-group to sys
resource apache-vol? set devname-mode to 600
resource apache-vol? done
```

Defining a Resource: XFS Filesystem

The following example shows an XFS filesystem definition in the FailSafe 1.2 `ha.conf` file. Parameters that you must use when configuring the same filesystem as a filesystem resource in a FailSafe 2.1.*x* system are indicated in bold.

```
filesystem apache-fs
{
   mount-point = /apache-fs
   mount-info
{
   fs-type = xfs
   volume-name = apache-vol
   mode = rw, noauto
}
}
```

In this configuration example, you will use the following values when you define the same filesystem in FailSafe 2.1.*x*:

- Resource name (mount point): `/apache-vol`
- XLV volume: `apache-vol`
- Mount options: `rw`, `noauto`

To create a filesystem resource, use the following `cmgr` commands:

```
cmgr> define resource /apache-fs of resource_type filesystem in cluster apache-cluster
Enter commands, when finished enter either "done" or "cancel"

resource /apache-fs? set volme-name to apache-vol
resource /apache-fs? set mount-options to "rw,noauto"
resource /apache-fs? done
```

Defining a Resource: IP Address

The following example shows an IP address definition in the FailSafe 1.2 `ha.conf` file. Parameters that you must use when configuring the same IP address as a highly available resource in a FailSafe 2.1.*x* system are indicated in bold.

```
interface-pair FDDI_1
{
   primary-interface = node-fxd
   secondary-interface = node2-fxd
   re-mac = false
   netmask = 0xffffff00
   broadcast-addr = 54.3.252.255
```

```
    ip-aliases = ( 54.3.252.7 )
}
```

In this configuration example, you will use the following values when you define the same IP Address in FailSafe 2.1.*x*:

- Resource name: `54.3.252.7`
- Broadcast address: `54.3.252.255`
- Network mask: `0xffffff00`

To create an IP address resource, use the following `cmgr` commands:

```
cmgr> define resource 54.3.252.7 of resource_type IP_address in cluster apache-cluster
Enter commands, when finished enter either "done" or "cancel"

resource 54.3.252.7? set interfaces to rns0
resource 54.3.252.7? set NetworkMask to 0xffffff00
resource 54.3.252.7? set BroadcasetAddress to 54.3.252.255
resource 54.3.252.7? done
```

Additional FailSafe 2.1.*x* Tasks

After you have defined your nodes, clusters, and resources, you define your resource groups, a task which has no equivalent in FailSafe 1.2. When you define a resource group, you specify the resources that will be included in the resource group and the failover policy that determines which node will take over the services of the resource group on failure.

For information on defining resource groups, see "Define a Resource Group with the GUI", page 193.

After you have configured your system, you can start FailSafe services, as described in "Start FailSafe HA Services", page 199.

Status

In FailSafe 1.2, you produced a display of the system status with the `ha_admin -a` command. In FailSafe 2.1.*x*, you can display the system status in the following ways:

- You can keep continuous watch on the state of a cluster using the GUI.
- You can query the status of an individual resource group, node, or cluster using either the GUI or `cmgr`.
- You can use the `/var/cluster/cmgr-scripts/ha Status` script provided with the `cmgr`to see the status of all clusters, nodes, resources, and resource groups in the configuration.

For information on performing these tasks, see "System Status", page 233.

IRIS FailSafe 2.1.*x* Software

This appendix summarizes software to be installed on systems used for IRIS FailSafe 2.1.*x*. It consists of the following sections:

- "Subsystems on the CD"
- "Subsystems for Servers and Workstations in the Pool", page 339
- "Additional Subsystems for Nodes in the FailSafe Cluster", page 340
- "Additional Subsystems for Administrative Workstations ", page 340

Note: "Install Software", page 55 contains step-by-step instructions for installing the software.

Subsystems on the CD

The IRIS FailSafe 2.1.*x* base CD requires about 10 MB.

Table C-1, page 338, lists FailSafe 2.1.*x* subsystems on the IRIS FailSafe 2.1.*x* CD.

Table C-1 IRIS FailSafe 2.1.*x* CD

Purpose	System

IRIS FailSafe 2.1.*x*	`failsafe2`
	`failsafe2.idb`
	`failsafe2.man`
	`failsafe2.sw`
	`failsafe2.books` (InSight versions of customer manuals)

FailSafe system administration	`sysadm_failsafe2`
	`sysadm_failsafe2.idb`
	`sysadm_failsafe2.man`
	`sysadm_failsafe2.sw`

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

The EL-8+ multiplexer driver subsystems are `el_serial`, `el_serial.man`, and `el_serial.sw`, which are on a CD accompanying the EL-8+ multiplexer.

Subsystems for Servers and Workstations in the Pool

The following table 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 C-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`
Cluster administration GUI	`sysadm_cluster.sw.server`	`sysadm_base.sw.server`

IRIS FailSafe 2.1.*x* administration server	`sysadm_failsafe2.sw.server`	`sysadm_base.sw.server`
		`sysadm_cluster.sw.server`
		`cluster_admin.sw.base`
		`cluster_services.sw.cli`
		`cluster_control.sw.cli`
		`failsafe2.sw.cli`

Cluster administration	`cluster_admin.sw` `cluster_control.sw`	`sysadm_base.sw.dso`

Web-based administration	`sysadm_failsafe2.sw.web`	`sysadm_failsafe2.sw.client`
		`sysadm_failsafe2.sw.server`
		`sysadmbase.sw.client`
		`java_eoe.sw`, version 3.1.1
		Web server

EL-8+ multiplexer driver (from CD included with multiplexer)	`el_serial`
	`el_serial.man`
	`el_serial.sw`	

Additional Subsystems for Nodes in the FailSafe Cluster

The following table lists additional subsystems required for each server that is a node in the cluster. A cluster is one or more nodes coupled with each other by networks. A node is a single UNIX image, usually, an individual server. A node can be a member of only one cluster.

Table C-3 Additional Subsystems Required for Nodes in the Cluster

Product	Images and Subsystems	Prerequisites
Highly available clustering software	`cluster_services.sw`	`cluster_admin.sw` `cluster_control.sw`
IRIS FailSafe 2.1.*x* software	`failsafe2.sw`	`cluster_services.sw`

Additional Subsystems for 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 the following table.

Table C-4 Subsystems Required for IRIX Administrative Workstations

Product	Subsystems	Prerequisites
Cluster administration GUI	`sysadm_cluster.sw.client`	`sysadm_base.sw.client`

FailSafe GUI	`sysadm_failsafe2.sw.client`
	`sysadm_failsafe2.sw.desktop`
		`sysadm_base.sw.client`
		`sysadm_cluster.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

Subsystems for Non-IRIX Administrative Workstations

From a non-IRIX workstation, the GUI can be launched from a web browser that supports Java.

Metrics Exported by PCP for FailSafe

This appendix lists the metrics implemented by `pmdafsafe`(1).

`fsafe.srm.all.*` metrics are the same as the `fsafe.srm.*` metrics, except that the latest values obtained for all resources will be available, even if `ha_srmd`(1M) or any of the resources themselves are not available.

Table D-1 PCP Metrics

Metric	Description
`fsafe.srm.status` `fsafe.srm.all.status`	Latest status of a monitoring event performed on a resource, for all resources configured to be monitored on this node.
`fsafe.srm.timeout` `fsafe.srm.all.timeout`	The prescribed timeout, in milliseconds, for monitoring a resource.
`fsafe.srm.probes` `fsafe.srm.all.probes`	Number of times a resource has been monitored, for all resources configured to be monitored on this node, since the time `ha_srmd`(1M) was started.
`fsafe.srm.recent.probes`	Number of times a resource has been monitored, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.timeouts` `fsafe.srm.all.timeouts`	Number of resource monitoring events that have timed out before declaring that resource as failed, for all resources configured to be monitored on this node, since the time the resources have last been available.
`fsafe.srm.recent.timeouts`	Number of resource monitoring events that have timed out before declaring that resource as failed, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.min_resp` `fsafe.srm.all.min_resp`	Approximate minimum time, in milliseconds, taken to complete a monitoring event on a resource, for all resources configured to be monitored.

Metric	Description
`fsafe.srm.max_resp` `fsafe.srm.all.max_resp`	Approximate maximum time, in milliseconds, taken to complete a monitoring event on a resource, for all resources configured to be monitored on this node.
`fsafe.srm.last_resp` `fsafe.srm.all.last_resp`	Approximate time, in milliseconds, taken in completing the most recent monitoring event on a resource, for all resources configured to be monitored on this node.
`fsafe.srm.cumm_timeouts` `fsafe.srm.all.cumm_timeouts`	Cumulative number of resource monitoring events that have timed out, for all resources configured to be monitored on this node, since the time `ha_srmd`(1M) has started.
`fsafe.srm.recent.cumm_timeouts`	Cumulative number of resource monitoring events that have timed out, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.histo_20` `fsafe.srm.all.histo_20`	Fraction of monitoring events that have been received within 0-20% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since the time `ha_srmd`(1M) has started.
`fsafe.srm.recent.histo_20`	Fraction of monitoring events that have been received within 0-20% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.histo_40` `fsafe.srm.all.histo_40`	Fraction of monitoring events that have been received within 20-40% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since the time `ha_srmd`(1M) has started.
`fsafe.srm.recent.histo_40`	Fraction of monitoring events that have been received within 20-40% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).

Metric	Description
`fsafe.srm.histo_60` `fsafe.srm.all.histo_60`	Fraction of monitoring events that have been received within 40-60% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since the time `ha_srmd(1M)` has started.
`fsafe.srm.recent.histo_60`	Fraction of monitoring events that have been received within 40-60% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.histo_80` `fsafe.srm.all.histo_80`	Fraction of monitoring events that have been received within 60-80% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since the time `ha_srmd`(1M) has started.
`fsafe.srm.recent.histo_80`	Fraction of monitoring events that have been received within 60-80% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.histo_100` `fsafe.srm.all.histo_100`	Fraction of monitoring events that have been received within 80-100% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since the time `ha_srmd`(1M) has started.
`fsafe.srm.recent.histo_100`	Fraction of monitoring events that have been received within 80-100% of the response time from 0 milliseconds to `fsafe.srm.timeout`, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.frac_timeouts` `fsafe.srm.all.frac_timeouts`	Fraction of monitoring events that have timed out before declaring that resource as failed, for all resources configured to be monitored on this node, since the time the resources have last been available.

Metric	Description
`fsafe.srm.recent.frac_timeouts`	Fraction of monitoring events that have timed out, before declaring that resource as failed, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.frac_cumm_timeouts` `fsafe.srm.all.frac_cumm_timeouts`	Fraction of cumulative number of monitoring events that have timed out, for all resources configured to be monitored on this node, since the time `ha_srmd`(1M) has started.
`fsafe.srm.recent.frac_cumm_timeouts`	Fraction of cumulative number of monitoring events that have timed out, for all resources configured to be monitored on this node, since a data collection reset (via `fsafe.control.reset_srm`).
`fsafe.srm.recent.timestamp`	The time when a new collection of statistics was started for the `fsafe.srm.recent.*` metrics, after issuing a store to the metric `fsafe.control.reset_srm`.
`fsafe.config.clustername`	The name of this cluster.
`fsafe.config.hostname`	The name of all hosts in the cluster specified by `fsafe.config.clustername`.
`fsafe.config.nnodes`	Number of nodes in the cluster specified by `fsafe.config.clustername`.
`fsafe.config.cms.interval`	The cluster heartbeat event interval, in milliseconds.
`fsafe.config.cms.timeout`	The heartbeat event timeout for all nodes in the cluster, in milliseconds.
`fsafe.config.cms.nbuckets`	The number of heartbeat event response intervals per node, where each interval covers a time equal to the heartbeat event interval (`fsafe.config.cms.interval`) for segments of time until the heartbeat event timeout (`fsafe.config.cms.timeout`).
`fsafe.control.debug`	Debugging flags for the `fsafe` PMDA when a decimal integer value is stored to this metric. It ultimately affects what information is put into the `fsafe` PMDA's log (normally at `/var/adm/pcplog/fsafe.log`). Reading this metric will return the currently assigned debugging flags as a decimal integer.

Metric	Description
`fsafe.control.reset_cms`	Resets data collection statistics for all metrics gathered from `ha_cmsd`(1M). When this metric is stored to, the data provided is ignored; it is the act of storing to this metric which causes the reset. Reading this metric will return zero (0).
`fsafe.control.reset_srm`	Resets data collection statistics for all metrics gathered from `ha_srmd`(1M). When this metric is stored, the data provided is ignored; it is the act of storing to this metric which causes the reset. Reading this metric will return zero (0).
`fsafe.control.retry`	Sets the number of retries permitted when contacting `ha_cmsd`(1M) or `ha_srmd`(1M), and when the daemons indicate that they are busy. Depending on which metrics are being read, and which daemon is required to obtain values for the required metrics, values for some metrics may not be available, possibly producing the message "`Try again. Information not currently available.`" This metric can be adjusted in order to increase the number of retries permitted when collecting metrics, before giving up and displaying this message. A retry is performed once every 100 ms (approximately). Note that setting this metric does not alter how the fsafe PMDA handles more serious errors from `ha_cmsd`(1M) or `ha_srmd`(1M). Reading this metric will return the current retry count.
`fsafe.cms.expected`	The number of heartbeat events expected to have been received for each node in the cluster (excluding the collector host), since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.expected`	The number of heartbeat events expected to have been received for each node in the cluster (excluding the collector host), since a data collection reset (via `fsafe.control.reset_cms`).

Metric	Description
`fsafe.cms.received`	The number of heartbeat events actually received for each node in the cluster (excluding the collector host), since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.received`	The number of heartbeat events actually received for each node in the cluster (excluding the collector host), since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.missed`	The number of heartbeat events determined not to have been received for each node in the cluster (excluding the collector host), since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.missed`	The number of heartbeat events determined not to have been received for each node in the cluster (excluding the collector host), since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.histo`	Histogram of heartbeat event response times for events that have occurred within discrete heartbeat response intervals for each node in the cluster (excluding the collector host), since the time `ha_cmsd`(1M) has started. The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (`fsafe.config.cms.interval`), for a number of intervals up to the configured heartbeat event timeout (`fsafe.config.cms.timeout`).
`fsafe.cms.recent.histo`	Histogram of heartbeat event response times for events that have occurred within discrete heartbeat response intervals for each node in the cluster (excluding the collector host), since a data collection reset (via `fsafe.control.reset_cms`). The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (`fsafe.config.cms.interval`), for a number of intervals up to the configured heartbeat event timeout (`fsafe.config.cms.timeout`).
`fsafe.cms.frac_received`	Fraction of heartbeat events received over all expected events for each node in the cluster, since the time `ha_cmsd`(1M) has started.

Metric	Description
`fsafe.cms.recent.frac_received`	Fraction of heartbeat events received over all expected events for each node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.frac_missed`	Fraction of heartbeat events determined not to have been received over all expected events for each node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.frac_missed`	Fraction of heartbeat events determined not to have been received over all expected events for each node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.recent.timestamp`	The time when a new collection of statistics was started for the `fsafe.cms.recent.*` metrics, after issuing a store to the metric `fsafe.control.reset_cms`.
`fsafe.cms.pernode.expected`	The number of heartbeat events expected to have been received for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.pernode.expected`	The number of heartbeat events expected to have been received for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.pernode.received`	The number of heartbeat events actually received for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.pernode.received`	The number of heartbeat events actually received for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.pernode.missed`	The number of heartbeat events determined not to have been received for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.pernode.missed`	The number of heartbeat events determined not to have been received for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).

Metric	Description
`fsafe.cms.pernode.histo`	Histogram of heartbeat event response times for events that have occurred within discrete heartbeat response intervals for a particular node in the cluster, since the time `ha_cmsd`(1M) has started. The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (`fsafe.config.cms.interval`), for a number of intervals up to the configured heartbeat event timeout (`fsafe.config.cms.timeout`).
`fsafe.cms.recent.pernode.histo`	Histogram of heartbeat event response times for events that have occurred within discrete heartbeat response intervals for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`). The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (`fsafe.config.cms.interval`), for a number of intervals up to the configured heartbeat event timeout (`fsafe.config.cms.timeout`).
`fsafe.cms.pernode.frac_received`	Fraction of heartbeat events received over all expected events for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.pernode.frac_received`	Fraction of heartbeat events received over all expected events for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).
`fsafe.cms.pernode.frac_missed`	Fraction of heartbeat events determined not to have been received over all expected events for a particular node in the cluster, since the time `ha_cmsd`(1M) has started.
`fsafe.cms.recent.pernode.frac_missed` Fraction of heartbeat events determined not to have been received over all expected events for a particular node in the cluster, since a data collection reset (via `fsafe.control.reset_cms`).	

Glossary

action scripts

The set of scripts that determine how a resource is started, monitored, and stopped. There must be a set of action scripts specified for each resource type. The possible set of action scripts is: `exclusive`, `start`, `stop`, `monitor`, and `restart`.

active/backup configuration

A configuration in which all resource groups have the same primary node. The backup node does not run any highly available resource groups until a failover occurs.

cluster

The set of nodes in the pool that have been defined as a cluster. A cluster is identified by a simple name; this name must be unique within the pool. All nodes in the cluster are also in the pool. However, all nodes in the pool are not necessarily in the cluster; that is, the cluster may consist of a subset of the nodes in the pool. There is only one cluster per pool.

cluster administrator

The person responsible for managing and maintaining a cluster.

cluster database

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

cluster process group

A group of application instances in a distributed application that cooperate to provide a service.

For example, distributed lock manager instances in each node would form a process group. By forming a process group, they can obtain membership and reliable, ordered, atomic communication services. There is no relationship between a UNIX process group and a cluster process group.

collector host

The nodes in the FailSafe cluster itself from which you want to gather statistics, on which PCP for FailSafe has installed the collector agents.

control messages

Messages that cluster software sends between the nodes to request operations on or distribute information about nodes and resource groups. FailSafe sends control messages for the purpose of ensuring that nodes and groups remain highly available. Control messages and heartbeat messages are sent through a node's network interfaces that have been attached to a control network. A node can be attached to multiple control networks.

control network

The network that connects nodes through their network interfaces (typically Ethernet) such that FailSafe can maintain a cluster's high availability by sending heartbeat messages and control messages through the network to the attached nodes. FailSafe uses the highest priority network interface on the control network; it uses a network interface with lower priority when all higher-priority network interfaces on the control network fail.

A node must have at least one control network interface for heartbeat messages and one for control messages (both heartbeat and control messages can be configured to use the same interface). A node can have no more than eight control network interfaces.

database

See *cluster database*.

dependency list

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

failover

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

failover attribute

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

failover domain

The ordered list of nodes on which a particular *resource group* can be allocated. The nodes listed in the failover domain must be within the same cluster; however, the failover domain does not have to include every node in the cluster. The administrator defines the *initial failover domain* when creating a failover policy. This list is transformed into the *run-time failover domain* by the *failover script* the run-time failover domain is what is actually used to select the failover node. FailSafe stores the run-time failover domain and uses it as input to the next failover script invocation. The initial and run-time failover domains may be identical, depending upon the contents of the failover script. In general, FailSafe allocates a given resource group to the first node listed in the run-time failover domain that is also in the FailSafe membership; the point at which this allocation takes place is affected by the *failover attributes*.

failover policy

The method used by FailSafe to determine the destination node of a failover. A failover policy consists of a *failover domain*, *failover attributes*, and a *failover script*. A failover policy name must be unique within the *pool*.

failover script

A failover policy component that generates a *run-time failover domain* and returns it to the FailSafe process. The process applies the failover attributes and then selects the first node in the returned failover domain that is also in the current FailSafe membership.

FailSafe membership

The list of FailSafe nodes in a cluster on which FailSafe can make resource groups online. It differs from the CXFS membership. For more information about CXFS, see the *CXFS Version 2 Software Installation and Administration Guide*.

FailSafe database

See *cluster database*.

`fs2d` database membership

Also known as *user-space membership*. The group of nodes in the pool that are accessible to `fs2d` and therefore can receive cluster database updates; this may be a subset of the nodes defined in the pool.

heartbeat messages

Messages that cluster software sends between the nodes that indicate a node is up and running. Heartbeat messages and *control messages* are sent through a node's network interfaces that have been attached to a control network. A node can be attached to multiple control networks.

heartbeat interval

Interval between heartbeat messages. The node timeout value must be at least 10 times the heartbeat interval for proper FailSafe operation (otherwise false failovers may be triggered). The higher the number of heartbeats (smaller heartbeat interval), the greater the potential for slowing down the network. Conversely, the fewer the number of heartbeats (larger heartbeat interval), the greater the potential for reducing availability of resources.

initial failover domain

The ordered list of nodes, defined by the administrator when a failover policy is first created, that is used the first time a cluster is booted. The ordered list specified by the initial failover domain is transformed into a *run-time failover domain* by the *failover script*; the run-time failover domain is used along with failover attributes to determine the node on which a resource group should reside. With each failure, the failover script takes the current run-time failover domain and potentially modifies it; the initial failover domain is never used again. Depending on the run-time conditions and contents of the failover script, the initial and run-time failover domains may be identical. See also *run-time failover domain*.

key/value attribute

A set of information that must be defined for a particular resource type. For example, for the resource type `filesystem` one key/value pair might be *mount_point=/fs1* where *mount_point* is the key and *fs1* is the value specific to the particular resource being defined. Depending on the value, you specify either a `string` or `integer` data type. In the previous example, you would specify `string` as the data type for the value *fs1*.

log configuration

A log configuration has two parts: a *log level* and a *log file*, both associated with a *log group*. The cluster administrator can customize the location and amount of log output, and can specify a log configuration for all nodes or for only one node. For example, the `crsd` log group can be configured to log detailed level-10 messages to the `/var/cluster/ha/log/crsd-foo` log only on the node `foo` and to write only minimal level-1 messages to the `crsd` log on all other nodes.

log file

A file containing notifications for a particular *log group*. A log file is part of the *log configuration* for a log group. By default, log files reside in the `/var/cluster/ha/log` directory, but the cluster administrator can customize this. Note: FailSafe logs both normal operations and critical errors to `/var/adm/SYSLOG`, as well as to individual logs for specific log groups.

log group

A set of one or more FailSafe processes that use the same log configuration. A log group usually corresponds to one daemon, such as `gcd`.

log level

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

LUN

Logical unit number

monitor host

A workstation that has a display and is running the IRIS Desktop, on which PCP for FailSafe has installed the monitor client.

node

A single IRIX kernel image. Usually, a node is an individual computer. This use of the term node does not have the same meaning as a node in an SGI 2000 or SGI Origin 3000 system.

node ID

A 16-bit positive integer that uniquely defines a node. During node definition, FailSafe will assign a node ID if one has not been assigned by the cluster administrator. Once assigned, the node ID cannot be modified.

node timeout

If no heartbeat is received from a node in this period of time, the node is considered to be dead. The node timeout value must be at least 10 times the heartbeat interval for proper FailSafe operation (otherwise false failovers may be triggered).

notification command

The command used to notify the cluster administrator of changes or failures in the cluster, nodes, and resource groups. The command must exist on every node in the cluster.

offline resource group

A resource group that is not highly available in the cluster. To put a resource group in offline state, FailSafe stops the group (if needed) and stops monitoring the group. An offline resource group can be running on a node, yet not under FailSafe control. If the cluster administrator specifies the *detach only* option while taking the group offline, then FailSafe will not stop the group but will stop monitoring the group.

online resource group

A resource group that is highly available in the cluster. When FailSafe detects a failure that degrades the resource group availability, it moves the resource group to another node in the cluster. To put a resource group in online state, FailSafe starts the group (if needed) and begins monitoring the group. If the cluster administrator specifies the *attach only* option while bringing the group online, then FailSafe will not start the group but will begin monitoring the group.

owner host

A system that can control a node remotely, such as power-cycling the node. At run time, the owner host must be defined as a node in the pool.

owner TTY name

The device file name of the terminal port (TTY) on the *owner host* to which the system controller serial cable is connected. The other end of the cable connects to the node with the system controller port, so the node can be controlled remotely by the owner host.

plug-in

The set of software required to make an application highly available, including a resource type and action scripts. There are plug-ins provided with the base FailSafe release, optional plug-ins available for purchase from SGI, and customized plug-ins you can write using the instructions in the *IRIS FailSafe Version 2 Programmer's Guide*.

pool

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

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

port password

The password for the system controller port, usually set once in firmware or by setting jumper wires. (This is not the same as the node's root password.)

powerfail mode

When powerfail mode is turned `on`, FailSafe tracks the response from a node's system controller as it makes reset requests to a node. When these requests fail to reset the node successfully, FailSafe uses heuristics to try to estimate whether the machine has been powered down. If the heuristic algorithm returns with success, FailSafe assumes the remote machine has been reset successfully. When powerfail mode is turned `off`, the heuristics are not used and FailSafe may not be able to detect node power failures.

process membership

A list of process instances in a cluster that form a process group. There can multiple process groups per node.

re-MACing

The process of moving the physical medium access control (MAC) address of a network interface to another interface. It is done by using the `macconfig` command.

resource

A single physical or logical entity that provides a service to clients or other resources. For example, a resource can be a single disk volume, a particular network address, or an application such as a web server. A resource is generally available for use over time on two or more nodes in a cluster, although it can be allocated to only one node at any given time. Resources are identified by a resource name and a resource type. Dependent resources must be part of the same resource group and are identified in a resource dependency list.

resource dependency

The condition in which a resource requires the existence of other resources.

resource dependency list

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

resource group

A collection of resources. A resource group is identified by a simple name; this name must be unique within a cluster. Resource groups cannot overlap; that is, two resource groups cannot contain the same resource. All interdependent resources must be part of the same resource group. If any individual resource in a resource group becomes unavailable for its intended use, then the entire resource group is considered unavailable. Therefore, a resource group is the unit of failover.

resource keys

Variables that define a resource of a given resource type. The action scripts use this information to start, stop, and monitor a resource of this resource type.

resource name

The simple name that identifies a specific instance of a resource type. A resource name must be unique within a given resource type.

resource type

A particular class of resource. All of the resources in a particular resource type can be handled in the same way for the purposes of failover. Every resource is an instance of exactly one resource type. A resource type is identified by a simple name; this name must be unique within a cluster. A resource type can be defined for a specific node or for an entire cluster. A resource type that is defined for a node overrides a cluster-wide resource type definition with the same name; this allows an individual node to override global settings from a cluster-wide resource type definition.

resource type dependency

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

resource type dependency list

A list of resource types upon which a resource type depends.

run-time failover domain

The ordered set of nodes on which the resource group can execute upon failures, as modified by the failover script. The run-time failover domain is used along with failover attributes to determine the node on which a resource group should reside. See also *initial failover domain*.

start/stop order

Each resource type has a start/stop order, which is a nonnegative integer. In a resource group, the start/stop orders of the resource types determine the order in which the resources will be started when FailSafe brings the group online and will be stopped when FailSafe takes the group offline. The group's resources are started in increasing order, and stopped in decreasing order; resources of the same type are started and stopped in indeterminate order. For example, if resource type `volume` has order 10 and resource type `filesystem` has order 20, then when FailSafe brings a resource group online, all volume resources in the group will be started before all file system resources in the group.

system controller port

A port located on a node that provides a way to power-cycle the node remotely. Enabling or disabling a system controller port in the cluster database (CDB) tells FailSafe whether it can perform operations on the system controller port. (When the port is enabled, serial cables must attach the port to another node, the owner host.) System controller port information is optional for a node in the pool, but is required if the node will be added to a cluster; otherwise resources running on that node never will be highly available.

tie-breaker node

A node identified as a tie-breaker for FailSafe to use in the process of computing the FailSafe membership for the cluster, when exactly half the nodes in the cluster are up and can communicate with each other. If a tie-breaker node is not specified, FailSafe will use the node with the lowest node ID in the cluster as the tie-breaker node.

type-specific attribute

Required information used to define a resource of a particular resource type. For example, for a resource of type `filesystem` you must enter attributes for the resource's volume name (where the file system is located) and specify options for how to mount the file system (for example, as readable and writable).

user-space membership

See `fs2d` *database membership*.

Index

A

action script
  execution of, 317
  set of, 13
action script timeouts, modifying, 165
activate FailSafe, 199
ACTIVE cluster status, 235
active/backup configuration, 37
add nic, 120
add/remove dependencies for a resource definition, 177
add/remove dependencies for a resource type, 159
add/remove nodes in the cluster, 125
administration daemon, 322
administration of FailSafe, 21
administrative workstation subsystem software, 340
aliasing IP addresses, 22
application failover domain, 12
application monitoring, 20
applications, highly available, 25
ATM LAN emulation failover, 16
attributes for failover, 13
Auto_Failback failover attribute, 13, 184
Auto_Recovery failover attribute, 184
AutoLoad boot parameter, 59, 67

B

backup and restore, 258
backup, CDB, 258
broadcast address, 171

C

cad
  verify it is running, 286
CAD options file, 62
cad process, 104, 285
CD contents, 337
CDB
  backup and restore, 258
  maintenance, 285
  recovery, 285
cdbBackup and cdbRestore, 258
cdbreinit command, 287
chkconfig, 72, 104
clconfd process, 104
CLI (cmgr), 94
cli log, 207
cluster
  convert, 142
  define, 137
  delete, 143
  display, 145
  error recovery, 278
  modify, 141
cluster (terminology), 3
cluster administration daemon, 322
cluster database
  backup and restore, 258
  recovery, 287
  sync failure, 285
  terminology, 4
cluster database security, 181
cluster environment, 3
Cluster Manager GUI
  See IRIS FailSafe Cluster Manager GUI, 87
cluster membership, 4
cluster process group, 14

cluster status, 235
cluster tasks, 136
cluster type, 49
cluster_admin subsystem, 308, 310
cluster_control subsystem, 308, 310
cluster_mgr command, 93
cluster_services subsystem, 308, 310
cluster_status, 234
cmgr
  -c option, 98
  command line execution, 98
  exiting, 96
  help, 94
  invoking a shell, 101
  prompt mode, 94
  scripts and, 98
  See cmgr, 94
  startup script, 97
  template files, 100
cmgr command, 93
cmgr-templates directory, 100
CMGR_START_FILE environment variable, 97
cmond
  verify it is running, 285
cmond process, 104, 285
cmond.options file, 66
coexecution with IRIS FailSafe, 48
collector host installation, 79
communication paths, 311
components, 321
concepts, 3
configuration overview, 27
configuration parameters
  filesystem, 44
  IP address, 48
  logical volumes, 42
configuration parameters for disks, 39
configuration planning
  disk, 33
  filesystem, 42
  IP address, 45
  logical volume, 39
  overview, 29
configuration tasks
  add/remove dependencies for a resource definition, 177
  add/remove dependencies for a resource type, 159
  add/remove nodes in the cluster, 125
  add/remove resources in resource group, 196
  cluster definition, 137
  cluster tasks, 136
  connectivity test, 262
  convert a CXFS cluster, 142
  convert a CXFS node to FailSafe, 131
  customize failure detection, 113
  customize resource failover behavior, 114
  customize resource group failover behavior, 114
  defaults in cmgr, 107
  define a custom resource, 113
  define a failover policy, 182
  define a node, 116
  define a resource, 169
  define a resource group, 193
  define a resource type, 146
  delete a cluster, 143
  delete a failover policy, 191
  delete a node, 132
  delete a resource, 180
  delete a resource group, 196
  delete a resource type, 167
  display a cluster, 145
  display a failover policy, 192
  display a node, 135
  display a resource, 181
  display a resource group, 198
  display a resource type, 168
  failover policy, 182
  fix or upgrade cluster nodes, 112
  guided configuration, 108
  HA services tasks, 199
  load a resource type, 162
  log groups, 210

make changes to existing cluster, 112
modify a cluster definition, 141
modify a failover policy definition, 188
modify a node definition, 126
modify a resource definition, 179
modify a resource group definition, 195
modify a resource type, 162
monitoring intervals, 106
mount a filesystem, 141
move a resource group, 197
name restrictions, 106
node addition to cluster, 126
node deletion, 132
node resets, 257
node tasks, 116
notify administrator of cluster changes, 137
optimize node usage, 112
preliminary steps, 103
redefine a resource for a specific node, 175
redefine a resource type for a specific node, 155
resource group tasks, 193
resource load redistribution, 115
resource tasks, 168
resource type tasks, 146
set FailSafe HA parameters, 204
set log configuration, 206
set up a new cluster, 109
set up an existing CXFS cluster for FailSafe, 111
set up an HA resource group, 110
start HA services, 199
stop HA services, 200
timeout values, 106
connectivity test, 261, 262
control network, 7, 16, 117
changing in cluster, 293
recovery, 284
Controlled_Failback failover attribute, 13, 184
controllers, 17
conversion between CXFS and FailSafe, 52
convert a FailSafe cluster, 111, 142
convert a FailSafe node, 111, 131
convert from FailSafe taskset, 111
corepluspid system parameter, 67
create a cluster, 137
Critical_RG failover attribute, 185
crsd
verify it is running, 286
crsd log, 207
crsd process, 104, 285, 310
ctrl+c ramifications, 226
custom failover scripts, 13
custom resource, 113
customize failure detection, 113
customize resource failover behavior, 114
customize resource group failover behavior, 114
CXFS, 309
and FailSafe, 221
configuration example, 221
exporting filesystems, 222
CXFS GUI, 52
CXFS membership, 5
CXFS metadata servers and failover domain, 50
CXFS resource type, 307
CXFS resource type for FailSafe, 50

D

daemons, 104, 285
database membership, 4
database security, 181
deactivate FailSafe, 200
deactivating HA services, 257
defaults, 107
define a cluster, 137
define a custom resource, 113
define a failover policy, 182
define a node, 116
define a resource, 169
define a resource group, 193
define a resource type, 146
delete a cluster, 143
delete a failover policy, 191

delete a node, 132
delete a resource, 180
delete a resource group, 196
delete a resource type, 167
dependency list, 10
deskside storage systems, 16
destructive mode, 267
detect failures, 113
developer's guide, 13
devname-group, 42
devname-mode, 42
devname-owner, 42
diagnostic command overview, 261
diags log, 207
diags_nodename log file, 261
DISCOVERY state, 240
disk configuration planning, 33
disk connections, 17
disk storage, 16
display a cluster, 145
display a failover policy, 192
display a resource, 181
display a resource group, 198
display a resource type, 168
display nodes, 135
DMF resource type, 307
DNS, 60
domain, 12, 186
domain name service , 60
DOWN node state, 242
driver subsystems, 338
dual controllers, 17
dual hubs, 17
dual pathing, 17
dual vaults, 17
dual-active, 325
dynamic management, 20

E

EL-16, 16
EL-8+, 16
EL-8+ multiplexer driver subsystems, 338
emulation failover, 16
ERROR cluster status, 235
error state, resource group, 240
ESP, 249
/etc/config/cad.options file, 62
/etc/config/cmond.options file, 66
/etc/config/fs2d.options file, 63
/etc/config/netif.options, 61
/etc/config/nfsd.options, 43
/etc/config/routed.options, 71
/etc/fstab, 43, 44
/etc/hosts, 60
/etc/hosts file, 45
/etc/inetd.conf, 286
/etc/inittab, 74
/etc/nsswitch.conf, 60, 73
/etc/services file, 62
/etc/sys_id, 60
Ethernet, 16
example
  startup script, 97
  XLV naming scheme, 40
examples
  add a node, 291
  add a node to the cluster, 142
  add new resource groups or resources in an active cluster, 297
  add/remove dependencies for a resource type, 161
  administrative communication within one node , 311
  administrative communication within one node under coexecution, 316
  attributes, 149
  bring a resource group online, 252
  change the log level, 211
  chkconfig, 72
  cluster information display, 98
  cluster_status, 234

commands used to create NFS, CXFS and statd_unlimited resources, 51
communication between nodes in the pool, 314
communication for a node not in the cluster, 314
configuration planning process, 31
configuration types, 18
configuration with four resource groups, 32
configuration with two resource groups, 34
configure network interfaces, 69
convert a node, 131
convert the cluster, 143
create a resource group, 223
daemon communication within one node, 313
daemon communication within one node under coexecution, 317
define a cluster, 140
define a node, 123
define a resource, 173
define a resource group, 194
define a resource type, 151
define failover policy, 188
define the log group, 210
delete a cluster, 144
delete a node, 134, 292
delete a resource group, 196
dependencies, 11
dependencies for a resource type, 160
detach a resource group, 278, 292
determine hostname, 105
display a cluster, 145
display a resource group, 198
/etc/config/cad.options, 63
/etc/config/fs2d.options, 65
/etc/config/routed.options, 71
/etc/hosts contents and hostname resolution, 60
/etc/inittab, 74
/etc/nsswitch.conf, 60, 73
/etc/services, 62, 290
export CXFS filesystems, 222
failover domain, 187
FailSafe configuration, 214
FailSafe Manager GUI, 92
FailSafe membership, 5, 6
failure of a resource's monitor action, 148
filesystem configuration, 44
filesystems and logical volumes, 45
GUI showing details for a resource, 93
HA IP address configuration, 48
haStatus, 243
heartbeat response statistics, 300
heterogeneous clusters for an IP_address resource, 175
increase the statd resource type monitor executable timeout, 165
interface configuration, 69
IRISconsole reset model, 18
Local failover of HA IP address, 220
log file management, 259
log file name, 208
logging information and /etc/config/fs2d.options, 65
logical volume configuration, 41
message paths for action scripts and failover policy scripts, 320
modify a cluster, 134
modify a cluster to include a CXFS filesystem, 221
modify a node, 131
modify a resource group, 195
modify the resource type timeouts, 166
monitoring system status, 234
move a resource group, 198, 281, 296
mutual dependency of resources is not allowed, 178
name of a filesystem resource, 170
network reset model, 18
nodes in the failover domain, 186
non-shared disk configuration and failover, 35
offline a resource group, 281
offline detatch, 232
output when just the initial daemons are running, 105
partition id determination, 118

partitioning, 129
patch installation, 74
PCP, 300
performing tasks, 91
pool and cluster concepts, 3
prompt mode, 94
redefine a resource for a specific node, 175
remove the error state of a resource, 283
reset models, 18
resource dependency, 177
resource group, 9
resource group maintenance and error recovery, 280
resource group recovery, 279
resource monitoring statistics, 301
resource type dependencies, 11
resources, 170
ring configuration, 18
rotating log files, 259
script files, 99
script to define a three-node cluster, 214
server-to-server reset model, 18
set group ID, 174
setting configuration defaults for cmgr, 107
sgi-cad, 62
sgi-cmsd, 62
sgi-crsd, 62
sgi-gcd, 62
shared disk configuration for active/backup use, 37
shared disk configuration for dual-active use, 38
show cluster, 133
show failover policy, 227
show nodes in pool/cluster, 136
single node, 227
software layers, 309
star configuration, 18
start cluster daemons, 104
start HA services, 200, 291, 295
start HA services on a subset of nodes, 200
stop HA services, 203, 280, 292
system components, 15
test a failover policy, 269
test a resource, 265
test a resource type, 266
test logical volumes, 266
test multiple nodes, 263
test network connectivity, 264
test serial connections, 263
testing the private network interface, 84
testing the serial reset connection, 84
three-node cluster, 213
tie-breaker and membership, 7
two-node cluster, 20
two-node configuration, 18
two-node use, 230
updating from IRIS FailSafe 1.2, 323
upgrade software in an active cluster, 296
upgrades, 327
/usr/lib/aliases, 73
verify cluster daemons are running, 286
verify that chkconfig flags are on, 104
exclusive action script, 13

F

failover, 12
and recovery processes, 26
behavior of a resource, 114
behavior of a resource group, 114
description, 26
resource group, 250
failover attributes, 13, 183, 189
failover domain, 12, 186
failover policy, 12
define, 182
delete, 191
display, 192
failover attributes, 183, 189
failover domain, 186
failover script, 189
modify, 188

tasks, 182
test, 263, 269
failover policy modification, 188
failover script, 13, 189, 317
FailSafe
membership, 274
FailSafe base software, 308
FailSafe coexecution, 48
FailSafe Manager
overview, 89
See "configuration tasks", 109
FailSafe Manager GUI overview, 87
FailSafe membership, 4, 5, 274, 275
failsafe2 subsystem, 310
failure detection, 113
fault-tolerant systems, definition, 1
FDDI, 16
features, 20
Fibre Channel, 16
filesystem
configuration parameters, 44
configuration planning, 42
resource, 170
test, 267
filesystem mounting, 141
filesystem resource type, 307
fine-grain failover, 21
fix cluster nodes, 112
FORE Systems ATM cards and switch, 16
fs2d
verify it is running, 285
fs2d database membership, 4
fs2d options file, 63
fs2d process, 104, 285
fsafe.srm* metrics, 343

G

giveaway, giveback, 326
GUI
recovery, 287
See IRIS FailSafe Cluster Manager GUI, 87
GUI overview, 87
GUI will not run, 285

H

HA parameters
set, 204
HA services
start, 199
stop, 200
HA services tasks, 199
ha.conf, 324, 326
ha_agent log, 207
ha_cfginfo, 326
ha_cilog, 327
ha_cmsd log, 207
ha_cmsd process, 310
ha_fsd log, 207
ha_fsd process, 13, 310
ha_gcd log, 207
ha_gcd process, 310
ha_get_field(), 326
ha_get_info(), 326
ha_ifd process, 310
ha_ifd log, 207
ha_ifmx2 process, 310
ha_script log, 207
ha_srmd log, 207
ha_srmd process, 310
ha_sybs2 process, 310
hardware components, 15
hardware device, adding to cluster, 298
haStatus script, 243
heartbeat interval, 204
heartbeat network, 7, 16, 117
help
for cmgr, 94
for GUI, 109
high-availability infrastructure, 308

highly available systems, definition, 1
hostname
  control network, 117
hostname determination, 105
hosts file, 60
hubs, 17

I

icons and states, 235
ifconfig, 265
IFD
  See Interface Agent Daemon, 311
INACTIVE cluster status, 235
INACTIVE node state, 242
Informix resource type, 307
informix_rdbms subsystem, 310
infrastructure, 308
initial cluster configuration
  GUI, 108
initial failover domain, 12, 186, 190
INITIALIZING state, 240
inittab file, 73
InPlace_Recovery failover attribute, 184
install resource type, 162
installation, 56
installing
  patches, 74
Interface Agent Daemon (IFD), 311
INTERNAL ERROR state, 239, 240
IP address
  configuration planning, 45
  fixed, 22
  highly available, 22
  local failover, 220
  overview, 22
  planning, 31, 45
  resource, 171
IP address and control network, 117
IP address resource type, 307
IP aliasing, 22
IRIS FailSafe coexecution, 48
IRISconsole reset model, 19
is_* commands, 120

J

Java Plug-in, 338
java supported release level, 57
java_plugin, 58
JBOD, 16, 17

K

ksh shell, 327

L

L1, 121
L2, 121
LAN emulation failover, 16
layers, system software, 307
load a resource type, 162
load redistribution, 115
local failover, IP address, 220
local restart, 21
log configuration, 206
log files, 208, 272
  management, 259
log levels, 207
log messages
  debug, 273
  error, 273
  normal, 273
  syslog, 273
  warning, 273
logical volume
  configuration planning, 39
  creation, 67

parameters, 42
logical volume testing, 266
lonely state, 227

M

MAC address impersonation, 23
MAC address resource, 171
MAC_address resource type, 307
maintenance mode, 255
make changes to an existing cluster, 112
managed resources, 20
membership, 4
FailSafe, 274
See "cluster membership", 4
metrics exported by PCP, 343
mgr
-p option, 94
migrating from IRIS FailSafe 1.2, 323
mirrored disks, 18
mkpart, 118, 121
MMSC, 121
mode, 44
modify a cluster, 112
modify a failover policy definition, 188
modify a node, 126
modify a resource definition, 179
modify a resource group definition, 195
modify a resource type, 162
monitor action script, 13
MONITOR ACTIVITY UNKNOWN error state, 240
monitor applications, 20
monitor host, 82
monitor license, 82
monitor-level, 44
monitoring interval, 106
monitoring system status, 234
move a resource group, 197
MSC, 121
multihosted RAID disks devices, 18
multiplexer driver subsystems, 338

N

name restrictions, 60, 106
name service daemon, 60
netif.options, 61
netif.options file, 71
Netscape resource type, 308
Netscape servers, testing with cmgr, 268
network connectivity, 262
network connectivity test, 264
network information service, 60
network interface
configuration, 69
overview, 22
network mask, 171
network reset model, 19
network segment, 7
networks, 7
NFS and CXFS filesystems, 222
NFS resource type, 307
NIS, 60
NIS database, 70
NO AVAILABLE NODES error state, 240
NO ERROR error state, 240
NO MORE NODES IN AFD error message, 282
node
add/remove, 125
adding to cluster, 289
configuration, 55
convert, 131
define, 116
delete, 132
deleting from cluster, 291
display, 135
error recovery, 279
highly available, 22
modify, 126
reset, 257, 274
resets, 257
state, 242
status, 242

terminology, 3
timeout, 204
wait time, 204
NODE NOT AVAILABLE error state, 240
node states, 326
node tasks, 116
node type, 50
NODE UNKNOWN error state, 240
node usage optimization, 112
node-specific resource, 175
node-specific resource type, 155
Node_Failures_Only failover attribute, 185
non-IRIX administrative workstation software, 343
notification, 141
notify administrator of cluster changes, 137
nsadmin, 61
nsd, 60
nsswitch.conf, 60
NVRAM variables, 67

O

OFFLINE state, 239
OFFLINE-PENDING state, 239
ONLINE state, 239
ONLINE-MAINTENANCE state, 239, 255
ONLINE-PENDING state, 239
ONLINE-READY state, 239, 251
optimize node usage, 112
Oracle resource type, 307
oracle_rdbms subsystem, 310
ordered failover script, 13
overlap of resource groups, 9
overview of FailSafe, 1

P

partition, 121, 127
partition id, 118
patch installation, 74
pathing, 17
PCP installation, 79
PCP metrics, 343
pcp_eoe.sw, 79, 82
PCPMON, 82
performance metrics, 82
Performance Metrics Domain Agent (PMDA), 81
pinging system controller, 242
plexed disks, 18
plug-in
custom, 308
terminology, 14
PMDA, 81
pool, 3
powerfail mode, 204
preliminary configuration steps, 103
primary node, 37
private network, 7
private network interface, 84
process group, 14
process membership, 5
programmer's guide, 13

Q

quorum, 5

R

rackmount storage systems, 16
RAID, 16, 17
re-MACing, 23
dedicated backup interfaces required, 46
determining if required, 46
recovery
overview, 271
procedures, 277
redefine a resource for a specific node, 175
redefine a resource type for a specific node, 155

redistribute resource load, 115
redundancy, 18
Remote System Control port, 16
remove a node from the pool, 132
remove dependencies for a resource definition, 177
remove dependencies for a resource type, 159
remove nic, 120
remove nodes, 141
remove resources in resource group, 196
reset connection, 84
reset hardware, 16
reset models, 18, 19
resetting nodes, 257, 274
resource
  adding to cluster, 297
  define, 169
  delete, 180
  dependencies, 177
  dependency list, 10
  display, 181
  filesystem, 170
  IP address, 171
  MAC address, 171
  modify, 179
  name, 9
  NFS, 223
  node-specific, 175
  owner, 241
  recovery, 283
  statd_unlimited, 173
  status, 238
  terminology, 8
  volume, 172
resource failover behavior, 114
resource group
  add/remove resources, 196
  adding to cluster, 297
  bringing online, 250
  creation example, 223
  define, 193
  delete, 196
  dependencies, 196
  detaching, 252
  display, 198
  error state, 240
  failover, 250
  forcing offline, 252
  modify, 195
  monitoring, 255
  move, 197
  moving, 254
  recovery, 280
  resume monitoring, 256
  state, 239
  status, 238
  suspend monitoring, 255
  taking offline, 250, 252
  terminology, 9
  test, 268
resource group failover behavior, 114
resource group tasks, 193
resource group taskset, 110
resource load redistribution, 115
resource tasks, 168
resource type
  define, 146
  delete, 167
  dependencies, 159
  dependency list, 10
  display, 168
  load, 162
  modify, 162
  NFS, 223
  node-specific, 155
  terminology, 8
resource type tasks, 146
resources, 21
resources in resource group, 196
restart (local), 21
restart action script, 13
restore, CDB, 258
ring configuration, 18
ring reset, 73

rotating log files, 259
round-robin failover script, 13
run-time failover domain, 12, 186, 190

S

Samba resource type, 308
/sbin/ksh, 327
/sbin/sh, 327
SCSI bus, 16
SCSI ID parameter, 67
security of the cluster database, 181
serial cable recovery, 284
serial connections, 262
serial connections test, 263
serial line, 16
serial port configuration, 73
serial reset connection, 84
server-to-server reset model, 19
set commands, 120
set FailSafe HA parameters, 204
set log configuration, 206
set up a new cluster, 109
SGI 200, 15
SGI 2000, 15
SGI Origin 3000, 15
sgi-cad, 62
sgi-cmsd, 62
sgi-crsd, 62
sgi-gcd, 62
sh shell, 327
shared disk issues, 42
shells, 326
show a cluster, 145
show a node, 135
show a resource type, 168
single controller, 17
single hub, 17
single pathing, 17
single vault, 17
single-node use, 227
software overview, 337
SPLIT RESOURCE GROUP (EXCLUSIVITY)
    error state, 240, 282
srmd executable error , 321
SRMD EXECUTABLE ERROR error state, 240
ST16XX, 16
standby, 326
star configuration, 17
start action script, 13, 321
start cluster daemons, 104
start HA services, 199
startup script for cmgr, 97
statd_unlimited resource, 173
statd_unlimited resource type, 307
state, resource group, 239
states and icons, 235
status
  cluster, 235
  node, 242
  resource, 238
  resource group, 238
  system controller, 238
  system, overview, 233
stop action script, 13, 321
stop HA services, 200
stopping HA services, 257
  force option, 202
storage connection, 16
subsystems on the CD, 337
sys_id, 60
sysctrl* commands, 120
SYSLOG, 65
syslog messages, 273
system
  software communication paths, 311
system configuration defaults, 107
system controller
  status, 238
system controller types, 121
system files, 59
system operation considerations, 226

system software
components, 321
layers, 307
system status, 233

T

takeover, takeback, 326
tasks, 91
See "configuration tasks", 103
TCP and NFS, 43
tcpmux, 286
tcpmux/sgi_sysadm, 286
template files, 100
terminology, 3
testing, 83
three-node cluster, example, 213
tie-breaker node, 6, 204, 274
timeout values, 106
timeouts, action script, 165
TMF resource type, 308
TP9100, 16
TP9400, 16
TP9900 , 16
troubleshooting
GUI will not run, 285
two-node clusters: single-node use, 227
two-node configuration, 20
two-node use, 230

U

UDP, 43
Unified Name Service, 61
unit of failover, 327
UNIX process group, 14
UNKNOWN cluster status, 235, 278
UNKNOWN node state, 242
UNS, 61
UP node state, 242
updating from IRIS FailSafe 1.2, 323
upgrade cluster nodes, 112
upgrade FailSafe, 21
upgrading
FailSafe software, 296
OS software, 295
user privileges, 181
user-space membership, 4
/usr/etc/ifconfig, 84, 265
/usr/lib/aliases, 73

V

/var/adm/SYSLOG, 65
/var/cluster/cdb/cdb/db, 326
/var/cluster/ha directory, 322
/var/cluster/ha/common_scripts/scriptlib, 326
/var/ha/actions/common.vars, 326
/var/ha/logs, 327
/var/pcp/pmdas/fsafe, 81
vaults, 17
verify cluster daemons are running, 285
volume
resource, 172
test, 266
volume resource type, 307
volume-name, 44

W

wsync mode, and NFS filesystems, 43

X

XFS filesystem creation, 67
XFS resource type, 307
XLV logical volume creation, 67
XLV logical volume resource type, 308

Y

ypmatch, 70