

IRIS FailSafe™ Version 2
Administrator's Guide

007-3901-008

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 Gemma Exton, Scott Henry, Vidula Iyer, Ashwinee Khaladkar, Harald Kaul, Tony Kavadias, Linda Lait, Michael Nishimoto, Nate Pearlstein, Alain Renaud, Wesley Smith, Bill Sparks, Paddy Sreenivasan, Dan Stekloff, Rebecca Underwood, Manish Verma

COPYRIGHT

© 1999–2003, Silicon Graphics, Inc. All rights reserved; provided portions may be copyright in third parties, as indicated elsewhere herein. No permission is granted to copy, distribute, or create derivative works from the contents of this electronic documentation in any manner, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

LIMITED RIGHTS LEGEND

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

TRADEMARKS AND ATTRIBUTIONS

Silicon Graphics, SGI, the SGI logo, IRIS, IRIX, Onyx, Onyx2, and Origin are registered trademarks and CXFS, FailSafe, IRIS FailSafe, NUMAlink, Performance Co-Pilot, SGI SAN Server, Silicon Graphics Fuel, and XFS are trademarks of Silicon Graphics, Inc., in the United States and/or other countries worldwide.

Informix is a registered trademark of IBM. Java is a trademark of Sun Microsystems, Inc. Linux is a registered trademark of Linus Torvalds, used with permission by Silicon Graphics, Inc. Netscape and Netscape FastTrack Server are trademarks of Netscape Communications Corporation. Oracle is a registered trademark of Oracle Corporation. UNIX is a registered trademark of The Open Group in the United States and other countries.

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:

- Changes to the FailSafe graphical user interface (GUI):
 - New login connection choices, including support for a remote shell connection, which connects to the server via a user-specified command shell, such as `rsh(1C)` or `ssh(1)`. See "Starting the GUI", page 87.
 - The ability for the `root` user to grant other users permission to execute specific GUI tasks. See:
 - "GUI Features", page 90
 - "Granting Task Execution Privileges to Users", page 286
 - "Key to Icons and States", page 258
 - Use of Java2 for the FailSafe GUI, which simplifies the Java installation and co-operation with third-party GUIs. This also enhances the ability to run the GUI through a web browser (via `http://server/FailSafeManager/`). See "Install Software", page 55.
 - Information about using the right mouse button to access tasks appropriate to the selected GUI item. See "Performing Tasks", page 94.
- A new attribute to the `filesystem` resource type allows you to specify whether or not the `nfsd(1M)` daemon will be stopped before unmounting the filesystem and then restarted after the unmount is complete. See "filesystem Attributes", page 176, and Table 5-2, page 182.
- References to the `statd` resource type have been replaced by `statd_unlimited`. There could be only two `statd` resource instances in a cluster; `statd_unlimited` resources have no such restriction.
- More information about memberships, quorums, and tiebreakers; see Appendix B, "Memberships and Quorums", page 355.

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
007	October 2002 Supports the IRIS FailSafe 2.1.4 release
008	April 2003 Supports the IRIS FailSafe 2.1.5 release

Contents

About This Guide	xxix
Audience	xxix
Assumptions	xxix
Structure of This Guide	xxix
Related Documentation	xxx
Obtaining Publications	xxxii
Conventions	xxxiii
Reader Comments	xxxiii
1. Overview	1
High Availability and FailSafe	1
Cluster Environment	2
Terminology	2
Cluster	2
Node	3
Pool	3
Cluster Database	4
Membership	4
Quorum	5
Private Network	6
Resource	7
Resource Type	7
Resource Name	8
Resource Group	8
Dependency	8

Failover	10
Failover Policy	11
Failover Domain	11
Failover Attribute	11
Failover Scripts	12
Action Scripts	12
Cluster Process Group	13
Plug-In	13
Hardware Components	14
Disk Connections	16
Supported Configurations	16
Additional Features	19
Dynamic Management	19
Fine-Grain Failover	20
Local Restarts	20
Administration	20
Highly Available Resources	20
Nodes	21
Network Interfaces and IP Addresses	21
Disks	23
Highly Available Applications	24
Failover and Recovery Processes	25
Overview of Configuring and Testing a New Cluster	26
2. Configuration Planning	27
Overview	27
Questions to be Answered	27
Example	29

Disk Configuration	30
Planning Disk Configuration	30
Configuration Parameters for Disks	36
XLV Logical Volume Configuration	36
Planning XLV Logical Volumes	36
Example Logical Volume Configuration	38
Resource Attributes for Logical Volumes	39
XFS Filesystem Configuration	39
Planning XFS Filesystems	39
Example XLV Filesystem Configuration	41
HA IP Address Configuration	42
Planning Network Interface and HA IP Address Configuration	42
Example HA IP Address Configuration	45
Local Failover of HA IP Addresses	45
Coexecution of CXFS and FailSafe	45
Size of the Coexecution Cluster	47
Cluster Type	47
Node Types for CXFS Metadata Servers	48
CXFS Metadata Servers and Failover Domain	48
CXFS Resource Type for FailSafe	48
Separate CXFS and FailSafe GUIs	50
Conversion Between CXFS and FailSafe	50
Network Interfaces	51
Local XVM Volumes	51
XVM Resource Type for FailSafe	51
Resource Attributes for Local XVM Volumes	52
Example Local XVM Volume Configuration	53

3. Installation and System Preparation	55
Install Software	55
Configure System Files	59
Hostname Resolution: /etc/sys_id, /etc/hosts, /etc/nsswitch.conf	59
/etc/services	61
/etc/config/cad.options	62
/etc/config/fs2d.options	63
Example 1	65
Example 2	65
/etc/config/cmond.options	66
Set the corepluspid System Parameter	66
Set NVRAM Variables	67
Create XLV Logical Volumes and XFS Filesystems	67
Configure Network Interfaces	68
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	78
Installing the Collector Host	79
Removing Performance Metrics from a Collector Host	81
Installing the Monitor Host	81
Test the System	82
Private Network Interface	83
Serial Reset Connection	83
4. Administration Tools	87
FailSafe Manager GUI	87

Starting the GUI	87
Launch Methods	87
Logging In	89
Making Changes from One Node	90
GUI Features	90
GUI Window Layout	91
File Menu	91
Edit Menu	91
Tasks Menu	92
Help Menu	92
View Menu	92
Selecting Items to View or Modify	93
Viewing Component Details	93
Performing Tasks	94
Getting More Information	94
Screens	95
cmgr Command	97
Getting Help	98
Using Prompt Mode	98
Completing Actions and Cancelling	100
Command Line Editing within cmgr	100
Long-Running Tasks	101
Startup Script	101
Entering Subcommands on the Command Line	102
Using Script Files	102
Creating a cmgr Script Automatically	104
Template Scripts	107

Invoking a Shell from within <code>cmgr</code>	108
5. Configuration	109
Preliminary Steps	109
Verify that the Cluster <code>chkconfig</code> Flag is On	110
Start the Cluster Daemons	110
Verify that the Cluster Daemons are Running	110
Determine the Hostname of the Node	111
Name Restrictions	112
Configuring Timeout Values and Monitoring Intervals	112
Setting Configuration Defaults with <code>cmgr</code>	113
Guided Configuration with the GUI	114
Set Up a New Cluster	115
Set Up a Highly Available Resource Group	116
Set Up an Existing CXFS Cluster for FailSafe	117
Fix or Upgrade Cluster Nodes	118
Make Changes to Existing Cluster	118
Optimize Node Usage	119
Define Custom Resource	119
Customize FailSafe Failure Detection	120
Customize Resource Group Failover Behavior	120
Customize Resource Failover Behavior	121
Redistribute Resource Load in Cluster	121
Node Tasks	122
Define a Node	122
Define a Node with the GUI	122
Define a Node with <code>cmgr</code>	126
Add or Remove Nodes in the Cluster	131

Add or Remove Nodes in the Cluster with the GUI	131
Modify a Node Definition	132
Modify a Node Definition with the GUI	132
Modify a Node with <code>cmgr</code>	134
Example of Partitioning	135
Convert a CXFS Node to FailSafe	137
Convert a CXFS Node to FailSafe with the GUI	137
Convert a Node to CXFS or FailSafe with <code>cmgr</code>	137
Delete a Node	138
Delete a Node with the GUI	138
Delete a Node with <code>cmgr</code>	139
Display a Node	141
Display a Node with the GUI	141
Display a Node with <code>cmgr</code>	141
Cluster Tasks	142
Define a Cluster	143
Define a Cluster with the GUI	143
Define a Cluster with <code>cmgr</code>	144
Modify a Cluster Definition	147
Modify a Cluster Definition with the GUI	147
Modify a Cluster Definition with <code>cmgr</code>	147
Convert a CXFS Cluster to FailSafe	148
Convert a CXFS Cluster to FailSafe with the GUI	148
Converting a CXFS Cluster to Failsafe with <code>cmgr</code>	149
Delete a Cluster	149
Delete a Cluster with the GUI	149
Delete a Cluster with <code>cmgr</code>	150
Display a Cluster	151

Display a Cluster with the GUI	151
Display a Cluster with <code>cmgr</code>	151
Resource Type Tasks	152
Define a Resource Type	152
Define a Resource Type with the GUI	152
Define a Resource Type with <code>cmgr</code>	155
Redefine a Resource Type for a Specific Node	161
Redefine a Resource Type for a Specific Node with the GUI	161
Define a Node-Specific Resource Type with <code>cmgr</code>	164
Add/Remove Dependencies for a Resource Type	164
Add/Remove Dependencies for a Resource Type with the GUI	165
Add/Remove Dependencies for a Resource Type with <code>cmgr</code>	167
Load a Resource Type	168
Load a Resource Type with the GUI	168
Load a Resource Type with <code>cmgr</code>	168
Modify a Resource Type Definition	168
Modify a Resource Type with the GUI	168
Modify a Resource Type with <code>cmgr</code>	171
Delete a Resource Type	173
Delete a Resource Type with the GUI	173
Delete a Resource Type with <code>cmgr</code>	174
Display a Resource Type	174
Display Resource Types with the GUI	174
Display Resource Types with <code>cmgr</code>	174
Resource Tasks	174
Define a Resource	175
Define a Resource with the GUI	175

CXFS Attributes	176
filesystem Attributes	176
IP_address Attributes	177
MAC_address Attributes	178
volume Attributes	178
XVM Attributes	179
Define a Resource with <code>cmgr</code>	180
Specify Resource Attributes with <code>cmgr</code>	181
Redefine a Resource for a Specific Node	183
Redefine a Resource for a Specific Node with the GUI	183
Redefine a Resource for a Specific Node with <code>cmgr</code>	184
Add/Remove Dependencies for a Resource Definition	184
Add/Remove Dependencies for a Resource Definition with the GUI	185
Add/Remove Dependencies for a Resource Definition with <code>cmgr</code>	186
Modify a Resource Definition	187
Modify a Resource Definition with the GUI	187
Modify a Resource Definition with <code>cmgr</code>	188
Delete a Resource	188
Delete a Resource with the GUI	188
Delete a Resource with <code>cmgr</code>	189
Display a Resource	189
Display a Resource with the GUI	189
Display a Resource with <code>cmgr</code>	189
Failover Policy Tasks	190
Define a Failover Policy	190
Define a Failover Policy with the GUI	190
Define a Failover Policy with <code>cmgr</code>	195

Modify a Failover Policy Definition	196
Modify a Failover Policy Definition with the GUI	196
Modify a Failover Policy Definition with <code>cmgr</code>	199
Delete a Failover Policy	199
Delete a Failover Policy with the GUI	199
Delete a Failover Policy with <code>cmgr</code>	200
Display a Failover Policy	200
Display a Failover Policy with the GUI	200
Display a Failover Policy with <code>cmgr</code>	200
Resource Group Tasks	201
Define a Resource Group	201
Define a Resource Group with the GUI	201
Define a Resource Group with <code>cmgr</code>	202
Modify a Resource Group Definition	203
Modify a Resource Group Definition with the GUI	203
Modify a Resource Group Definition with <code>cmgr</code>	203
Delete a Resource Group	203
Delete a Resource Group with the GUI	204
Delete a Resource Group with <code>cmgr</code>	204
Add/Remove Resources in Resource Group	204
Display a Resource Group	205
Display a Resource Group with the GUI	205
Display a Resource Group with <code>cmgr</code>	205
FailSafe HA Services Tasks	206
Start FailSafe HA Services	206
Start FailSafe HA Services with the GUI	207
Start FailSafe HA Services with <code>cmgr</code>	207

Stop FailSafe HA Services	208
Stop FailSafe HA Services with the GUI	208
Stopping HA Services on One Node	209
Stopping HA Services on All Nodes in a Cluster	210
Stop FailSafe HA Services with cmgr	210
Set FailSafe HA Parameters	210
Set FailSafe HA Parameters with the GUI	211
Set FailSafe HA Parameters with cmgr	212
Set Log Configuration	213
Set Log Configuration with the GUI	213
Default Log File Names	213
Display Log Group Definitions with the GUI	216
Define Log Groups with cmgr	216
Configure Log Groups with cmgr	217
Modify Log Groups with cmgr	218
Display Log Group Definitions	219
Display Log Group Definitions with cmgr	219
6. Configuration Examples	221
Example: Script to Define an SGI File Server 850 Cluster	221
Example: Script to Define an SGI SAN Server 1000 Cluster	229
Example: Script to Define a Three-Node Cluster	235
Example: Local Failover of HA IP Address	242
Example: Modify a Cluster to Include a CXFS Filesystem	243
Example: Export CXFS Filesystems	244
Example: Create a Resource Group	245
7. FailSafe System Operation	247

Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C Console Support	247
Interrupting FailSafe Commands	248
Two-Node Clusters: Single-Node Use	249
Using a Single Node	249
Resuming Two-Node Use	252
System Status	255
Monitoring System Status with <code>cluster_status</code>	256
Monitoring System Status with the GUI	257
Key to Icons and States	258
Querying Cluster Status with <code>cmgr</code>	260
Monitoring Resource and Reset Serial Line with <code>cmgr</code>	260
Querying Resource Status with <code>cmgr</code>	260
Performing a <code>ping</code> of a System Controller with <code>cmgr</code>	261
Resource Group Status	261
Resource Group State	261
Resource Group Error State	262
Resource Owner	263
Monitoring Resource Group Status with GUI	263
Querying Resource Group Status with <code>cmgr</code>	264
Node Status	264
Monitoring Node Status with <code>cluster_status</code>	264
Monitoring Cluster Status with the GUI	264
Querying Node Status with <code>cmgr</code>	265
Performing a <code>ping</code> of the System Controller with <code>cmgr</code>	265
Viewing System Status with the <code>haStatus</code> Script	265
Embedded Support Partner (ESP) Logging of FailSafe Events	271
Resource Group Failover	273

Bring a Resource Group Online	273
Bring a Resource Group Online with the GUI	273
Bring a Resource Group Online with <code>cmgr</code>	274
Take a Resource Group Offline	274
Take a Resource Group Offline with the GUI	275
Take a Resource Group Offline with <code>cmgr</code>	276
Move a Resource Group	277
Move a Resource Group with the GUI	277
Move a Resource Group with <code>cmgr</code>	277
Suspend and Resume Monitoring of a Resource Group	278
Suspend Monitoring a Resource Group with the GUI	278
Resume Monitoring of a Resource Group with the GUI	279
Putting a Resource Group into Maintenance Mode with <code>cmgr</code>	279
Resume Monitoring of a Resource Group with <code>cmgr</code>	279
Stopping FailSafe	279
Resetting Nodes	279
Reset a Node with the GUI	280
Reset a Node with <code>cmgr</code>	280
Cluster Database Backup and Restore	281
Restoring the Database from Another Node	281
Using <code>build_cmgr_script</code> for the Cluster Database	281
Using <code>cdbBackup</code> and <code>cdbRestore</code> for the Cluster Database and Logging Information	283
Log File Management	285
Rotating All Log Files	285
Rotating Large Log Files	285
Granting Task Execution Privileges to Users	286
Grant Task Access to a User or Users	286

Granting Access to a Few Tasks	287
Granting Access to Most Tasks	289
Revoke Task Access from a User or Users	290
8. Testing the Configuration	291
Overview of FailSafe Diagnostic Commands	291
Performing Diagnostic Tasks with the GUI	292
Test Connectivity with the GUI	292
Test Resources with the GUI	292
Test Failover Policies with the GUI	293
Performing Diagnostic Tasks with cmgr	293
Test the Serial Connections with cmgr	293
Test Network Connectivity with cmgr	294
Test Resources with cmgr	295
Test Logical Volumes with cmgr	296
Test Filesystems with cmgr	297
Test Resource Groups with cmgr	298
Test Failover Policies with cmgr	299
9. System Recovery and Troubleshooting	301
Overview of System Recovery	301
Disabling Resource Groups for Maintenance	302
Ensuring that Resource Groups are Deallocated	303
Checking for Exclusivity	303
Stopping Resources Manually	303
FailSafe Log Files	303
FailSafe Membership and Resets	304
FailSafe Membership and Tie-Breaker Node	305

No Membership Formed	306
Status Monitoring	306
XVM Alternate Path Failover	307
Dynamic Control of FailSafe HA Services	308
Recovery Procedures	309
Single-Node Recovery	310
Cluster Error Recovery	310
Resource Group Recovery	311
Node Error Recovery	311
Resource Group Maintenance and Error Recovery	312
Clear Resource Error State	315
Control Network Failure Recovery	316
Serial Cable Failure Recovery	316
Cluster Database Sync Failure	317
Cluster Database Maintenance and Recovery	317
GUI Will Not Run	318
GUI and cmgr Inconsistencies	319
GUI Does Not Report Information	319
Using the cdbreinit Command	320
Action Script Configuration Errors	320
CXFS Metadata Server Relocation	321
Other Problems with CXFS Coexecution	321
10. Upgrading and Maintaining Active Clusters	323
Adding a Node to an Active Cluster	323
Deleting a Node from an Active Cluster	325
Changing Control Networks in a Cluster	327
Upgrading OS Software in an Active Cluster	329

Upgrading FailSafe Software in an Active Cluster	330
Adding New Resource Groups or Resources in an Active Cluster	331
Adding a New Hardware Device in an Active Cluster	332
11. Performance Co-Pilot for FailSafe	333
Using the Visualization Tools	333
PCP for FailSafe Performance Metrics	337
PCP Gray Display	337
Appendix A. Software Overview	339
Software Layers	339
Interface Agent Daemon (IFD)	343
Communication Paths	343
Communication Paths in a Coexecution Cluster	348
Execution of FailSafe Action and Failover Scripts	349
When a start Script Fails	353
When a stop Script Fails	353
Components	353
Appendix B. Memberships and Quorums	355
Membership Types	356
Cluster Database Membership and Quorum	357
FailSafe Membership, Quorum, and Tiebreaker	358
CXFS Kernel Membership, Quorum, and Tiebreaker	359
Cluster Database Membership Logs	361
Quorum and Tiebreaker Examples	366
Coexecution Example	366
FailSafe Tiebreaker Node Example	368

Cases when the FailSafe Tiebreaker is Not in the Membership	370
Heartbeat Considerations	373
Appendix C. Updating from IRIS FailSafe 1.2 to IRIS FailSafe 2.1.x	375
Hardware Changes	375
Software Changes	376
Configuration Changes	376
Scripts	377
Operational Comparison	378
Upgrade Examples	379
Defining a Node	380
Defining a Cluster	381
Setting HA Parameters	382
Defining a Resource: XLV Volume	384
Defining a Resource: XFS Filesystem	385
Defining a Resource: IP Address	385
Additional FailSafe 2.1.x Tasks	386
Status	387
Appendix D. FailSafe Software	389
Subsystems on the CD	389
Subsystems for Servers and Workstations in the Pool	391
Additional Subsystems for Nodes in the FailSafe Cluster	392
Additional Subsystems for Workstations	392
Subsystems for IRIX Workstations	393
Subsystems for Other Operating System Workstations	393
Appendix E. Metrics Exported by PCP for FailSafe	395

Appendix F. System Messages	403
SYSLOG Messages	403
Normal Messages After Successfully Starting HA Services	405
cli Error Messages	405
crsd Error Messages	406
cmond Error Messages	407
fs2d Error Messages	408
ha_srmd Error Message	409
Log File Error Messages	409
cad Messages	410
cli Messages	412
crsd Errors	413
fs2d Errors	414
Glossary	415
Index	427

Figures

Figure 1-1	Pool and Cluster Concepts	4
Figure 1-2	FailSafe Membership	6
Figure 1-3	Resource Type Dependencies	10
Figure 1-4	Sample System Components	14
Figure 1-5	Configuration Types	17
Figure 1-6	Reset Types	18
Figure 1-7	Disk Storage Failover in a Two-Node Cluster	24
Figure 2-1	Example Configuration with Four Resource Groups	30
Figure 2-2	Non-Shared Disk Configuration and Failover	32
Figure 2-3	Shared Disk Configuration for Active/Backup Use	34
Figure 2-4	Shared Disk Configuration for Dual-Active Use	35
Figure 2-5	Example Logical Volume Configuration	38
Figure 2-6	Filesystems and Logical Volumes	42
Figure 2-7	Contents of a Coexecution Cluster Database	46
Figure 3-1	Example Interface Configuration	69
Figure 4-1	GUI Showing Details for a Resource	95
Figure 4-2	Pop-up Menu that Appears After Clicking the Right Mouse Button	96
Figure 5-1	Dependencies	166
Figure 5-2	Example of Resource Dependency	185
Figure 5-3	Mutual Dependency of Resources Is Not Allowed	186
Figure 6-1	SGI File Server 850 Configuration	222
Figure 6-2	Resource Dependencies for SGI File Server 850	224
Figure 6-3	SGI SAN Server Configuration	229

Figure 6-4	Resource Dependencies for SGI SAN Server	231
Figure 6-5	FailSafe Configuration Example	236
Figure 7-1	Results of Granting a User Privilege	288
Figure 7-2	Displaying the Privileged Commands a User May Execute	289
Figure 11-1	Heartbeat Response Statistics	334
Figure 11-2	Resource Monitoring Statistics	335
Figure A-1	Software Layers	341
Figure A-2	Administrative Communication within One Node	344
Figure A-3	Daemon Communication within One Node	345
Figure A-4	Communication between Nodes in the Pool	346
Figure A-5	Communication for a Node Not in the Cluster	347
Figure A-6	Administrative Communication within One Node under Coexecution	348
Figure A-7	Daemon Communication within One Node under Coexecution	349
Figure A-8	Message Paths for Action Scripts and Failover Policy Scripts	352
Figure B-1	One Node is Out of Date: Most Recent Log is Replicated	363
Figure B-2	Unequally Sized Pools are Joined: Larger Pool Log is Replicated	364
Figure B-3	Equally Sized Pools are Joined: Lowest Node ID Log is Replicated	366
Figure B-4	Example Memberships in a Coexecution Cluster	368
Figure B-5	FailSafe Tiebreaker Node	370
Figure B-6	Forming a Membership without the FailSafe Tiebreaker	372

Tables

Table 1-1	Example webgroup Resource Group	8
Table 2-1	XLV Logical Volume Resource Attributes	39
Table 2-2	Filesystem Configuration Parameters	41
Table 2-3	HA IP Address Configuration Parameters	45
Table 2-4	Local XVM Volume Resource Attributes	52
Table 3-1	fs2d.options File Options	63
Table 3-2	PCP for FailSafe Collector Subsystems	79
Table 3-3	PCP for FailSafe Monitor Subsystems	81
Table 4-1	GUI Platforms	89
Table 4-2	Template Scripts for cmgr	107
Table 5-1	System Controller Types	129
Table 5-2	Resource Type Attributes	182
Table 5-3	Failover Attributes	192
Table 5-4	Log Levels	215
Table 5-5	Default Log File Names	215
Table 6-1	SGI File Server 850 Resources and Failover Policies for RG1 and RG2	223
Table 6-2	SGI SAN Server Resources and Failover Policies for RG1 and RG2	230
Table 6-3	Resources and Failover Policies for RG1 and RG2	236
Table 7-1	Key to Icons	258
Table 7-2	Key to States	259
Table 8-1	FailSafe Diagnostic Test Summary	291
Table 9-1	Message Levels	303
Table A-1	Provided Plug-Ins	339

Table A-2	Optional Plug-Ins	340
Table A-3	Contents of /usr/cluster/bin	342
Table A-4	Contents of the /var/cluster/ha directory	354
Table C-1	Differences between IRIS FailSafe 1.2 and IRIS FailSafe 2.1.x	378
Table D-1	FailSafe CD	390
Table D-2	Subsystems Required for Nodes in the Pool (Servers and GUI Client(s))	391
Table D-3	Additional Subsystems Required for Nodes in the Cluster	392
Table D-4	Subsystems Required for IRIX Workstations	393
Table E-1	PCP Metrics	395
Table F-1	SYSLOG Error Message Format	404
Table F-2	Log File Error Message Format	410

About This Guide

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

This guide was prepared in conjunction with IRIS FailSafe 2.1.5, which supports IRIX 6.5.20 and later.

Audience

This guide is written for the person who administers the FailSafe system. The FailSafe administrator must be familiar with the operation of Origin servers, as well as optional Origin Vault, Fibre Channel RAID, JBOD, SGI TP9100, or SGI TP9400 storage systems, whichever is used in the 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

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

- Chapter 1, "Overview", introduces the components of the 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` 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, "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 C, "Updating from IRIS FailSafe 1.2 to IRIS FailSafe 2.1.x", describes the upgrade procedure.
- Appendix D, "FailSafe Software", summarizes the systems to install on each component of a cluster.
- Appendix E, "Metrics Exported by PCP for FailSafe", lists the metrics implemented by `pmdafsafe`.

Related Documentation

The following documentation will be useful in a FailSafe environment:

- *IRIS FailSafe Version 2 Programmer's Guide*
- *Performance Co-Pilot for IRIX Advanced 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*
- *IRIX FailSafe 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*

Personal System Administration Guide

The 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 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 for DMF
failsafe2_informix	IRIS FailSafe for INFORMIX
failsafe2_nfs	IRIS FailSafe for NFS
failsafe2_oracle	IRIS FailSafe for Oracle
failsafe2_samba	IRIS FailSafe for Samba
failsafe2_tmf	IRIS FailSafe for TMF

Obtaining Publications

You can obtain SGI documentation in the following ways:

- See the SGI Technical Publications Library at <http://docs.sgi.com>. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.
- If it is installed on your SGI system, you can use InfoSearch, an online tool that provides a more limited set of online books, release notes, and man pages. With an IRIX system, select **Help** from the Toolchest, and then select **InfoSearch**. Or you can type `infosearch` on a command line.

- You can also view release notes by typing either `grelnotes` or `relnotes` on a command line.
- You can also view man pages by typing `man title` on a command line.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
<code>command</code>	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
<code>manpage(x)</code>	Man page section identifiers appear in parentheses after man page names. (1) indicates a user command, (1M) and (8) indicate an administrator command
<i>variable</i>	Italic typeface denotes variable entries and words or concepts being defined.
GUI	This font denotes the names of graphical user interface (GUI) elements such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, fields, and lists.
user input	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
[]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this publication, contact SGI. Be sure to include the title and document number of the publication with your comments. (Online, the document number is located in the

front matter of the publication. In printed publications, the document number is located at the bottom of each page.)

You can contact SGI 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 Web page:

`http://docs.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 Parkway, M/S 535
Mountain View, California 94043-1351

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

SGI values your comments and will respond to them promptly.

Overview

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

- "High Availability and FailSafe"
- "Cluster Environment", page 2
- "Additional Features", page 19
- "Administration", page 20
- "Highly Available Resources", page 20
- "Highly Available Applications", page 24
- "Failover and Recovery Processes", page 25
- "Overview of Configuring and Testing a New Cluster", page 26

Also see Appendix A, "Software Overview", page 339

High Availability and FailSafe

FailSafe provides a general facility for providing *highly available services*. This type of system survives a single point of failure by using redundant components and FailSafe software to provide highly available services for a cluster that contains multiple nodes.

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 systems 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. 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"
- "Hardware Components", page 14
- "Disk Connections", page 16
- "Supported Configurations", page 16

Terminology

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

Cluster

A *cluster* is the set of systems (nodes) configured to work together as a single computing resource. A cluster is identified by a simple name and a cluster ID. There is only one cluster that may be formed from a given pool of nodes.

Disks or logical units (LUNs) are assigned to clusters by recording the name of the cluster on the disk or LUN. Thus, if any disk is accessible (via a Fibre Channel connection) from machines in multiple clusters, then those clusters must have unique names. When members of a cluster send messages to each other, they identify their cluster via the cluster ID. Thus, if two clusters will be sharing the same network for communications, then they must have unique cluster IDs.

Because of the above restrictions on cluster names and cluster IDs, and because cluster names and cluster IDs cannot be changed once the cluster is created (without deleting the cluster and recreating it), SGI advises that you choose unique names and cluster IDs for each of the clusters within your organization.

Node

A *node* is an operating system (OS) image, usually an individual computer. (This use of the term *node* does not have the same meaning as a node in an SGI Origin 3000 or SGI 2000 system.)

A given node can be a member of only one pool and therefore only one cluster.

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.)

A pool is formed when you connect to a given node and define that node in the cluster database using the FailSafe GUI or `cmgr` command. You can then add other nodes to the pool by defining them while still connected to the first node, or to any other node that is already in the pool. (If you were to connect to another node and then define it, you would be creating a second pool).

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

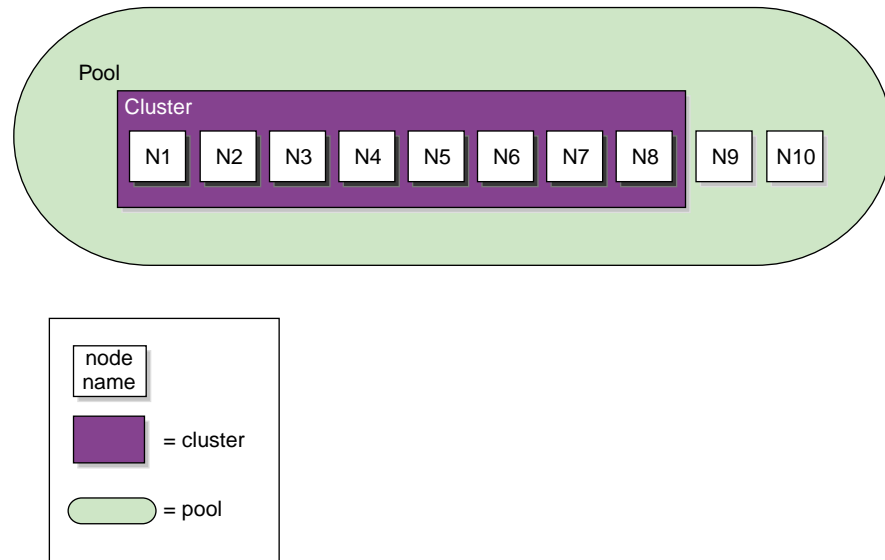


Figure 1-1 Pool and Cluster Concepts

Cluster Database

The *cluster database* contains configuration information about resources, resource groups, failover policy, nodes, clusters, logging information, and configuration parameters.

The database consists of a collection of files; you can view and modify the contents of the database by using the FailSafe Manager GUI and the `cmgr`, `cluster_status`, and `clconf_info` commands.

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.
- *Cluster database membership* (also known as *fs2d membership*) is a group of nodes in the **pool** where the cluster database is replicated. The `fs2d` daemon is the cluster database daemon that maintains membership of nodes where the database is replicated and keeps the database synchronized across database transactions.

With CXFS coexecution, there is also *CXFS membership*. For more information about CXFS, see "Coexecution of CXFS and FailSafe", page 45, 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) to form and maintain a cluster.
- For cluster 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 cluster database 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 cluster database membership consists of N1 through N3, three of six nodes in the pool (50%).
- The FailSafe membership also consists of nodes N1 through N3, three of four nodes where HA services were started.

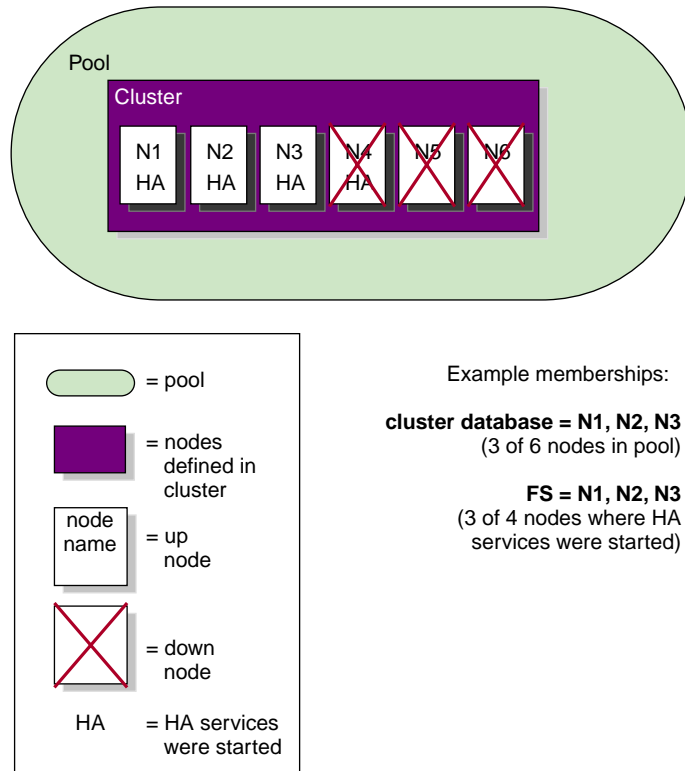


Figure 1-2 FailSafe Membership

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 tiebreaker node will attempt to reset a node in the other set in order to maintain a quorum. For more information, see Appendix B, "Memberships and Quorums", page 355.

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 the 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 1-3.

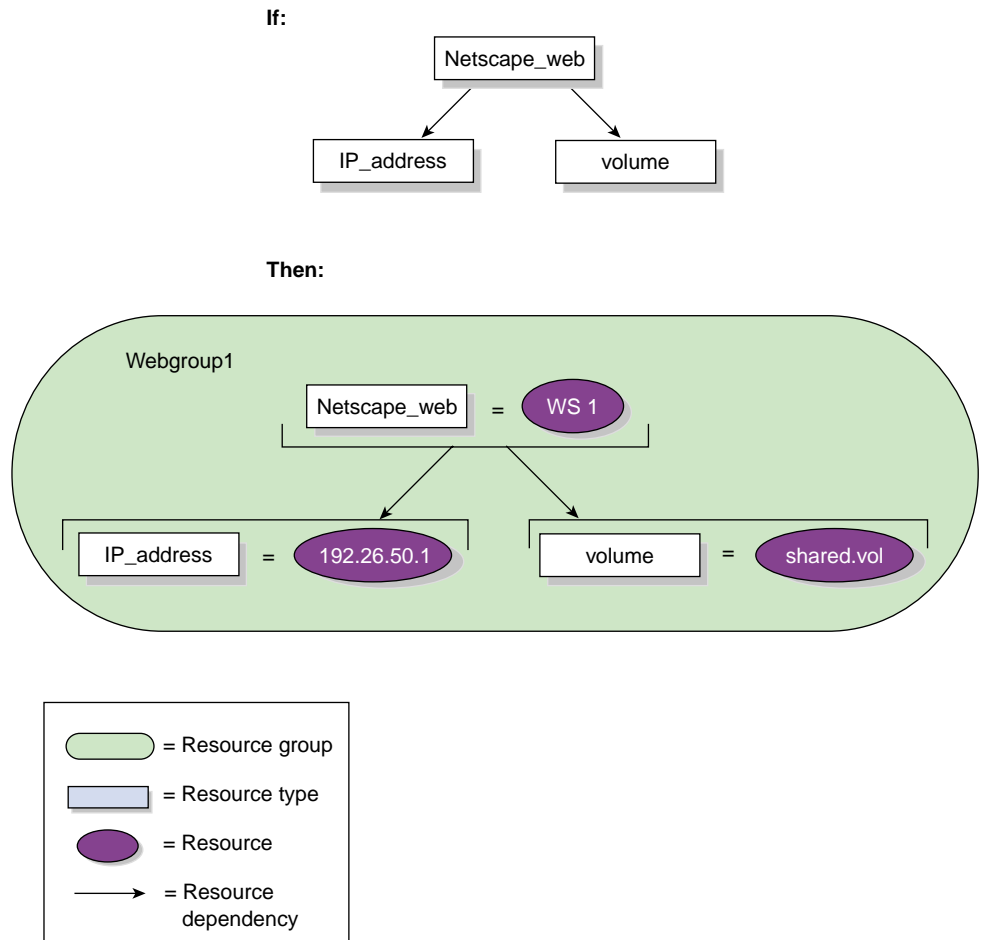


Figure 1-3 Resource Type Dependencies

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 a UNIX process group and a 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 339.

Hardware Components

Figure 1-4 shows an example of FailSafe hardware components, in this case for a two-node system.

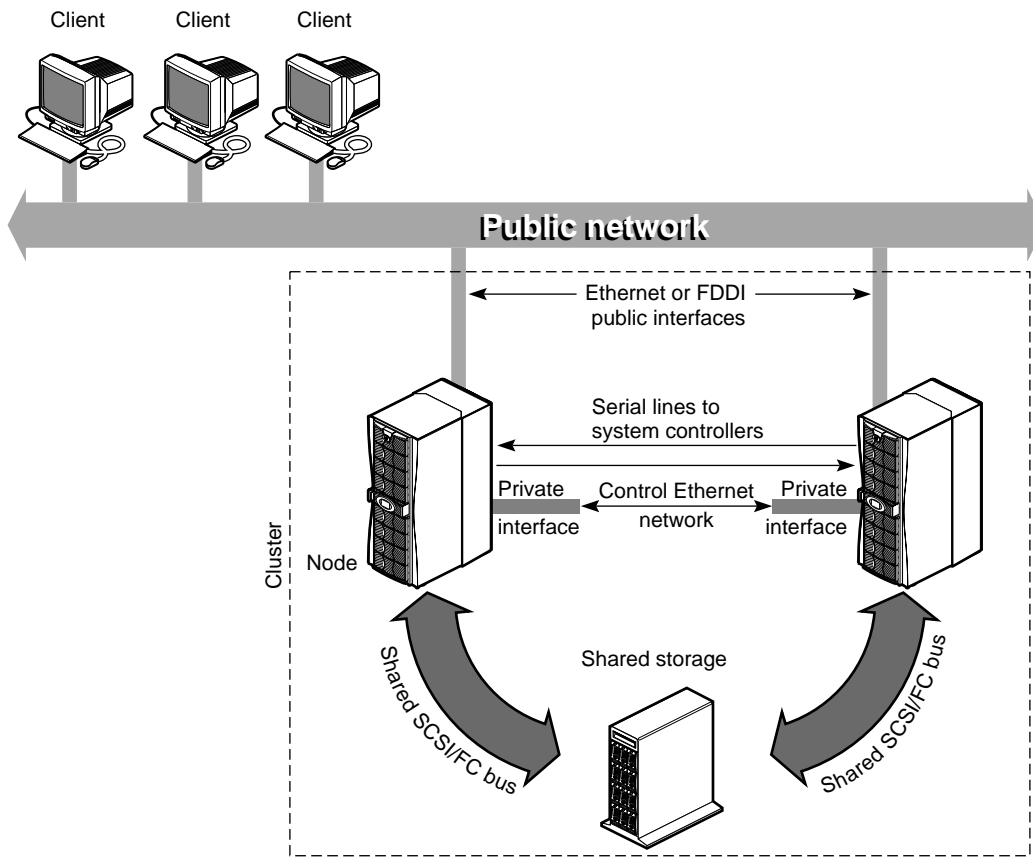


Figure 1-4 Sample System Components

The hardware components are as follows:

- Up to eight nodes: Origin 300 and Origin 3000 series, Onyx 300 and Onyx 3000 series, Origin 200, Onyx2 deskside, SGI 2000 series, and Onyx2.
- 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 multi-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
- TP9500

For an exact list of storage supported, please contact SGI support.

- An Etherlite network-based serial multiplexer (EL-16) (FAILSAFE-N_NODE) hardware component to reset machines in a cluster.

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

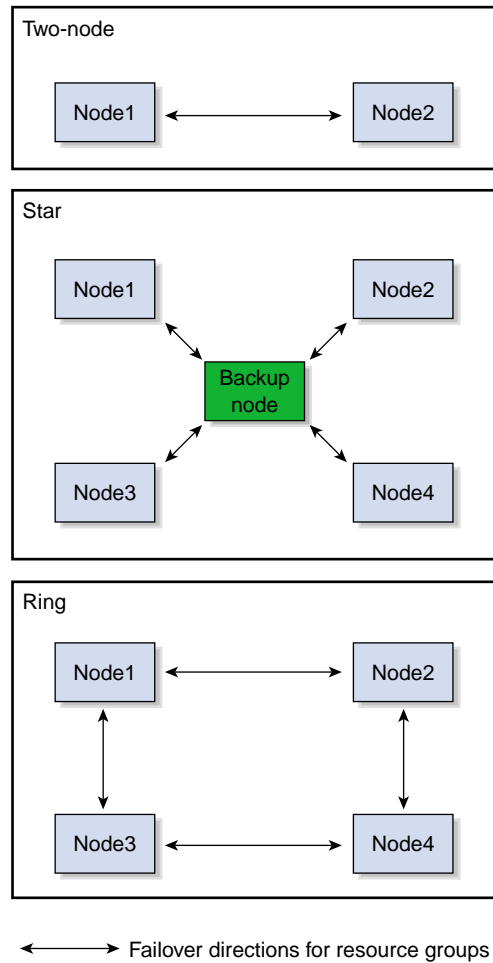


Figure 1-5 Configuration Types

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.

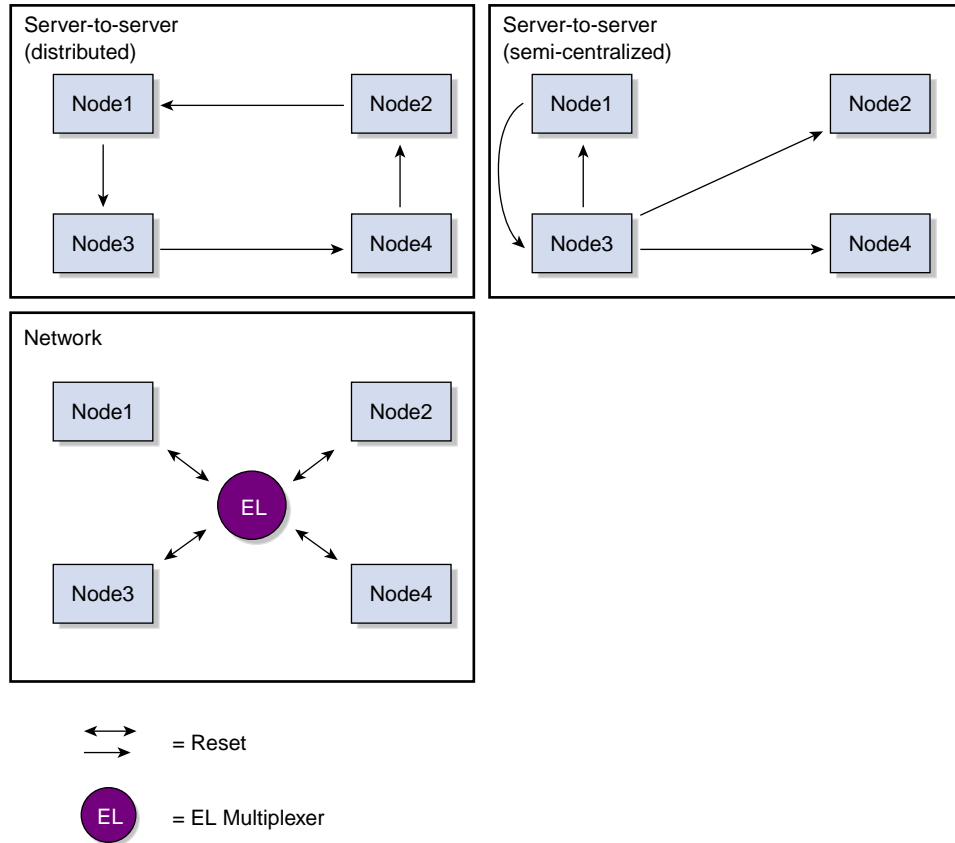


Figure 1-6 Reset Types

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` 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 109, and Chapter 7, "FailSafe System Operation", page 247.

Highly Available Resources

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

- Nodes
- Network interfaces and IP addresses
- Disks

Nodes

FailSafe detects if a node crashes or hangs (for example, due to a parity error or bus error). A different node, determined by the failover policy, resets the failed node and takes over the failed node's services.

If a node fails, its interfaces, access to storage, and services also become unavailable. See the following 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, FailSafe moves them to another network interface in the other node. 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 42, 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

FailSafe 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 it cannot be used with CXFS.

The storage components should not have a single point of failure. All data should be in a 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 1-7 shows an example of disk storage failover in a two-node cluster.

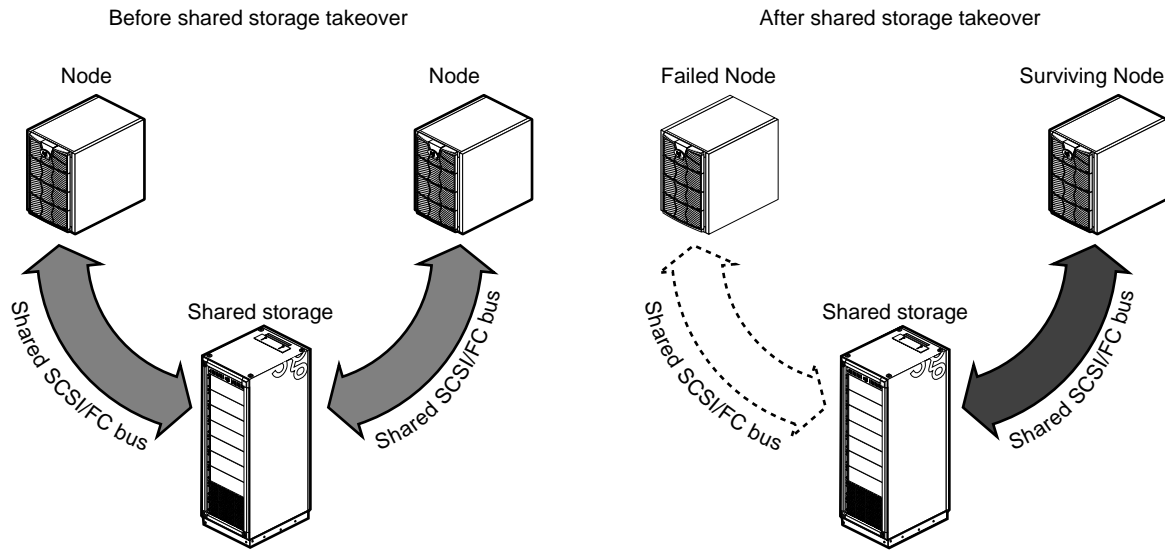


Figure 1-7 Disk Storage Failover in a Two-Node Cluster

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 a *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` command. Information on configuring the system is provided in Chapter 5, "Configuration", page 109. 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 hardware rest 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 or not the node automatically starts to provide highly available services again depends on the failover policy that you define.

For more information, see "Define a Failover Policy with the GUI", page 190.

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 349.

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", page 27.
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", page 55.
4. Define the configuration as explained in Chapter 5, "Configuration", page 109.
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.

See Chapter 8, "Testing the Configuration", page 291.

Configuration Planning

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

- "Overview"
- "Disk Configuration", page 30
- "XLV Logical Volume Configuration", page 36
- "XFS Filesystem Configuration", page 39
- "HA IP Address Configuration", page 42
- "Coexecution of CXFS and FailSafe", page 45
- "Local XVM Volumes", page 51

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 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 339. 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 or Fibre Channel disk storage that have plexed XLV logical volumes on them? Shared disks must be part of a RAID storage system or in a SCSI or 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 FailSafe", page 45.

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? 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. For more information about failover domains, see "Failover Domain", page 11.
- How many 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 the nodes, so that 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` network interfaces in each node are connected to each other to form a private network.

Figure 2-1 depicts this configuration.

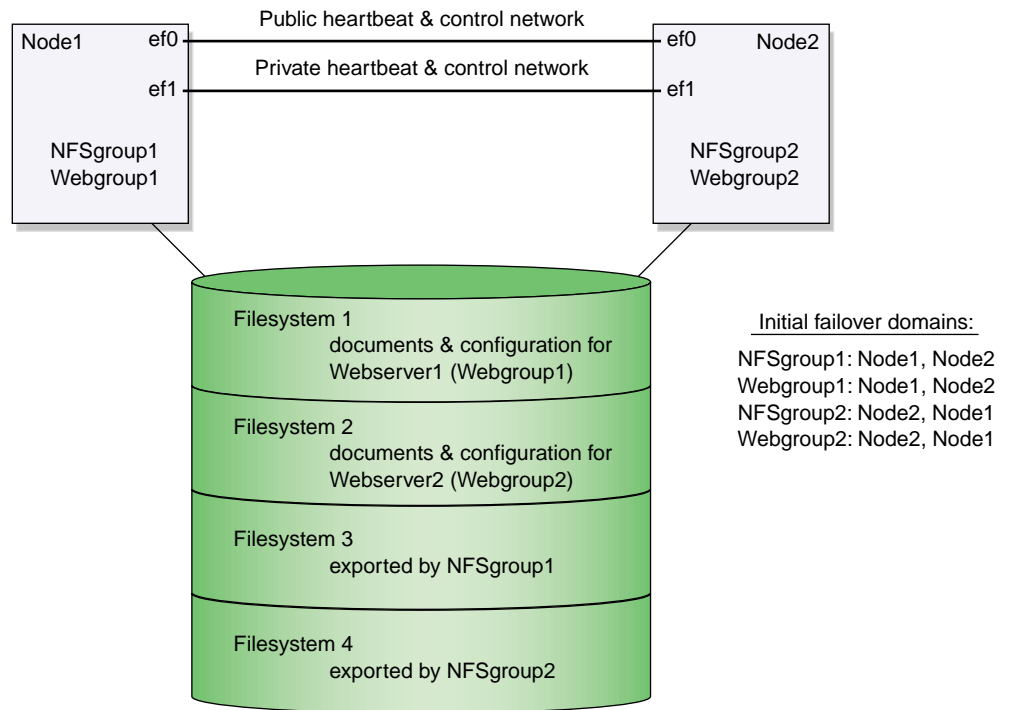


Figure 2-1 Example Configuration with Four Resource Groups

Disk Configuration

This section contains the following:

- "Planning Disk Configuration"
- "Configuration Parameters for Disks", page 36

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 (such as `/tmp`, `/var`, `/usr`, `/bin`, and `/dev`) should be in a 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 either 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, 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 an HA IP address (alias). When a failover occurs, HA IP addresses 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 address 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

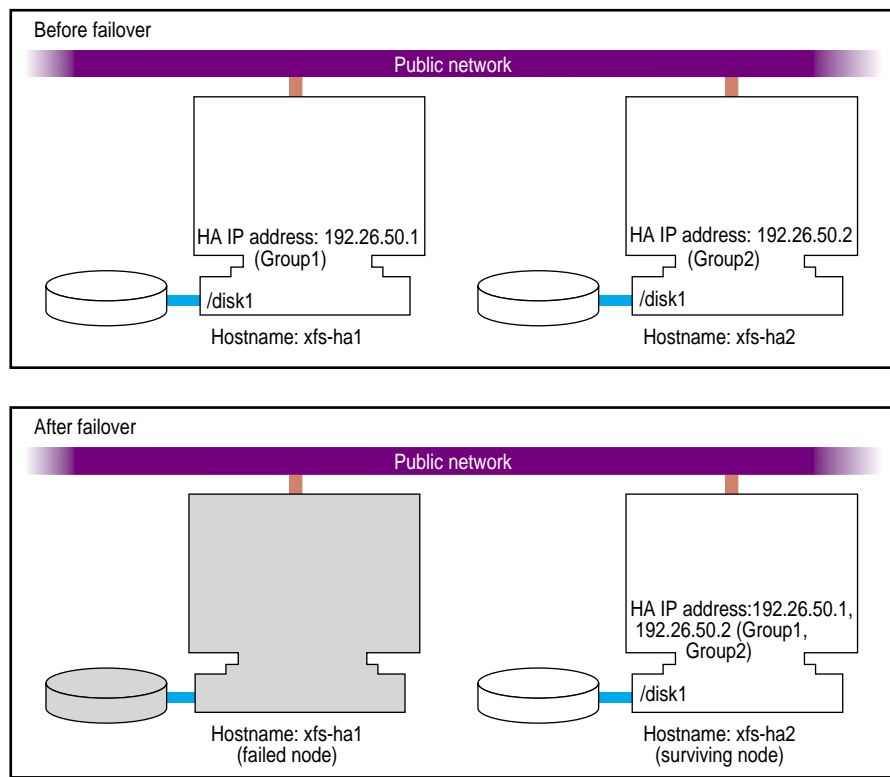


Figure 2-2 Non-Shared Disk Configuration and Failover

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.

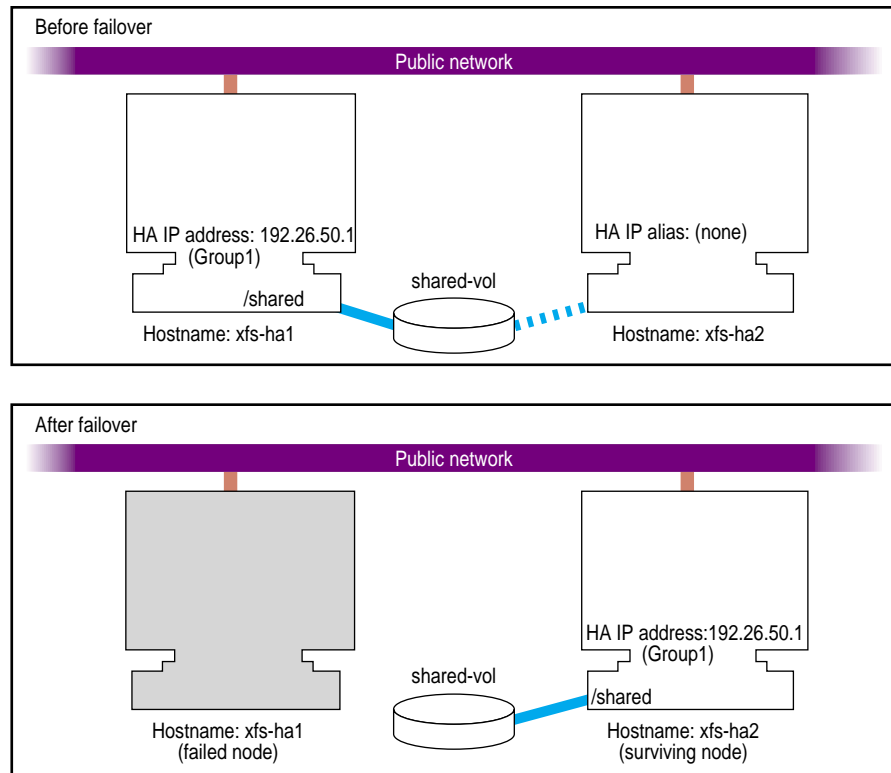


Figure 2-3 Shared Disk Configuration for Active/Backup Use

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.

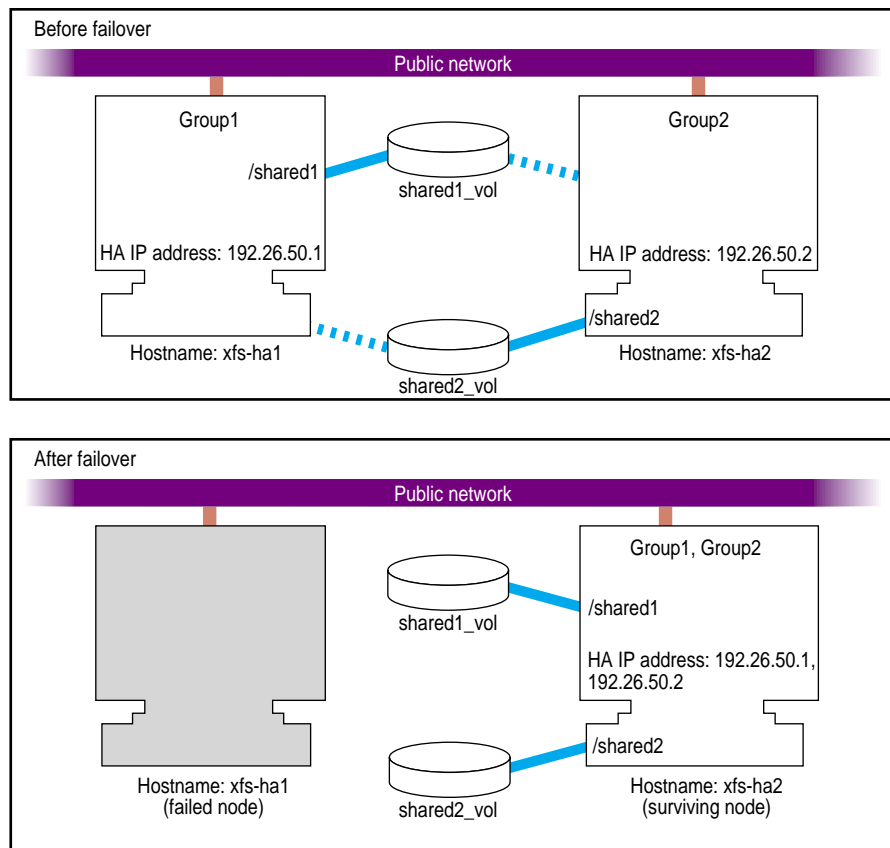


Figure 2-4 Shared Disk Configuration for Dual-Active Use

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 "Resource Attributes for Logical Volumes", page 39.

For information on using CXFS filesystems (which use XVM logical volumes) in a FailSafe configuration, see "Coexecution of CXFS and FailSafe", page 45.

XLV Logical Volume Configuration

Note: This section describes logical volume configuration using XLV logical volumes. See also "Coexecution of CXFS and FailSafe", page 45, and "Local XVM Volumes", page 51.

This section contains the following:

- "Planning XLV Logical Volumes"
- "Example Logical Volume Configuration", page 38
- "Resource Attributes for Logical Volumes", page 39

Planning XLV Logical Volumes

All shared disks with XFS filesystems 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 of 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

where:

- *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 name must have owner, group, and mode values that are different from the defaults (`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) 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.

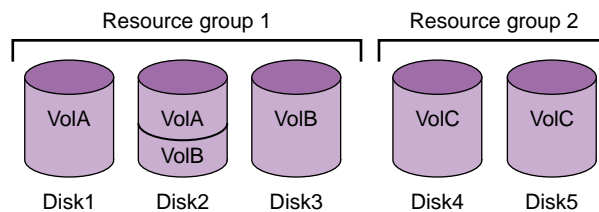


Figure 2-5 Example Logical Volume Configuration

Resource Attributes for Logical Volumes

Table 2-1 lists the resource attributes for XLV logical volumes.

Table 2-1 XLV Logical Volume Resource Attributes

Resource Attribute	Default	Description
devname-owner	root	Owner of the XLV device name
devname-group	sys	Group of the XLV device name
devname-mode	0600	File permission mode of the XLV device name

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

XFS 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 FailSafe", page 45.

Planning XFS 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 FailSafe filesystems on each node, ensure 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, verify 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 verifying 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 CXFS, which provides this functionality. For more information about NFS, see *IRIX FailSafe 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 XLV Filesystem Configuration

Continuing with the scenario from in "Example Logical Volume Configuration", page 38, 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 <code>/etc/mtab</code> file 2 – checks if the filesystem is mounted using the <code>stat</code> command
volume-name	VolA	VolB	VolC	The label of the logical volume on which the filesystem was created
mode	<code>rw, noauto</code>	<code>rw, noauto, wsync</code>	<code>rw, noauto</code>	The modes of the filesystem (identical to the modes specified in <code>/etc/fstab</code>)

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`)

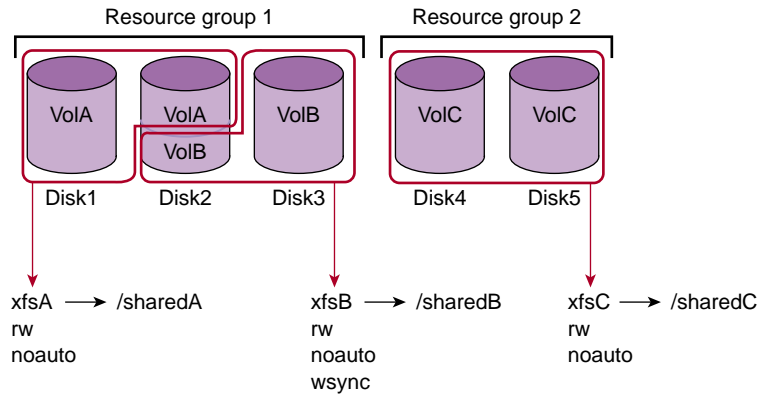


Figure 2-6 Filesystems and Logical Volumes

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 45
- "Local Failover of HA IP Addresses", page 45

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-xfp-ha1` and `priv-xfp-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. 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`. For more information about re-MACing, see "Network Interfaces and IP Addresses", page 21.

1. Configure an HA IP address on one of the interfaces of `node1`. For example:

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

interface is the interface to be used to 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` command:

```
# ping -c 2 ip_address
```

For example, if the value for *ip_address* is `190.0.2.1`:

```
# ping -c 2 190.0.2.1
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` 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 example 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	0xffffffff00	0xffffffff00
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 242.

Coexecution of CXFS and FailSafe

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

The cluster database contains configuration information about nodes, clusters, logging information, and configuration parameters. If you are running CXFS, it also contains information about CXFS filesystems and CXFS metadata servers. If you are running FailSafe, it also contains information about resources, resource groups, and failover policies. A cluster running both CXFS and FailSafe is known as a coexecution cluster. Figure 2-7 depicts the contents of a coexecution cluster database.

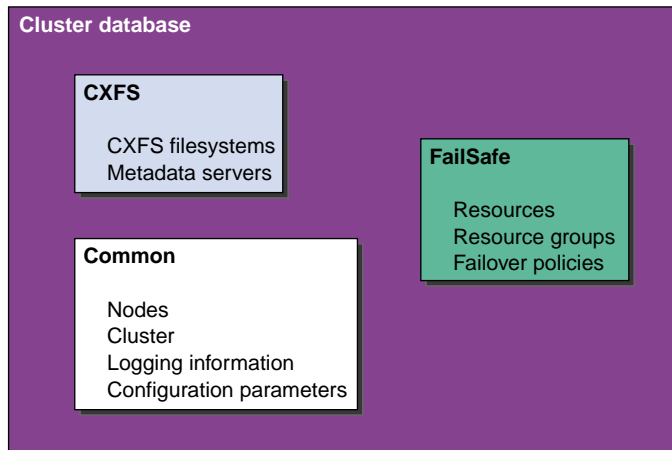


Figure 2-7 Contents of a Coexecution Cluster Database

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.

Note: In IRIX 6.5.18f, CXFS provides a system tunable parameter (`cxfs_relocation_ok`) to allow users to disable CXFS metadata server relocation. CXFS filesystem relocation is disabled by default. In FailSafe/CXFS coexecution clusters, it is recommended that the CXFS filesystem relocation is disabled using the tunable parameter.

The system tunable parameter is different from the `relocate-mds` attribute of a CXFS resource. FailSafe uses a different procedure to initiate metadata server relocation for a filesystem that is not impacted by the value of the `cxfs_relocation_ok` system tunable parameter.

This section contains the following:

- "Size of the Coexecution Cluster", page 47
- "Cluster Type", page 47

- "Node Types for CXFS Metadata Servers", page 48
- "CXFS Metadata Servers and Failover Domain", page 48
- "CXFS Resource Type for FailSafe", page 48
- "Separate CXFS and FailSafe GUIs", page 50
- "Conversion Between CXFS and FailSafe", page 50
- "Network Interfaces", page 51

See also "Communication Paths in a Coexecution Cluster", page 348.

Size of the Coexecution Cluster

A coexecution cluster is supported with as many as 32 nodes. All of these nodes must run CXFS and up to eight can also run FailSafe. As many as 16 of the nodes can be CXFS administration nodes and all other nodes can be client-only nodes. FailSafe must be run on a CXFS administration node; FailSafe cannot run on a client-only node.

You should configure a production cluster with an odd number of CXFS metadata server-capable administration nodes. (A cluster with serial hardware reset cables and only two server-capable nodes is supported, but there are inherent issues with this configuration; see the *CXFS Version 2 Software Installation and Administration Guide*.)

Even when you are running CXFS and FailSafe, there is still only one pool, one cluster, and one cluster configuration.

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, the set of 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. Resources of type `CXFS` must be added to the resource group that contain the resources that depend on a `CXFS` filesystem. The name of the 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` graphical user interface (GUI) or `cmgr` 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 filesystem's potential metadata server 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` command.

For example, following are the commands used to create resources named `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
```

2: Configuration Planning

```
=====
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` command but separate 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`, 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.

Local XVM Volumes

FailSafe also supports local XVM; you cannot use local XVM with CXFS XVM.

XVM Resource Type for FailSafe

FailSafe provides the `XVM` resource type, which can be used to fail over applications that use local XVM volumes without CXFS. (Do not use the `XVM` resource type with the `CXFS` resource type.)

For each local XVM resource, the name of the resource is the name of the XVM volume without the preceding `vol/` characters. The resource name must be unique for all XVM domains in the FailSafe cluster.

Resource Attributes for Local XVM Volumes

Table 2-4 provides the XVM resource attributes. There are no resource type dependencies for the XVM resource type.

Table 2-4 Local XVM Volume Resource Attributes

Resource Attribute	Default	Description
<code>devname_owner</code>	<code>root</code>	Owner of the XVM device name. Databases that use XVM volumes often require that the owner be changed to the database user. This attribute is optional.
<code>devname_group</code>	<code>sys</code>	Group of the XVM device name. Databases that use XVM volumes request the device group to be the database user group. This attribute is optional.
<code>devname_mode</code>	<code>0600</code>	File permission mode of the XVM device file. This attribute is optional.
<code>fs_xvm_owner</code>	<code>fake_owner</code>	FailSafe XVM temporary owner. The value can be any string other than the name of cluster or the hostname of any of the nodes in the FailSafe cluster.
<code>physvol_names</code>	(no default)	The list of physical volumes that make up the XVM volume, separated by commas. You can obtain the list of physical volumes by using the <code>xvm show</code> command.

Example Local XVM Volume Configuration

Following is an example local XVM volume configuration:

```
cmgr> show resource lun1 of resource_type XVM in cluster lab-cluster
```

```
fs_xvm_owner: fake_owner  
devname_group: sys  
physvol_names: lun1,lun2,lun3  
devname_mode: 600  
devname_owner: root
```


Installation and System Preparation

Note: The procedures in this chapter assume that you have done the work described in Chapter 2, "Configuration Planning", page 27.

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

- "Install Software"
- "Configure System Files", page 59
- "Set the `corepluspid` System Parameter", page 66
- "Set NVRAM Variables", page 67
- "Create XLV Logical Volumes and XFS Filesystems", page 67
- "Configure Network Interfaces", page 68
- "Configure the Serial Ports for a Ring Reset", page 73
- "Install Patches", page 74
- "Install Performance Co-Pilot (PCP) Software", page 78
- "Test the System", page 82

Install Software

Installing the 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:

```
irix# relnotes failsafe2 [chapter_number]
```

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

```
irix# 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 the order shown:

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

When `sysadm_base` is installed, `tcpmux` service is 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 the order shown:

```
java2_eoe.sw
java2_eoe.sw32
sysadm_base.sw.client
sysadm_cluster.sw.client
sysadm_failsafe2.sw.client
sysadm_failsafe2.sw.web
```



Caution: The GUI only operates with Java2 v1.4.1 Execution Environment (Sun JRE v1.4.1). This is the version of Java that is provided with the IRIX 6.5.x release.

The SGI website also contains Java 1. However, you cannot use this version of Java with the GUI. Using a Java version other than 1.4.1 will cause the GUI to fail.

5. On each node, install application software and appropriate optional FailSafe plugin software. For example, for NFS install the following:

```
nfs.sw.nfs (If necessary; from IRIX, might already be present)
failsafe2_nfs.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
java2_eoe.sw
java2_eoe.sw32
```

If the administrative workstation is an IRIX machine that launches the GUI client from a web browser that supports Java, install the 1.4.1 `java_plugin` from the IRIX applications 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.)

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 IRIX node in the `/var/flexlm/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 `xlvmgr` command:

```
irix# xlvmgr
xlvmgr> show config
```

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

```
Plexing license: present
```

- c. Quit `xlvmgr`.
9. Install recommended patches for FailSafe.

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

10. 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 D, "FailSafe Software", page 389, 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 61
- "`/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:

- The hostname must be configured on a network interface connected to the public network and should be resolved using `/etc/hosts`.
- Hostnames cannot begin with an underscore (`_`) or include any whitespace characters.
- The value of the `/etc/sys_id` file must match the node's 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 *hostname* can be the simple hostname or the fully qualified domain name:

IP_address hostname

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`.

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` 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` 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.
- The `/etc/sys_id` file contains the hostname of the machine and should not be associated with an HA IP 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      # FailSafe Membership Daemon
sgi-gcd       8000/udp      # Group Communication Daemon
```

`/etc/config/cad.options`

The `/etc/config/cad.options` file contains the list of parameters that the `cad` 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` 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:

```
irix# /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` 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
<code>-logevents <i>event name</i></code>	Log selected events. The following event names may be used: <code>all</code> , <code>internal</code> , <code>args</code> , <code>attach</code> , <code>chandle</code> , <code>node</code> , <code>tree</code> , <code>lock</code> , <code>datacon</code> , <code>trap</code> , <code>notify</code> , <code>access</code> , <code>storage</code> . The default is <code>all</code> .
<code>-logdest <i>log destination</i></code>	Set log destination. The following log destinations may be used: <code>all</code> , <code>stdout</code> , <code>stderr</code> , <code>syslog</code> , <code>logfile</code> . If multiple destinations are specified, the log messages are written to all of them. If <code>logfile</code> is specified, it has no effect unless the <code>-logfile</code> option is also specified. The default is <code>logfile</code> .
<code>-logfile <i>filename</i></code>	Set log filename. The default is <code>/var/cluster/ha/log/fs2d_log</code> .
<code>-logfilemax <i>maximum size</i></code>	Set log file maximum size (in bytes). If the file exceeds the maximum size, any preexisting <code>filename.old</code> will be deleted, the current file will be renamed to <code>filename.old</code> , and a new file will be created. A single message will not be split across files. If <code>-logfile</code> is set, the default is 1000000.

Option	Description
-loglevel <i>loglevel</i>	Set log level. The following log levels may be used: <i>always</i> , <i>critical</i> , <i>error</i> , <i>warning</i> , <i>info</i> , <i>moreinfo</i> , <i>freq</i> , <i>morefreq</i> , <i>trace</i> , <i>busy</i> . The default is <i>info</i> .
-trace <i>trace_class</i>	Trace selected events. The following trace classes may be used: <i>all</i> , <i>rpcs</i> , <i>updates</i> , <i>transactions</i> , <i>monitor</i> . If you specify this option, you must also specify <i>-tracefile</i> and/or <i>-tracelog</i> . No tracing is done, even if it is requested for one or more classes of events, unless either or both of <i>-tracefile</i> or <i>-tracelog</i> is specified. The default is <i>transactions</i> .
-tracefile <i>filename</i>	Set trace filename. There is no default.
-tracefilemax <i>maximum_size</i>	Set trace file maximum size (in bytes). If the file exceeds the maximum size, any preexisting <i>filename.old</i> will be deleted, the current file will be renamed to <i>filename.old</i> , 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 <i>-tracelog</i> .
-[no]parent_timer	[Do not] exit when parent exits. The default is <i>-noparent_timer</i> .
-[no]daemonize	[Do not] run as a daemon. The default is <i>-daemonize</i> .
-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 initial configuration, 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:

```
irix# /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` 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` 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:

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

```
irix# nvrmap AutoLoad
Y
irix# nvrmap scsihostid
0
```

To set these variables, use the following commands:

```
irix# nvrmap AutoLoad yes
irix# nvrmap 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 FailSafe", page 45. 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 36). 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` and `chmod` 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.

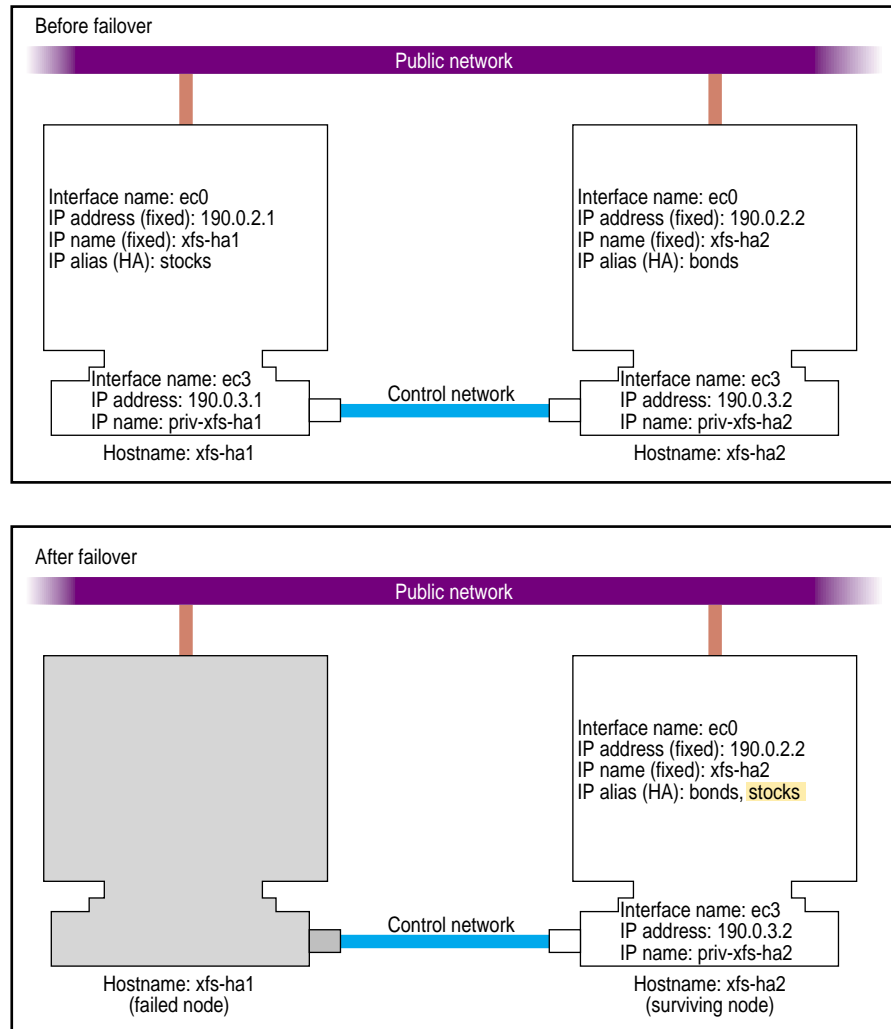


Figure 3-1 Example Interface Configuration

1. If possible, add every IP address, IP name, and highly available (HA) IP address (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: HA IP addresses 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 HA IP addresses 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:

```
irix# 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 HA IP addresses 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 the following command for each address:

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

`address` is an IP address, IP name, or HA IP address. If `ypmatch` 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 is as follows:

```
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 FailSafe 2.x is turned off on each node, using the `chkconfig` command:

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

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

```
irix# 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:

```
irix# 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 `xfsh-ha1` and `xfsh-ha2`, add the following to `/usr/lib/aliases` on `xfsh-ha1`:

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

On `xfsh-a2`, add the following line to `/usr/lib/aliases`:

```
fsafe_admin:operations@console.xyz.com,admin_user@xfsh-hal.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` — 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

When using a ring reset configuration, you must turn off the `getty` process for the tty ports to which the reset serial cables. . 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:

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

Install Patches

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

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:

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

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

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

5. Start cluster processes on the node:

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

You are now ready to run the FailSafe Manager GUI or the `cmgr` 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:

```
patchSGxxxxxx.sysadm_base_sw.client
patchSGxxxxxx.sysadm_failsafe2_sw.client
patchSGxxxxxx.sysadm_failsafe2_sw.desktop
```

2. Choose a node on which to install the patch. Start up the FailSafe GUI or `cmgr` 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 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 HA services stop, FailSafe will no longer be able to report current cluster and node state if the FailSafe GUI is connected to that node. To monitor the cluster state during installation, connect the FailSafe GUI 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 optional step 3, 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 an error occurs, either complete step 3 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:

```
irix# /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:

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

7. Use `chkconfig` to turn off the `cluster` flag:

```
irix# 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 208.

8. Install the patch on the node.
9. Use `chkconfig` to turn on the `cluster` flag:

```
irix# chkconfig cluster on
```

10. Start cluster processes on the node:

```
irix# /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 view area. 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
- 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
<code>pcp_fsafe.man.pages</code>	40
<code>pcp_fsafe.man.relnotes</code>	32
<code>pcp_fsafe.sw.collector</code>	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` command:


```
irix# 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_fsafes` 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/fsafes` directory:

```
irix# cd /var/pcp/pmdas/fsafes
```

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

```
irix# ./Install
```

10. Choose an appropriate configuration for installation of the `fsafes` 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_fsafes` subsystem by performing the following commands:

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

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

2. Run the Remove utility:

```
irix# ./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
<code>pcp_fsafes.man.pages</code>	40
<code>pcp_fsafes.man.relnotes</code>	32
<code>pcp_fsafes.sw.monitor</code>	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` :

```
irix# 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_fsafesw` package for a monitor configuration:

```
Inst> keep pcp_fsafesw.collector
```

```
Inst> install pcp_fsafesw.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 83

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:

```
irix# /usr/etc/ping -c 3 nodeIPAddress
```

Typical ping 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 ttl=254 time=3 ms
64 bytes from 190.x.x.x: icmp_seq=1 ttl=254 time=2 ms
64 bytes from 190.x.x.x: icmp_seq=2 ttl=254 time=2 ms
```

If ping fails, follow these steps:

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

```
irix# /usr/etc/ifconfig ec3
ec3: flags=c63<UP,BROADCAST,NOTRAILERS,RUNNING,FILTMULTI,MULTICAST>
inet 190.x.x.x netmask 0xfffff00 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 hardware reset connections, do the following:

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

```
irix# 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 FailSafe administration tasks using either the FailSafe Manager graphical user interface (GUI) or the `cmgr` 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 Features", page 90.

FailSafe Manager GUI

The FailSafe Manager GUI lets you configure, administer, and monitor a FailSafe cluster. This section contains the following:

- "Starting the GUI"
- "GUI Features", page 90
- "Screens", page 95

Starting the GUI

There are several methods to start the GUI and connect to a node.

Launch Methods

To start the GUI, use one of the following methods:

- On an IRIX system where the FailSafe GUI-client software (`sysadm_failsafe2.sw.client`) and desktop support software (`sysadm_failsafe.sw.desktop`) are installed, do one of the following:

Note: Do not use this method across a wide-area network (WAN) or virtual private network (VPN), or if the IRIX system has an R5000 or earlier CPU and less than 128-MB memory.

– Enter the following command line:

```
# /usr/sbin/fsmgr
```

(The `fsdetail` and `fstask` commands perform the identical function as `fsmgr`; these command names are kept for historical purposes.)

- Choose the following from the Toolchest:

System

> FailSafe Manager

You must restart the Toolchest after installing FailSafe in order to see the **FailSafe** entry on the Toolchest display. Enter the following commands to restart the Toolchest:

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

- On a PC or if you want to perform administration from a remote location via VPN or WAN, do the following:
 - Install a web server (such as Apache) and the `sysadm_failsafe2.sw.web` package on one of the administration nodes with an R5000 or later and at least 128-MB memory.
 - Install the Java2 v1.4.1 plug-in on your PC.
 - Close any existing Java windows and restart the Web browser on the PC.
 - Enter the URL `http://server/FailSafeManager/` (where *server* is the name of a administration node in the pool)
 - At the resulting webpage, click the FailSafe Manager icon.

Note: This method can be used on IRIX systems, but it is not the preferred method unless you are using WAN or VPN. If you load the GUI using Netscape on IRIX and then switch to another page in Netscape, CXFS Manager GUI will not operate correctly. To avoid this problem, leave the CXFS Manager GUI web page up and open a new Netscape window if you want to view another web page.

Table 4-1 describes the platforms where the GUI may be started, connected to, and displayed.

Table 4-1 GUI Platforms

GUI Mode	Where You Start the GUI	Where You Connect the GUI	Where the GUI Displays
fsmgr(1) or Toolchest	An IRIX system (such as an SGI 2000 series, SGI O2 workstation, or Silicon Graphics Fuel visual workstation) with <code>sysadm_failsafe2.sw.client</code> and <code>sysadm_failsafe.sw.desktop</code> software installed	The FailSafe administration node in the pool that you want to use for cluster administration	The system where the GUI was invoked
Web	Any system with a web browser and Java 1.1 plug-in installed and enabled	The FailSafe administration node in the pool that you want to use for cluster administration	The same system with the web browser

Logging In

To ensure that the required GUI 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 other user using the GUI privilege tasks; see "Granting Task Execution Privileges to Users", page 286. (This functionality is also available with the Privilege Manager, part of the IRIX Interactive Desktop System Administration `sysadmdesktop` product. For more information, see the *Personal System Administration Guide*.)

A dialog box will appear, prompting you to log in to a FailSafe host. You can choose one of the following connection types:

- **Local** runs the server-side process on the local host instead of going over the network.
- **Direct** creates a direct socket connection using the `tcpmux` TCP protocol.
- **Remote Shell** connects to the server via a user-specified command shell, such as `rsh(1C)` or `ssh(1)`.

Note: For a secure connection, choose **Remote Shell** and type a secure connection command using a utility such as `ssh(1)`. Otherwise, the FailSafe Manager GUI will not encrypt communication and transferred passwords will be visible to users of the network.

- **Proxy** connects to the server through a firewall via a proxy server.

Making Changes from One Node

You should only make changes from one GUI process running at any given time; changes made by a second GUI process (a second invocation of `fsmgr`) may overwrite changes made by the first instance. However, multiple FailSafe Manager windows accessed via the **File** menu are all part of the same application process; you can make changes from any of these windows.

The FailSafe administration node to which you connect the GUI affects your view of the cluster. You should wait for a change to appear in the *details area* before making another change; the change is not guaranteed to be propagated across the cluster until it appears in the view area. (To see the location of the view and details areas, see Figure 4-1, page 95.) The entire cluster status information is sent to every FailSafe administration node each time a change is made to the cluster database.

GUI Features

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 FailSafe cluster:

- *Tasks* let you set up and monitor individual components of a CXFS cluster, including XVM volumes. For details about XVM tasks, see the *XVM Volume Manager Administrator's Guide*.
- *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.

This section discusses the following:

- "GUI Window Layout"
- "File Menu", page 91
- "Edit Menu", page 91
- "Tasks Menu", page 92
- "Help Menu", page 92
- "View Menu", page 92
- "Performing Tasks", page 94
- "Getting More Information", page 94

GUI Window Layout

By default, the window is divided into two sections: the *view area* and the *details area*. The details area shows generic overview text if no item is selected in the view area. You can use the arrows in the middle of the window to shift the display.

File Menu

The **File** menu lets you display multiple windows for this instance of the GUI, the `/var/adm/SYSLOG` system log file, and the `/var/sysadm/salog` system administration log file. (The `salog` file shows the commands run directly by this instance of the GUI or run as a side effect of someone running `cmgr` commands on the system or some other instance of the GUI running commands on the system. Changes should not be made simultaneously by multiple instances of the GUI or the GUI and `cmgr`.) The **File** menu also lets you close the current window and exit the GUI completely.

Edit Menu

The **Edit** menu lets you expand and collapse the contents of the view area. You can choose to automatically expand the display to reflect new nodes added to the pool or cluster. You can also use this menu to select all items in the view menu or clear the current selections.

Tasks Menu

The **Tasks** menu contains the following:

- **Guided Configuration**, which contains the tasks to set up your cluster, define filesystems, create volumes, check status, and modify an existing cluster
- **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 manage or modify existing resource types, or create new ones
- **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 tiebreaker node, and set the log configuration
- **Diagnostics**, which contains the tasks to test connectivity, resources, and failover policies
- **Privileges**, which lets you grant or revoke access to a specific task for one or more users
- **Find Tasks**, which lets you use keywords to search for a specific task

Help Menu

The **Help** menu provides an overview of the GUI and a key to the icons. You can also get help for certain items in blue text by clicking on them.

View Menu

Choose what you want to view from the **View** menu:

- Resources in groups
- Groups owned by nodes

- Resources owned by nodes
- Resources by type
- Groups by failover policies
- Groups
- Nodes in the cluster
- Nodes in the pool (that is, all defined nodes)
- Users
- Task privileges

Selecting Items to View or Modify

You can use the following methods to select items:

- Click to select one item at a time
- Shift+click to select a block of items
- Ctrl+click to toggle the selection of any one item

Another way to select one or more items is to type a name into the **Find** text field and then press **Enter** or click the **Find** button.

Viewing Component Details

To view the details on any component, click its name in the view area; see "Selecting Items to View or Modify", page 93.

The configuration and status details for the component will appear in the details area to the right. At the bottom of the details area will be the **Applicable Tasks** list, which displays tasks you may wish to launch after evaluating the component's configuration details. To launch a task, click the task name; based on the component selected, default values will appear in the task window.

To see more information about an item in the details area, select its name (which will appear in blue); details will appear in a new window. Terms with glossary definitions also appear in blue.

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 view area. For example:

Task
 > **Guided Configuration**
 > **Set Up a New Cluster**

The task window appears.

As a shortcut, you can right-click an item in the view area to bring up a list of tasks applicable to that item; information will also be displayed in the details area.

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**.)

Note: 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.

Getting More Information

Click blue text to see term definitions, instructions on what to input, or item configuration details, or to launch tasks.

In general, clicking on blue text will display one of the following:

- Term definitions
- Input instructions
- Item details
- The selected task window

Screens

Figure 4-1 shows a sample GUI window.

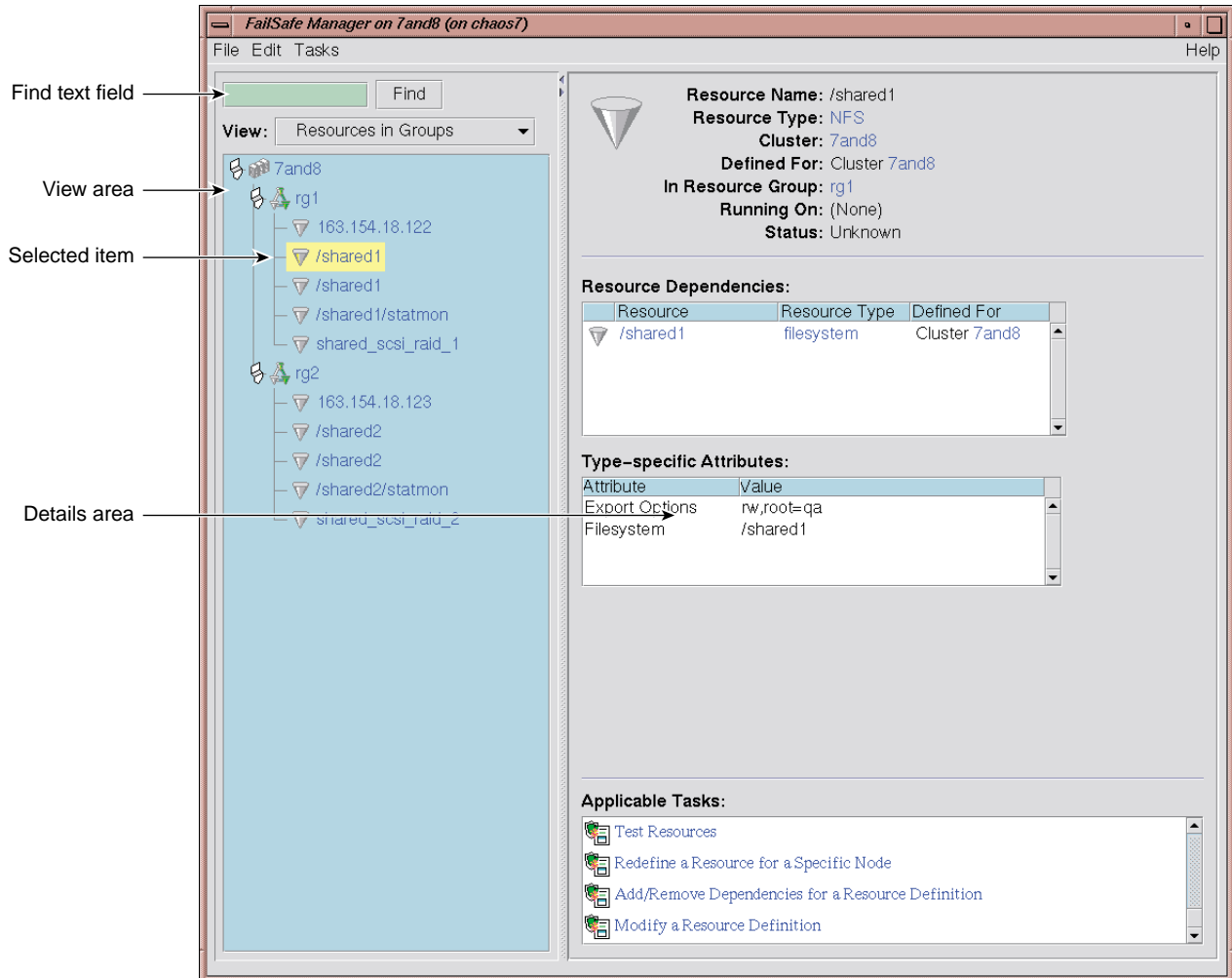


Figure 4-1 GUI Showing Details for a Resource

Figure 4-2 shows an example of the pop-up menu of applicable tasks that appears when you click the right mouse button on a selected item; in this example, clicking on the resource group name `bartest-group` displays a list of applicable resource-group tasks.

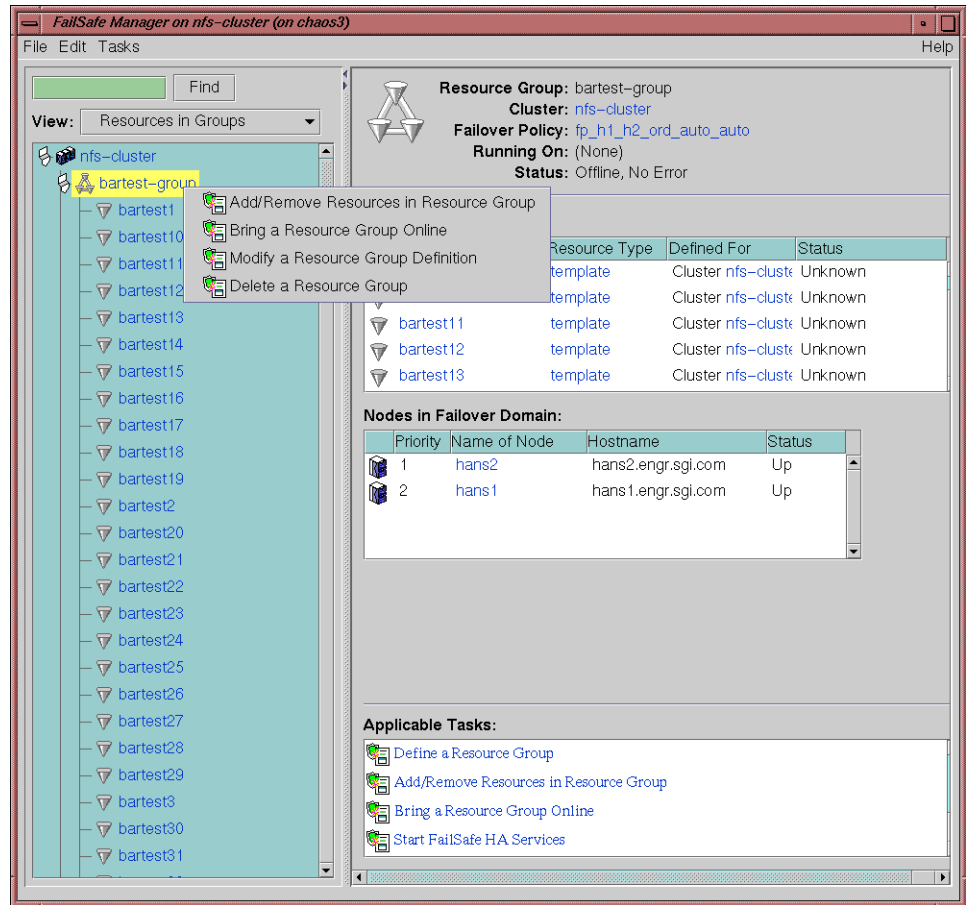


Figure 4-2 Pop-up Menu that Appears After Clicking the Right Mouse Button

cmgr Command

The `cmgr` command enables you to configure and administer a FailSafe system using a command-line interface 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 the following:

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

For more assistance, you can use the `-p` option on the command line; see "Using Prompt Mode", page 98.

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.

This section contains the following:

- "Getting Help", page 98
- "Using Prompt Mode", page 98
- "Completing Actions and Cancelling", page 100
- "Command Line Editing within `cmgr`", page 100
- "Long-Running Tasks", page 101
- "Startup Script", page 101
- "Entering Subcommands on the Command Line", page 102

- "Using Script Files", page 102
- "Creating a cmgr Script Automatically", page 104
- "Template Scripts", page 107
- "Invoking a Shell from within cmgr", page 108

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` command provides an option which displays detailed prompts for the required inputs 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 cmla
```

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

```
cmla?
```

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 126). 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|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
```

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 command-line editing commands:

<code>history [n]</code> or <code>h [n]</code>	Displays command line history. The optional <i>n</i> can be used to set the number commands that will be remembered.
<code>!!</code>	Refers to the previous command. By itself, this substitution repeats the previous command.
<code>!n</code>	Refers to command line <i>n</i> .
<code>!-n</code>	Refers to the current command line minus <i>n</i> .
<code>!string</code>	Refers to the most recent command starting with <i>string</i> .
<code>exit</code>	Exits from the shell.
<code>Ctrl-W</code>	Deletes the previous word.
<code>Ctrl-D</code>	Deletes the current character.
<code>Ctrl-A</code>	Goes to the beginning of the line.
<code>Ctrl-E</code>	Goes to the end of the line.
<code>Ctrl-F</code>	Moves forward one character.
<code>Ctrl-B</code>	Moves backward one character.
<code>Ctrl-H</code>	Deletes the previous character.
<code>Ctrl-N</code>	Moves down in the history.
<code>Ctrl-K</code>	Erases to the end of the line from the cursor.
<code>Ctrl-L</code>	Clears the screen and redisplay the prompt.
<code>Ctrl-P</code>	Moves up in the history.
<code>Ctrl-U</code>	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_STARTUP_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_STARTUP_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 HA services and sets the configuration so that HA services will be automatically restarted upon reboot
- `stop`, which stops 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 109, 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
```

Creating a `cmgr` Script Automatically

After you have configured the cluster database, you can use the `build_cmgr_script` command to automatically create a `cmgr` script based on the contents of the cluster database. The generated script will contain the following:

- Node definitions
- Cluster definition
- Resource definitions
- Resource type definitions
- Resource group definitions
- Failover policy definitions
- HA parameters settings

- Any changes made using either the `cmgr` command or the GUI
- CXFS information (only in a coexecution cluster)

When you use the `-s` option, the command also generates `create_resource_type` scripts for resource types.

As needed, you can then use the generated script to recreate the cluster database after performing a `cdbreinit`.

By default, the generated script is placed in the following location:

```
/tmp/cmgr_create_cluster_clustername_processID
```

You can specify an alternative path name by using the `-o` option:

```
build_cmgr_script [-o script_pathname]
```

For more details, see the `build_cmgr_script` man page.

For example:

```
# /var/cluster/cmgr-scripts/build_cmgr_script -o /tmp/newcdb
Building cmgr script for cluster test-cluster ...
build_cmgr_script: Generated cmgr script is /tmp/newcdb
```

The example script file contents are as follows:

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

# Node node1 definition
define node node1
    set hostname to node1.dept.company.com
    set is_failsafe to true
    set nodeid to 32065
    set hierarchy to Reset,Shutdown
    set reset_type to powerCycle
    set sysctrl_type to msc
    set sysctrl_status to enabled
    set sysctrl_owner to node2
    set sysctrl_device to /dev/ttyd2
    set sysctrl_owner_type to tty
    add nic 192.0.2.58
```

```
        set heartbeat to true
        set ctrl_msgs to true
        set priority to 1
    done
    add nic 160.0.2.15
        set heartbeat to true
        set ctrl_msgs to true
        set priority to 2
    done
done

# Node node2 definition
define node node2
    set hostname to node2.dept.company.com
    set is_failsafe to true
    set nodeid to 24140
    set hierarchy to Reset,Shutdown
    set reset_type to powerCycle
    set sysctrl_type to msc
    set sysctrl_status to enabled
    set sysctrl_owner to node1
    set sysctrl_device to /dev/ttyd2
    set sysctrl_owner_type to tty
    add nic 192.0.2.59
        set heartbeat to true
        set ctrl_msgs to true
        set priority to 1
    done
    add nic 160.0.2.16
        set heartbeat to true
        set ctrl_msgs to true
        set priority to 2
    done
done

# Define cluster and add nodes to the cluster
define cluster test-cluster
    set is_failsafe to true
    set ha_mode to normal
done
```

```

modify cluster test-cluster
    add node node1
    add node node2
done

set cluster test-cluster

quit

```

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 4-2 shows the template scripts for `cmgr` that are found in the `/var/cluster/cmgr-templates` directory.

Table 4-2 Template Scripts for `cmgr`

File name	Description
<code>cmgr-create-cluster</code>	Creates a cluster
<code>cmgr-create-failover_policy</code>	Creates a failover policy
<code>cmgr-create-node</code>	Creates a node
<code>cmgr-create-resource_group</code>	Creates a resource group
<code>cmgr-create-resource_type</code>	Creates a resource type
<code>cmgr-create-resource-ResourceType</code>	Creates a the specified resource of 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`

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

```
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` command.

Note: For the initial installation, SGI **highly** recommends that you use the GUI guided configuration tasks. See "Guided Configuration with the GUI", page 114.

SGI also recommendeds that you perform all FailSafe administration 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. Complete the following steps to ensure that you are ready to configure the initial cluster:

- "Verify that the Cluster `chkconfig` Flag is On", page 110
- "Start the Cluster Daemons", page 110
- "Verify that the Cluster Daemons are Running", page 110
- "Determine the Hostname of the Node", page 111

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` shows the following flag set to on:

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

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

```
# 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/cmmond_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` when logging in to the GUI and when defining the nodes in the pool. The value of `/etc/sys_id` must match the name of the IP address for the node in `/etc/hosts`. The value of `/etc/sys_id` is displayed by the `hostname` command. For example:

```
# hostname
fs6
```

Also, if you use `nsd`, 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 59.



Caution: It is critical that these files are configured properly and that you enter the hostname 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 apply 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:

$$\text{application_downtime} = \text{failure_detection} + \text{time_to_handle_failure} + \text{failure_recovery_time}$$

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 operating system 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 152.

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 RTname
```

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 view area before making another change; the change is not guaranteed to be propagated across the cluster until the icons appear in the view area.

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 `fsmgr`) 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 **OK**.

The **Set Up a New Cluster** task 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. Click **Define a Node** to define the node to which you are connected (that is, the local node). The hostname that appears in `/etc/sys_id` is used for all node definitions; see "Determine the Hostname of the Node", page 111. See "Define a Node", page 122.
-

Note: If you attempt to define a cluster or other object before the local node has been defined, you will get an error message that says:

```
No nodes are registered on servername. You cannot define a cluster until you define the node to which the GUI is connected. To do so, click "Continue" to launch the "Set Up a New Cluster" task.
```

2. (Optional) **After** the first node icon appears in the view area, 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 a second node until the icon for the first node appears in the view area. 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 three nodes and then continue on to the next step; add the remaining nodes after you have a successful small cluster.

3. Click **Define a Cluster** to create the cluster definition. See "Define a Cluster", page 143. Verify that the cluster appears in the view area; choose **View: Nodes in Cluster**.
4. Click **Add/Remove Nodes in Cluster** to add the nodes to the new cluster. See "Add or Remove Nodes in the Cluster with the GUI", page 131.

Click **Next** to move to the second page of tasks.

5. (*Optional*) Click **Test Connectivity** to verify that the nodes are physically connected. See "Test Connectivity with the GUI", page 292. (This test requires the proper configuration of the `/etc/.rhosts` file.)
6. Click **Start HA Services**.
7. Click **Close**. Clicking on **Close** exits the task; 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 **OK**.

The **Set Up a Highly Available Resource Group** task 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 Resource", page 175.
2. Add any required resource dependencies. See "Add/Remove Dependencies for a Resource Definition", page 184.
3. Verify the resources and dependencies. See "Test Resources with the GUI", page 292.
4. Define a failover policy to specify where the resources can run. See "Define a Failover Policy", page 190.

5. Test the failover policies. See "Test Failover Policies with the GUI", page 293.
Click **Next** to move to the next page.
6. Define a resource group that uses the failover policy you defined earlier. See "Define a Resource Group", page 201.
7. Add or remove resources in resource group. See "Test Failover Policies with the GUI", page 293.
8. Set the resources in the resource group to start when HA services are started. See "Bring a Resource Group Online", page 273.
9. Start FailSafe HA services if they have not already been started. See "Start FailSafe HA Services with the GUI", page 207.

Repeat these steps for each resource group.

Set Up an Existing CXFS Cluster for FailSafe

This task 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 **OK**.

The **Set Up an Existing CXFS Cluster for FailSafe** task 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 **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 148.
2. Use the CXFS GUI (or `cmgr` command) to stop CXFS services on the nodes to be converted. See the *CXFS Version 2 Software Installation and Administration Guide*.

3. Click **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 137.
4. Click **Convert a CXFS Node to FailSafe** to convert another node. Repeat this step for each node you want to convert.
5. Click **Start HA Services**.

Fix or Upgrade Cluster Nodes

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

- Migrate all resource groups off the node. This might require using the **Force** option. See "Move a Resource Group with the GUI", page 277.
- Stop FailSafe HA services on the cluster nodes. See "Stop FailSafe HA Services", page 208.
- Perform the necessary maintenance on the node. Only if required, see "Reset a Node with the GUI", page 280.
- Turn HA services back on so that the node and its resource groups are monitored. See "Start FailSafe HA Services with the GUI", page 207.
- Monitor the state of the cluster components in the view area. See "System Status", page 255.

Make Changes to Existing Cluster

You can make most cluster changes when HA services are active. To use the destructive option in FailSafe diagnostics, you must stop HA services on all nodes in the cluster. To make changes to network configuration (IP address, hostname, network interfaces) in a FailSafe node, you must stop HA and cluster services on all nodes in the pool.

See the following:

- "Modify a Cluster Definition", page 147
- "Define a Node", page 122

- "Test Connectivity with the GUI", page 292
- "Add or Remove Nodes in the Cluster with the GUI", page 131
- "Set FailSafe HA Parameters", page 210

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 196.
- "Move a Resource Group with the GUI", page 277.
- Create a resource that has a custom definition on a specific node. See "Redefine a Resource for a Specific Node", page 183.
- 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 161.

Define Custom Resource

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

- "Define a Resource Type", page 152
- "Redefine a Resource for a Specific Node", page 183
- "Add/Remove Dependencies for a Resource Type", page 164
- "Define a Resource", page 175
- "Add/Remove Dependencies for a Resource Definition", page 184
- "Test Resources with the GUI", page 292

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 interval at which FailSafe sends messages between nodes). See "Set FailSafe HA Parameters", page 210.
- Change the monitor action timeout and the restart action timeout used by a resource type. See "Modify a Resource Type Definition", page 168.

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 210.
- To change the nodes and their ordering in the failover domain, see "Modify a Failover Policy Definition", page 196.
- To change monitoring settings for the resource types used in the resource group, see "Modify a Resource Type Definition", page 168.

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 196.
5. Select **View: Groups owned by Nodes** in the GUI view area.
6. Test the script by moving a resource group from one node to another, simulating failover. Watch the resource group behavior in the view area to confirm that failover behavior works as expected. See "Move a Resource Group with the GUI", page 277.

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 152, and "Modify a Resource Type Definition", page 168.
6. Define resources using the new resource type. See "Define a Resource", page 175.
7. Verify that FailSafe can manage the new custom resources. See "Test Resources with the GUI", page 292.
8. Add the new resource. See "Add or Remove Nodes in the Cluster", page 131.

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. Do the following:

1. Determine the current load. For example, invoke the **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 with the GUI", page 277.
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 190.
- Use the new failover policy for the resource group. See "Modify a Resource Group Definition", page 203.
- 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 with the GUI", page 277.

Node Tasks

A *node* is an operating system (OS) image, usually an individual computer. 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 or Remove Nodes in the Cluster", page 131
- "Modify a Node Definition", page 132
- "Convert a CXFS Node to FailSafe", page 137
- "Delete a Node", page 138
- "Display a Node", page 141

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 **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` 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 59.

Note: If you attempt to define a cluster or other object before the local node has been defined, you will get an error message that says:

No nodes are registered on *servername*. You cannot define a cluster until you define the node to which the GUI is connected. To do so, click "Continue" to launch the "Set Up a New Cluster" task.

- **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 59. For information about why a private network is required, see "Private Network", page 6.

- **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 **Add** to add the network to the list.

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

If you want to delete a network from the list, click the network in the list to select it, then click **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` 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 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 247
 - **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 the **Set System Controller Parameters** box to deselect (uncheck) it.

2. Click **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 view area. 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 LogicalHostname
  set hostname to Hostname
  set nodeid to NodeID
  set node_function to server_admin|client_admin
  set partition_id to PartitionID
  set reset_type to powerCycle
  set sysctrl_type to msc|mmsc|I2|I1 (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
  add nic IPaddressOrHostname (if DNS)
    set heartbeat to true|false
    set ctrl_msgs to true|false
    set priority to integer
  remove nic IPaddressOrHostname (if DNS)
```

There are additional commands that apply to CXFS; if you are running a coexecution cluster, see *CXFS Version 2 Software Installation and Administration Guide* for more information.

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` 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 *LogicalHostname*; therefore, you must supply a value for this command if you use a value other than the hostname or an abbreviation of it for *LogicalHostname*.
- `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.
- `node_function` specifies the CXFS function of the node. If you use prompting mode, you must enter one of the following:
 - `server_admin` for a node that you wish to use as a CXFS metadata server in a coexecution cluster.
 - `client_admin`, for a node that will not be used as a CXFS metadata serverA FailSafe node cannot have the `client-only` function; this function is for CXFS-only nodes.
- `partition_id` uniquely defines a partition in a partitioned SGI Origin 3000 system.

Note: Use the `mkpart` 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 129.
- `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 hardware reset 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_failsafe` to `true`. If you are running both CXFS and FailSafe on this node in a coexecution cluster, set both values to `true`.
- `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. SGI recommends that this network be private; see "Private Network", page 6.

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 59. For information about why a private network is required, see "Private Network", page 6.

Table 5-1 System Controller Types

11	12	mmsc	msc
Origin 300 (see also "Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C Console Support", page 247)	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 IPaddress
```

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 false  
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 Function <server_admin|client_admin ? client_admin
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? 192.56.50.1
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? 192.56.50.2
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 or Remove Nodes in the Cluster

This section describes how to add or remove nodes.

Add or 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 until the cluster icon appears in the view area; 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 **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 **Remove**.
2. Click **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 **Add**.
 - If you want to modify a network that is already in the list, click 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 **Add**.
 - If you want to delete a network, click on the network in the priority list in order to select it. Then click **Delete**.

- If you want to change the priority of a network, click the network in the priority list in order to select it. Then click 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` 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 **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 247

- **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 the **Set System Controller Parameters** box to deselect (uncheck) it.

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

Modify a Node with `cmgr`

To modify an existing node, use the following commands:

```
modify node LogicalHostname
  set hostname to Hostname
  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
add nic IPaddress_Or_Hostname (if DNS)
    set heartbeat to true|false
    set ctrl_msgs to true|false
    set priority to integer
remove nic IPaddress_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 126.

There are additional commands that apply to CXFS; if you are running a coexecution cluster, see *CXFS Version 2 Software Installation and Administration Guide* for more information.



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

```

5: Configuration

```
cmgr> modify node n_preston
```

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

```
Hostname[optional] ? (preston.dept.company.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 ? (192.168.168.2)
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.dept.company.com
Node Is FailSafe: true
Node Is CXFS: true
Nodeid: 606
Partition id: 1
Reset type: powerCycle
ControlNet Ipaddr: 192.168.168.2
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 FailSafe, 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 **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 132. 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 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 147.

To delete a node, do the following:

1. **Node to Delete:** select the logical name of the node to be deleted.
2. Click **OK** to complete the task.

Delete a Node with `cmgr`

To delete a node, use the following command:

```
delete node Nodename
```

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
```

```
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:

- Nodes that have been defined (**Nodes in Pool**)
- Nodes that are members of a specific cluster (**Nodes in Cluster**)
- Attributes of a node

Click any name or icon in the view area to see detailed status and configuration information in the details area.

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: 192.56.50.1  
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
   cxf8
   cxf6
   cxf7
```

You can show the nodes in the cluster with the following command:

```
show nodes in cluster Clustername
```

For example, if node `cxf8` was in the pool but not in `clusterA`, you would see:

```
cmgr> show nodes in cluster clusterA
```

```
Cluster clusterA has following 2 machine(s)
   cxf6
   cxf7
```

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 143
- "Modify a Cluster Definition", page 147
- "Convert a CXFS Cluster to FailSafe", page 148

- "Delete a Cluster", page 149
- "Display a Cluster", page 151

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. Clusters that share a network and use XVM must have unique names.
 - **Cluster Mode:** usually, you should choose the default `Normal` mode.
Choosing `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 (by default, `/usr/sbin/Mail`) 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 **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 143.

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 NotifyCommand
  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_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 (by default, `/usr/sbin/Mail`) as the *NotifyCommand*.
- `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 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
```

5: Configuration

```
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 143, 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 (by default `/usr/sbin/Mail`) 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 **OK**.

To modify the nodes that make up a cluster, see "Add or Remove Nodes in the Cluster with the GUI", page 131.

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 EmailAddress
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 **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 or Remove Nodes in the Cluster", page 131.

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 first remove all nodes from the cluster. See "Add or Remove Nodes in the Cluster with the GUI", page 131.

To delete a cluster, do the following:

1. **Cluster to Delete:** select the cluster name.
2. Click **OK** to complete the task.

Delete a Cluster with cmgr

You cannot delete a cluster that contains nodes; you must first remove all nodes from the cluster.

To delete a cluster, use the following command:

```
delete cluster Clustername
```

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 details area on the right side of the GUI screen.

Display a Cluster with `cmgr`

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

```
show clusters
show cluster Clustername
```

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
```

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 161
- "Add/Remove Dependencies for a Resource Type", page 164
- "Load a Resource Type", page 168
- "Modify a Resource Type Definition", page 168
- "Delete a Resource Type", page 173
- "Display a Resource Type", page 174

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 168.

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 page.

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.
- **Monitor 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.
- **Monitor Start Time:** the amount of time between starting a resource and beginning monitoring of that resource.

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

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 278.

- **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` 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` 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` command. For example:

```
filesystem = /dev/xlv/xlv_object
```

Click **Add** to add the attribute, and repeat as necessary for other attributes.

5. Click **OK** to complete the task.

Define a Resource Type with `cmgr`

Use the following commands:

```
define resource_type RTname on node Nodename [in cluster Clustername]  
  
define resource_type RTname [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 ActionScriptname
    set exec_time to ExecutionTimeout
    set monitor_interval to MonitorInterval
    set monitor_time to MonitorTime
add type_attribute Type-specific_Attributename
    set data_type to string|integer
    set default_value to Default
add dependency RTname
remove action ActionScriptname
remove type_attribute Type-specific_Attributename
remove dependency DependencyName
```

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 12. 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 (*RTname*)

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. **Local 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 page.

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.
- **Monitor 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.
- **Monitor Start Time:** the amount of time between starting a resource and beginning monitoring of that resource.

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

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 278.

- **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` 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` 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` command. For example:

```
filesystem = /dev/xlv/xlv_object
```

Click **Add** to add the attribute, and repeat as necessary for other attributes.

6. Click **OK** to complete the task.

Define 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 RTname 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.

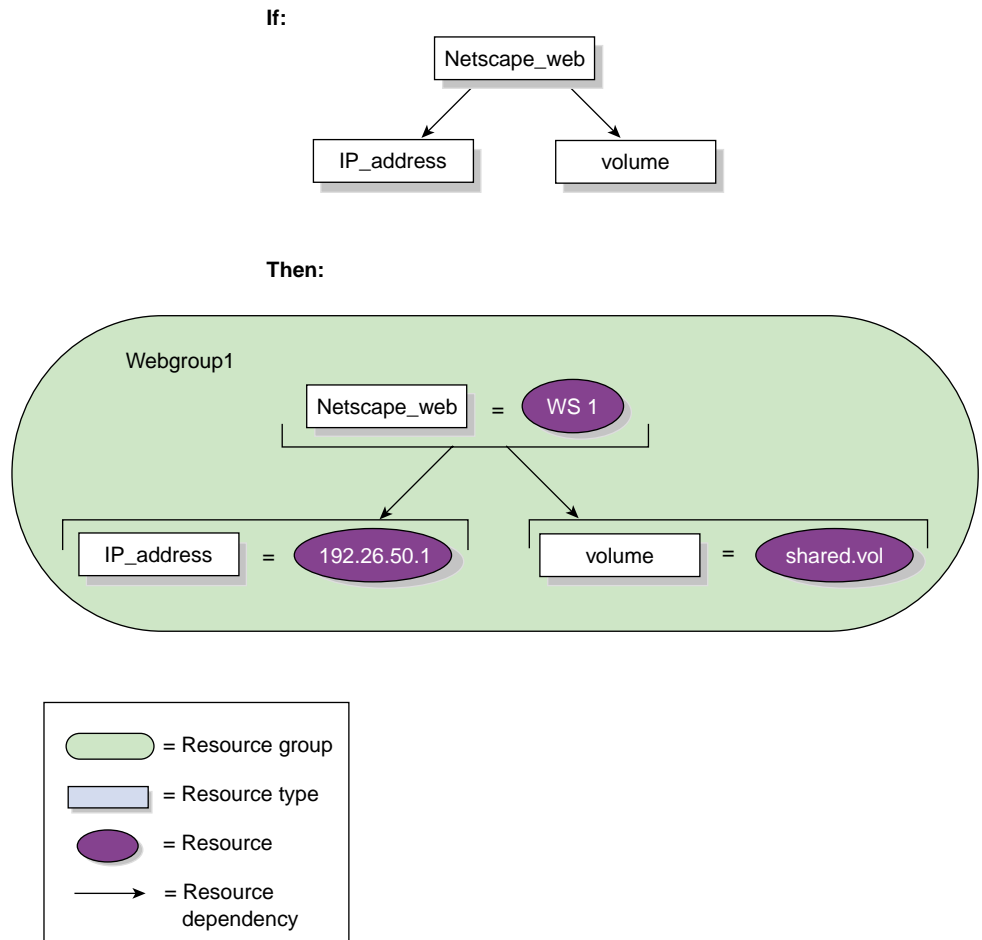


Figure 5-1 Dependencies

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 RTname [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 page. 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. **Monitor 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. **Monitor Start Time:** the amount of time between starting a resource and beginning monitoring of that resource.

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

9. **Enable Restart:** 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 278.

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 RTname [in cluster Clustername]
  set order to start/stop_OrderNumber
  set restart_mode to 0|1
  set restart_count to Number_Of_Attempts
  add action ActionScriptname
    set exec_time to ExecutionTimeout
    set monitor_interval to MonitorInterval
    set monitor_time to MonitorTime
  modify action ActionScriptname
    set exec_time to ExecutionTimeout
    set monitor_interval to MonitorInterval
    set monitor_time to MonitorTime
  add type_attribute Type-specificAttributename
    set data_type to string|integer
    set default_value to Default
  add dependency RTname
  remove action ActionScriptname
  remove type_attribute Type-specific_Attributename
  remove dependency Dependencyname

```

You modify a resource type using the same commands you use to define a resource type. See "Define a Resource", page 175.

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_unlimited` resource type monitor executable timeout from 40 seconds to 60 seconds.

```

#cmgr> modify resource_type statd_unlimited in cluster test-cluster
Enter commands, when finished enter either "done" or "cancel"

```

```

resource_type statd_unlimited? 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_unlimited ? done
Successfully modified resource_type statd_unlimited
```

The following examples show how to modify the resource type timeouts in prompt mode.

```
#cmgr> modify resource_type statd_unlimited 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_unlimited

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 be used instead.

You cannot delete a clusterwide resource type if there are any resources of that type.

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

Delete a Resource Type with `cmgr`

Use the following command to delete a resource type:

```
delete resource_type RTname [in cluster Clustername]
```

Display a Resource Type

This section describes how to display resource types.

Display Resource Types with the GUI

Select **View: Resource Types**. You can then click on any of the resource type icons in the view area 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 (*RTname*):

```
show resource_type RTname [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 Resource", page 175
- "Redefine a Resource for a Specific Node", page 183

- "Add/Remove Dependencies for a Resource Definition", page 184
- "Modify a Resource Definition", page 187
- "Delete a Resource", page 188
- "Display a Resource", page 189

Define a Resource

This section describes how to define a new resource.

Define a 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 183.

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 339. 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 an HA 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. XVM resource names must not begin with a slash (/).

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 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 page.

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 **OK** to complete the task.

CXFS Attributes

The CXFS resource is the mount point of the CXFS filesystem, such as `/shared_CXFS`. In the **Relocate Metadata server?** field, you must specify 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`.

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

command. The list of available options is provided on the `fstab(4)` man page. The default is `rw`.

- **kill-nfsd-before-umount:**
 - `true` stops the `nfsd` NFS server processes in the server before the filesystem is unmounted and then restarts the `nfsd` daemons after the `umount` is completed. This is the default (`true`).
 - `false` unmounts the filesystem without changing the `nfsd` status.
- **Monitoring Level:** the monitoring level to be used for the filesystem.
 - `1` specifies to check whether the filesystem exists in `/etc/mtab`, as described in the `mtab(4)` man page.
 - `2` specifies to check whether the filesystem is mounted using the `stat` command. This is a more-intrusive check that is more reliable if it completes on time; however, 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.

Specify the following parameters:

- **Network Mask:** 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.

- **Broadcast Address:** 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` command.

You must specify attribute **Interface:**, which is 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 `xl_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 `xl_vmgr` to see which machine has the volume assembled.

Specify the following parameters:

- **Device Group:** the group name of the XLV device file. The `sys` group is the default group name for XLV device files.
- **Device Owner:** the user name (login name) of the owner of the XLV device file. `root` is the default owner for XLV device files.
- **Device Mode:** the device file permissions, specified in octal notation. `600` mode is the default value for XLV device file permissions.

xvm Attributes

The `xvm` resource type is the local XVM volume used by FailSafe applications.

When you define an `xvm` resource, the resource name must be a unique string for all local XVM domains in the FailSafe cluster. The name should be the name of the volume without the preceding `vol/` characters.

Specify the following parameters:

- **FS XVM owner:** is the XVM temporary owner for FailSafe. This value must not be the cluster name or a hostname known to any machine in the cluster. The default is `fake_owner`.
- **Device group:** is the XVM device group. The default is `sys`.
- **Physvol names:** the names of the physical volume that comprise an XVM volume, separated by commas. This name does not contain the `vol/` prefix that is displayed by the `xvm` command. (For example, if you enter `bigvol` here, the `xvm show` command would display this physvol as `vol/bigvol`.) You must enter a value in this field, there is no default.
- **Device mode:** is the XVM device mode permissions. The default is `600`.
- **Device owner:** is the XVM device owner. The default is `root`.

For more details about XVM, see the *XVM Volume Manager Administrator's Guide*.

Define a Resource with cmgr

Use the following command to define a clusterwide resource:

```
define resource Resourcename [of resource_type RTname] [in cluster Clustername]  
  set Key to AttributeValue  
  add dependency Dependencyname of type RTname  
  remove dependency Dependencyname of type RTname
```

Usage notes:

- The resource name has a maximum length of 255 characters and cannot begin with an underscore. XVM resource names must not begin with a slash (/).
- set *Key* specifies the name of the attribute, and *AttributeValue* sets its value
- add dependency adds a dependency of the specified resource type (*RTname*)
- 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 *AttributeValue* will depend on the type of resource you are defining, as described in "Define a Resource", page 175. For detailed information on how to determine the format for defining resource attributes, see "Specify Resource Attributes with cmgr", page 181.

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 RTname [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 AttributeValue` 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.
	mount-options	Specifies the mount options to be used for mounting the filesystem.
	kill-nfsds-before-umount	When set to <code>true</code> , stops the <code>nfsd</code> NFS server processes running in the server before the filesystem is unmounted and then restarts <code>nfsd</code> after the <code>umount</code> is completed. When set to <code>false</code> , the <code>umount</code> takes place without changing the <code>nfsd</code> status. The default is <code>true</code> .
	monitoring-level	Specifies the monitoring level to be used for the filesystem: <ul style="list-style-type: none"> • 1 specifies to check whether the filesystem exists in <code>/etc/mtab</code>, as described in the <code>mtab(4)</code> man page. • 2 specifies to check whether the filesystem is mounted using the <code>stat</code> command. This is a more-intrusive check that is more reliable if it completes on time; however, some loaded systems have been known to have problems with this level.
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.
MAC_address	BroadcastAddress	Specifies the broadcast address for the IP 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.

Resource Type	Attribute	Description
XVM	<code>fs_xvm_owner</code>	Specifies the XVM temporary owner for FailSafe. This value must not be the cluster name or a hostname known to any node in the cluster. The default is <code>fake_owner</code> .
	<code>devname_group</code>	Specifies the XVM device group. The default is <code>sys</code> .
	<code>physvol_names</code>	Specifies the names of the physical volumes, separated by commas. There is no default.
	<code>devname_mode</code>	Specifies the permission mode of the XVM device in octal. The default is <code>600</code> .
	<code>devname_owner</code>	Specifies the XVM device owner. The default is <code>root</code> .

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 100BASE-T 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. **Local 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 `fsmgr` — 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 page.
4. **Type-specific attributes:** change the information for each attribute as needed. For information about each attribute, see "Define a Resource with the GUI", page 175.
5. Click **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 RTname on node Nodename [in cluster Clustname]
```

If you have already 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.

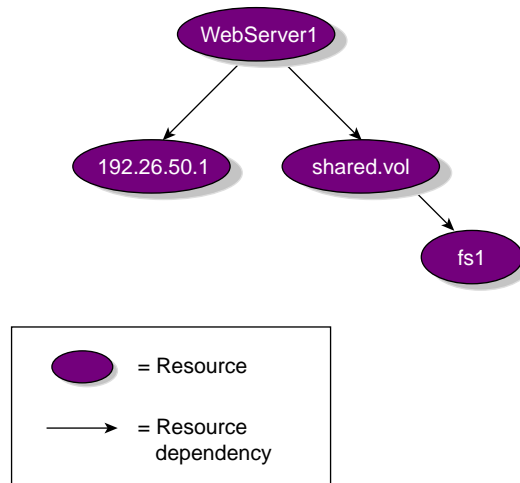


Figure 5-2 Example of Resource Dependency

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.

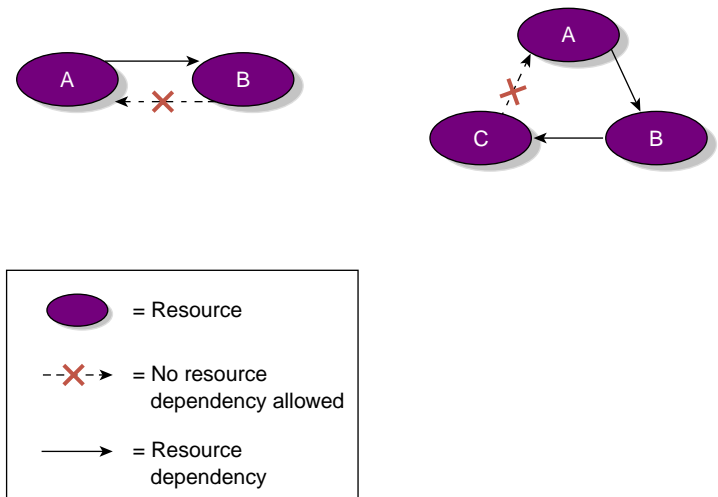


Figure 5-3 Mutual Dependency of Resources Is Not Allowed

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 **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 after 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 page.
3. **Type-specific attributes:** modify the attributes as needed. For information about attributes, see "Define a Resource with the GUI", page 175.
4. Click **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 RTname] on node Nodename [in cluster Clustername]
```

```
modify resource Resourcename [of resource_type RTname] [in cluster Clustername]  
  set key to AttributeValue  
  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 Resource", page 175.

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 "Add/Remove Resources in Resource Group", page 204.

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 **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 RTtype [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: Anyone 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 view area. 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 RTname
```

- To view all of the defined resources in a resource group:

```
show resources in resource_group RGname [in cluster Clustername]
```

- To view all of the defined resources of a particular resource type in a specified cluster:

```
show resources of resource_type RTname [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 196
- "Delete a Failover Policy", page 199
- "Display a Failover Policy", page 200

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. The 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` never changes the initial failover domain; when using this script, the initial and run-time failover domains are equivalent.
- `round-robin` 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 scripts provided with the release do 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. 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*, which 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 differ 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 <code>srmd executable error</code> status. If the <code>Critical_RG</code> 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 should 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 **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 190.

Use the following to define a failover policy:

```
define failover_policy Polycyname
  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 190.

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 190.

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` never changes the initial domain; when using this script, the initial and run-time domains are equivalent.
 - `round-robin` 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 scripts provided with the release do 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. 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 192 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 **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 195.

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 **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 view area. Select the name of a specific policy in the view area in order to see details about it in the details area.

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 203
- "Delete a Resource Group", page 203
- "Add/Remove Resources in Resource Group", page 204
- "Display a Resource Group", page 205

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 **OK** to complete the task.

To add resources to the group, see "Add/Remove Resources in Resource Group", page 204.

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.

Define a Resource Group with `cmgr`

Use the following command to define a resource group:

```
define resource_group RGname [in cluster Clustername]  
    set failover_policy to Policyname  
    add resource Resourcename of resource_type RTname  
    remove resource Resourcename of resource_type RTname
```

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 245.

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 204.

Modify a Resource Group Definition with `cmgr`

Use the following commands to modify a resource group:

```
modify resource_group RGname [in cluster Clustername]  
    set failover_policy to Policyname  
    add resource Resourcename of resource_type RTname  
    remove resource Resourcename of resource_type RTname
```

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 "Define a Resource Group with `cmgr`", page 202.

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. **Resource Group:** select the name of the resource group you want to delete. Only offline resource groups are listed.
2. Click **OK** to complete the task.

Delete a Resource Group with `cmgr`

Use the following command to delete a resource group:

```
delete resource_group RGname [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 after 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.

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 RGname [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 208
- "Set FailSafe HA Parameters", page 210
- "Set Log Configuration", page 213

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 ensure 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 verify 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 FailSafe 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. In 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 207.

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-HA 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. In 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 tiebreaker 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 tiebreaker node, the node with the lowest node ID number is used.

You should configure a tiebreaker node even if there is an odd number of nodes in the cluster because 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 tiebreaker 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 TimeoutValue  
  set heartbeat to HeartbeatInterval  
  set run_pwrfail to true|false  
  set node_wait to NodeWaitTime  
  set tie_breaker to TieBreakerNodename
```

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 tiebreaker.

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 tiebreaker 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 `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:

<code>cli</code>	Commands log
<code>crsd</code>	Cluster reset services (<code>crsd</code>) log
<code>diags</code>	Diagnostics log
<code>ha_agent</code>	HA monitoring agents (<code>ha_ifmx2</code>) log
<code>ha_cmsd</code>	FailSafe membership daemon (<code>ha_cmsd</code>) log
<code>ha_fsd</code>	FailSafe daemon (<code>ha_fsd</code>) log
<code>ha_gcd</code>	Group communication daemon (<code>ha_gcd</code>) log
<code>ha_ifd</code>	Network interface monitoring daemon (<code>ha_ifd</code>) log
<code>ha_script</code>	Action and failover policy scripts log
<code>ha_srmd</code>	System resource manager (<code>ha_srmd</code>) 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, page 215 shows the logging levels as you specify them with the GUI and `cmgr`.

Notifications of critical errors and normal operations are always sent to the `SYSLOG` file. 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`.

Table 5-5 shows the default log file names.

4. Click **OK** to complete the task.

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 through Debug9	10 through 19	Logs 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.

Table 5-5 Default Log File Names

Log File Name	Log File Owner
<i>/var/cluster/ha/log/cmsd_Nodename</i>	FailSafe membership services daemon in node <i>Nodename</i> .
<i>/var/cluster/ha/log/gcd_Nodename</i>	Group communication daemon in node <i>Nodename</i> .
<i>/var/cluster/ha/log/srmd_Nodename</i>	System resource manager daemon in node <i>Nodename</i> .
<i>/var/cluster/ha/log/failsafe_Nodename</i>	FailSafe daemon, a policy implementor for resource groups, in node <i>Nodename</i> .
<i>/var/cluster/ha/log/AgentNodename</i>	Monitoring agent named <i>Agent</i> in node <i>Nodename</i> . For example, <i>ifd_Nodename</i> is the log file for the interface daemon monitoring agent that monitors interfaces and IP addresses and performs local failover of IP addresses.
<i>/var/cluster/ha/log/crsd_Nodename</i>	Reset daemon in node <i>nodename</i> .

Log File Name	Log File Owner
<code>/var/cluster/ha/log/script_Nodename</code>	Scripts in node <code>nodename</code> .
<code>/var/cluster/ha/log/cli_Nodename</code>	Internal administrative commands in node <code>nodename</code> invoked by the GUI and <code>cmgr</code> .

Display Log Group Definitions with the GUI

To display log group definitions with the GUI, select 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 Logfilename
    remove log_file Logfilename
```

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 213.
- `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 215.

Configure Log Groups with `cmgr`

You can configure a log group with the following command:

```
define log_group LogGroup on node Nodename [in cluster Clustername]
```

The `LogGroup` 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 LogGroupname on node Nodename [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 216.

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 levels:

```
show log_groups
```

This command shows the currently defined log group names, their logging levels, and the log files. For example:

```
cmgr> show log_groups

ha_cmsd 13 /var/cluster/ha/log/cmsd
crsd 5 /var/cluster/ha/log/crsd
ha_gcd 5 /var/cluster/ha/log/gcd
ha_ifd 5 /var/cluster/ha/log/ifd
ha_srmd 14 /var/cluster/ha/log/srmd
ha_fsd 11 /var/cluster/ha/log/failsafe
cli 2 /var/cluster/ha/log/cli
ha_script 13 /var/cluster/ha/log/script
ha_agent 5 CommandName
diags 2 /var/cluster/ha/log/diags
clconfd 5 /var/cluster/ha/log/clconfd
```

In this example, `ha_cmsd` is the name of logging group used by the `ha_cmsd` daemon to log messages. The log verbosity level for `ha_cmsd` is 13; the verbosity level range from 0 (no message) to 19 (most verbose). The log file used is `/var/cluster/ha/log/cmsd`. A node name suffix is added to all log file names.

Use the following command to messages logged by a specific daemon on a specific node:

```
show log_group LogGroupName [on node Nodename]
```

To exit from the message display, enter `Cntrl-C`.

Configuration Examples

This chapter contains the following sections:

- "Example: Script to Define an SGI File Server 850 Cluster"
- "Example: Script to Define an SGI SAN Server 1000 Cluster", page 229
- "Example: Script to Define a Three-Node Cluster", page 235
- "Example: Local Failover of HA IP Address", page 242
- "Example: Modify a Cluster to Include a CXFS Filesystem", page 243
- "Example: Export CXFS Filesystems", page 244
- "Example: Create a Resource Group", page 245

Example: Script to Define an SGI File Server 850 Cluster

The script in this section configures an SGI File Server 850 cluster, which has two Origin 300 nodes in a FailSafe configuration attached to an SGI TP9100 RAID. Figure 6-1 shows an example of this configuration.

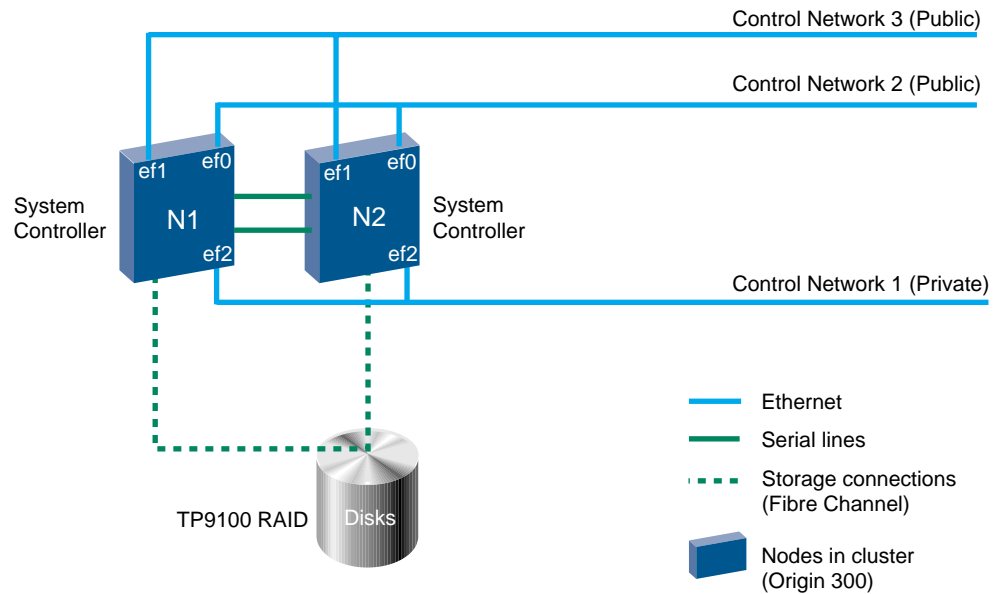


Figure 6-1 SGI File Server 850 Configuration

All highly available (HA) services run on one node and 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 HA services.

This configuration uses the following component names:

- Nodes names:
 - N1
 - N2
- Cluster name: `fileserver850cluster`
- Resource groups:
 - RG1
 - RG2

Table 6-1 SGI File Server 850 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	ordered
Attributes	Auto_Failback, Auto_Recovery	Auto_Failback, Auto_Recovery
Failover domain	N1, N2	N1, N2

Figure 6-2 describes the resource dependencies.

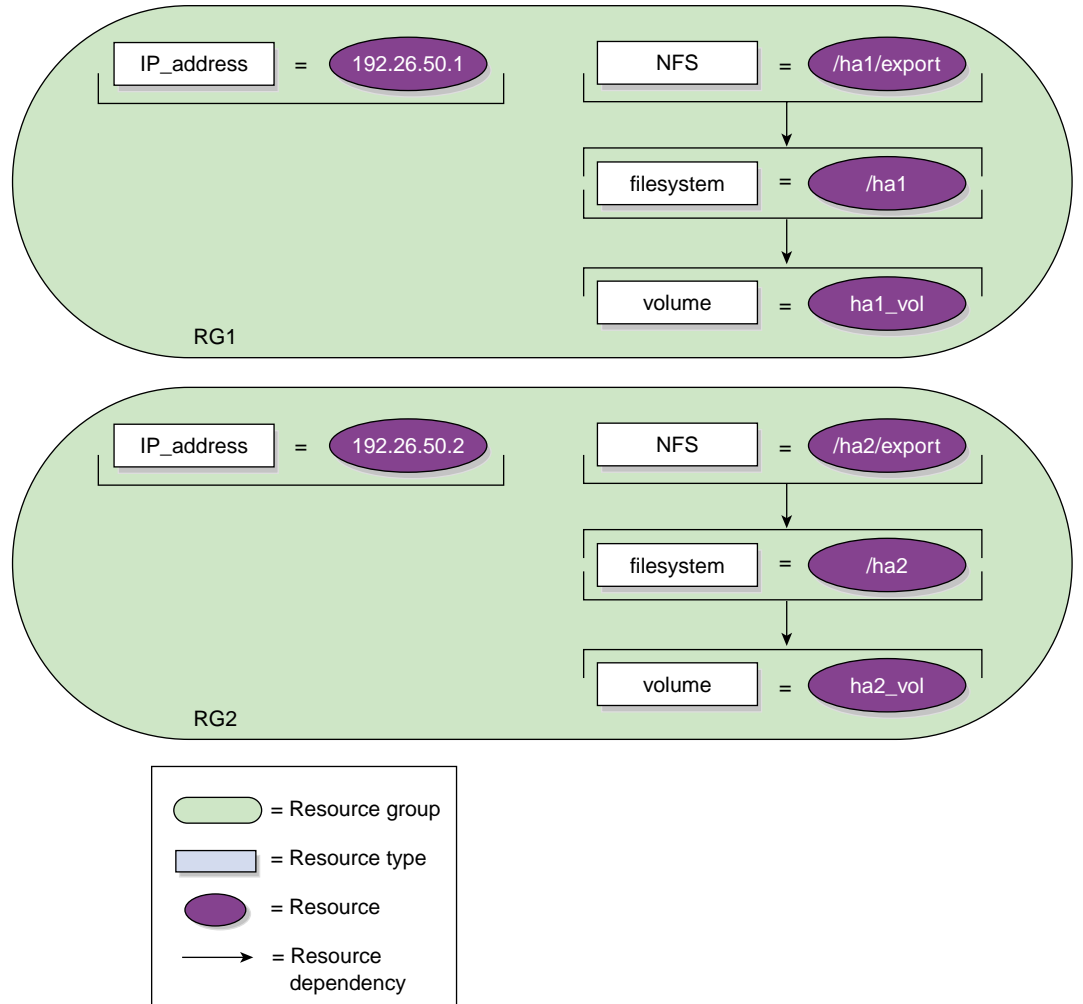


Figure 6-2 Resource Dependencies for SGI File Server 850

The cmgr script used 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 l1
    set sysctrl_status to enabled
    set sysctrl_password to none
    set sysctrl_owner to N2
    set sysctrl_device to /dev/ttyd2
    set sysctrl_owner_type to tty
    add nic ef0-N1
        set heartbeat to true
        set ctrl_msgs to true
        set priority to 1
    done
    add nic ef1-N1
        set heartbeat to true
        set ctrl_msgs to true
        set priority to 2
    done
done

define node N2
    set hostname to N2
    set is_failsafe to true
    set sysctrl_type to l1
    set sysctrl_status to enabled
    set sysctrl_password to none
    set sysctrl_owner to N1
    set sysctrl_device to /dev/ttyd2
    set sysctrl_owner_type to tty
    add nic ef0-N2
        set heartbeat to true
        set ctrl_msgs to true
        set priority to 1
    done
    add nic ef1-N2
        set heartbeat to true
```

```
                set ctrl_msgs to true
                set priority to 2
            done
done

define cluster fileserver850cluster
    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
done

define failover_policy fp1
    set attribute to Auto_Failback
    set attribute to Auto_Recovery
    set script to ordered
    set domain to N1 N2
done

define failover_policy fp2
    set attribute to Auto_Failback
    set attribute to Auto_Recovery
    set script to ordered
    set domain to N2 N1
done

set cluster fileserver850cluster

define resource 192.26.50.1 of resource_type IP_address
    set NetworkMask to 0xffffffff
    set interfaces to ef0,ef1
    set BroadcastAddress to 192.26.50.255
done

define resource hal_vol of resource_type volume
    set devname-owner to root
    set devname-group to sys
    set devname-mode to 600
```

```
done

define resource /hal of resource_type filesystem
    set volume-name to hal_vol
    set mount-options to rw,noauto
    set monitor-level to 2
done

modify resource /hal of resource_type filesystem
    add dependency hal_vol of type volume
done

define resource /hal/export of resource_type NFS
    set export-info to rw,wsync
    set filesystem to /hal
done

modify resource /hal/export of resource_type NFS
    add dependency /hal of type filesystem
done

define resource_group RG1
    set failover_policy to fp1
    add resource 192.26.50.1 of resource_type IP_address
    add resource hal_vol of resource_type volume
    add resource /hal of resource_type filesystem
    add resource /hal/export of resource_type NFS
done

define resource 192.26.50.2 of resource_type IP_address
    set NetworkMask to 0xffffffff
    set interfaces to ef0
    set BroadcastAddress to 192.26.50.255
done

define resource ha2_vol of resource_type volume
    set devname-owner to root
    set devname-group to sys
    set devname-mode to 600
done
```

```
define resource /ha2 of resource_type filesystem
    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
    add dependency ha2_vol of type volume
done

define resource /ha2/export of resource_type NFS
    set export-info to rw,wsync
    set filesystem to /ha2
done

modify resource /ha2/export of resource_type NFS
    add dependency /ha2 of type filesystem
done

define resource_group RG2
    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: Script to Define an SGI SAN Server 1000 Cluster

The script in this section configures an SGI SAN Server 1000 cluster, which has two Origin 300 nodes in a FailSafe configuration, shown in Figure 6-3. Both Origin 300 nodes are attached to a SGI TP9400 RAID using a Brocade Fibre Channel switch. Both nodes act as CXFS metadata servers. There can be one or more CXFS metadata clients.

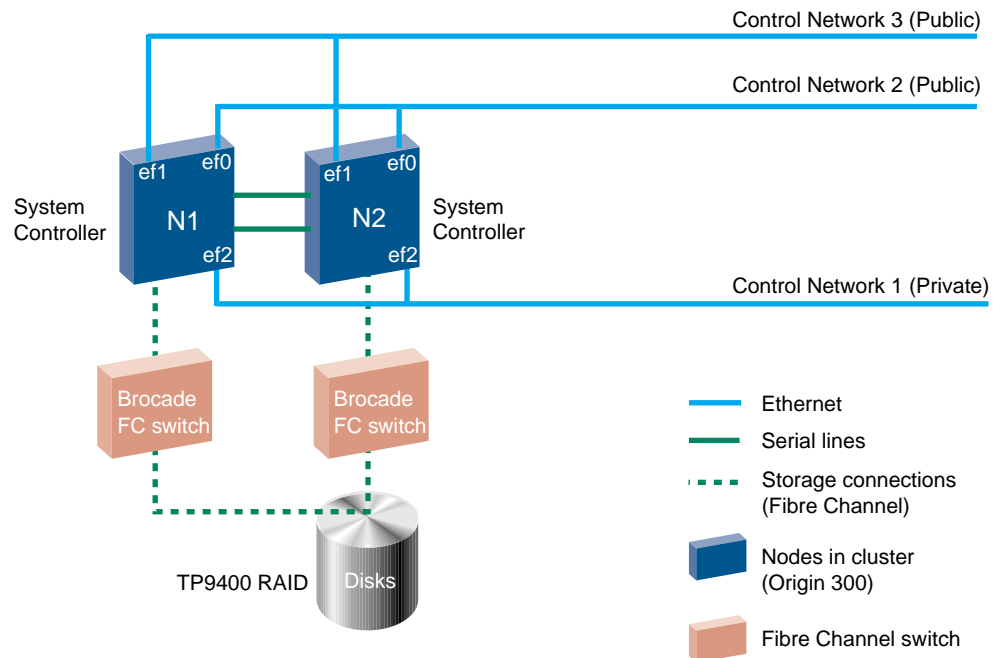


Figure 6-3 SGI SAN Server Configuration

All HA services run on one node and 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 HA services.

The resource groups contain a CXFS filesystem, NFS export point, and an IP address. The /ha_cxfs1 and /ha_cxfs2 CXFS filesystems use metadata servers named N1 and N2.

This configuration uses the following component names:

- Node names:
 - N1
 - N2
- Cluster name: sanserver1000cluster
- Resource groups:
 - RG1
 - RG2

Table 6-2 SGI SAN Server 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
CXFS	/ha_cxfs1	/ha_cxfs2
NFS	/ha_cxfs1/export	/ha_cxfs2/export
Failover policy		
Name	fp1	fp2
Script	ordered	ordered
Attributes	Auto_Failback, Auto_Recovery	Auto_Failback, Auto_Recovery
Failover domain	N1, N2	N1, N2

Figure 6-4 describes the resource dependencies.

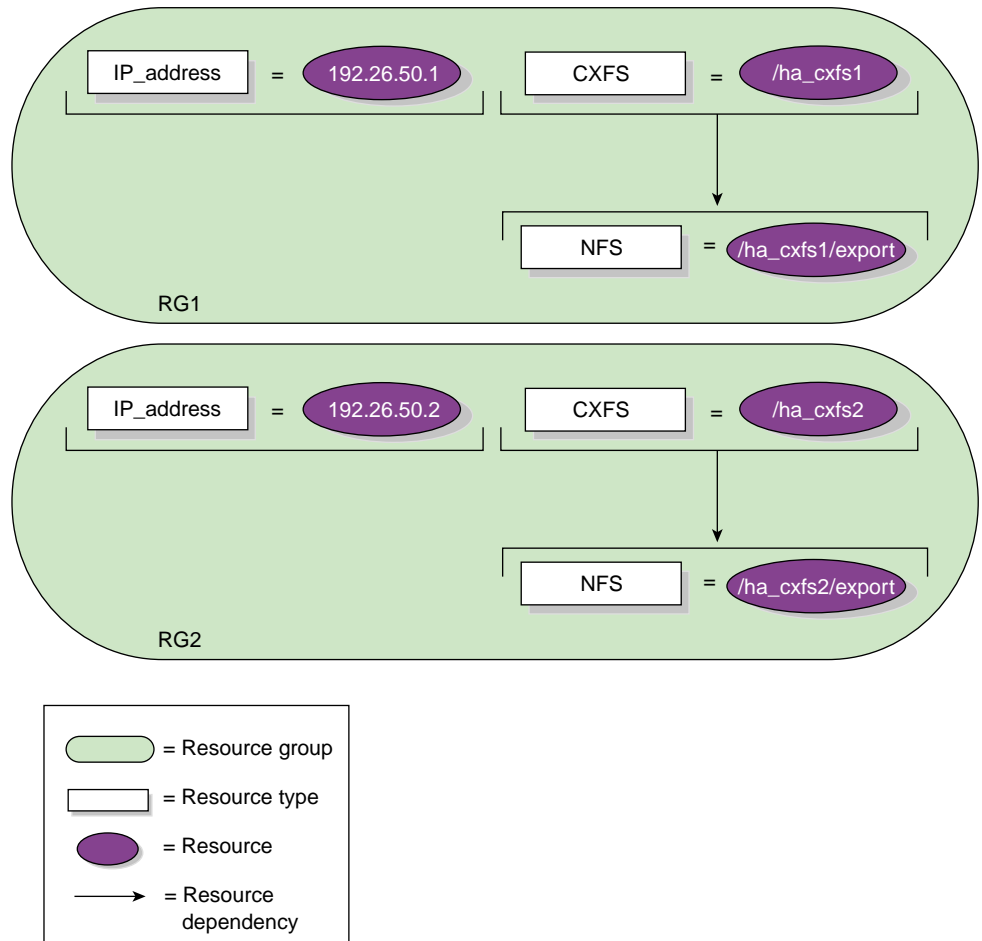


Figure 6-4 Resource Dependencies for SGI SAN Server

The cmgr script used 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 is_cxfs to true
```

```
set sysctrl_type to l1
set sysctrl_status to enabled
set sysctrl_password to none
set sysctrl_owner to N2
set sysctrl_device to /dev/ttyd2
set sysctrl_owner_type to tty
set operating_system to IRIX
set weight to 1
set hierarchy to Reset

add nic ef0-N1
    set heartbeat to true
    set ctrl_msgs to true
    set priority to 1
done
add nic ef1-N1
    set heartbeat to true
    set ctrl_msgs to true
    set priority to 2
done
done

define node N2
    set hostname to N2
    set is_failsafe to true
    set is_cxfs to true
    set sysctrl_type to l1
    set sysctrl_status to enabled
    set sysctrl_password to none
    set sysctrl_owner to N1
    set sysctrl_device to /dev/ttyd2
    set sysctrl_owner_type to tty
    set operating_system to IRIX
    set weight to 1
    set hierarchy to Reset

    add nic ef0-N2
        set heartbeat to true
        set ctrl_msgs to true
        set priority to 1
```

```
done
add nic efl-N2
    set heartbeat to true
    set ctrl_msgs to true
    set priority to 2
done
done

define cluster sanserver1000cluster
    set is_failsafe to true
    set is_cxfs to true
    set clusterid to 1
    set notify_cmd to /usr/bin/mail
    set notify_addr to sanserver1000_sysadm@company.com
    add node N1
    add node N2
done

define failover_policy fp1
    set attribute to Auto_Failback
    set attribute to Auto_Recovery
    set script to ordered
    set domain to N1 N2
done

define failover_policy fp2
    set attribute to Auto_Failback
    set attribute to Auto_Recovery
    set script to ordered
    set domain to N2 N1
done

set cluster sanserver1000cluster

define resource 192.26.50.1 of resource_type IP_address
    set NetworkMask to 0xffffffff
    set interfaces to ef0,ef1
    set BroadcastAddress to 192.26.50.255
done
```

```
define resource /ha_cxfs1 of resource_type CXFS
    set relocate-mds to true
done

define resource /ha_cxfs1/export of resource_type NFS
    set export-info to rw,wsync
    set filesystem to /ha_cxfs1
done

modify resource /ha_cxfs1/export of resource_type NFS
    add dependency /ha_cxfs1 of type CXFS
done

define resource_group RG1
    set failover_policy to fp1
    add resource 192.26.50.1 of resource_type IP_address
    add resource /ha_cxfs1 of resource_type CXFS
    add resource /ha_cxfs1/export of resource_type NFS
done

define resource 192.26.50.2 of resource_type IP_address
    set NetworkMask to 0xffffffff
    set interfaces to ef0
    set BroadcastAddress to 192.26.50.255
done

define resource /ha_cxfs2 of resource_type CXFS
    set relocate-mds to true
done

define resource /ha_cxfs2/export of resource_type NFS
    set export-info to rw,wsync
    set filesystem to /ha_cxfs2
done

modify resource /ha_cxfs2/export of resource_type NFS
    add dependency /ha_cxfs2 of type CXFS
done
```

```
define resource_group RG2
    set failover_policy to fp2
    add resource 192.26.50.2 of resource_type IP_address
    add resource /ha_cxfs2 of resource_type CXFS
    add resource /ha_cxfs2/export of resource_type NFS
done

quit
```

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 6-5. 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 Etherlite network-based serial port multiplexer, which is also connected to the private control network.

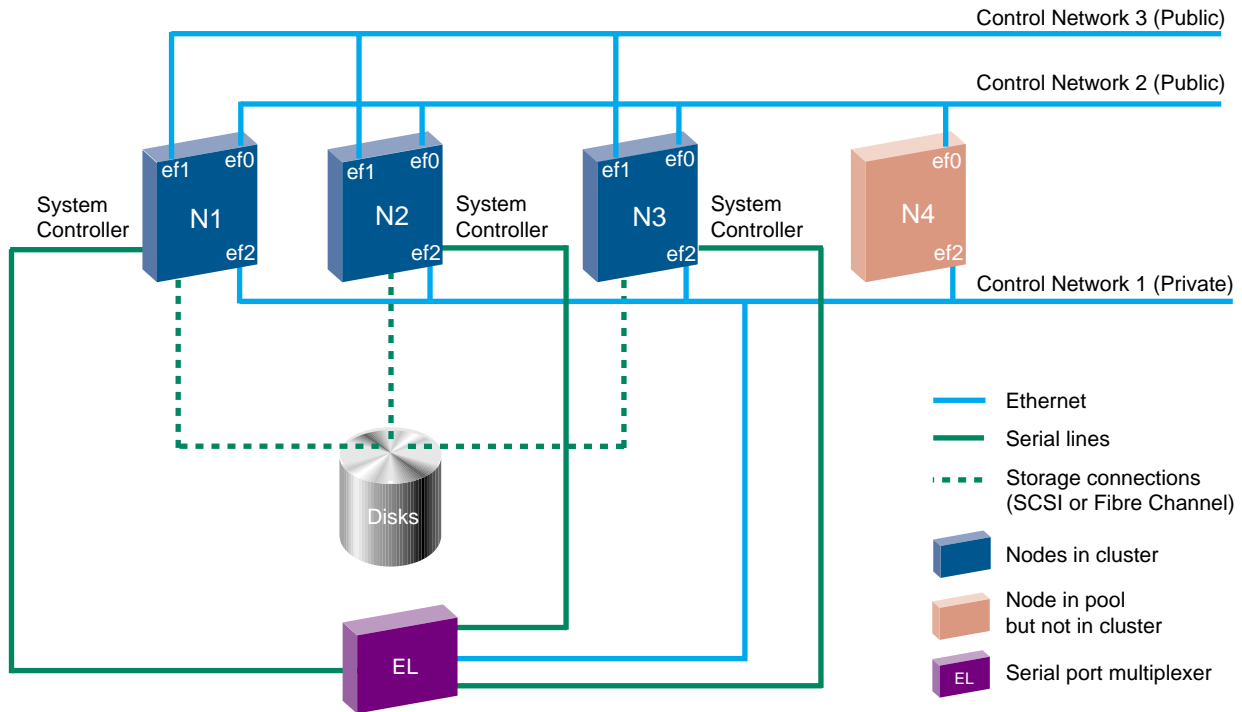


Figure 6-5 FailSafe Configuration Example

For general information on `cmgr` scripts see "Using Script Files", page 102. For information on the template files that you can use to create your own configuration script, see "Template Scripts", page 107.

This cluster has two resource groups, RG1 and RG2, as shown in Table 6-3.

Table 6-3 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

Resources and Failover Policy	RG1	RG2
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 used 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 0xffffffff
    set interfaces to ef0,ef1
    set BroadcastAddress to 192.26.50.255
done

define resource hal_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 /hal of resource_type filesystem in cluster TEST
    set volume-name to hal_vol
    set mount-options to rw,noauto
    set monitor-level to 2
done

modify resource /hal of resource_type filesystem in cluster TEST
    add dependency hal_vol of type volume
done

define resource /hal/export of resource_type NFS in cluster TEST
    set export-info to rw,wsync
    set filesystem to /hal
done

modify resource /hal/export of resource_type NFS in cluster TEST
    add dependency /hal 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 0xffffffff
    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 an 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 177.

The following example configures local failover of an HA IP address. It uses the configuration illustrated in Figure 6-5, page 236.

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 0xffffffff
    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 6-5, page 236, so that it includes HA 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 the *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 clustered XVM volumes. Clustered XVM volumes should not be added to the FailSafe resource groups. (Local XVM volumes use the XVM resource type.)

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 the *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 (if any) of the FailSafe NFS 2.2 release are installed on all FailSafe nodes in the cluster.
2. Perform all of the steps mentioned in "Example: Modify a Cluster to Include a `CXFS` Filesystem", page 243.

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 example 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 havoll 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` command.

FailSafe System Operation

This chapter describes administrative tasks you perform to operate and monitor an FailSafe system. It describes how to perform tasks using the FailSafe Manager GUI and the `cmgr` command. The major sections in this chapter are as follows:

- "Origin 300, Origin 3200C, Onyx 300, and Onyx 3200C Console Support"
- "Interrupting FailSafe Commands", page 248
- "Two-Node Clusters: Single-Node Use", page 249
- "System Status", page 255
- "Embedded Support Partner (ESP) Logging of FailSafe Events", page 271
- "Resource Group Failover", page 273
- "Stopping FailSafe", page 279
- "Resetting Nodes", page 279
- "Cluster Database Backup and Restore", page 281
- "Log File Management", page 285
- "Granting Task Execution Privileges to Users", page 286

Note: SGI recommends that you perform all FailSafe administration 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 the console when IRIX is not running (miniroot), you must physically reconnect the machine: unplug the serial hardware reset cable from the console/L1 port and then connect the console cable.

Interrupting FailSafe Commands

After 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 in "System Status") 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 *ActiveNode* is the name of the node using the policy (in the examples, `nodeA`) and *DownNode* is the name of the nonfunctioning node (in the examples, `nodeB`):

```
cmgr> show failover_policy emergency-ActiveNode
```

```
Failover Policy: emergency-ActiveNode
Version: 1
Script: ordered
Attributes: Controlled_Failback InPlace_Recovery
Initial AFD: ActiveNode
```

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 190.

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 RGname in cluster Clustername
set failover_policy to emergency-ActiveNode
```

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 because the other node is not functional. This step tells the database to label the state of the resource groups appropriately in preparation for later steps.)

Use the following command:

```
admin offline_force resource_group RGname in cluster Clustername
```

For example:

```
cmgr> set cluster test-cluster
cmgr> show resource_groups in test-cluster
```

```
Resource Groups:
    group1
    group2
```

```
cmgr> admin offline_force resource_group group1
cmgr> admin offline_force resource_group group2
```

4. Force the stop of HA services for the down node:

```
stop ha_services on DownNode for cluster Clustername force
```

Note: This is a long running task that and might take a few minutes to complete. The cmgr command 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 RGname 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 ActiveNode 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 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:DownNode cmond[1074]: Notification can not be processed, local machine and cluster name is not known.
```

```
Dec 14 18:23:16 6D:DownNode cmond[1074]: Local machine belongs to cluster Clustername.
```

```
Dec 14 18:23:16 6D:DownNode cmond[1074]: Local machine name is DownNode"
```

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 command to set all cluster services to off:

```
# chkconfig | grep cluster
```

3. Boot the *DownNode* to multiuser mode.
4. Start HA services for the down node:

```
cmgr> start ha_services on node DownNode for cluster Clustername
```

5. Use the `tail` command to watch the `SYSLOG` file for messages that indicate the formerly down node is now in FailSafe membership and that services are in the UP state. Do not continue until you see messages such as the following in this example:

```
# tail -f /var/adm/SYSLOG
...
blue node ActiveNode [81] : UP incarnation 7 age 2:0
blue node DownNode [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 RGname in cluster Clustername
set failover_policy to OriginalFailoverPolicy
```

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: Doing an `offline_detach` operation on a resource group leaves the resources in the resource group running on the node. FailSafe does not track these resources any longer. You should take care should to prevent resources from running on multiple nodes in the cluster.

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 (that is, 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, and C are in UP state and node D is DOWN state. If resource group RG is made offline using the `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 RGname [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 *RGname* 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 RGname in cluster Clustername
```

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 *RGname* in cluster *Clustername* to node *PrimaryOwner*

For example:

```
cmgr> admin move resource_group group1 in cluster test-cluster to node nodeB
```

If you run into errors after entering the `admin move` command, see "Ensuring that Resource Groups are Deallocated", page 303.

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:

- Keep continuous watch on the state of a cluster using the `cluster_status` command or the GUI.

- Query the status of an individual resource group, node, or cluster using either the GUI or the `cmgr` command.
- 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 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      Online   Offline    No error
    footest-group      Online   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 view area.

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. Green indicates FailSafe HA services active without an error, gray indicates FailSafe HA services are inactive, and red indicates an error state.

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 is as follows:

Table 7-1 Key to Icons

Icon	Entity
	IRIX Node
	Cluster
	Resource
	Resource group
	Resource type
	Failover policy












Icon	Entity
	Expanded tree
	Collapsed tree
	User name
	GUI task for which execution privilege may be granted or revoked
	Privileged command executed by a given GUI task

Table 7-2 Key to States

Icon	State
	Inactive or unknown (HA services may not be active)
	Online-ready state for a resource group

Icon	State
	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 on 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 Resourcename of resource_type RTname [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` operation on a system controller by providing the device name, use the following command:

```
admin ping dev_name devicename of dev_type Devicetype with sysctrl_type SystemControllerType
```

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:

<code>ONLINE</code>	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 an error state.
<code>ONLINE-PENDING</code>	FailSafe is running on the local nodes and the resource group is in the process of being allocated. This is a transient state.
<code>OFFLINE</code>	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	An internal error occurred; notify SGI if this condition arises.

NODE UNKNOWN	Node that had the resource group online is in an 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 communicate with 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 FailSafe 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 view area 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 RGname [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 node 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 view area 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 of 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_unlimited)
        150.166.41.95 (type: IP_address)
        /hafs1 (type: filesystem)
        havoll (type: volume)
```

```
# haStatus -i
Tue Nov 30 14:13:52 PST 1999
Cluster test-cluster:
Node hans2:
    Logical Machine Name: hans2
    Hostname: hans2.dept.company.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.dept.company.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_unlimited)
150.166.41.95 (type: IP_address)
/hafs1 (type: filesystem)
havoll (type: volume)
Resource /hafs1 (type NFS):
export-info: rw,wsync
filesystem: /hafs1
Resource dependencies
statd_unlimited /hafs1/nfs/statmon
```

```
filesystem /hafs1
Resource /hafs1/nfs/statmon (type statd_unlimited):
  InterfaceAddress: 150.166.41.95
  Resource dependencies
  IP_address 150.166.41.95
  filesystem /hafs1
Resource 150.166.41.95 (type IP_address):
  NetworkMask: 0xffffffff00
  interfaces: efl
  BroadcastAddress: 150.166.41.255
  No resource dependencies
Resource /hafs1 (type filesystem):
  volume-name: havoll
  mount-options: rw,noauto
  monitor-level: 2
  Resource dependencies
  volume havoll
Resource havoll (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.dept.company.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.dept.company.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_unlimited)
    150.166.41.95 (type: IP_address)
    /hafs1 (type: filesystem)
    havoll (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_unlimited /hafs1/nfs/statmon
    filesystem /hafs1
Resource /hafs1/nfs/statmon (type statd_unlimited):
    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: 0xffffffff0
    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: havoll
    mount-options: rw,noauto
```



```

        monitor-level: 2
        Resource dependencies
        volume havoll
Resource havoll (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_unlimited)
    150.166.41.95 (type: IP_address)
    /hafs1 (type: filesystem)
    havoll (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:

```
# espsconfig -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` or `launchESPartner` 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 274.



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 274.

Do the following:

1. **Group to Bring Online:** 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 RGname [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 278.

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. **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.
5. 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 RGname [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 RGname [in cluster Clustername]
```

Note: Doing an `offline_force` operation on a resource group can leave resources in the resource group running on the cluster. The `offline_force` operation will succeed even though all resources in the resource group have not been stopped. FailSafe does not track these resources any longer. You should take care to prevent resources from running on multiple nodes in the cluster.

To detach a resource group, use the following command:

```
admin offline_detach resource_group RGname [in cluster Clustername]
```

To detach the resource group and force the error state to be cleared:

```
admin offline_detach_force resource_group RGname [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.

Enter 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 RGname [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.

If you run into errors after entering the `admin move` command, see "Ensuring that Resource Groups are Deallocated", page 303.

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 a resource group.

Once monitoring is resumed and assuming that the restart action is enabled, if the resource group or one of its resources fails, FailSafe will restart each failed component based on the failover policy.

Perform the following steps:

1. **Group to Start Monitoring:** select the name of the group you want to start 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 RGname [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 RGname [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 208.

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:** 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.

Cluster Database Backup and Restore

You should perform a database backup whenever you want to save the database and be able to restore it to the current state at a later point.

You can use the following methods to restore the database:

- If the database is accidentally deleted from a node, use the `fs2d` daemon to replicate the database from another node in the pool.
- If you want to be able to recreate the current configuration, use the `build_cmgr_script` script. You can then recreate this configuration by running the script generated.
- If you want to retain a copy of the database and all node-specific information such as local logging, use the `cdbBackup` and `cdbRestore` commands.

Restoring the Database from Another Node

If the database has been accidentally deleted from an individual node, you can replace it with a copy from another node. Do not use this method if the cluster database has been corrupted.

Do the following:

1. Stop the cluster database daemons by running the following command on each node:

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

2. Run `cdbreinit` on nodes that are missing the cluster database.
3. Restart the cluster database daemons by running the following command on each node:

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

The `fs2d` daemon will then replicate the cluster database to those nodes from which it is missing

Using `build_cmgr_script` for the Cluster Database

You can use the `build_cmgr_script` command from one node in the cluster to create a `cmgr` script that will recreate the node, cluster, switch, and filesystem

definitions for all nodes in the cluster database. You can then later run the resulting script to recreate a database with the same contents; this method can be used for missing or corrupted cluster databases.

Note: The `build_cmgr_script` script does not contain local logging information, so it cannot be used as a complete backup/restore tool.

To perform a database backup, use the `build_cmgr_script` script from one node in the cluster, as described in "Creating a `cmgr` Script Automatically", page 104.



Caution: Do not make configuration changes while you are using the `build_cmgr_script` command.

By default, this creates a `cmgr` script in the following location:

```
/tmp/cmgr_create_cluster_clustername_processID
```

You can specify another filename by using the `-o` option.

To perform a restore on all nodes in the pool, do the following:

1. Stop HA services for all nodes in the cluster.
 2. Stop the cluster database daemons on each node.
 3. Remove all copies of the old database by using the `cdbreinit` command on each node.
 4. Execute the `cmgr` script (which was generated by the `build_cmgr_script` script) on the node that is defined first in the script. This will recreate the backed-up database on each node.
-

Note: If you want to run the generated script on a different node, you must modify the generated script so that the node is the first one listed in the script.

5. Restart cluster database daemons on each node.

For example, to backup the current database, clear the database, and restore the database to all nodes, do the following:

On one node in the cluster:

```
# /var/cluster/cmgr-scripts/build_cmgr_script -o /tmp/newcdb
Building cmgr script for cluster clusterA ...
build_cmgr_script: Generated cmgr script is /tmp/newcdb
```

On one node:

```
# stop ha_services for cluster clusterA
```

On each node:

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

On each node:

```
# /usr/cluster/bin/cdbreinit
On each node:
# /etc/init.d/cluster start
```

*On the *first* node listed in the /tmp/newcdb script:*

```
# /tmp/newcdb
```

Using cdbBackup and cdbRestore for the Cluster Database and Logging Information

The `cdbBackup` and `cdbRestore` commands backup and restore the cluster database and node-specific information, such as local logging information. You must run these commands individually for each node.

To perform a backup of the cluster, use the `cdbBackup` command on each node.



Caution: Do not make configuration changes while you are using the `cdbBackup` command.

To perform a restore, run the `cdbRestore` command on each node. You can use this method for either a missing or corrupted cluster database. Do the following:

1. Stop HA services.
2. Stop cluster services on each node.
3. Remove the old database by using the `cdbreinit` command on each node.

4. Stop cluster services again (these were restarted automatically by `cdbreinit` in the previous step) on each node.
5. Use the `cdbRestore` command on each node.
6. Start cluster services on each node.

For example, to backup the current database, clear the database, and then restore the database to all nodes, do the following:

On each node:

```
# /usr/cluster/bin/cdbBackup
```

On one node in the cluster:

```
# stop ha_services for cluster clusterA
```

On each node:

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

On each node:

```
# /usr/cluster/bin/cdbreinit
```

On each node (again):

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

On each node:

```
# /usr/cluster/bin/cdbRestore
```

On each node:

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

For more information, see the `cdbBackup(1M)` and `cdbRestore(1M)` man page.

Note: Do 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.

For information about log levels, see "Set Log Configuration", page 213.

Rotating All Log Files

You can run the `/var/cluster/cmgr-scripts/rotatelogs` script to copy all files to a new location. This script saves log files with the day and the month name as a suffix. If you run the script twice in one day, it will append the current log file to the previous saved copy. The `root crontab` file has an entry to run this script weekly.

The script syntax is as follows:

```
/var/cluster/cmgr-scripts/rotatelogs [-h] [-d|-u]
```

If no option is specified, the log files will be rotated. Options are as follows:

- h Prints the help message. The log files are not rotated and other options are ignored.
- d Deletes saved log files that are older than one week before rotating the current log files. You cannot specify this option and `-u`.
- u Unconditionally deletes all saved log files before rotating the current log files. You cannot specify this option and `-d`.

By default, the `rotatelogs` script will be run by `crontab` once a week, which is sufficient if you use the default log levels. If you plan to run with a high debug level for several weeks, you should reset the `crontab` entry so that the `rotatelogs` script is run more often.

On heavily loaded machines, or for very large log files, you may want to move resource groups and stop HA services before running `rotatelogs`.

Rotating Large Log Files

You can use a script such as the following to copy large files to a new location. The files in the new location will be overwritten each time this script is run.

```
#!/bin/sh
# Argument is maximum size of a log file (in characters) - default: 500000

size=${1:-500000}
find /var/cluster/ha/log -type f ! -name '*.OLD' -size +${size}c -print | while read log_file; do
    cp ${log_file} ${log_file}.OLD
    echo '*** LOG FILE ROTATION ' `date` '***' > ${log_file}
done
```

Granting Task Execution Privileges to Users

The GUI lets you grant or revoke access to a specific GUI task for one or more specific users. By default, only `root` may execute tasks in the GUI. You cannot grant or revoke tasks for users with a user ID of 0.

Access to the task is allowed only on the node to which the GUI is connected; if you want to allow access on another node in the pool, you must connect the GUI to that node and grant access again.

GUI tasks and the `cmgr(1M)` command operate by executing underlying privileged commands which are normally accessible only to `root`. When granting access to a task, you are in effect granting access to all of its required underlying commands, which results in also granting access to the other GUI tasks that use the same underlying commands.

To see which tasks a specific user can currently access, select **View: Users**. Select a specific user to see details about the tasks available to that user.

To see which users can currently access a specific task, select **View: Task Privileges**. Select a specific task to see details about the users who can access it and the privileged commands it requires.

Grant Task Access to a User or Users

You can grant access to a specific task to one or more users at a time.

Note: Access to the task is only allowed on the node to which the GUI is connected; if you want to allow access on another node in the pool, you must connect the GUI to that node and grant access again.

Do the following:

1. Select the user or users for whom you want to grant access. You can use the following methods to select users:
 - Click to select one user at a time
 - Shift+click to select a block of users
 - Ctrl+click to toggle the selection of any one user, which allows you to select multiple users that are not contiguous
 - Click **Select All** to select all users

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

2. Select the task or tasks to grant access to, using the above selection methods. Click **Next** to move to the next page.
3. Confirm your choices by clicking **OK**.

Note: If more tasks than you selected are shown, then the selected tasks run the same underlying privileged commands as other tasks, such that access to the tasks you specified cannot be granted without also granting access to these additional tasks.

To see which tasks a specific user can access, select **View: Users**. Select a specific user to see details about the tasks available to that user.

To see which users can access a specific task, select **View: Task Privileges**. Select a specific task to see details about the users who can access it and the privileged commands it requires.

Granting Access to a Few Tasks

Suppose you wanted to grant user `guest` permission to define clusterwide and node-specific resources. You would do the following:

1. Select `guest` and click **Next** to move to the next page.
2. Select the tasks you want `sys` to be able to execute:
 - a. Click **Define a Resource**
 - b. Ctrl+click **Redefine a Resource for a Specific Node**

c. Ctrl+click **Add or Remove Resources in Resource Group**

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

3. Confirm your choices by clicking **OK**.

Figure 7-1 shows the privileged commands that were granted to user *guest*.

Note: Modify a Resource Definition is also displayed, even though the administrator did not explicitly select it; the privilege commands for the tasks selected also require this command.

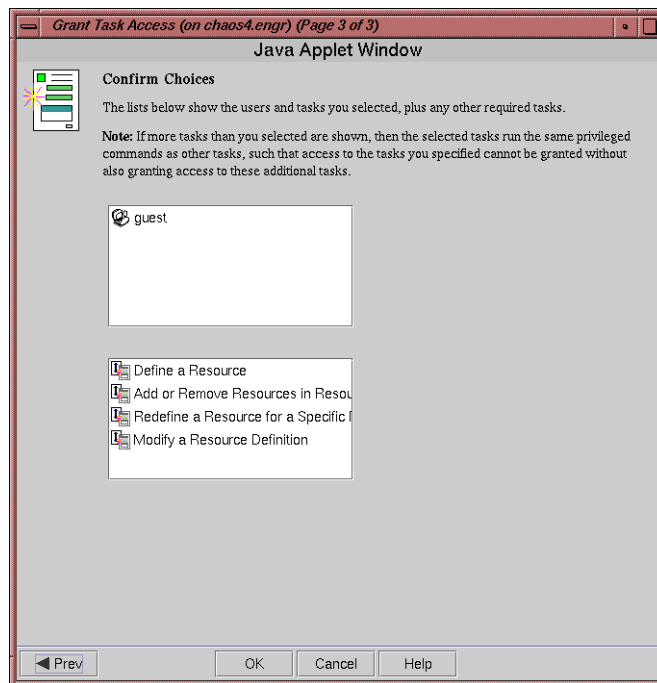


Figure 7-1 Results of Granting a User Privilege

Figure 7-2 shows the screen that is displayed when you select **View: Users** and click *guest* to display information in the details area of the GUI window. The privileged commands listed are the underlying commands executed by the GUI tasks.

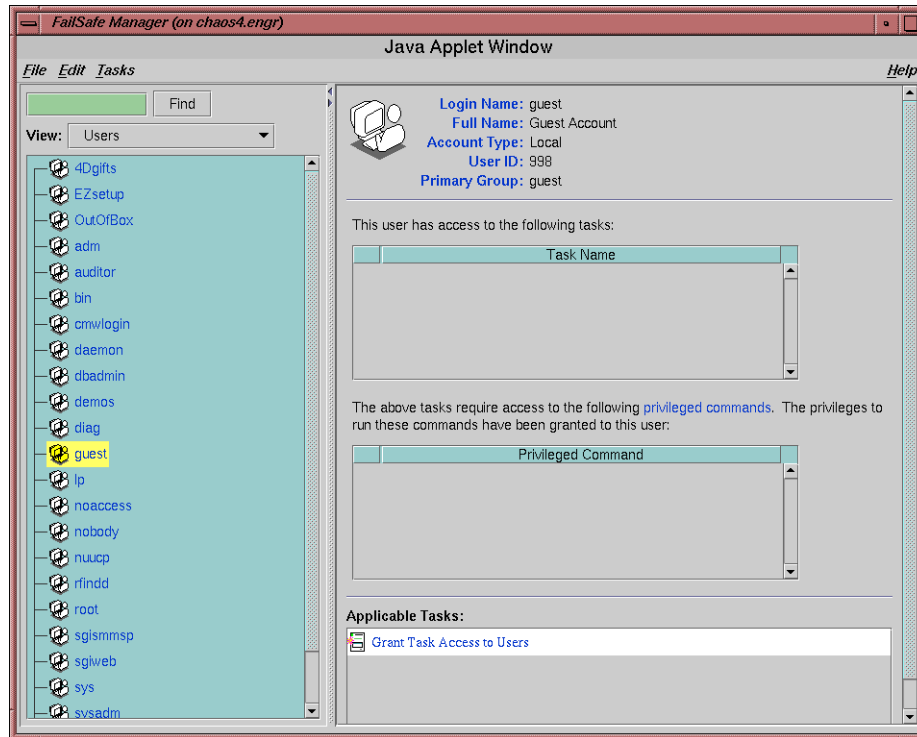


Figure 7-2 Displaying the Privileged Commands a User May Execute

Granting Access to Most Tasks

Suppose you wanted to give user *sys* access to all tasks **except** adding or removing nodes from a cluster. The easiest way to do this is to select all of the tasks and then deselect the one you want to restrict. You would do the following:

1. Select *sys* and click **Next** to move to the next page.
2. Select the tasks you want *sys* to be able to execute:
 - a. Click **Select All** to highlight all tasks.
 - b. Deselect the task to which you want to restrict access. **Ctrl+click Add/Remove Nodes in Cluster.**

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

3. Confirm your choices by clicking **OK**.

Revoke Task Access from a User or Users

You can revoke task access from one or more users at a time.

Note: Access to the task is only revoked on the node to which the GUI is connected; if a user has access to the task on multiple nodes in the pool, you must connect the GUI to those other nodes and revoke access again.

Do the following:

1. Select the user or users from whom you want to revoke task access. You can use the following methods to select users:
 - Click to select one user at a time
 - Shift+click to select a block of users
 - Ctrl+click to toggle the selection of any one user, which allows you to select multiple users that are not contiguous
 - Click **Select All** to select all users

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

2. Select the task or tasks to revoke access to, using the above selection methods. Click **Next** to move to the next page.
 3. Confirm your choices by clicking **OK**.
-

Note: If more tasks than you selected are shown, then the selected tasks run the same underlying privileged commands as other tasks, such that access to the tasks you specified cannot be revoked without also revoking access to these additional tasks.

To see which tasks a specific user can access, select **View: Users**. Select a specific user to see details about the tasks available to that user.

To see which users can access a specific task, select **View: Task Privileges**. Select a specific task to see details about the users who can access it.

Testing the Configuration

This chapter explains how to test the FailSafe system configuration using the FailSafe Manager GUI and the `cmgr` 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 292
- "Performing Diagnostic Tasks with `cmgr`", page 293

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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • Failover policy exists • Failover domain contains a valid list of hosts
Network connectivity	Checks that: <ul style="list-style-type: none"> • 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` access between nodes. To test connectivity, do the following from the **FailSafe Manager**:

Note: The **Test Node Connectivity** screen requires `rsh` 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 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 **Delete**.
- To start the tests, click on **Start Tests**. To stop the tests, click **Stop Tests**.
- To run another test, click **Clear Output** to clear the status screen.
- To exit from the window, click **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` 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 Clustername [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 cml
Success: testing serial...
Success: Ensuring Node Can Get IP Addresses For All Specified Hosts
Success: Number of IP addresses obtained for <cml> = 1
Success:      The first IP address for <cml> = 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 cml
Error: Cannot run the serial tests, diagnostics has detected FailSafe (ha_cmds) 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 Clustername [on node node1 node node2 ...]
```

The following shows an example of the `test connectivity` command:

```
cmgr> test connectivity in cluster eagan on node cml
Success: testing connectivity...
Success: checking that the control IP_addresses are on the same networks
Success: pinging address cml-priv interface ef0 from host cml
Success: pinging address cml interface ef1 from host cml
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 0xffffffff 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 RTname in cluster Clustername [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 RTname in cluster Clustername [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 RTname in cluster Clustername [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 Clustername [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 Clustername [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 cml destructive
Warning: executing the tests in destructive mode
Success: *** testing node resources on node cml ***
Success: *** testing all filesystem resources on node cml ***
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 RGname in cluster Clustername [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 cml
Success: *** testing node resources on node cml ***
Success: testing resource /disk4 of resource type NFS on node cml
Success: testing resource /disk3 of resource type NFS on node cml
Success: testing resource /disk3/statmon of resource type statd_unlimited on node cml
Success: testing resource 128.162.19.45 of resource type IP_address on node cml
Success: testing resource /disk4 of resource type filesystem on node cml
Success: testing resource /disk3 of resource type filesystem on node cml
Success: testing resource dmfl of resource type volume on node cml
Success: testing resource dmfjournals of resource type volume on node cml
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 FPname in cluster Clustername [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 302
- "FailSafe Log Files", page 303
- "FailSafe Membership and Resets", page 304
- "Status Monitoring", page 306
- "XVM Alternate Path Failover", page 307
- "Dynamic Control of FailSafe HA Services", page 308
- "Recovery Procedures", page 309
- "CXFS Metadata Server Relocation", page 321
- "Other Problems with CXFS Coexecution", page 321

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 HA 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 311 , and "Resource Group Maintenance and Error Recovery", page 312.
2. Perform the needed maintenance.
3. Reboot the node.
4. Online the resource group.

Ensuring that Resource Groups are Deallocated

Performing an `admin offline_force` does not guarantee that all resource groups are offline. If you run into errors, such as with an `admin move` command, you should verify that the resource groups have been deallocated.

Checking for Exclusivity

After performing an `admin offline_force`, you should run the `exclusive` script with the appropriate arguments to verify that the resource in question is not running, or perform a check similar to that done by the script.

Stopping Resources Manually

You must stop resources according to their execution order, from highest to lowest. Use the `exclusive` scripts to verify whether or not a resource needs to be stopped.

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 213 in Chapter 5, "Configuration", page 109.

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)

Message Level	Description
Error/Warning	<p>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 indicates a warning. <E indicates an error.):</p> <pre>Wed Sep 2 13:45:47.199 <W crsd crs 9908:0 crs_config.c:634> CI_ERR_NOTFOUND, safer - no such node</pre>
SYSLOG	<p>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:</p> <pre>Wed Sep 2 12:22:57 6X:safe syslog: <<CI> ha_cmds misc 10435:0> CI_FAILURE, I am not part of the enabled cluster anymore</pre>
Debug	<p>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:</p> <pre>Thu Sep 27 14:43:24.233 <D0 ha_fsd fsd 57540:0 fs_failsafe.c:1471> Determine oldest state: coordinator: perf22/0x10001</pre>

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.

See Appendix F, "System Messages", page 403.

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 tiebreaker node when you configured your FailSafe parameters. If no tiebreaker 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 tiebreaker node.

If a tied subset of nodes in the cluster had no previous membership, then the subset of nodes in the cluster with the tiebreaker 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 tiebreaker 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` 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 view area. 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 255 of Chapter 7, "FailSafe System Operation".

XVM Alternate Path Failover

Messages in the `/var/adm/SYSLOG` file indicate that XVM has detected a failure in the disk path and has successfully used alternate path failover for `lun0`; that is, the cluster is in a degraded state and requires attention from the system administrator. For example (line breaks added for readability):

```
Jun 26 13:21:55 5A:gold2 unix: NOTICE: xvm_serverpal_iodone: done with retry ior 0xa800000041afe100
for physvol 0xa8000000008eae00
Jun 26 13:21:55 6A:gold2 unix: dksc 50050cc002004b23/lun0vol/c4p3: <6>SCSI driver error: device
does not respond to selection
Jun 26 13:21:55 6A:gold2 unix: dksc 50050cc002004b23/lun0vol/c4p3: <6>SCSI driver error: device
does not respond to selection
Jun 26 13:21:55 4A:gold2 unix: WARNING: XVM: WRITE I/O error - errno 5, dev 0x134, bp 0xa800000059bcdf80,
b_flags 0x100400c, b_addr 0x0, b_pages 0xa800000202127600, io_resid -4611686018427387904, io_error
0xa800000000000000
Jun 26 13:21:55 6A:gold2 unix: dksc 50050cc002004b23/lun0vol/c4p3: <6>SCSI driver error: device does
not respond to selection
Jun 26 13:21:55 4A:gold2 unix: WARNING: XVM: WRITE I/O error - errno 5, dev 0x134, bp 0xa80000024bef9980,
b_flags 0x100400c, b_addr 0x0, b_pages 0xa8000000025aa040, io_resid -4611686018427387904, io_error
0xa800000200000000
Jun 26 13:21:55 4A:gold2 unix: WARNING: XVM: WRITE I/O error - errno 5, dev 0x134, bp 0xa800000011037600,
b_flags 0x100400c, b_addr 0x0, b_pages 0xa8000000024fc3c0, io_resid -6341068275337592832, io_error
0xa800000000000000
Jun 26 13:21:55 4A:gold2 unix: WARNING: XVM: WRITE I/O error - errno 5, dev 0x134, bp 0xa800000259171680,
b_flags 0x100400c, b_addr 0x0, b_pages 0xa8000000024e5a80, io_resid 0, io_error 0x0
Jun 26 13:21:55 4A:gold2 unix: WARNING: XVM: failover successful. Failover from dev 0x134 to dev 0x156
(/hw/module/001c01/Ibrick/xtalk/14/pci/1/scsi_ctlr/0/node/50050cc002004b23/port/2/lun/0/disk/volume/block)
physvol 0x1b7
Jun 26 13:21:55 5A:gold2 unix: NOTICE: xvm_serverpal_iodone: done with retry ior 0xa800000041afe100 for
physvol 0xa8000000008eae00
```

These messages in the SYSLOG file would be produced by the following /etc/failover.conf file:

```
#ident $Revision: 1.22 $
#
#   This is the configuration file for table configured failover support.
#
#   Please see the failover (7m) manual page for details on failover and
#   on how to use this file.
#
#disable_target_lun_check
lun2    50050cc002004b23/lun2/c4p3 \
        50050cc002004b23/lun2/c3p2 \
        50050cc002004b23/lun2/c3p3 \
        50050cc002004b23/lun2/c4p2
lun1    50050cc002004b23/lun1/c4p3 \
        50050cc002004b23/lun1/c3p2 \
        50050cc002004b23/lun1/c3p3 \
        50050cc002004b23/lun1/c4p2
lun0    50050cc002004b23/lun0/c4p3 \
        50050cc002004b23/lun0/c3p2 \
        50050cc002004b23/lun0/c3p3 \
        50050cc002004b23/lun0/c4p2
```

Dynamic Control of FailSafe HA 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 HA 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 HA 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 109, and Chapter 7, "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"
- "Cluster Error Recovery", page 310
- "Resource Group Recovery", page 311
- "Node Error Recovery", page 311
- "Resource Group Maintenance and Error Recovery", page 312
- "Clear Resource Error State", page 315
- "Control Network Failure Recovery", page 316
- "Serial Cable Failure Recovery", page 316
- "Cluster Database Sync Failure", page 317
- "Cluster Database Maintenance and Recovery", page 317
- "GUI Will Not Run", page 318
- "GUI and cmgr Inconsistencies", page 319
- "GUI Does Not Report Information", page 319

- "Using the `cdbreinit` Command", page 320
- "Action Script Configuration Errors", page 320

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 249.

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 316).
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 311

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 RGname [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. Verify that the control networks in the node are working (see "Control Network Failure Recovery", page 316).

2. Verify that the serial reset cables to reset the node are working (see "Serial Cable Failure Recovery", page 316).
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 261, 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:RGname]:Exclusivity failed -- RUNNING on Node1 and Node2
```

```
[Resource Group:RGname]: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 274, 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_Nodename`. Follow this procedure when the control network fails:

1. Use the `ping` 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. You should use IP addresses instead of IP names.
4. Check whether the heartbeat interval and node timeouts are correctly set for the cluster. These HA parameters can be 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 `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 "Cluster Database Backup and Restore", page 281.

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 `fsmgr` 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 `cmnd` 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` 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 few 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 HA 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 few minutes for the cluster database to synchronize. There will be cluster database sync long messages in the `SYSLOG` on the node.
6. Start FailSafe HA services in the cluster.

Action Script Configuration Errors

If you try to execute an action script that is missing or does not have the correct permissions, you will get an error message (see "ha_srmd Error Message", page 409.) After fixing the problem, you must send a `SIGHUP` signal to the `ha_srmd` process on each node so that it rereads the configuration. Use the following command line on each node:

```
# killall -HUP ha_srmd
```

When `ha_srmd` receives the `SIGHUP` signal, it will reread the resource type configuration. If `ha_srmd` finds errors in the resource type configuration, the errors will be logged in the `SYSLOG` or `ha_srmd` logs.

CXFS Metadata Server Relocation

FailSafe uses a `umount` 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 a 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:

- "Adding a Node to an Active Cluster"
- "Deleting a Node from an Active Cluster", page 325
- "Changing Control Networks in a Cluster", page 327
- "Upgrading OS Software in an Active Cluster", page 329
- "Upgrading FailSafe Software in an Active Cluster", page 330
- "Adding New Resource Groups or Resources in an Active Cluster", page 331
- "Adding a New Hardware Device in an Active Cluster", page 332

Adding 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` 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 291.
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      # Cluster admin daemon
sgi-crsd         7500/udp      # Cluster reset services daemon
sgi-cmsd         7000/udp      # FailSafe membership Daemon
sgi-gcd          8000/udp      # 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. For example:

```
# /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 SGI 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 SGI 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 SGI 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 template script (`/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 or 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, you must redefine the entire failover policy to include the additional node in the failure domain.

Deleting a Node from an Active Cluster

Use the following procedure to delete a node from an active cluster. This procedure 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 that 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:

```
# /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:

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

Changing Control Networks in a Cluster

Use the following procedure to change the control networks in a currently active cluster. This procedure is valid for a two-node cluster consisting of nodes `node1` and `node2`. In this procedure, you must complete each step before proceeding to the next step.

Note: Do not perform any other administration operations during this procedure.

1. From either 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_nodeID>
...
Finished receiving CDB sync series from machine <node1_nodeID>
```

Wait for approximately 60 seconds for the synchronization to complete.

9. From any node, start HA services in the cluster.

Upgrading OS Software in an Active Cluster

When you upgrade your OS software in an active cluster, you perform the upgrade on one node at a time.

If the OS software upgrade does not require reboot or does not impact the FailSafe software, there is no need to use the OS upgrade procedure. If you do not know whether the upgrade will impact FailSafe software or if the OS upgrade requires a machine reboot, follow the upgrade procedure described below.

The following procedure upgrades the OS software on node `web-node3`:

1. If resource groups are online on the node, use a `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 (`cmnd`, `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.

Upgrading FailSafe Software in an Active Cluster

When you upgrade FailSafe software in an active cluster, you upgrade one node at a time in the cluster.

The following procedure upgrades FailSafe on node `web-node3`:

1. If resource groups are online on the node, use a `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 (`cmnd`, `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.

Adding New Resource Groups or Resources in an Active Cluster

The following procedure adds 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 291.
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 "Upgrading OS Software in an Active Cluster", page 329. 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 a FailSafe cluster. For information about installing PCP for FailSafe, see "Install Performance Co-Pilot (PCP) Software", page 78.

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` and `hbvis` commands.

The `hbvis` command constructs a display showing the distribution of heartbeat response times for every node in the cluster. Figure 11-1 shows an example display.

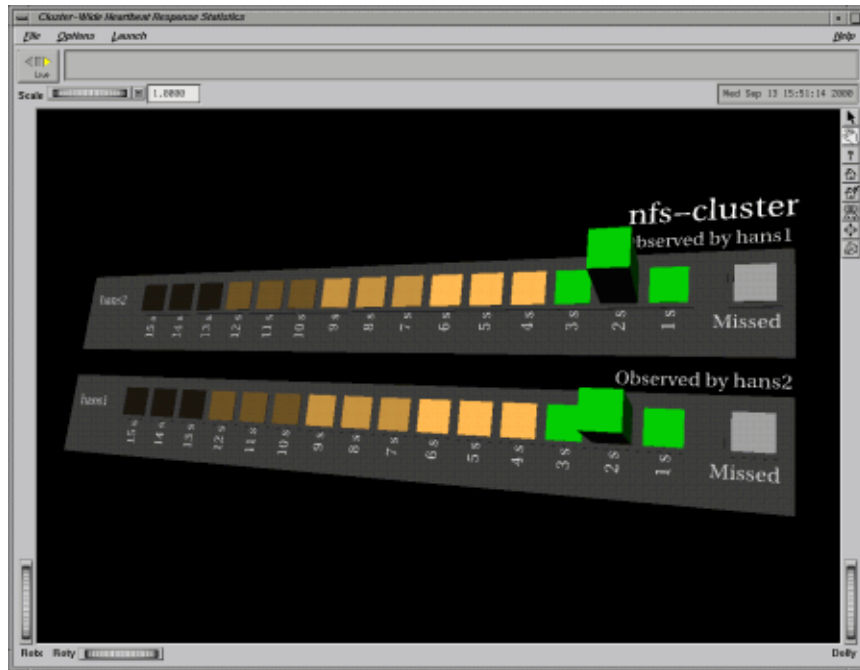


Figure 11-1 Heartbeat Response Statistics

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` 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.

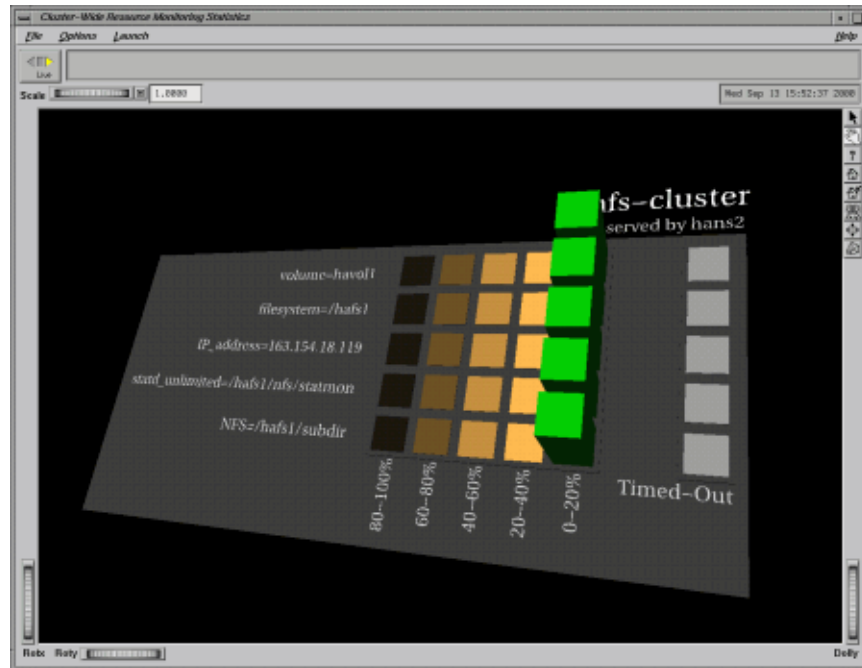


Figure 11-2 Resource Monitoring Statistics

The display is similar in concept to that of `hbvis`, 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` and `rmvis` when launched from the command line:

- | | |
|---------------------------------|---|
| <code>-H <i>hostfile</i></code> | 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. |
| <code>-t <i>interval</i></code> | 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, <code>hbvis</code> and <code>rmvis</code> will always show the latest statistics available for the sampling time selected. For details about the <i>interval</i> option, see the <code>pmview</code> and <code>PCPIntro</code> man pages. |
| <code>-r</code> | Selects the FailSafe metrics that present a sampling of statistics taken from the time of the last statistical reset. This enables <code>hbvis</code> and <code>rmvis</code> to improve the sensitivity of the visualization when abrupt changes appear in the FailSafe monitoring statistics.

Without the <code>-r</code> option, the statistics presented are from a sampling of FailSafe metrics collected from the time <code>ha_cmsd</code> and/or <code>ha_srmd</code> was last restarted. |
| <code>-R</code> | Starts a new statistical sampling. |
| <code>-v</code> | (<code>hbvis</code> 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 <code>-h</code> option). There is a |

graphical representation of heartbeat statistics for each node in the cluster as observed by the selected collector host.

`-w` (hbvis 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` and `rmvis` commands use the command `pmview` 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 page for `pmview(1)`.

PCP for FailSafe Performance Metrics

PCP tools such as `pmlogger`, `pmchart`, and `pminfo` can use the metrics exported by PCP for FailSafe.

Appendix E, "Metrics Exported by PCP for FailSafe", page 395, 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` or `rmvis` 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`/`rmvis` 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/rmvis` man pages for more details on the -H option.

- The collector daemons have been killed on that node.

To solve this problem, restart `pmdafsafe` in one of the following ways:

- If `pmcd` is still running, send `pmcd` the `SIGHUP` signal by entering the following:

```
# killall -HUP pmcd
```

- If `pmcd` 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` 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` and `pmcd`, or between `pmcd` and `pmview`. Refer to the man pages for `pmcd` and `PCPIntro` for information on how to change the timeout settings for the various PCP tools.

- The resource has failed over (for `rmvis`).

In this case, restart `rmvis` 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 343
- "Communication Paths", page 343
- "Communication Paths in a Coexecution Cluster", page 348
- "Execution of FailSafe Action and Failover Scripts", page 349
- "When a start Script Fails", page 353
- "When a stop Script Fails", page 353
- "Components", page 353

Software Layers

A FailSafe system has the following software layers:

- Plug-ins, which create highly available services. The following tables show the provided and optional FailSafe plug-ins and their associated resource types.

Table A-1 Provided Plug-Ins

Provided Plug-In	Resource Type
CXFS file system	CXFS
IP addresses	IP_address
MAC addresses	MAC_address
XFS filesystems	filesystem
XLV logical volumes	volume
XVM volume manager	XVM

Table A-2 Optional Plug-Ins

Optional Plug-In	Resource Type
IRIS FailSafe for DMF	DMF
IRIS FailSafe for NFS	NFS and statd_unlimited
IRIS FailSafe for Informix	INFORMIX_DB
IRIS FailSafe for Oracle	Oracle_DB
IRIS FailSafe for Samba	Samba
IRIS FailSafe for TMF	TMF
IRIS FailSafe for Web (Netscape)	Netscape_web

See the release notes for information about the specific releases of these products that are supported.

Note: The Samba `interfaces` parameter allows Samba to support multiple IP interfaces. It takes the following format, where *IP* **must be** a dotted decimal IP address and *netmask* **must be** a dotted decimal netmask such as 255.255.255.0:

```
interfaces = IP1/netmask1 IP2/netmask2
```

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-3, page 342, describes the contents of the `/usr/cluster/bin` directory. For more information about CXFS, see the *CXFS Version 2 Software Installation and Administration Guide*.

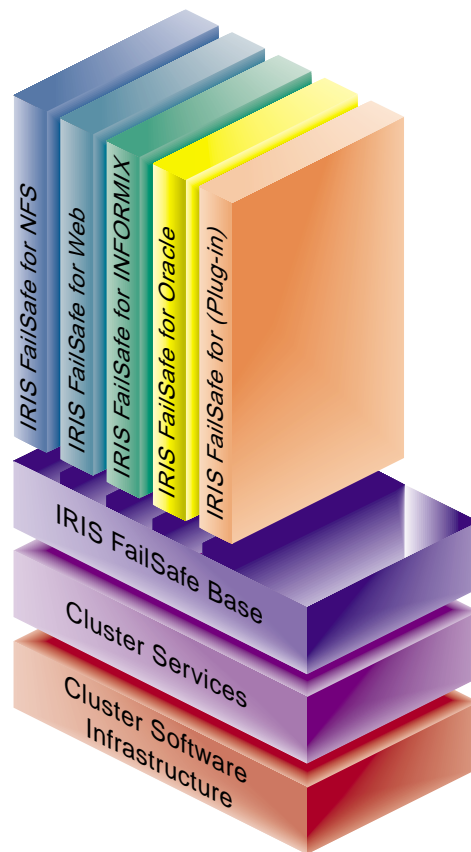


Figure A-1 Software Layers

Table A-3 Contents of /usr/cluster/bin

Layer	Subsystem	Process	Description
Plug-ins	failsafe_informix failsafe2_oracle	ha_ifmx2	FailSafe database agents. Each database agent monitors all instances of one type of database.
FailSafe Base	failsafe2	ha_fsd	FailSafe daemon. Provides basic component of the FailSafe software.
Cluster services (high-availability processes)	cluster_services	ha_cmds	The FailSafe membership daemon. Provides the list of nodes, called <i>FailSafe membership</i> , available to the cluster.
		ha_gcd	Group membership daemon. Provides group membership and reliable communication services in the presence of failures to 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 343.
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 interface. 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` 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.

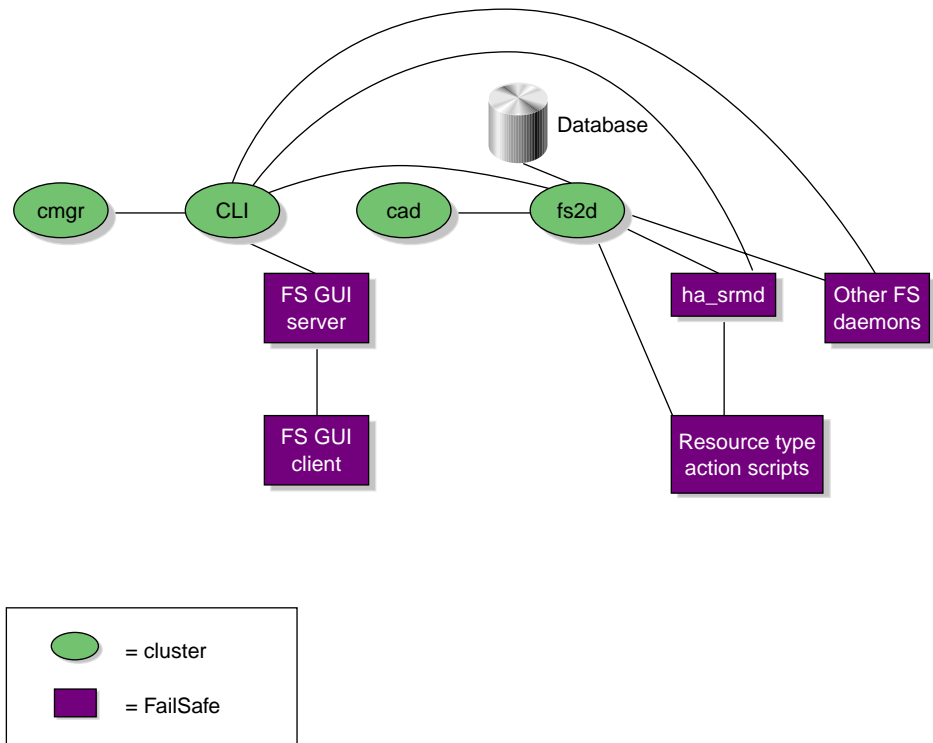


Figure A-2 Administrative Communication within One Node

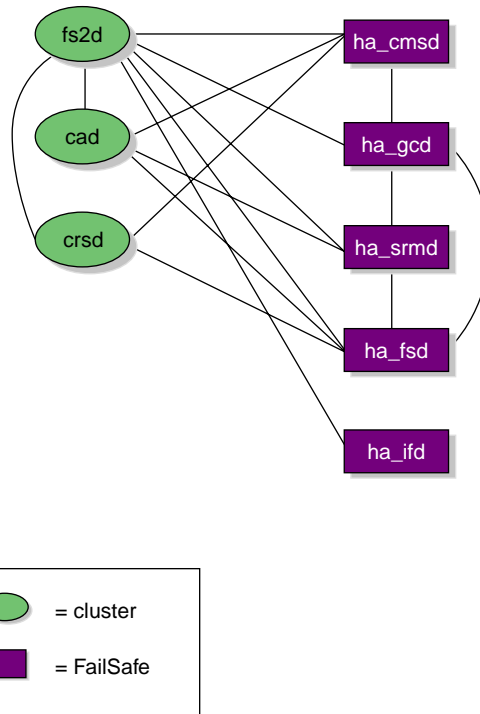


Figure A-3 Daemon Communication within One Node

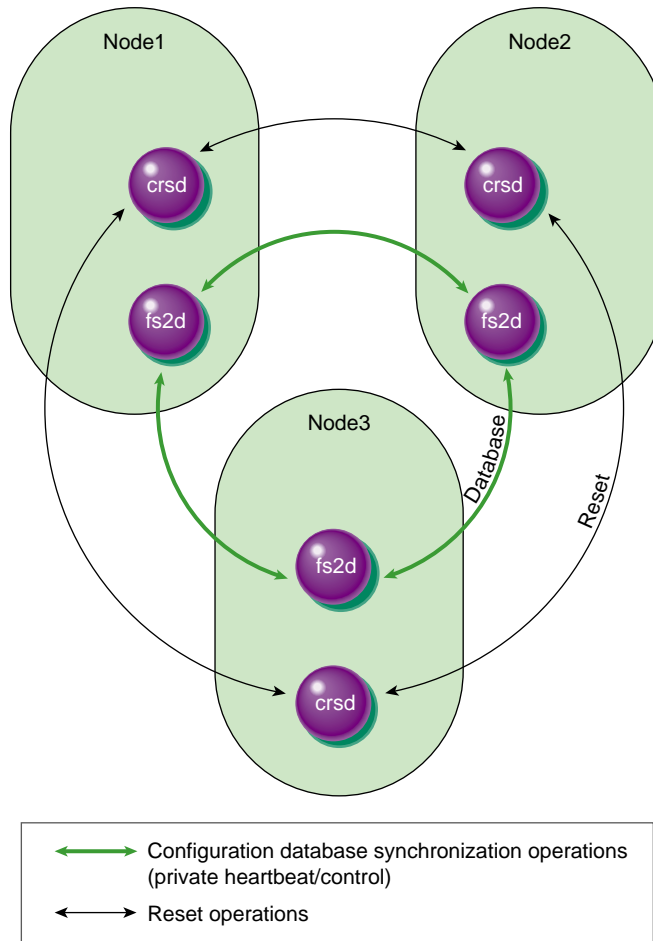


Figure A-4 Communication between Nodes in the Pool

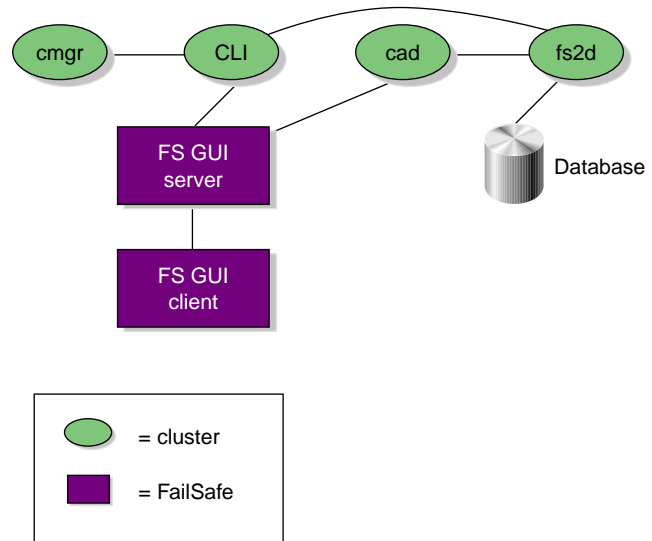


Figure A-5 Communication for a Node Not in the Cluster

Communication Paths in a Coexecution Cluster

The following figures show the communication paths within one node in a coexecution cluster.

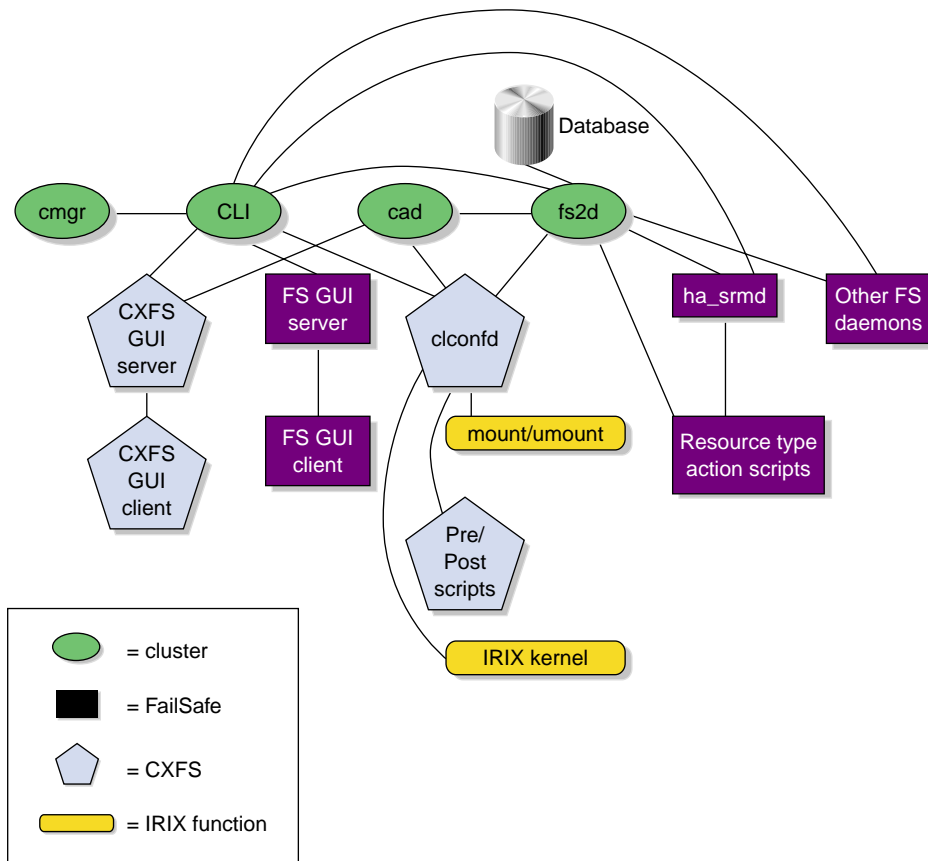


Figure A-6 Administrative Communication within One Node under Coexecution

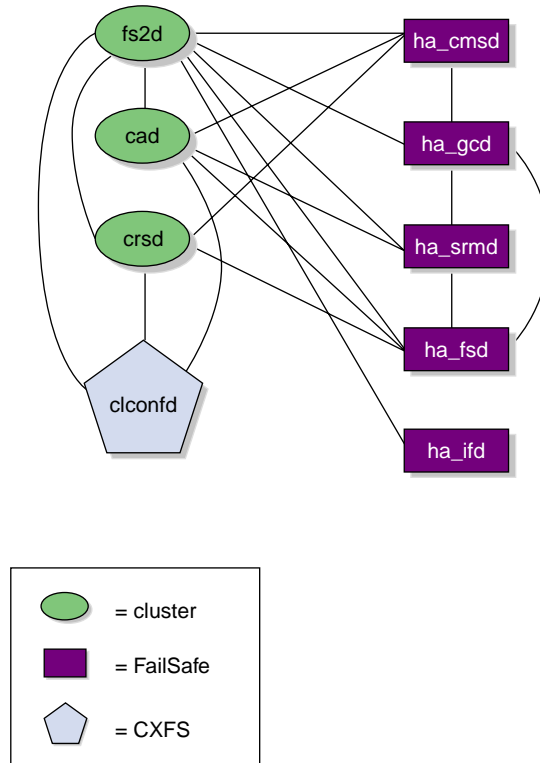


Figure A-7 Daemon Communication within One Node under Coexecution

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 script 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.

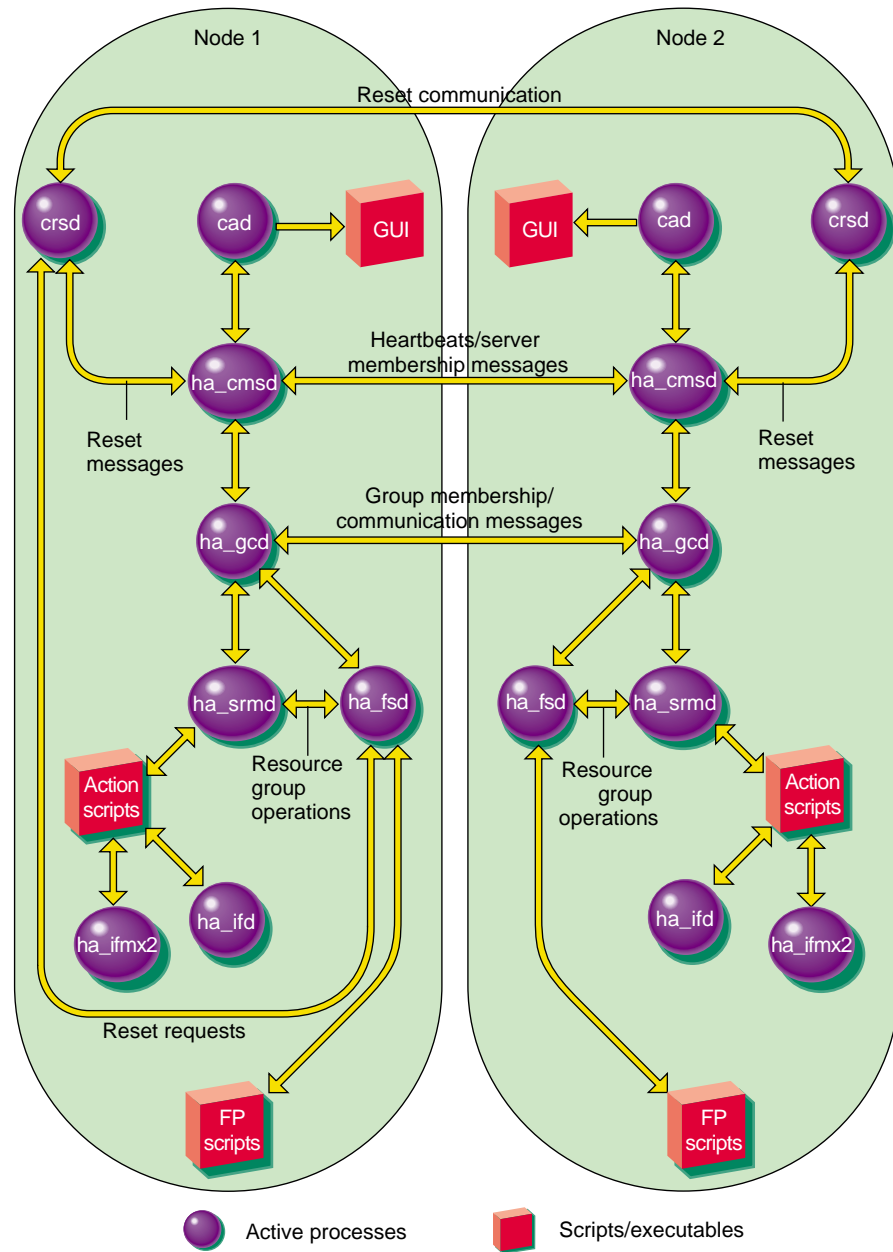


Figure A-8 Message Paths for Action Scripts and Failover Policy Scripts

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-4 Contents of the `/var/cluster/ha` directory

Directory or File	Purpose
<code>comm/</code>	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.
<code>common_scripts/</code>	Directory that contains the script library (the common functions that may be used in action scripts).
<code>log/</code>	Directory that contains the logs of all scripts and daemons executed by FailSafe. The outputs and errors from the commands within the scripts are logged in the <code>script_Nodename</code> file.
<code>policies/</code>	Directory that contains the failover scripts used for resource groups.
<code>resource_types/template</code>	Directory that contains the template action scripts.
<code>resource_types/RTname</code>	Directory that contains the action scripts for the <code>RTname</code> resource type. For example, <code>/var/cluster/ha/resource_types/filesystem</code> .
<code>resource_types/RTname/exclusive</code>	Script that verifies that a resource of this resource type is not already running.
<code>resource_types/RTname/monitor</code>	Script that monitors a resource of this resource type.
<code>resource_types/RTname/restart</code>	Script that restarts a resource of this resource type on the same node after a monitoring failure.
<code>resource_types/RTname/start</code>	Script that starts a resource of this resource type.
<code>resource_types/RTname/stop</code>	Script that stops a resource of this resource type.

Memberships and Quorums

The nodes in a FailSafe or CXFS cluster must act together to provide a service. To act in a coordinated fashion, each node must know about all the other nodes currently active and providing the service. The set of nodes that are currently working together to provide a service is called a membership. Cluster activity is coordinated by a configuration database that is replicated or at least accessible on all nodes in the cluster. The cluster software sends heartbeat messages between the nodes to indicate that a node is up and running. Heartbeat messages for each membership type are exchanged via a private network so that each node can verify each membership.

Nodes within the cluster must have the correct memberships in order to provide services. This appendix discusses the different types of membership and the effect they have on the operation of your cluster.

Nodes might not be able to communicate for reasons such as the following:

- They are down.
- The communication daemons have failed or have been turned off.
- Software has not been configured, or has been misconfigured.
- The network is misconfigured (in this case, some heartbeat messages may fail while others succeed).
- The network router or cable fails (in this case, all heartbeat messages will fail).

Nodes that cannot communicate must be excluded from the membership because the other nodes will not be able to verify their status.

It is critical that only one membership of each type exist at any one time, as confusion and corruption will result if two sets of nodes operate simultaneously but independently. There is a risk of this happening whenever a segmentation of the private network occurs, or any other network problem occurs that causes the nodes eligible for membership to be divided into two or more sets, where the nodes in each set can communicate with themselves, but not with nodes outside of the set. Thus, in order to form a membership, the nodes must have a quorum, the minimum number of nodes required to form a membership. The quorum is typically set at half the total eligible members.

For example, consider the case of six nodes eligible for a membership:

- If all six nodes can communicate with each other, they will form a membership of six and begin offering the membership's services.
- If a network segmentation occurs that causes four nodes to be in one set and two in another set, the two-node set will try to form its own membership but will be unable to do so because it does not have enough nodes to form a quorum; these nodes will therefore stop offering services. The four-node set will be able to form a new membership of four nodes and will continue to offer the membership's services.
- If a network segmentation occurs that divides the nodes into three sets of two nodes each, no set will be able to form a membership because none contains enough nodes to form a quorum. In this case, the membership services will be unavailable; this situation is unavoidable, as each set of two nodes thinks that the four other nodes may have formed a quorum, and so no set may safely offer the membership's services.
- If a network segmentation occurs that divides the nodes into two sets of three, then both could have a quorum, which could cause problems. To prevent this situation from occurring, some memberships may require a majority (>50%) of nodes or a tiebreaker node to form or maintain a membership. Tiebreaker nodes are used when exactly half the nodes can communicate with each other.

The following sections provide more information about the specific requirements for membership.

Note: Because the nodes are unable to distinguish between a network segmentation and the failure of one or more nodes, the quorum must always be met, regardless of whether a partition has actually occurred or not.

Membership Types

There are three types of membership:

- "Cluster Database Membership and Quorum"
- "FailSafe Membership, Quorum, and Tiebreaker", page 358
- "CXFS Kernel Membership, Quorum, and Tiebreaker", page 359

Each provides a different service using a different heartbeat. Nodes are usually part of more than one membership.

Cluster Database Membership and Quorum

The nodes that are part of the the cluster database membership (also known as *fs2d membership*) work together to coordinate configuration changes to the cluster database:

- The *potential* cluster database membership is all of the administration nodes (installed with `cluster_admin` and running `fs2d`) that are defined using the GUI or the `cmgr` command as nodes in the pool. (CXFS client-only nodes are not eligible for cluster database membership.)
- The *actual* membership is the subset of eligible nodes that are up and running and accessible to each other, as determined by heartbeats on the private network. If the primary private network is unavailable, the cluster database heartbeat will fail over to the next available heartbeat network defined for the node, if any (CXFS nodes are limited to a single heartbeat network).

The cluster database heartbeat messages use remote procedure calls (RPCs). Heartbeats are performed among all nodes in the pool. You cannot change the heartbeat timeout or interval.

If a node loses its cluster database membership, the cluster database write-operations from the node will fail; therefore, FailSafe and CXFS configuration changes cannot be made from that node.

The *cluster database membership quorum* ensures atomic write-operations to the cluster database that `fs2d` replicates in all administration nodes in the pool.

The quorum allows an initial membership to be formed when a majority (>50%) of the eligible members are present. If there is a difference in the membership log between members, the cluster database tiebreaker node is used to determine which database is replicated. (See "Cluster Database Membership Logs", page 361.) The tiebreaker node is always the administration node in the membership with the lowest node ID; you cannot reconfigure the tiebreaker for cluster database membership.

When the quorum is lost, the cluster database cannot be updated. This means that FailSafe and CXFS configuration changes cannot be made; although FailSafe and CXFS may continue to run, the loss of the cluster database quorum usually results in the loss of quorum for FailSafe and/or CXFS, because the nodes that drop from the cluster database membership will probably also drop from other memberships.

FailSafe Membership, Quorum, and Tiebreaker

The nodes that are part of the FailSafe membership provide highly available (HA) resources for the cluster:

- The *potential* FailSafe membership is the set of all FailSafe nodes that are defined in the cluster and on which HA services have been enabled. Nodes are enabled when HA services are started. The enabled status is stored in the cluster database; if an enabled node goes down, its status will remain enabled to indicate that it is supposed to be in the membership.
- The *actual* membership consists of the eligible nodes whose state is known and that are communicating with other FailSafe nodes using heartbeat and control networks. If the primary private network is unavailable, the FailSafe heartbeat will fail over to the next available heartbeat network defined for the node.

The FailSafe heartbeat uses user datagram protocol (UDP). Heartbeats are performed among all FailSafe-enabled nodes in the cluster. You can change the FailSafe heartbeat timing with the GUI **Set FailSafe HA Parameters** task or the `cmgr` command `modify ha_parameters` (the `node_timeout` parameter is the heartbeat timeout and the heartbeat is the heartbeat interval).

If a node loses its FailSafe membership, FailSafe will fail over its HA resources to another node in the cluster.

The *FailSafe membership quorum* ensures that a FailSafe resource is available only on one node in the cluster. The quorum requires that the state of a majority (>50%) of eligible nodes to be known and that half (50%) of the eligible nodes be present to form or maintain membership.

If a network partition results in a tied membership, in which there are two sets of nodes (each consisting of 50% of the potential FailSafe membership), then a node from the set containing the *FailSafe tiebreaker node* will attempt to perform a serial hardware reset on a node in the other set. *Serial hardware reset* is the failure action that performs a system reset via a serial line connected to the system controller.

If the node can verify that the other node was reset, then the membership will continue on the set with the tiebreaker. However, containing the tiebreaker is not a guarantee of membership; for more information, see the *IRIS FailSafe Version 2 Administrator's Guide*. The default FailSafe tiebreaker is the node with the lowest node ID in the cluster.

When FailSafe membership quorum is lost, the resources will continue to run but they are no longer highly available.

CXFS Kernel Membership, Quorum, and Tiebreaker

The nodes that are part of the CXFS kernel membership can share CXFS filesystems:

- The *potential* CXFS kernel membership is the group of all CXFS nodes defined in the cluster and on which CXFS services have been enabled. Nodes are enabled when CXFS services are started. The enabled status is stored in the cluster database; if an enabled node goes down, its status will remain enabled to indicate that it is supposed to be in the membership.
- The *actual* membership consists of the eligible nodes on which CXFS services have been enabled and that are communicating with other nodes using the heartbeat/control network. CXFS supports only one private network, and that network is the only network used for CXFS kernel membership heartbeats (but remember that the CXFS nodes may use multiple networks for the cluster database membership heartbeats).

Note: CXFS metadata also uses the private network. The multiple heartbeats on the private network therefore reduce the bandwidth available for CXFS metadata.

During the boot process, a CXFS node applies for CXFS kernel membership. Once accepted, the node can actively share the filesystems in the cluster.

The CXFS heartbeat uses multicast. Heartbeats are performed among all CXFS-enabled nodes in the cluster.

If a node loses its CXFS kernel membership, it can no longer share CXFS filesystems.

The *CXFS kernel membership quorum* ensures that only one metadata server is writing the metadata portion of the CXFS filesystem over the storage area network:

- For the *initial* CXFS kernel membership quorum, a majority (>50%) of the server-capable administration nodes with CXFS services enabled must be available to form a membership. (*Server-capable administration* nodes are those that are installed with the `cluster_admin` product and are also defined with the GUI or `cmgr` as capable of serving metadata. Client administration nodes are those that are installed with the `cluster_admin` product but are not defined as server-capable.)

Note: Client administration nodes and client-only nodes can be part of the CXFS kernel membership, but they are not considered when forming a CXFS kernel membership quorum. Only server-capable nodes are counted when forming the quorum.

- To *maintain* the existing CXFS kernel membership quorum requires at least half (50%) of the server-capable nodes that are eligible for membership. If CXFS kernel quorum is lost, the shared CXFS filesystems are no longer available.

If you do not use serial hardware reset or I/O fencing to prevent a problem node from accessing I/O devices, you should set a CXFS tiebreaker node to avoid multiple CXFS kernel memberships in the event of a network partition. In CXFS, there is no default tiebreaker. Any node in the cluster can be a CXFS tiebreaker node. You can set the tiebreaker node by using the GUI's **Set Tiebreaker Node** task or by using the `set tie_breaker` command in `cmgr`.

Note: Suppose you have a cluster with only two server-capable nodes with one of them being the CXFS tiebreaker node. If the tiebreaker node fails or if the administrator stops CXFS services on the tiebreaker node, the other node will not be able to maintain a membership and will do a forced shutdown of CXFS services, which unmounts all CXFS filesystems.

If I/O fencing or serial hardware reset is used, the quorum is maintained by whichever side wins the reset/fence race.

If a tiebreaker node is set and the network being used for heartbeat/control is divided in half, only the group that has the CXFS tiebreaker node will remain in the CXFS kernel membership. Nodes on any portion of the heartbeat/control network that are not in the group with the tiebreaker node will exit from the membership. Therefore, if the heartbeat/control network is cut in half, you will not have an active metadata server on each half of the heartbeat/control network trying to access the same CXFS metadata over the storage area network at the same time.

Note: A tiebreaker node must be configured individually for CXFS and for FailSafe. In a coexecution cluster, these could be different nodes.

Cluster Database Membership Logs

Each `fs2d` daemon keeps a *membership log* that contains a history of each database change (write transaction), along with a list of nodes that were part of the membership when the write transaction was performed. All nodes that are part of the cluster database membership will have identical membership logs.

When a node is defined in the database, it must obtain a current copy of the cluster database and the membership log from a node that is already in the cluster database membership. The method used to choose which node's database is replicated follows a hierarchy:

1. If the membership logs in the pool share a common transaction history, but one log does not have the most recent transactions and is therefore incomplete, the database from a node that has the complete log will be chosen to be replicated.
2. If there are two different sets of membership logs, the database from the set with the most number of nodes will be chosen.
3. If there are two different sets of membership logs, and each set has an equal number of nodes, then the set containing the node with the lowest node ID will be chosen.

To ensure that the complete transaction history is maintained, do not make configuration changes on two different administration nodes in the pool simultaneously. You should connect the CXFS or FailSafe GUI to (or run the `cmgr` command on) a single administration node in the pool when making changes. However, you can use any node in the pool when requesting status or configuration information.

The following figures describe potential scenarios using the hierarchies.

Figure B-1, page 363, shows:

- Time 1: An established pool of three administration nodes sharing heartbeats, with node IDs 1-3, represented by the node names N1-N3. The tiebreaker node is the node in the membership with the lowest node ID. Each successive database write is identified by a letter in the membership log.
- Time 2: A new node, N4, is defined using `cmgr` or the GUI connected to node N1. Node N4 (node ID = 4) joins the pool. Its membership log is empty.
- Time 3: Because N1/N2/N3 have identical membership logs, the database is replicated from one of them. In this case, N2 is randomly chosen.

- Time 4: All nodes in the pool have identical membership logs.
- Time 5: A network partition occurs that isolates N1. Therefore, N1 can no longer receive database updates. Configuration changes are made by connecting the GUI to N2 (or running `cmgr` on node N2); this results in updates to the membership logs in N2, N3, and N4, but not to N1 because it is isolated.
- Time 6: The partition is resolved and N1 is no longer isolated. Because N2/N3/N4 have identical membership logs, and share the beginning history with N1, the database is replicated from one of them. N4 is chosen at random.
- Time 7: All nodes in the pool have identical membership logs.

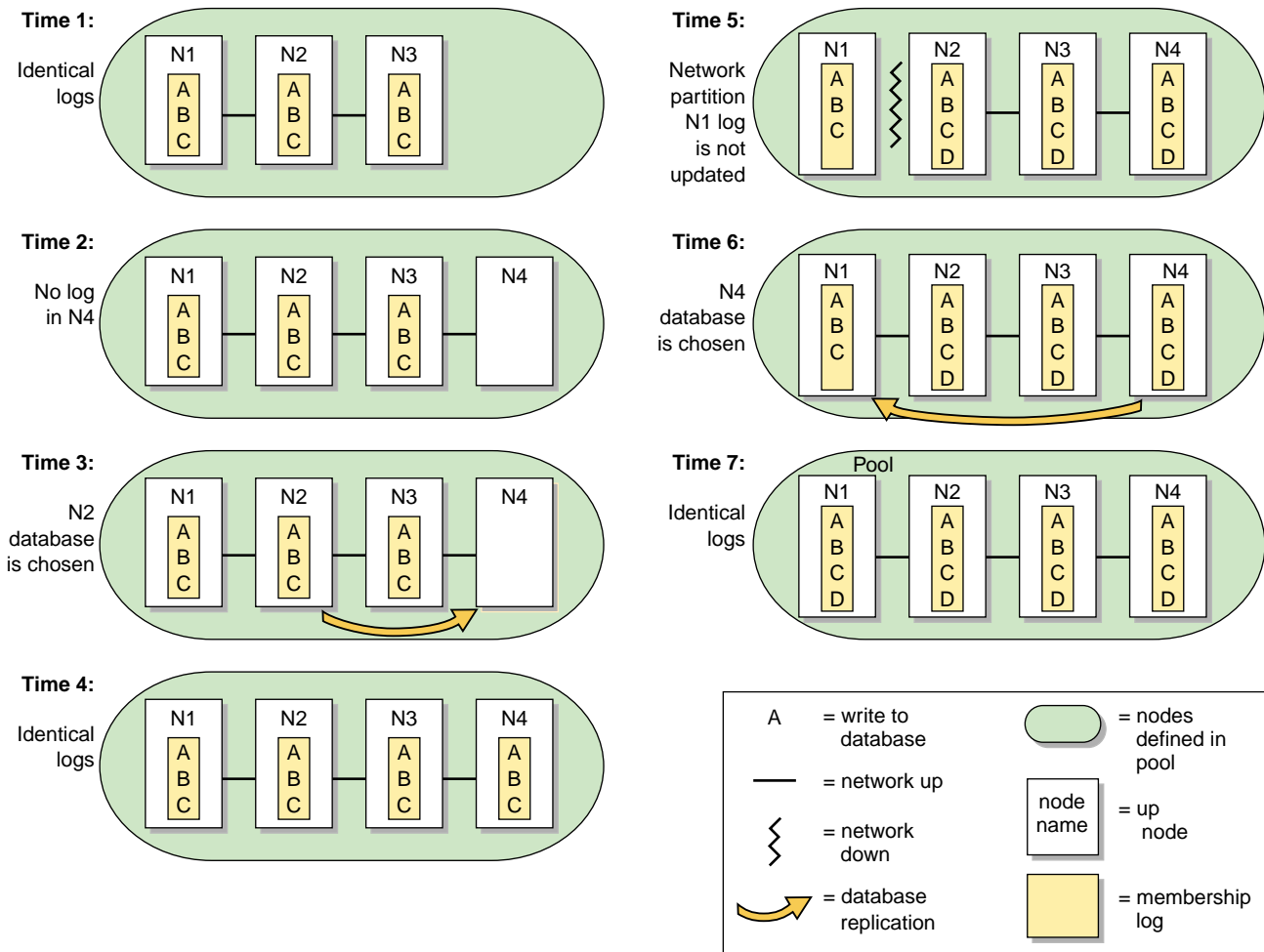


Figure B-1 One Node is Out of Date: Most Recent Log is Replicated

Recall that a node can be in only one pool at a time. If there are two separate pools, and from a node in one pool you define one or more nodes that are already in the other pool, the result will be that nodes from one of the pools will move into the other pool. **This operation is not recommended**, and determining which nodes will move into which other pool can be difficult. Figure B-2, page 364 illustrates what to expect in this situation.

- Time 1: There are two pools that do not share membership log contents. One pool has two nodes (N1/N2), the other has three (N3/N4/N5).
- Time 2: N1 and N2 are defined as part of the second pool by running `cmgr` or connecting the GUI to node N3, N4, or N5. This results in a new pool with five nodes with different membership logs.
- Time 3: The database from the larger set of nodes is the one that must be replicated. N3 is chosen at random from the N3/N4/N5 set.
- Time 4: All nodes in the pool have identical membership logs.

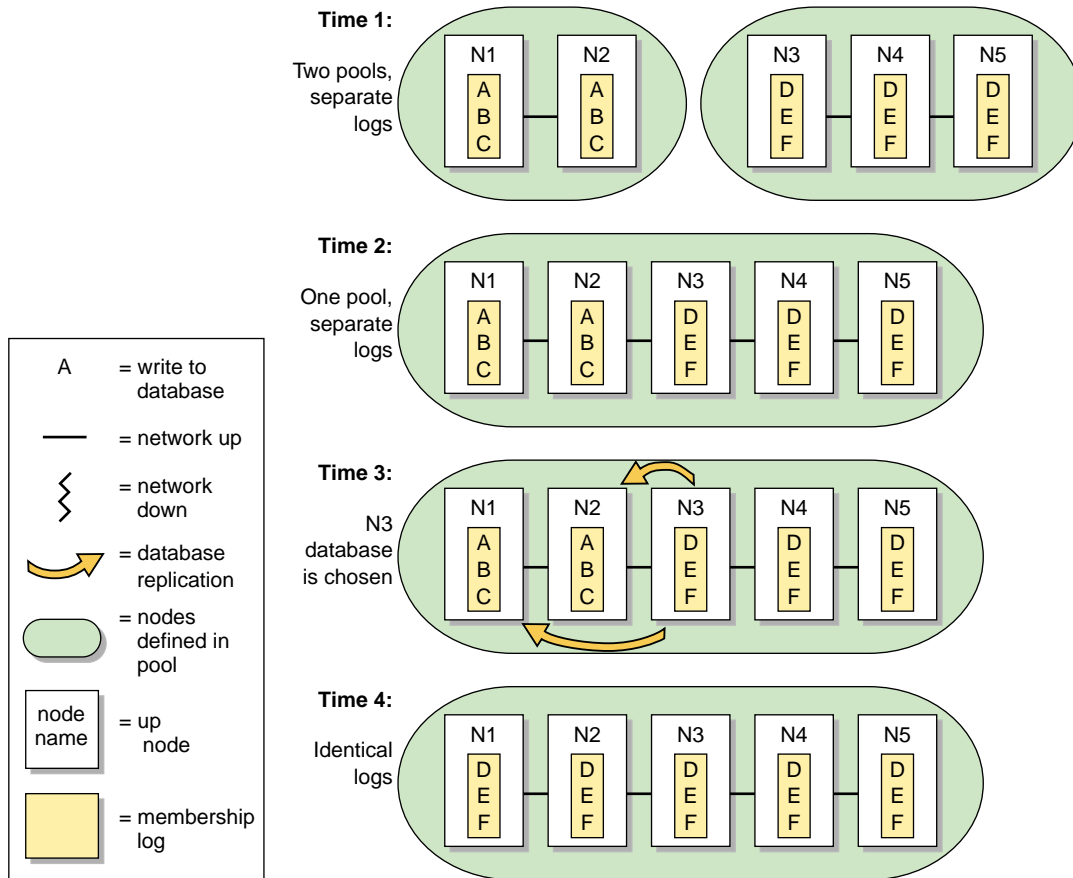


Figure B-2 Unequally Sized Pools are Joined: Larger Pool Log is Replicated

Figure B-3, page 366, shows a similar situation in which two nodes are defined in two pools, but the pools are of equal size:

- Time 1: There are two pools that do not share membership log contents. Each pool has two nodes (N1/N2 in pool 1, and N3/N4 in pool 2).
- Time 2: N1 and N2 are defined as part of the second pool by connecting the GUI or running `cmgr` on node N3 or N4. This results in a new pool with four nodes with different membership logs.
- Time 3: Because each set has the same number of nodes, the tiebreaker node (the node with the lowest node ID in the membership) must be used to determine whose database will be chosen. Because node N1 is the lowest node ID (node ID=1), the database from N1 is chosen.
- Time 4: All nodes in the pool have identical membership logs.

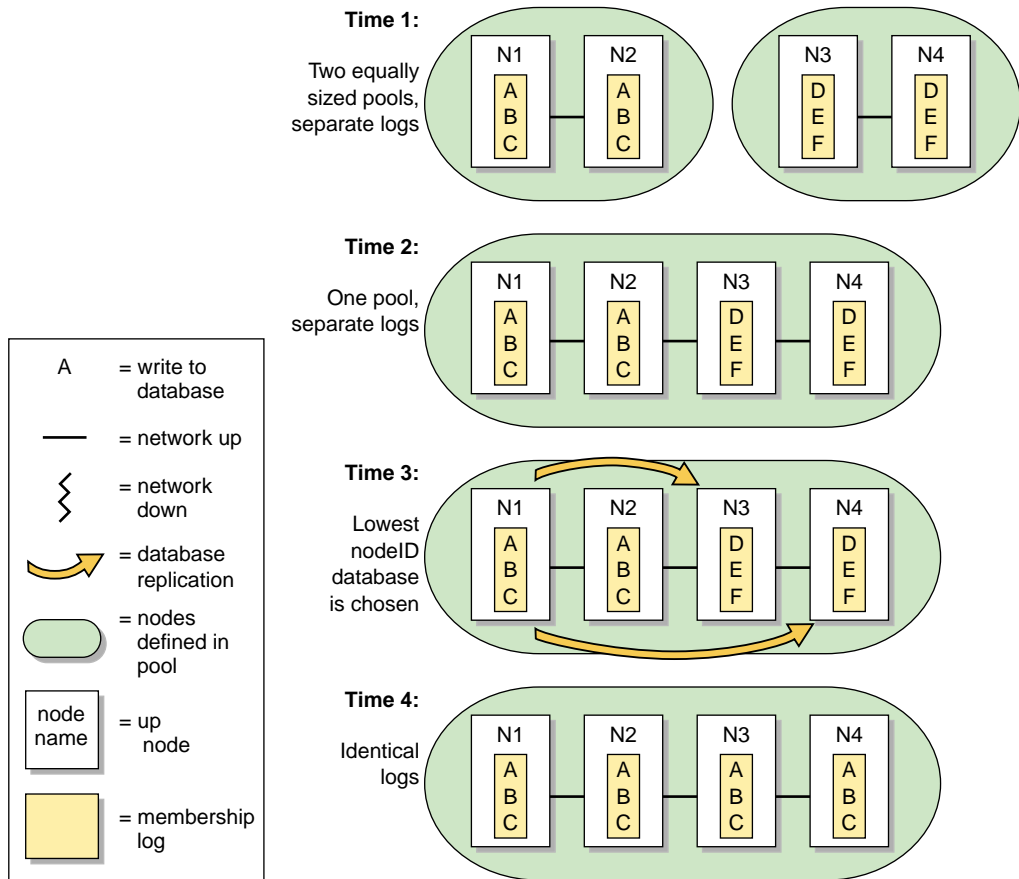


Figure B-3 Equally Sized Pools are Joined: Lowest Node ID Log is Replicated

Quorum and Tiebreaker Examples

Coexecution Example

Figure B-4, page 368, shows an example of the different memberships in a cluster running CXFS and FailSafe. The pool contains 15 nodes (named N1 through N15). N1 has the lowest node ID number. There are CXFS nodes running IRIX, Solaris, and Windows; only the nodes running IRIX are administration nodes containing the

cluster database. The FailSafe nodes are those where HA services are enabled; each of these is an administration node.

- Cluster database membership:
 - Eligible: N1, N2, N3, N5, N9, and N10 (that is, all nodes containing the cluster database)
 - Actual: N1, N2, N3, N4, and N10 (because N5 and N9 are down)
 - Quorum: N1, N2, N3, and N10 (>50% of eligible nodes)
- FailSafe membership:
 - Eligible: N1, N2, and N3 (that is, those nodes with HA services enabled and defined as part of the cluster)
 - Actual: N1, N2, N3
 - Quorum: N1, N2, N3 (>50% of eligible nodes)
- CXFS kernel membership:
 - Eligible: N1-N8 and N11-N15 (N9 and N10 are not defined as part of the cluster)
 - Actual: N1, N2, N3, N4, N6, and N11-N15 (because N5, N7, and N8 are down)
 - Quorum: N1, N2 (>50% of server-capable eligible nodes)

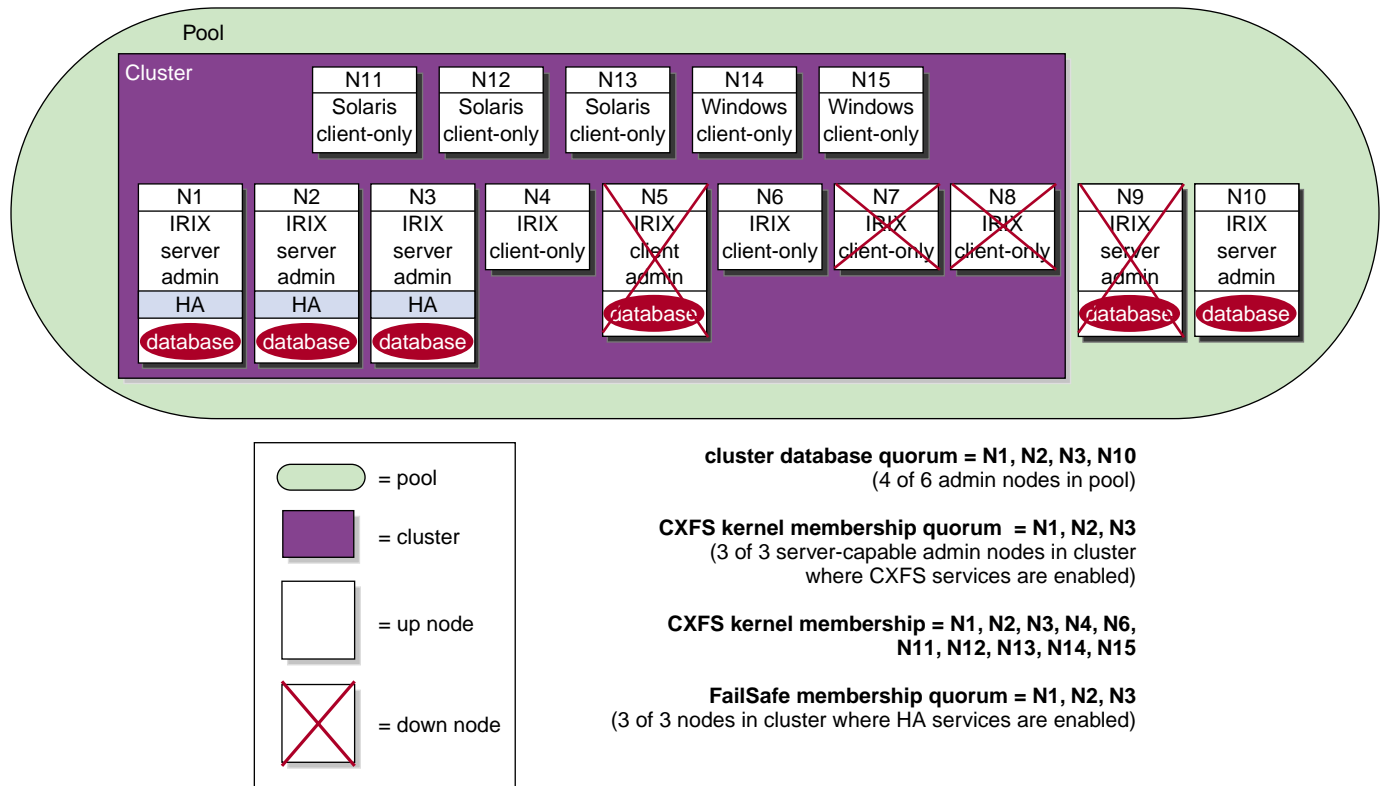


Figure B-4 Example Memberships in a Coexecution Cluster

FailSafe Tiebreaker Node Example

Figure B-5, page 370, assumes the following application failover domains for resource groups RG1-RG4:

- RG1: N1, N3
- RG2: N2, N4
- RG3: N3, N1
- RG4: N4, N2

Figure B-5, page 370. shows the following:

- Time 1: The nodes in the cluster are all part of the FailSafe membership.
- Time 2: A router dies and the heartbeat/control network is effectively split in two, leaving two sets of nodes that can communicate heartbeat messages (set 1 is N1/N2; set 2 is N3/N4)
- Time 3: Because nodes N1/N2 are the set containing the tiebreaker, they will both try to reset the nodes in the other set (N3/N4) (assuming that this is how the nodes were defined for reset purposes). The `crsd` daemon will filter out duplicate requests. N1 is able to reset N3 before the reset timeout takes place.
- Time 4: After the reset is successful and the membership of N1/N2 is confirmed, the resource group RG3 on the reset node N3 will be moved according to its failover policy to N1. Assuming N4 was also successfully reset, its resource group RG4 will similarly move to N2. If N4 does not get reset, it will continue to serve the resources in RG4, but will be in UNKNOWN state in the database in the membership formed by N1/N2.

Once N3 and N4 are running again after the reset, they will try to form a membership because they will be unaware that N1/N2 are already in a membership. However, they will not be able to form a membership because they will not have a >50% quorum.

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

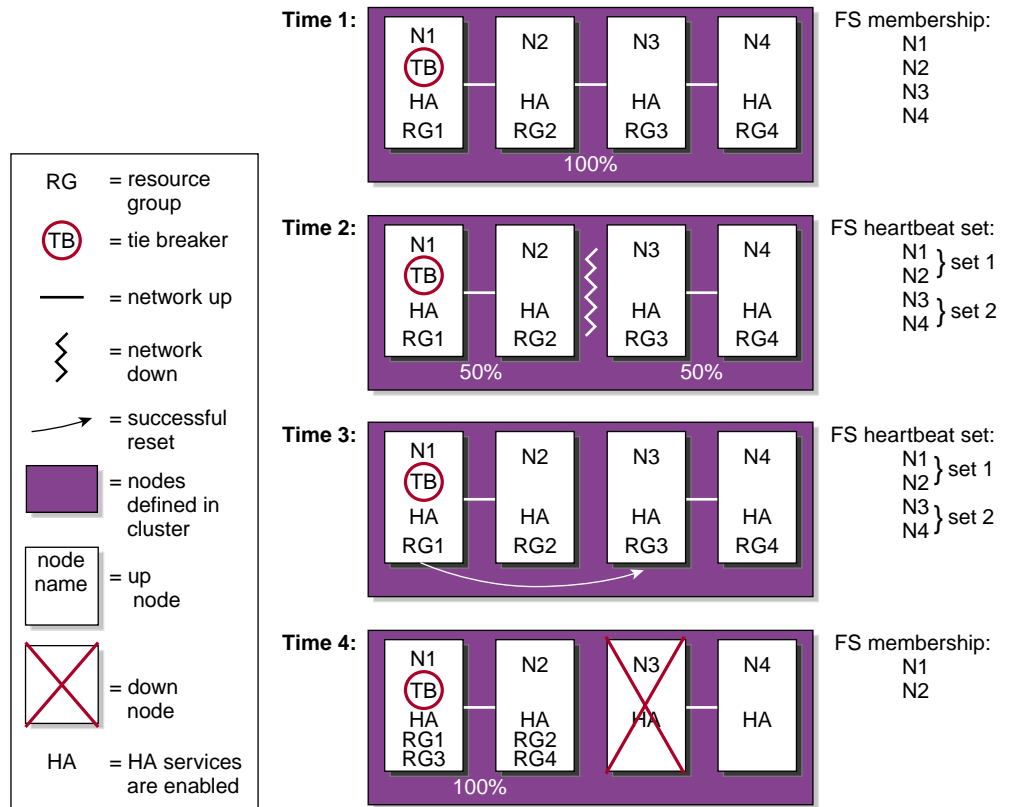


Figure B-5 FailSafe Tiebreaker Node

Cases when the FailSafe Tiebreaker is Not in the Membership

Just because a group of FailSafe nodes contains the FailSafe tiebreaker, that does not guarantee that it will form or maintain a FailSafe membership. If one group of nodes has the quorum (>50% of the eligible members), it will form the membership whether or not the group contains the tiebreaker node.

In the case of two sets of nodes, each with 50% of the eligible members, the group of nodes that contains the tiebreaker will have the first opportunity to form the membership; however, if the group with the tiebreaker is unable to reset a node in the other group before the reset timeout expires, then the other group will have the

opportunity to reset a node in the first group, and if successful will form a membership even though it does not have the tiebreaker node.

Figure B-6, assumes the same application failover domains as Figure B-5, page 370. Figure B-6, shows the following:

- Time 1: The nodes in the cluster are all part of the FailSafe membership.
- Time 2: A router dies and the heartbeat/control network is effectively split in two, leaving two sets of nodes that can communicate heartbeat messages (set 1 is N1/N2; set 2 is N3/N4)
- Time 3: Because nodes N1/N2 are the set containing the tiebreaker, they will both try to reset the nodes in the other set (N3/N4), assuming that this is how the nodes were defined for reset purposes. The `crsd` daemon will filter out duplicate requests. However, set 1 is unable to reset a node in set 2 before the reset timeout expires.
- Time 4: Set 2 is now free to try to reset set 1. Both N3 and N4 will try to reset both nodes in set 1; the `crsd` daemon will filter out duplicate requests. N3 is successful in resetting N2.
- Time 5: The membership of N3/N4 is confirmed, the resource group RG2 on the reset node N2 will be moved according to its failover policy to N3. Assuming N1 was also successfully reset, its resource group RG1 will similarly move to N4. If N1 does not get reset, it will continue to serve the resources in RG1, but will be in UNKNOWN state in the database in the membership formed by N3/N4.

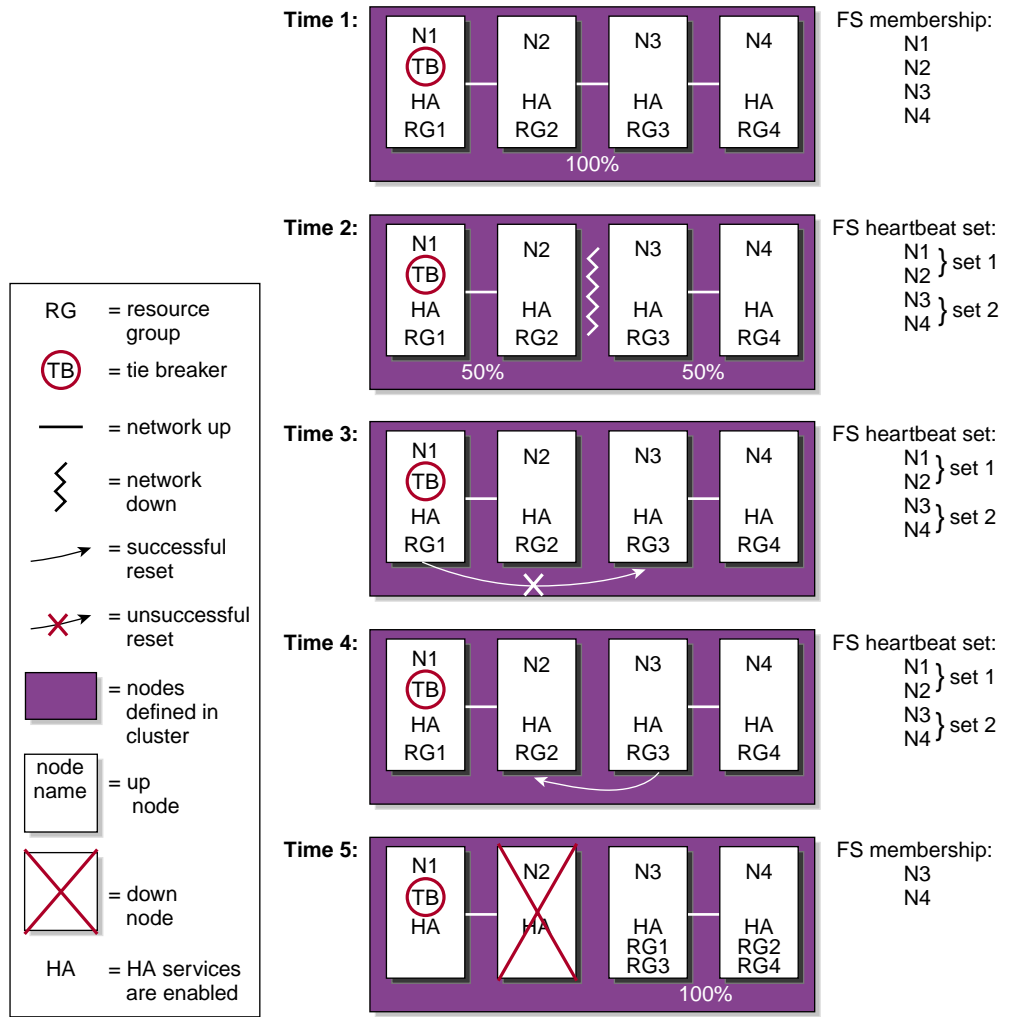


Figure B-6 Forming a Membership without the FailSafe Tiebreaker

Heartbeat Considerations

There are different heartbeats for each membership type, and each uses a different networking method. Therefore, certain network misconfiguration can cause one heartbeat to fail while another succeeds.

At least two networks should be designated as FailSafe heartbeat networks. FailSafe uses only the highest priority working network for heartbeats; the other network is for heartbeat failover. Usually the private network is used as the highest priority heartbeat network.

In a coexecution cluster, there must be two networks as required by FailSafe; at least one private network is recommended for FailSafe and a private network is required by CXFS.

In a coexecution cluster, CXFS metadata, CXFS heartbeat, and FailSafe heartbeat can use the same network. The heartbeat intervals and timeouts should be appropriately adjusted, if possible, so that all network traffic has sufficient bandwidth. You cannot change the heartbeat timeout or interval for the cluster database membership. Before you adjust the heartbeat settings for the FailSafe membership or CXFS kernel membership, you should consider the impact on the other heartbeats.

If the highest priority network fails, the FailSafe and cluster database memberships will continue using just the next priority network, but the CXFS kernel membership will fail.

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 376
- "Configuration Changes", page 376
- "Scripts", page 377
- "Operational Comparison", page 378
- "Upgrade Examples", page 379
- "Additional FailSafe 2.1.x Tasks", page 386
- "Status", page 387

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 291, 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_ttypty 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 379.

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 C-1 Differences between IRIS FailSafe 1.2 and IRIS FailSafe 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 <code>/var/ha/logs</code> .	Action scripts use cluster logging and all scripts log to the same file using the <code>ha_cilog</code> 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 382.

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:

$$\text{node_timeout} = (\text{probetime} + \text{timeout}) * \text{lostcount}$$

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 volume-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 = 0xffffffff00
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: 0xffffffff00

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 0xffffffff00
resource 54.3.252.7? set BroadcastAddress 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 201.

After you have configured your system, you can start FailSafe services, as described in "Start FailSafe HA Services", page 206.

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 255.

FailSafe Software

This appendix summarizes software to be installed on systems used for FailSafe. It consists of the following sections:

- "Subsystems on the CD"
- "Subsystems for Servers and Workstations in the Pool", page 391
- "Additional Subsystems for Nodes in the FailSafe Cluster", page 392
- "Additional Subsystems for Workstations ", page 392

Note: "Install Software", page 55 contains step-by-step instructions for installing the software.

Subsystems on the CD

The FailSafe base CD requires about 10 MB.

Table D-1, page 390, lists FailSafe subsystems on the FailSafe CD.

Table D-1 FailSafe 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 Etherlite network based serial multiplexer driver subsystems are `el_serial.man` and `el_serial.sw`, which are on a CD accompanying the 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 D-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
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
Etherlite network-based serial multiplexer driver (from CD included with multiplexer)	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 D-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 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 workstations
- Other operating system workstations

Subsystems for IRIX Workstations

On a workstation used to run the GUI client from an IRIX desktop, install subsystems listed in the following table.

Table D-4 Subsystems Required for IRIX 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 Other Operating System Workstations

From a workstation running an operating system other than IRIX, 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`.

`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` or any of the resources themselves are not available.

Table E-1 PCP Metrics

Metric	Description
<code>fsafe.srm.status</code> <code>fsafe.srm.all.status</code>	Latest status of a monitoring event performed on a resource, for all resources configured to be monitored on this node.
<code>fsafe.srm.timeout</code> <code>fsafe.srm.all.timeout</code>	The prescribed timeout, in milliseconds, for monitoring a resource.
<code>fsafe.srm.probes</code> <code>fsafe.srm.all.probes</code>	Number of times a resource has been monitored, for all resources configured to be monitored on this node, since the time <code>ha_srmd</code> was started.
<code>fsafe.srm.recent.probes</code>	Number of times a resource has been monitored, for all resources configured to be monitored on this node, since a data collection reset (via <code>fsafe.control.reset_srm</code>).
<code>fsafe.srm.timeouts</code> <code>fsafe.srm.all.timeouts</code>	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.
<code>fsafe.srm.recent.timeouts</code>	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 <code>fsafe.control.reset_srm</code>).
<code>fsafe.srm.min_resp</code> <code>fsafe.srm.all.min_resp</code>	Approximate minimum time, in milliseconds, taken to complete a monitoring event on a resource, for all resources configured to be monitored.

E: Metrics Exported by PCP for FailSafe

Metric	Description
<code>fsafe.srm.max_resp</code> <code>fsafe.srm.all.max_resp</code>	Approximate maximum time, in milliseconds, taken to complete a monitoring event on a resource, for all resources configured to be monitored on this node.
<code>fsafe.srm.last_resp</code> <code>fsafe.srm.all.last_resp</code>	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.
<code>fsafe.srm.cumm_timeouts</code> <code>fsafe.srm.all.cumm_timeouts</code>	Cumulative number of resource monitoring events that have timed out, for all resources configured to be monitored on this node, since the time <code>ha_srm</code> has started.
<code>fsafe.srm.recent.cumm_timeouts</code>	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 <code>fsafe.control.reset_srm</code>).
<code>fsafe.srm.histo_20</code> <code>fsafe.srm.all.histo_20</code>	Fraction of monitoring events that have been received within 0-20% of the response time from 0 milliseconds to <code>fsafe.srm.timeout</code> , for all resources configured to be monitored on this node, since the time <code>ha_srm</code> has started.
<code>fsafe.srm.recent.histo_20</code>	Fraction of monitoring events that have been received within 0-20% of the response time from 0 milliseconds to <code>fsafe.srm.timeout</code> , for all resources configured to be monitored on this node, since a data collection reset (via <code>fsafe.control.reset_srm</code>).
<code>fsafe.srm.histo_40</code> <code>fsafe.srm.all.histo_40</code>	Fraction of monitoring events that have been received within 20-40% of the response time from 0 milliseconds to <code>fsafe.srm.timeout</code> , for all resources configured to be monitored on this node, since the time <code>ha_srm</code> has started.
<code>fsafe.srm.recent.histo_40</code>	Fraction of monitoring events that have been received within 20-40% of the response time from 0 milliseconds to <code>fsafe.srm.timeout</code> , for all resources configured to be monitored on this node, since a data collection reset (via <code>fsafe.control.reset_srm</code>).

Metric	Description
<code>fsafe.srm.histo_60</code> <code>fsafe.srm.all.histo_60</code>	Fraction of monitoring events that have been received within 40-60% of the response time from 0 milliseconds to <code>fsafe.srm.timeout</code> , for all resources configured to be monitored on this node, since the time <code>ha_srmd (1M)</code> has started.
<code>fsafe.srm.recent.histo_60</code>	Fraction of monitoring events that have been received within 40-60% of the response time from 0 milliseconds to <code>fsafe.srm.timeout</code> , for all resources configured to be monitored on this node, since a data collection reset (via <code>fsafe.control.reset_srm</code>).
<code>fsafe.srm.histo_80</code> <code>fsafe.srm.all.histo_80</code>	Fraction of monitoring events that have been received within 60-80% of the response time from 0 milliseconds to <code>fsafe.srm.timeout</code> , for all resources configured to be monitored on this node, since the time <code>ha_srmd</code> has started.
<code>fsafe.srm.recent.histo_80</code>	Fraction of monitoring events that have been received within 60-80% of the response time from 0 milliseconds to <code>fsafe.srm.timeout</code> , for all resources configured to be monitored on this node, since a data collection reset (via <code>fsafe.control.reset_srm</code>).
<code>fsafe.srm.histo_100</code> <code>fsafe.srm.all.histo_100</code>	Fraction of monitoring events that have been received within 80-100% of the response time from 0 milliseconds to <code>fsafe.srm.timeout</code> , for all resources configured to be monitored on this node, since the time <code>ha_srmd</code> has started.
<code>fsafe.srm.recent.histo_100</code>	Fraction of monitoring events that have been received within 80-100% of the response time from 0 milliseconds to <code>fsafe.srm.timeout</code> , for all resources configured to be monitored on this node, since a data collection reset (via <code>fsafe.control.reset_srm</code>).
<code>fsafe.srm.frac_timeouts</code> <code>fsafe.srm.all.frac_timeouts</code>	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
<code>fsafe.srm.recent.frac_timeouts</code>	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 <code>fsafe.control.reset_srm</code>).
<code>fsafe.srm.frac_cumm_timeouts</code> <code>fsafe.srm.all.frac_cumm_timeouts</code>	Fraction of cumulative number of monitoring events that have timed out, for all resources configured to be monitored on this node, since the time <code>ha_srm</code> has started.
<code>fsafe.srm.recent.frac_cumm_timeouts</code>	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 <code>fsafe.control.reset_srm</code>).
<code>fsafe.srm.recent.timestamp</code>	The time when a new collection of statistics was started for the <code>fsafe.srm.recent.*</code> metrics, after issuing a store to the metric <code>fsafe.control.reset_srm</code> .
<code>fsafe.config.clustername</code>	The name of this cluster.
<code>fsafe.config.hostname</code>	The name of all hosts in the cluster specified by <code>fsafe.config.clustername</code> .
<code>fsafe.config.nnodes</code>	Number of nodes in the cluster specified by <code>fsafe.config.clustername</code> .
<code>fsafe.config.cms.interval</code>	The cluster heartbeat event interval, in milliseconds.
<code>fsafe.config.cms.timeout</code>	The heartbeat event timeout for all nodes in the cluster, in milliseconds.
<code>fsafe.config.cms.nbuckets</code>	The number of heartbeat event response intervals per node, where each interval covers a time equal to the heartbeat event interval (<code>fsafe.config.cms.interval</code>) for segments of time until the heartbeat event timeout (<code>fsafe.config.cms.timeout</code>).
<code>fsafe.control.debug</code>	<p>Debugging flags for the <code>fsafe</code> PMDA when a decimal integer value is stored to this metric. It ultimately affects what information is put into the <code>fsafe</code> PMDA's log (normally at <code>/var/adm/pcplog/fsafe.log</code>).</p> <p>Reading this metric will return the currently assigned debugging flags as a decimal integer.</p>

Metric	Description
<code>fsafe.control.reset_cms</code>	<p>Resets data collection statistics for all metrics gathered from <code>ha_cmsd</code>. When this metric is stored to, the data provided is ignored; it is the act of storing to this metric which causes the reset.</p> <p>Reading this metric will return zero (0).</p>
<code>fsafe.control.reset_srm</code>	<p>Resets data collection statistics for all metrics gathered from <code>ha_srmd</code>. When this metric is stored, the data provided is ignored; it is the act of storing to this metric which causes the reset.</p> <p>Reading this metric will return zero (0).</p>
<code>fsafe.control.retry</code>	<p>Sets the number of retries permitted when contacting <code>ha_cmsd</code> or <code>ha_srmd</code>, and when the daemons indicate that they are busy.</p> <p>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).</p> <p>Note that setting this metric does not alter how the <code>fsafe</code> PMDA handles more serious errors from <code>ha_cmsd</code> or <code>ha_srmd</code>.</p> <p>Reading this metric will return the current retry count.</p>
<code>fsafe.cms.expected</code>	<p>The number of heartbeat events expected to have been received for each node in the cluster (excluding the collector host), since the time <code>ha_cmsd</code> has started.</p>
<code>fsafe.cms.recent.expected</code>	<p>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 <code>fsafe.control.reset_cms</code>).</p>

Metric	Description
<code>fsafe.cms.received</code>	The number of heartbeat events actually received for each node in the cluster (excluding the collector host), since the time <code>ha_cmsd</code> has started.
<code>fsafe.cms.recent.received</code>	The number of heartbeat events actually received for each node in the cluster (excluding the collector host), since a data collection reset (via <code>fsafe.control.reset_cms</code>).
<code>fsafe.cms.missed</code>	The number of heartbeat events determined not to have been received for each node in the cluster (excluding the collector host), since the time <code>ha_cmsd</code> has started.
<code>fsafe.cms.recent.missed</code>	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 <code>fsafe.control.reset_cms</code>).
<code>fsafe.cms.histo</code>	<p>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 <code>ha_cmsd</code> has started.</p> <p>The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (<code>fsafe.config.cms.interval</code>), for a number of intervals up to the configured heartbeat event timeout (<code>fsafe.config.cms.timeout</code>).</p>
<code>fsafe.cms.recent.histo</code>	<p>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 <code>fsafe.control.reset_cms</code>).</p> <p>The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (<code>fsafe.config.cms.interval</code>), for a number of intervals up to the configured heartbeat event timeout (<code>fsafe.config.cms.timeout</code>).</p>
<code>fsafe.cms.frac_received</code>	Fraction of heartbeat events received over all expected events for each node in the cluster, since the time <code>ha_cmsd</code> has started.

Metric	Description
<code>fsafe.cms.recent.frac_received</code>	Fraction of heartbeat events received over all expected events for each node in the cluster, since a data collection reset (via <code>fsafe.control.reset_cms</code>).
<code>fsafe.cms.frac_missed</code>	Fraction of heartbeat events determined not to have been received over all expected events for each node in the cluster, since the time <code>ha_cmsd</code> has started.
<code>fsafe.cms.recent.frac_missed</code>	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 <code>fsafe.control.reset_cms</code>).
<code>fsafe.cms.recent.timestamp</code>	The time when a new collection of statistics was started for the <code>fsafe.cms.recent.*</code> metrics, after issuing a store to the metric <code>fsafe.control.reset_cms</code> .
<code>fsafe.cms.pernode.expected</code>	The number of heartbeat events expected to have been received for a particular node in the cluster, since the time <code>ha_cmsd</code> has started.
<code>fsafe.cms.recent.pernode.expected</code>	The number of heartbeat events expected to have been received for a particular node in the cluster, since a data collection reset (via <code>fsafe.control.reset_cms</code>).
<code>fsafe.cms.pernode.received</code>	The number of heartbeat events actually received for a particular node in the cluster, since the time <code>ha_cmsd</code> has started.
<code>fsafe.cms.recent.pernode.received</code>	The number of heartbeat events actually received for a particular node in the cluster, since a data collection reset (via <code>fsafe.control.reset_cms</code>).
<code>fsafe.cms.pernode.missed</code>	The number of heartbeat events determined not to have been received for a particular node in the cluster, since the time <code>ha_cmsd</code> has started.
<code>fsafe.cms.recent.pernode.missed</code>	The number of heartbeat events determined not to have been received for a particular node in the cluster, since a data collection reset (via <code>fsafe.control.reset_cms</code>).

E: Metrics Exported by PCP for FailSafe

Metric	Description
<code>fsafe.cms.pernode.histo</code>	<p>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 <code>ha_cmsd</code> has started.</p> <p>The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (<code>fsafe.config.cms.interval</code>), for a number of intervals up to the configured heartbeat event timeout (<code>fsafe.config.cms.timeout</code>).</p>
<code>fsafe.cms.recent.pernode.histo</code>	<p>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 <code>fsafe.control.reset_cms</code>).</p> <p>The heartbeat response intervals are defined to be equal to the configured heartbeat event interval (<code>fsafe.config.cms.interval</code>), for a number of intervals up to the configured heartbeat event timeout (<code>fsafe.config.cms.timeout</code>).</p>
<code>fsafe.cms.pernode.frac_received</code>	<p>Fraction of heartbeat events received over all expected events for a particular node in the cluster, since the time <code>ha_cmsd</code> has started.</p>
<code>fsafe.cms.recent.pernode.frac_received</code>	<p>Fraction of heartbeat events received over all expected events for a particular node in the cluster, since a data collection reset (via <code>fsafe.control.reset_cms</code>).</p>
<code>fsafe.cms.pernode.frac_missed</code>	<p>Fraction of heartbeat events determined not to have been received over all expected events for a particular node in the cluster, since the time <code>ha_cmsd</code> has started.</p>
<code>fsafe.cms.recent.pernode.frac_missed</code>	<p>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 <code>fsafe.control.reset_cms</code>).</p>

System Messages

This appendix discusses the following:

- "SYSLOG Messages"
- "Log File Error Messages", page 409

SYSLOG Messages

FailSafe logs both normal operations and critical errors to SYSLOG, as well as to individual log files for each log group.

In general, errors in the SYSLOG file take the following form:

timestamp priority_&_facility : hostname process[ID]: <internal_info> CODE message_text

For example:

```
Sep  7 11:12:59 6X:fs0 cli[5830]: < E clconf 0> CI_IPCERR_NOSERVER, clconf
ipc: ipccInt_connect() failed, file /var/cluster/ha/comm/clconfd-ipc_fs0
```

The following table shows the parts of the preceding SYSLOG message.

Table F-1 SYSLOG Error Message Format

Content	Part	Meaning
Sep 7 11:12:59	Timestamp	September 7 at 11:12 AM.
6X	Facility and level	6X indicates an informational message. See the <code>syslogd(1M)</code> man page and the file <code>/usr/include/sys/syslog.h</code> .
fs0	Node name	The node whose logical name is <code>fs0</code> is the node on which the process is running.
cli[5830]	Process[ID]	The process sending the message is <code>cli</code> and its process ID number is 5830.
<CI>E clconf 0	Internal information: message source, logging subsystem, and thread ID	The message is from the cluster infrastructure (CI). <code>E</code> indicates that it is an error. The <code>clconf</code> command is the logging subsystem. <code>0</code> indicates that it is not multithreaded.
CI_IPCERR_NOSERVER, clconf ipc	Internal error code	Information about the type of message; in this case, a message indicating that the server is missing. No error code is printed if it is a normal message.
ipcclnt_connect() failed, file /var/cluster/ha/comm/clconfd-ipc_fs0	Message text	A connection failed for the <code>clconfd-ipc_fs0</code> file.

Some of the following sections present only the message identifiers and text.

Normal Messages After Successfully Starting HA Services

When you start HA services successfully, you will see a series of message. The following shows messages you would see after successfully starting HA services for nodes named hans1 and hans2 (line breaks added for readability):

```
Aug 14 15:01:23 6X:hans1 ha_cmsd[6431]: < N cms 0> FailSafe: ha_cmsd process started.
Aug 14 15:01:24 6X:hans1 ha_ifd[6430]: < N ifd 0> FailSafe: ha_ifd monitoring network interfaces
Aug 14 15:02:00 6X:hans1 ha_cmsd[6431]: < N cms 0> FailSafe Node Confirmed Membership: sqn 1 G_sqn = 1,
    ack false change 5
Aug 14 15:02:00 5B:hans1 Configuration changes
Aug 14 15:02:00 5B:hans1 Membership changes
Aug 14 15:02:00 5B:hans1 node hans1 [1] : UP incarnation 69 age 1:0
Aug 14 15:02:00 5B:hans1 node hans2 [2] : UP incarnation 168 age 1:0
Aug 14 15:02:04 6X:hans1 ha_gcd[6435]: < N gcd 0> FailSafe: ha_gcd initialization complete
Aug 14 15:02:17 6X:hans1 ha_srmd[6424]: < N srm 2> FailSafe: SRM ready to accept clients
Aug 14 15:02:48 6X:hans1 ha_fsd[6408]: < N fsd 0> FailSafe initialization complete
```

cli Error Messages

For all cli messages, only the last message from the command (which begins with CLI private command failed) is meaningful. You can ignore all other cli messages.

The following are example errors from the cli daemon.

```
CI_ERR_INVALID, CLI private command: failed (Machine (fs0)
exists.)
```

You tried to create a new node definition with logical name fs0; however, that node name already exists in the cluster database. Choose a different name.

```
CI_ERR_INVALID, CLI private command: failed (IP address
(128.162.89.33) specified for control network is fs0 is assigned
to control network of machine (fs0).)
```

You specified the same IP address for two different control networks of node fs0. Use a different IP address.

CI_FAILURE, CLI private command: failed (Unable to validate hostname of machine (fs0) being modified.)

The DNS resolution of the fs0 name failed. To solve this problem, add an entry for fs0 in /etc/hosts on all nodes.

CI_IPCERR_NOPULSE, CLI private command: failed (Cluster state is UNKNOWN.)

The cluster state is UNKNOWN and the command could not complete. This is a transient error. However, if it persists, stop and restart the cluster daemons.

crsd Error Messages

The following errors are sent by the crsd daemon.

CI_ERR_NOTFOUND, No logging entries found for group crsd, no logging will take place - Database entry #global#logging#crsd not found.

No crsd logging definition was found in the cluster database. This can happen if you start cluster processes without creating the database.

CI_ERR_RETRY, Could not find machine listing.

The crsd daemon could not find the local node in the cluster database. You can ignore this message if the local node definition has not yet been created.

CI_ERR_SYS:125, bind() failed.

The sgi-crsd port number in the /etc/services file is not unique, or there is no sgi-crsd entry in the file.

CI_FAILURE, Entry for sgi-crsd is missing in /etc/services.

The sgi-crsd entry is missing from the /etc/services file.

CI_FAILURE, Initialization failed, exiting.

A sequence of messages will be ended with this message; see the messages prior to this one in order to determine the cause of the failure.

cmond Error Messages

The following errors are sent by the cmond daemon.

```
Could not register for notification.cdb_error = 7
```

An error number of 7 indicates that the cluster database was not initialized when the cluster process was started.

This may be caused if you execute the `cdbreinit` on one administration node while some other administration nodes in the pool are still running `fs2d` and already have the node listed in the database.

Do the following:

1. Execute the following command on the nodes that show the error:

```
# /usr/cluster/bin/cdb-init-std-nodes
```

This command will recreate the missing nodes without disrupting the rest of the database.

2. If the error persists, force the daemons to restart by executing the following command:

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

Verify that cmond is restarted.

3. If the error persists, reinitialize the database on just the node that is having problems.
4. If the error still persists, reinitialize all nodes in the cluster.

```
Process clconfd:343 of group cluster_cx exited, status = 3.
```

The `clconfd` process exited with status 3, meaning that the process will not be restarted by cmond. No corrective action is needed.

Process crsd:1790 of group cluster_control exited, status = 127

The crsd process exited with an error (nonzero) status. Look at the corresponding daemon logs for error messages.

fs2d Error Messages

The following errors are sent by the fs2d daemon.

Error 9 writing CDB info attribute for node
#cluster#elaine#machines#fs2#HA#status

An internal error occurred when writing to the cluster database.
Retry the operation. If the error persists, stop and restart the cluster daemons.

If the problem persists, clear the database, reboot, and re-create the database.

Error 9 writing CDB string value for node
#cluster#elaine#machines#fs2#HA#status

An internal error occurred when writing to the cluster database.
Retry the operation. If the error persists, stop and restart the cluster daemons.

If the problem persists, clear the database, reboot, and re-create the database.

Failed to update CDB for node
#cluster#elaine#HA#FileSystems#fs1#FSStatus

An internal error occurred when writing to the cluster database.
Retry the operation. If the error persists, stop and restart the cluster daemons.

If the problem persists, clear the database, reboot, and re-create the database.

Failed to update CDB for node
#cluster#elaine#machines#fs2#HA#status

An internal error occurred when writing to the cluster database.
Retry the operation. If the error persists, stop and restart the cluster daemons.

If the problem persists, clear the database, reboot, and re-create the database.

```
Machine 101 machine_sync failed with lock_timeout error
```

The `fs2d` daemon was not able to synchronize the cluster database and the `sync` process timed out. This operation will be retried automatically by `fs2d`.

ha_srmd Error Message

The following error is sent by the `ha_srmd` daemon:

```
Executable /var/cluster/ha/resource_types/template/stop does not
have execute file permission
Skipping resource type template configuration because of
configuration errors
Fix the configuration errors and send SIGHUP signal to ha_srmd
process on all nodes in the FailSafe cluster
```

The `stop` action script does not have the correct execution permission and therefore cannot be run. You must change the mode of the script to allow execute permission and then send a `SIGHUP` signal the `ha_srmd` process on each node so that `ha_srmd` will reread the resource type configuration. If `ha_srmd` finds errors in the resource type configuration, errors will be sent to the `SYSLOG` or `ha_srmd` logs. Use the following command line on each node:

```
# killall -HUP ha_srmd
```

Log File Error Messages

FailSafe maintains logs for each of the FailSafe daemons.

Log file messages take the following form:

```
daemon_log timestamp internal_process: message_text
```

For example:

```
cad_log:Thu Sep 2 17:25:06.092 cclconf_poll_clconfd: clconf_poll failed with error CI_IPCERR_NOPULSE
```

Table F-2, page 410, shows the parts in the preceding message.

Table F-2 Log File Error Message Format

Content	Part	Meaning
cad_log	Daemon identifier	The message pertains to the cad daemon
Sep 2 17:25:06.092	Timestamp and process ID	September 2 at 5:25 PM, process ID 92.
cclconf_poll_clconfd	Internal process information	Internal process information
clconf_poll failed with error CI_IPCERR_NOPULSE	Message text	The clconfd daemon could not be contacted to get an update on the cluster's status.

cad Messages

The following are examples of messages from `/var/cluster/ha/log/cad_log`:

```
ccacdb_cam_open: failed to open connection to CAM
server error 4
```

Internal message that can be ignored because the cad operation is automatically retried.

```
ccamail_cam_open: failed to open connection to CAM
server error 4
```

Internal message that can be ignored because the cad operation is automatically retried.

```
ccicdb_cam_open: failed to open connection to CAM
server error 4
```

Internal message that can be ignored because the cad operation is automatically retried.

```
cclconf_cam_open: failed to open connection to CAM
server error 4
```

Internal message that can be ignored because the cad operation is automatically retried.

```
cclconf_poll_clconfd: clconf_poll failed with error  
CI_IPCERR_NOCONN
```

The clconfd daemon is not running or is not responding to external requests. If the error persists, stop and restart the cluster daemons.

```
cclconf_poll_clconfd: clconf_poll failed with error  
CI_IPCERR_NOPULSE
```

The clconfd daemon could not be contacted to get an update on the cluster's status. If the error persists, stop and restart the cluster daemons.

```
cclconf_poll_clconfd: clconf_poll failed with error  
CI_CLCONFERR_LONELY
```

The clconfd daemon does not have enough information to provide an accurate status of the cluster. It will automatically restart with fresh data and resume its service.

```
csrm_cam_open: failed to open connection to CAM server error 4
```

Internal message that can be ignored because the cad operation is automatically retried.

```
Could not execute notification cmd. system() failed. Error:  
No child processes
```

No mail message was sent because cad could not fork processes. Stop and restart the cluster daemons.

```
error 3 sending event notification to client 0x000000021010f078
```

GUI process exited without cleaning up.

```
error 8 sending event notification to client 0x000000031010f138
```

GUI process exited without cleaning up.

cli Messages

The following are examples of messages from
`/var/cluster/ha/log/cli_Hostname:`

```
CI_CONFERR_NOTFOUND, No machines found in the CDB.
```

The local node is not defined in the cluster database.

```
CI_ERR_INVALID, Cluster (bob) not defined
```

The cluster called bob is not present in the cluster database.

```
CI_ERR_INVALID, CLI private command: failed (Cluster (bob) not defined)
```

The cluster called bob is not present in the cluster database.

```
CI_IPCERR_AGAIN, ipccnt_connect(): file  
/var/cluster/ha/comm/clconfd-ipc_fs0 lock failed - Permission  
denied
```

The underlying command line interface (CLI) was invoked by a login other than root. You should only use cmgr(1M) when you are logged in as root.

```
CI_IPCERR_NOPULSE, CLI private command: failed (Cluster state  
is UNKNOWN.)
```

The cluster state could not be determined. Check if the clconfd(1M) daemon is running.

```
CI_IPCERR_NOPULSE, ipccnt_pulse_internal(): server failed to  
pulse
```

The cluster state could not be determined. Check if the clconfd(1M) daemon is running.

```
CI_IPCERR_NOSERVER, clconf ipc: ipccnt_connect() failed, file  
/var/cluster/ha/comm/clconfd-ipc_fs0
```

The local node (fs0) is not defined in the cluster database.

```
CI_IPCERR_NOSEVER, Connection file
/var/cluster/ha/comm/clconfd-ipc_fs0 not present.
```

The local node (fs0) is not defined in the cluster database.

crsd Errors

The following are examples of messages
from /var/cluster/ha/log/crsd_*Hostname*:

```
CI_CONFERR_INVALID, Nodeid -1 is invalid.
I_CONFERR_INVALID, Error from ci_security_init().
CI_ERR_SYS:125, bind() failed.
CI_ERR_SYS:125, Initialization failed, exiting.
CI_ERR_NOTFOUND, Nodeid does not have a value.
CI_CONFERR_INVALID, Nodeid -1 is invalid.
```

For each of these messages, either the node ID was not provided in the node definition or the cluster processes were not running in that node when node definition was created in the cluster database. This is a warning that optional information is not available when expected.

```
CI_ERR_NOTFOUND, SystemController information for node fs2 not
found, requests will be ignored.
```

System controller information (optional information) was not provided for node fs2. Provide system controller information for node fs2 by modifying node definition. This is a warning that optional information is not available when expected. Without this information, the node will not be reset if it fails, which might prevent the cluster from properly recovering from the failure.

```
CI_ERR_NOTFOUND, SystemController information for node fs0 not
found, requests will be ignored.
```

The owner node specified in the node definition for the node with a node ID of 101 has not been defined. You must define the owner node.

```
CI_CRSEERR_NOTFOUND, Reset request 0x10087d48 received for node
101, but its owner node does not exist.
```

The owner node specified in the node definition for the node with a node ID of 101 has not been defined. You must define the owner node.

fs2d Errors

The following are examples of messages from
`/var/cluster/ha/log/fs2d_Hostname:`

```
Failed to copy global CDB to node fs1 (1), error 4
```

There are communication problems between the local node and node fs2. Check the control networks of the two nodes.

```
Communication failure send new quorum to machine fs2 (102)
(error 6003)
```

There are communication problems between the local node and node fs2. Check the control networks of the two nodes.

```
Failed to copy CDB transaction to node fs2 (1)
```

There are communication problems between the local node and node fs2. Check the control networks of the two nodes.

```
Outgoing RPC to Hostname : NULL
```

If you see this message, check your Remote Procedure Call (RPC) setup. For more information, see the `rpcinfo(1M)` and `portmap(1M)` man pages .

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.

administration node

A node in the pool that is installed with the `cluster_admin.sw.base` software product, allowing the node to perform cluster administration tasks and contain a copy of the cluster database. Within the context of CXFS, there are two types of administration nodes: server-capable administration nodes and client administration nodes.

cluster

A *cluster* is the set of systems (nodes) configured to work together as a single computing resource. A cluster is identified by a simple name and a cluster ID.

There is only one cluster that may be formed from a given pool of nodes.

Disks or logical units (LUNs) are assigned to clusters by recording the name of the cluster on the disk (or LUN). Thus, if any disk is accessible (via a Fibre Channel connection) from machines in multiple clusters, then those clusters must have unique names. When members of a cluster send messages to each other, they identify their cluster via the cluster ID. Thus, if two clusters will be sharing the same network for communications, then they must have unique cluster IDs.

Because of the above restrictions on cluster names and cluster IDs, and because cluster names and cluster IDs cannot be changed once the cluster is created (without deleting the cluster and recreating it), SGI advises that you choose unique names and cluster IDs for each of the clusters within your organization. Clusters that share a network and use XVM must have unique names.

cluster administration node

A node in a coexecution cluster that is installed with the `cluster_admin` software product, allowing the node to perform cluster administration tasks and contain a copy of the cluster database. Also known as a *CXFS administration node*.

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 database 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. Also known as *user-space membership* and *fs2d membership*.

cluster ID

A unique number within your network in the range 1 through 128. The cluster ID is used by the IRIX kernel to make sure that it does not accept cluster information from any other cluster that may be on the network. The kernel does not use the database for communication, so it requires the cluster ID in order to verify cluster communications. This information in the kernel cannot be changed after it has been initialized; therefore, you must not change a cluster ID after the cluster has been defined.

cluster node

A node that is defined as part of the cluster. See also *node*.

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.

CXFS client administration node

A node that is installed with the `cluster_admin` software product, allowing the node to perform cluster administration tasks and contain a copy of the cluster database, but is not capable of coordinating CXFS metadata. FailSafe can run on a CXFS client-administration node.

CXFS client-only node

A node that is installed with the `cxfs_client.sw.base` software product; it does not run cluster administration daemons and is not capable of coordinating cluster activity and metadata. FailSafe cannot run on a client-only node.

CXFS server-capable administration node

A node in a coexecution cluster that is installed with the `cluster_admin` product and is also capable of coordinating CXFS metadata. FailSafe can run on a CXFS server-capable administration node.

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*.

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 *node* is an operating system (OS) image, usually an individual computer. (This use of the term *node* does not have the same meaning as a node in an SGI Origin 3000 or SGI 2000 system.)

A given node can be a member of only one pool (and therefore) only one cluster.

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 *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.)

A pool is formed when you connect to a given node and define that node in the cluster database using the CXFS GUI or `cmgr` command. You can then add other nodes to the pool by defining them while still connected to the first node, or to any other node that is already in the pool. (If you were to connect to another node and then define it, you would be creating a second pool).

port password

The password for the system controller port, usually set once in firmware or by setting jumper wires. (This is not the same as the node's `root` password.)

powerfail mode

When powerfail mode is turned on, 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*.

server-capable administration node

See *CXFS server-capable administration node*

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.

tiebreaker node

A node identified as a tiebreaker 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 tiebreaker node is not specified, FailSafe will use the node with the lowest node ID in the cluster as the tiebreaker 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).

Index

A

- action script
 - execution of, 349
 - set of, 12
- action script timeouts, modifying, 171
- activate FailSafe, 206
- ACTIVE cluster status, 257
- active/backup configuration, 34
- add nic, 126
- add/remove
 - dependencies for a resource definition, 185
 - dependencies for a resource type, 165
 - nodes in the cluster, 131
- administration daemon, 354
- administration node, 47
- administration of FailSafe, 20
- aliasing IP addresses, 21
- alternate path failover, 307
- application failover domain, 11
- application monitoring, 19
- ATM LAN emulation failover, 15
- attributes for failover, 12
- Auto_Failback failover attribute, 12, 192
- Auto_Recovery failover attribute, 192
- AutoLoad boot parameter, 59, 67

B

- blue text, 92
- broadcast address, 178

C

- cad

- messages, 410
- process, 110, 318
- verify it is running, 318
- CAD options file, 62
- CD contents, 389
- cdbreinit command, 320
- chkconfig, 72, 110
- clconfd process, 110
- CLI
 - See "cmgr", 98
- cli
 - errors, 405
 - log, 214
 - messages, 412
- client administration node, 127
- client-only node, 47
- cluster
 - convert, 148
 - define, 143
 - delete, 149
 - display, 151
 - environment, 2
 - error recovery, 310
 - membership, 4
 - modify, 147
 - process group, 13
 - status, 257
 - tasks, 142
 - terminology, 2
 - type, 47
- cluster administration daemon, 354
- cluster database
 - backup/restore, 281
 - maintenance, 317
 - membership, 4
 - recovery, 317, 319
 - security, 189

- sync failure, 317
- terminology, 4
- Cluster Manager GUI
 - See "GUI", 87
- cluster_admin subsystem, 340, 342
- cluster_control subsystem, 340, 342
- cluster_mgr command, 97
- cluster_services subsystem, 340, 342
- cluster_status, 256
- cmgr
 - c option, 102
 - command line execution, 102
 - exiting, 100
 - help, 98
 - invoking a shell, 108
 - overview, 97
 - p option, 98
 - prompt mode, 98
 - scripts and, 102
 - See cmgr, 98
 - startup script, 101
 - template files, 107
- cmgr-templates directory, 107
- CMGR_STARTUP_FILE environment variable, 101
- cmond
 - errors, 407
 - process, 110, 318
 - verify it is running, 318
- cmond.options file, 66
- coexecution with CXFS, 46
- collector host installation, 79
- command-line editing in cmgr, 100
- communication paths, 343
- components, 353
- concepts, 2
- configuration overview, 26
- configuration parameters
 - disks, 36
 - filesystem, 41
 - IP address, 45
 - local XVM volumes, 52
 - XLV logical volumes, 39
- configuration planning
 - disk, 30
 - filesystem, 39
 - IP address, 42
 - logical volume, 36
 - overview, 27
- configuration tasks
 - add/remove
 - dependencies for a resource definition, 185
 - dependencies for a resource type, 165
 - nodes in the cluster, 131
 - resources in resource group, 204
 - cluster definition, 143
 - cluster tasks, 142
 - connectivity test, 292
- convert
 - CXFS cluster, 148
 - CXFS node to FailSafe, 137
- custom
 - failure detection, 120
 - resource failover behavior, 121
 - resource group failover behavior, 120
- defaults in cmgr, 113
- define
 - custom resource, 119
 - failover policy, 190
 - node, 122
 - resource, 175
 - resource group, 201
 - resource type, 152
- delete
 - cluster, 149
 - failover policy, 199
 - node, 138
 - resource, 188
 - resource group, 204
 - resource type, 173
- display
 - cluster, 151
 - failover policy, 200
 - node, 141

- resource, 189
- resource group, 205
- resource type, 174
- failover policy, 190
- fix or upgrade cluster nodes, 118
- guided configuration, 114
- HA services tasks, 206
- load a resource type, 168
- log groups, 217
- make changes to existing cluster, 118
- modify
 - cluster definition, 147
 - failover policy definition, 196
 - node definition, 132
 - resource definition, 187
 - resource group definition, 203
 - resource type, 168
- monitoring intervals, 112
- mount a filesystem, 147
- name restrictions, 112
- node
 - addition to cluster, 132
 - deletion, 139
 - resets, 280
 - tasks, 122
- notify administrator of cluster changes, 143
- optimize node usage, 119
- preliminary steps, 109
- redefine
 - resource for a specific node, 183
 - resource type for a specific node, 161
- resource group
 - tasks, 201
- resource load redistribution, 121
- resource tasks, 174
- resource type
 - tasks, 152
- set FailSafe HA parameters, 211
- set log configuration, 213
- set up
 - existing CXFS cluster for FailSafe, 117
 - HA resource group, 116
 - new cluster, 115
 - start
 - HA services, 206
 - stop HA services, 208
 - timeout values, 112
 - connectivity test, 291, 292
 - control network, 6, 15, 123
 - changing in cluster, 327
 - recovery, 316
 - Controlled_Failback failover attribute, 12, 192
 - controllers, 16
 - conversion between CXFS and FailSafe, 51
 - convert
 - FailSafe cluster, 117, 148
 - FailSafe node, 117, 137
 - FailSafe task, 117
 - corepluspid system parameter, 66
 - create a cluster, 143
 - Critical_RG failover attribute, 193
 - crsd
 - errors, 406, 413
 - log, 214
 - process, 110, 318, 342
 - verify it is running, 318
 - ctrl+c ramifications, 248
 - custom
 - failover scripts, 12
 - failure detection, 120
 - resource, 119
 - resource failover behavior, 121
 - resource group failover behavior, 120
 - CXFS, 341
 - configuration example, 243
 - exporting filesystems, 244
 - FailSafe and, 243
 - GUI, 50
 - membership, 5
 - metadata servers and failover domain, 48
 - node types, 47
 - resource type, 339
 - resource type for FailSafe, 48

D

- daemons, 110, 318
- database membership
 - See "cluster database membership", 4
- deactivate FailSafe HA services, 208, 279
- defaults, 113
- define
 - cluster, 143
 - custom resource, 119
 - failover policy, 190
 - node, 122
 - resource, 175
 - resource group, 201
 - resource type, 152
- delete
 - cluster, 149
 - failover policy, 199
 - node, 138
 - resource, 188
 - resource group, 204
 - resource type, 173
- dependency list, 9
- deskside storage systems, 15
- destructive mode, 297
- details area, 91
- detect failures, 120
- developer's guide, 12
- devname-group
 - XLV, 39
- devname-mode
 - XLV, 39
- devname-owner
 - XLV, 39
- devname_group
 - local XVM, 52
- devname_mode
 - local XVM, 52
- devname_owner
 - local XVM, 52
- diagnostic command overview, 291
- diags log, 214

- diags_nodename log file, 291
- DISCOVERY state, 262
- disk
 - configuration planning, 30
 - connections, 16
 - storage, 15
 - storage types, 23
- display
 - cluster, 151
 - failover policy, 200
 - nodes, 141
 - resource, 189
 - resource group, 205
 - resource type, 174
- DMF resource type, 340
- DNS, 60
- domain, 11, 194
- domain name service , 60
- DOWN node state, 264
- driver subsystems, 390
- dual controllers, 16
- dual hubs, 16
- dual pathing, 16
- dual vaults, 16
- dual-active, 377
- dynamic management, 19

E

- Edit menu, 91
- EL-16, 15
- emulation failover, 15
- ERROR cluster status, 257
- error state, resource group, 262
- ESP, 271
 - /etc/config/cad.options file, 62
 - /etc/config/cmond.options file, 66
 - /etc/config/fs2d.options file, 63
 - /etc/config/netif.options, 60
 - /etc/config/nfsd.options, 40

- /etc/config/routed.options, 71
- /etc/failover.conf, 308
- /etc/fstab, 40, 41
- /etc/hosts, 60
- /etc/hosts file, 42
- /etc/inetd.conf, 318
- /etc/inittab, 73
- /etc/nsswitch.conf, 60, 73
- /etc/services file, 61
- /etc/sys_id, 60
- Etherlite multiplexer driver subsystems, 390
- Ethernet, 15
- examples
 - add
 - node, 148, 325
 - resource groups or resources in an active cluster, 331
 - add/remove
 - dependencies for a resource type, 167
 - administrative communication
 - within one node, 343
 - within one node under coexecution, 348
 - attributes, 155
 - bring a resource group online, 274
 - change the log level, 218
 - chkconfig, 72
 - cluster information display, 102
 - cluster_status, 256
 - commands used to create NFS, CXFS and statd_unlimited resources, 49
 - communication
 - between nodes in the pool, 346
 - for a node not in the cluster, 346
 - configuration planning process, 29
 - configuration types, 17
 - configuration with four resource groups, 29
 - configuration with two resource groups, 32
 - configure network interfaces, 69
 - convert
 - cluster, 149
 - node, 138
 - create a resource group, 245
 - daemon communication
 - within one node, 345
 - within one node under coexecution, 349
 - define
 - a cluster, 145
 - a resource, 180
 - a resource group, 202
 - failover policy, 196
 - log group, 217
 - node, 130
 - resource type, 157
 - delete
 - cluster, 150
 - node, 139, 326
 - resource group, 204
 - dependencies, 10
 - dependencies for a resource type, 166
 - detach a resource group, 311, 326
 - determine hostname, 111
 - display
 - cluster, 151
 - resource group, 205
 - /etc/config/cad.options, 62
 - /etc/config/fs2d.options, 65
 - /etc/config/routed.options, 71
 - /etc/hosts contents and hostname resolution, 60
 - /etc/inittab, 73
 - /etc/nsswitch.conf, 60, 73
 - /etc/services, 62, 324
 - export CXFS filesystems, 244
 - failover domain, 195
 - FailSafe configuration, 236
 - FailSafe Manager GUI, 95
 - FailSafe membership, 5, 6
 - failure of a resource's monitor action, 154
 - filesystem configuration, 41
 - filesystems and logical volumes, 42
 - GUI showing details for a resource, 95
 - HA IP address configuration, 45
 - haStatus, 265
 - heartbeat response statistics, 334

- heterogeneous clusters for an IP_address
 - resource, 183
- increase the statd_unlimited resource type
 - monitor executable timeout, 171
- interface configuration, 69
- Local failover of HA IP address, 242
- log file
 - management, 285
 - name, 214
- logging information and
 - /etc/config/fs2d.options, 65
- logical volume configuration, 38
- message paths for action scripts and failover
 - policy scripts, 352
- modify
 - cluster, 139
 - cluster to include a CXFS filesystem, 243
 - node, 138
 - resource group, 203
 - resource type timeouts, 172
- monitoring system status, 256
- move a resource group, 278, 313, 330
- mutual dependency of resources is not allowed, 186
- name of a filesystem resource, 176
- network reset model, 17
- nodes in the failover domain, 194
- non-shared disk configuration and failover, 32
- offline a resource group, 313
- offline detach, 254
- output when just the initial daemons are running, 111
- partition ID determination, 124
- partitioning, 135
- patch installation, 74
- PCP, 334
- pool and cluster concepts, 2
- prompt mode, 98
- redefine resource for a specific node, 183
- remove the error state of a resource, 316
- reset models, 17
- resource
 - defining, 176
 - dependency, 185
 - monitoring statistics, 335
- resource group, 8
 - maintenance and error recovery, 312
 - recovery, 311
- resource type dependencies, 10
- ring configuration, 17
- script files, 103
- script to define a three-node cluster, 236
- server-to-server reset model, 17
- set group ID, 181
- setting configuration defaults for cmgr, 113
- sgi-cad, 62
- sgi-cmsd, 62
- sgi-crsd, 62
- sgi-gcd, 62
- shared disk configuration
 - active/backup use, 34
 - dual-active use, 35
- show
 - cluster, 139
 - failover policy, 249
 - nodes in pool/cluster, 142
- single node, 249
- software layers, 341
- star configuration, 17
- start
 - cluster daemons, 110
 - HA services, 207, 325, 329
 - HA services on a subset of nodes, 207
- startup script, 101
- stop HA services, 210, 312, 326
- system components, 14
- test
 - a resource, 295
 - failover policy, 299
 - logical volumes, 296
 - multiple nodes, 293
 - network connectivity, 294
 - private network interface, 83

- resource type, 296
- serial connections, 293
- serial reset connection, 83
- three-node cluster, 235
- two-node
 - cluster, 19
 - configuration, 17
 - resuming use of, 252
- updating from FailSafe 1.2, 375
- upgrades, 379
 - software in an active cluster, 330
 - /usr/lib/aliases, 72
 - verify cluster daemons are running, 318
 - verify that chkconfig flags are on, 110
 - XLV naming scheme, 37
- exclusive action script, 12

F

- failover, 11
 - and recovery processes, 25
 - behavior of a resource, 121
 - behavior of a resource group, 120
 - description, 25
 - resource group, 273
- failover attributes, 12, 191, 197
- failover domain, 11, 194
- failover policy, 11
 - define, 190
 - delete, 199
 - display, 200
 - failover attributes, 191, 197
 - failover domain, 194
 - failover script, 197
 - modification, 196
 - modify, 196
 - tasks, 190
 - test, 293, 299
- failover script, 12, 197, 349
- failover.conf, 308
- FailSafe base software, 340

- FailSafe Manager
 - See "GUI", 90
- FailSafe membership, 4, 304
- failsafe2 subsystem, 342
- failure detection, 120
- fake_owner, 52
- FDDI, 15
- features, 19
- Fibre Channel, 15
- Fibre Channel RAID storage, 23, 40
- File menu, 90, 91
- filesystem
 - configuration parameters, 41
 - configuration planning, 39
 - mounting, 147
 - resource, 176
 - resource type, 339
 - test, 297
- Find text field, 93
- fine-grain failover, 20
- fix cluster nodes, 118
- FORE Systems ATM cards and switch, 15
- fs2d
 - errors, 408, 414
 - options file, 63
 - process, 110, 318
 - verify it is running, 318
- fs2d membership
 - See "cluster database membership", 4
- fs_xvm_owner, 52
- fsafe.srm* metrics, 395
- fsdetail, 87
- fsdetail (fsmgr), 90
- fsmgr, 87, 90
- fstask, 87
- fstask (fsmgr), 90
- function of a node, 127

G

- giveaway, giveback, 378
- GUI
 - multiple instances, 90
 - overview, 87
 - problems, 318
 - recovery, 319
 - See "configuration tasks", 115
 - starting, 89
- guided configuration tasks, 90

H

- HA parameters
 - set, 211
- HA services
 - start, 206
 - stop, 208
- HA services tasks, 206
- ha.conf, 376, 378
- ha_agent log, 214
- ha_cfginfo, 378
- ha_cilog, 379
- ha_cmds log, 214
- ha_cmds process, 342
- ha_fsd log, 214
- ha_fsd process, 12, 342
- ha_gcd log, 214
- ha_gcd process, 342
- ha_get_field(), 378
- ha_get_info(), 378
- ha_ifd process, 342
- ha_ifd log, 214
- ha_ifmx2 process, 342
- ha_script log, 214
- ha_srmd
 - errors, 409
- ha_srmd log, 214
- ha_srmd process, 342
- ha_sybs2 process, 342

- hardware components, 14
- hardware device, adding to cluster, 332
- haStatus script, 265
- heartbeat interval, 211
- heartbeat network, 6, 15, 123
- help
 - for cmgr, 98
 - for GUI, 115
- help menu, 92
- high-availability infrastructure, 340
- history command in cmgr, 100
- hostname
 - control network, 123
 - determination, 111
- hosts file, 60
- hubs, 16

I

- icons and states, 258
- ifconfig, 295
- IFD
 - See Interface Agent Daemon, 343
- INACTIVE
 - cluster status, 257
 - node state, 264
- Informix resource type, 340
- informix_rdbms subsystem, 342
- infrastructure, 340
- initial cluster configuration, 114
- initial failover domain, 11, 194, 198
- INITIALIZING state, 262
- inittab file, 73
- InPlace_Recovery failover attribute, 192
- input instructions, 94
- install resource type, 168
- installation, 56
- installing patches, 74
- Interface Agent Daemon (IFD), 343
- INTERNAL ERROR state, 262

IP address
 aliasing, 21
 configuration planning, 42
 control network and, 123
 fixed, 21
 highly available, 21
 local failover, 242
 overview, 21
 planning, 28, 42
 resource, 177
 IP address resource type, 339
 is_* commands, 126
 item view
 See "details view", 91

J

Java Plug-in, 390
 java_plugin, 58
 JBOD, 15, 16, 23

K

ksh shell, 379

L

L1, 128
 L2, 128
 LAN emulation failover, 15
 layers, system software, 339
 load a resource type, 168
 load redistribution, 121
 local failover of IP address, 45, 242
 local restart, 20
 local XVM volumes
 resource attributes, 52
 log configuration, 213
 log files, 214, 303

errors, 409
 management, 285
 log levels, 214
 log messages
 debug, 304
 error, 304
 normal, 303
 syslog, 304
 warning, 304
 logical volume
 configuration planning, 36
 creation, 67
 parameters, 39
 testing, 296
 lonely state, 249
 long-running tasks, 101

M

MAC address
 impersonation, 22
 MAC address resource, 178
 MAC_address
 resource type, 339
 maintenance mode, 278
 make changes to an existing cluster, 118
 managed resources, 19
 membership, 4, 304–306
 metadata server-capable administration node, 47
 metrics exported by PCP, 395
 migrating from FailSafe 1.2, 375
 mirror disks, 23
 mirrored disks, 17, 40
 mkpart, 124, 128
 MMSC, 128
 mode, 41
 modify
 cluster, 118
 failover policy definition, 196
 node, 132

- resource definition, 187
- resource group definition, 203
- modify a resource type, 168
- monitor
 - action script, 12
 - applications, 19
 - host, 81
 - license, 81
 - system status, 256
- MONITOR ACTIVITY UNKNOWN error state, 263
- monitor-level, 41
- monitoring interval, 112
- MSC, 128
- multihosted RAID disks devices, 17
- multiplexer driver subsystems, 390

N

- name restrictions, 59, 112
- name service daemon, 60
- netif.options, 60, 70
- Netscape
 - resource type, 340
 - servers, testing with cmgr, 298
- network connectivity, 292, 294
- network information service, 60
- network interface
 - configuration, 68
 - overview, 21
- network mask, 177
- network reset model, 18
- network segment, 6
- networks, 6
- NFS and CXFS filesystems, 244
- NFS resource type, 340
- NIS, 60
- NIS database, 70
- NO AVAILABLE NODES error state, 263
- NO ERROR error state, 262
- NO MORE NODES IN AFD error message, 314
- node

- add/remove, 131
- adding to cluster, 323
- configuration, 55
- convert, 137
- define, 122
- delete, 138
- deleting from cluster, 325
- display, 141
- error recovery, 311
- function, 127
- highly available, 21
- modify, 132
- reset, 279, 305
- resets, 280
- state, 264
- states, 378
- status, 264
- tasks, 122
- term defined, 3
- timeout, 211
- type, 48
- usage optimization, 119
- wait time, 211
- NODE NOT AVAILABLE error state, 263
- NODE UNKNOWN error state, 263
- node-specific resource, 183
- node-specific resource type, 161
- Node_Failures_Only failover attribute, 193
- notification, 147
- notify administrator of cluster changes, 143
- nsadmin, 60
- nsd, 60
- nsswitch.conf, 60
- NVRAM variables, 67

O

- OFFLINE state, 261
- OFFLINE-PENDING state, 262
- ONLINE state, 261

ONLINE-MAINTENANCE state, 262, 278
 ONLINE-PENDING state, 261
 ONLINE-READY state, 261, 262, 273
 optimize node usage, 119
 Oracle resource type, 340
 oracle_rdbms subsystem, 342
 ordered failover script, 12
 overlap of resource groups, 8
 overview of FailSafe, 1

P

partition, 128, 133
 partition ID , 124
 patch installation, 74
 pathing, 16
 pcp_eoe.sw, 79, 81
 PCPMON, 81
 perform tasks, 94
 Performance Co-Pilot (PCP)
 installation, 78
 metrics, 395
 performance metrics, 81
 Performance Metrics Domain Agent (PMDA), 80
 physvol_names, 52
 pinging system controller, 265
 plexed disks, 17
 plexing, 23, 40
 plug-in
 custom, 340
 terminology, 13
 PMDA, 80
 pool, 3
 powerfail mode, 211
 preliminary configuration steps, 109
 primary node, 28, 34
 private network
 interface, 83
 use of, 6
 Privilege Manager, 89
 privilege tasks, 286

process group, 13
 programmer's guide, 12

Q

quorum, 5

R

rackmount storage systems, 15
 RAID storage, 15, 16, 23, 40
 re-MACing, 22
 dedicated backup interfaces required, 43
 determining if required, 43
 recovery
 overview, 301
 procedures, 309
 redefine
 resource for a specific node, 183
 resource type for a specific node, 161
 redistribute resource load, 121
 redundancy, 17
 Remote System Control port, 15
 remove
 dependencies for a resource definition, 185
 dependencies for a resource type, 165
 nic, 126
 node from the pool, 138
 nodes, 147
 remove resources in resource group, 204
 reset
 connection test, 83
 hardware, 15
 models, 17, 18
 nodes, 279, 305
 resource
 adding to cluster, 331
 define, 175
 delete, 188

- dependencies, 185
 - dependency list, 9
 - display, 189
 - failover behavior, 121
 - filesystem, 176
 - in resource group, 204
 - IP address, 177
 - MAC address, 178
 - modify, 187
 - name, 8
 - NFS, 245
 - node-specific, 183
 - overview, 20
 - owner, 263
 - recovery, 315
 - statd_unlimited, 180
 - status, 260
 - terminology, 7, 20
 - volume, 178
 - XVM, 179
 - resource attributes
 - local XVM volumes, 52
 - XLV logical volumes, 39
 - resource group
 - add/remove
 - resources, 204
 - adding to cluster, 331
 - bringing online, 273
 - creation example, 245
 - define, 201
 - delete, 204
 - dependencies, 204
 - detaching, 275
 - display, 205
 - error state, 262
 - failover, 273
 - failover behavior, 120
 - forcing offline, 275
 - modify, 203
 - monitoring, 278
 - moving, 277
 - recovery, 312
 - resume monitoring, 279
 - state, 261
 - status, 260
 - suspend monitoring, 278
 - taking offline, 273, 275
 - task, 116
 - tasks, 201
 - terminology, 8
 - test, 298
 - resource load redistribution, 121
 - resource tasks, 174
 - resource type
 - define, 152
 - delete, 173
 - dependencies, 165
 - dependency list, 9
 - display, 174
 - load, 168
 - modify, 168
 - NFS, 245
 - node-specific, 161
 - tasks, 152
 - terminology, 7
 - restart (local), 20
 - restart action script, 12
 - revoke privileges to access tasks, 290
 - ring configuration, 17
 - ring reset, 73
 - rotatelog, 285
 - rotating log files, 285
 - round-robin failover script, 12
 - run-time failover domain, 11, 194, 198
- ## S
- Samba resource type, 340
 - /sbin/ksh, 379
 - /sbin/sh, 379
 - scripts and cmgr, 102
 - SCSI

- bus, 15
- ID parameter, 67
- storage, 23
- security of the cluster database, 189
- selecting items to view or modify, 93, 287, 290
- serial connections, 293
- serial hardware reset connections, 15, 83
 - cable recovery, 316
 - cables, 47
 - diagnostic tests, 292
 - port configuration, 73
- server-capable administration node, 47, 127
- server-to-server reset model, 18
- set commands, 126
- set FailSafe HA parameters, 211
- set log configuration, 213
- set up new cluster, 115
- SGI 200, 14
- SGI 2000, 14
- SGI Origin 3000, 14
- sgi-cad, 62
- sgi-cmsd, 62
- sgi-crsd, 62
- sgi-gcd, 62
- sh shell, 379
- shared disk issues, 39
- shells, 378
- show
 - cluster, 151
 - node, 141
 - resource type, 174
- single controller, 16
- single hub, 16
- single pathing, 16
- single vault, 16
- single-node use, 249
- software overview, 389
- SPLIT RESOURCE GROUP (EXCLUSIVITY)
 - error state, 263, 314
 - srmd executable error , 353
 - SRMD EXECUTABLE ERROR error state, 263
 - standby, 378
- star configuration, 16
- start
 - cluster daemons, 110
 - HA services, 206
- start action script, 12, 353
- startup script for cmgr, 101
- std_unlimited resource, 180
- std_unlimited resource type, 340
- state, resource group, 261
- states and icons, 258
- status
 - cluster, 257
 - node, 264
 - resource, 260
 - resource group, 261
 - system controller, 261
 - system, overview, 255
- stop action script, 12, 353
- stop HA services, 208, 279
 - force option, 209
- storage connection, 15
- subsystems on the CD, 389
- sys_id, 60
- sysadm_failsafe2.sw.desktop, 88
- sysadmdesktop, 89
- sysctrl* commands, 126
- SYSLOG, 91
 - error messages, 403
 - events logged to, 65
- SYSLOG error messages, 304
- system configuration defaults, 113
- system controller
 - status, 261
 - types, 128
- system files, 59
- system log file, 91
- system operation considerations, 248
- system software
 - communication paths, 343
 - components, 353
 - layers, 339

system status, 255

T

takeover, takeback, 378
task access privileges, 286
tasks, 94
 See "configuration tasks", 109
Tasks menu, 92
TCP and NFS, 40
tcpmux, 318
tcpmux/sgi_sysadm, 318
template files, 107
term definitions, 94
terminology, 2
testing, 82
three-node cluster, example, 235
tiebreaker node, 211, 305
timeout values, 112
timeouts, action script, 171
TMF resource type, 340
Toolchest, 88
TP9100, 15
TP9400, 15
TP9900, 15
tree view
 See "view area", 91
troubleshooting
 cad messages, 410
 cli errors, 405
 cli messages, 412
 cmond errors, 407
 crsd errors, 406, 413
 fs2d errors, 408, 414
 GUI will not run, 318
 ha_srmd errors, 409
 log file errors, 409
 SYSLOG errors, 403
two-node
 clusters and single-node use, 249
 configuration, 19

resuming use of the cluster, 252

U

UDP, 40
Unified Name Service, 60
unit of failover, 379
UNIX process group, 13
UNKNOWN
 cluster status, 257, 310
 node state, 264
UNS, 60
UP node state, 264
updating from FailSafe 1.2, 375
upgrade cluster nodes, 118
upgrade FailSafe, 20
upgrading
 FailSafe software, 330
 OS software, 329
user privileges, 189
user-space membership
 See "cluster database membership", 4
 /usr/etc/ifconfig, 83, 295
 /usr/lib/aliases, 72

V

/var/adm/SYSLOG, 65, 91
/var/cluster/cdb/cdb/db, 378
/var/cluster/cmgr-scripts/rotatelog, 285
/var/cluster/ha directory, 354
/var/cluster/ha/common_scripts/scriptlib, 378
/var/ha/actions/common.vars, 378
/var/ha/logs, 379
/var/pcp/pmdas/fsafe, 80
vaults, 16
verify cluster daemons are running, 318
view area, 91
view cluster components, 92

view component details, 94
volume
 resource, 178
 resource type, 339
 test, 296
volume-name, 41

W

workstation subsystem software, 392
wsync mode, and NFS filesystems, 40

X

XFS
 filesystem creation, 67

 resource type, 339
XLV
 logical volume creation, 67
 parameters, 39
 plexing, 23, 40
 resource type, 339

XVM

 alternate path failover, 307
 resource, 179
 resource attributes, 52
 resource type, 339, 340

Y

ypmatch, 70