



**CXFS™ 6 Client-Only Guide for
SGI® InfiniteStorage**

007-5619-001

COPYRIGHT

© 2010 SGI. 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 SGI.

LIMITED RIGHTS LEGEND

The software described in this document is "commercial computer software" provided with restricted rights (except as to included open/free source) as specified in the FAR 52.227-19 and/or the DFAR 227.7202, or successive sections. Use beyond license provisions is a violation of worldwide intellectual property laws, treaties and conventions. This document is provided with limited rights as defined in 52.227-14.

TRADEMARKS AND ATTRIBUTIONS

Altix, CXFS, IRIX, Performance Co-Pilot, SGI, SGI ProPack, the SGI logo, Silicon Graphics, Supportfolio, and XFS are trademarks or registered trademarks of Silicon Graphics International Corp. or its subsidiaries in the United States and other countries.

Active Directory, Microsoft, Windows, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. AIX and IBM are registered trademarks of IBM Corporation. Brocade is a trademark of Brocade Communication Systems, Inc. AMD, AMD Athlon, and AMD Opteron are trademarks of Advanced Micro Devices, Inc. Apple, Leopard, Mac, Mac OS, Power Mac, Tiger, and Xserve are registered trademarks of Apple Inc. LSI Logic is a trademark or registered trademark of LSI Corporation. InstallShield is a registered trademark of InstallShield Software Corporation in the United States and/or other countries. Intel, Intel Xeon, Itanium, and Pentium are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Legato NetWorker is a registered trademark of Legato Systems, Inc. Linux is a registered trademark of Linus Torvalds in several countries. Norton Ghost is a trademark of Symantec Corporation. Novell is a registered trademark, and SUSE is a trademark of Novell, Inc. in the United States and other countries. OpenLDAP is a registered trademark of OpenLDAP Foundation. Red Hat and all Red Hat-based trademarks are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries. SANsurfer and QLogic are registered trademarks of QLogic Corporation. Solaris and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the United States and other countries. UNIX and the X device are registered trademarks of The Open Group in the United States and other countries. All other trademarks mentioned herein are the property of their respective owners.

The `lsOf` command is written by Victor A. Abell and is copyright of Purdue Research Foundation.

New Features in this Guide

Note: Be sure to read the release notes to learn about any late-breaking changes to the installation and configuration procedures.

This guide supersedes *CXFS 5 Client-Only Guide for SGI InfiniteStorage* (007-4507-019). This new guide includes the following:

- New support for the following:
 - Mac OS X Snow Leopard 10.6.2 or later
 - RHEL 4 U3
 - SGI Foundation 2
 - SGI ProPack 7
 - Windows 7
- Removal of information about the AIX, IRIX, and Solaris client-only platforms

Note: AIX, IRIX, and Solaris clients are not supported in ISSP 2.0 and 2.X releases going forward. The AIX, IRIX, and Solaris clients are still fully supported in the CXFS 5.X series in ISSP 1.X.

Record of Revision

Version	Description
001	January 2010 Original publication with the CXFS 6.0 and SGI InfiniteStorage Software Platform (ISSP) release 2.0.

Contents

About This Guide	xxiii
Prerequisites	xxiii
Related Publications	xxiii
Obtaining Publications	xxv
Conventions	xxvi
Reader Comments	xxvii
1. Introduction	1
CXFS on Client-Only Nodes	1
Client-Only Platforms	2
Client-Only Commands	2
Client-Only Installation and Configuration Overview	3
Cluster Administration	4
CXFS Client Processes	5
User Administration for CXFS	5
User and Group Quotas	5
Requirements	6
License Keys	7
Guaranteed-Rate I/O (GRIO) and CXFS	7
XVM Failover and CXFS	8
Monitoring CXFS	9
2. Best Practices for Client-Only Nodes	11
Configuration Best Practices	11
Use CXFS when Appropriate	12
007-5619-001	vii

When to Use CXFS	12
When CXFS May Be Inappropriate	13
Understand Hostname Resolution and Network Configuration Rules	14
Fix Network Issues First	15
Use a Private Network	15
Make Most Nodes Client-Only Nodes	16
Use the Correct Mix of Software Releases	16
Protect Data Integrity	17
Use a Client-Only Tiebreaker	17
Enable Forced Unmount When Appropriate	18
Configure Firewalls for CXFS Use	19
Use the LSI Drivers that Ship with the Linux OS	19
Administration Best Practices	19
Upgrade the Software Properly	20
Understand the Platform-Specific Limitations and Considerations	21
Shut Down Client-Only Nodes Properly	21
Do Not Run Backups on a Client Node	21
Use <code>cron</code> Jobs Properly	21
Repair Filesystems with Care	22
Disable CXFS Before Maintenance	22
Do Not Run Power Management Software	23
Use Fast Copying for Large CXFS Files	23
Appropriately Map Physical Device Names to XVM Physvols	23
Do Not Overfill CXFS Filesystems	24
Limit Client Accounts to 32 Groups	24
Turn Off Local XVM on Linux Nodes if Unused	24
Access the Correct Cluster at a Multiple-Cluster Site	25

3. Linux Platform	27
CXFS on Linux	28
Requirements for Linux	28
CXFS Commands on Linux	30
Log Files on Linux	30
CXFS Scripts on Linux	31
Mount Scripts	31
cxfs-reprobe Script	31
cxfs-enumerate-wwns Script	32
Limitations and Considerations for Linux	33
Using the dmi Mount Option on a SLES Node	34
Access Control Lists and Linux	35
HBA Installation for Linux	35
Preinstallation Steps for Linux	36
Adding a Private Network for Linux	36
Using CXFS GUI Connectivity Diagnostics for Linux	39
Verifying the Private and Public Networks for Linux	40
Client Software Installation for Linux	41
Installing CXFS Software for Linux	41
Installing the Performance Co-Pilot Agent	44
Verifying the Linux Installation	44
I/O Fencing for Linux	44
Start/Stop cxfs_client for Linux	45
Maintenance for Linux	47
Modifying the CXFS Software for Linux	47
Recognizing Storage Changes for Linux	48
Using cxfs-reprobe with RHEL	50

GRIO on Linux	51
XVM Failover V2 on Linux	52
Troubleshooting for Linux	53
Device Filesystem Enabled for Linux	53
The <code>cxfs_client</code> Daemon is Not Started on Linux	53
Filesystems Do Not Mount on Linux	54
Unable to use the <code>dmi</code> Mount Option	55
Large Log Files on Linux	55
<code>xfss off</code> Output from <code>chkconfig</code>	55
<code>crash</code> Dumps	56
Reporting Linux Problems	56
4. Mac OS X Platform	59
CXFS on Mac OS X	59
Requirements for Mac OS X	60
CXFS Commands on Mac OS X	60
Log Files on Mac OS X	61
Limitations and Considerations on Mac OS X	62
Configuring Hostnames on Mac OS X	63
Mapping User and Group Identifiers for Mac OS X	63
Access Control Lists and Mac OS X	64
Displaying ACLs	65
Comparing POSIX ACLs with Mac OS X ACLs	65
Editing POSIX ACLs on Mac OS X	68
Default or Inherited ACLs on Mac OS X	70
HBA Installation for Mac OS X	73
Installing the Apple HBA	73
Installing the Fibre Channel Utility for Mac OS X	73

Configuring Two or More Apple HBA Ports	74
Using point-to-point Fabric Setting for Apple HBAs	74
Preinstallation Steps for Mac OS X	75
Adding a Private Network for Mac OS X Nodes	75
Verifying the Private and Public Networks for Mac OS X	76
Disabling Power Saving Modes for Mac OS X	77
Client Software Installation for Mac OS X	77
I/O Fencing for Mac OS X	79
Start/Stop <code>cxfs_client</code> for Mac OS X	81
Maintenance for Mac OS X	82
Updating the CXFS Software for Mac OS X	82
Modifying the CXFS Software for Mac OS X	82
Removing the CXFS Software for Mac OS X	83
Recognizing Storage Changes for Mac OS X	83
Switching Between 64-bit Kernel and 32-bit Kernel on Snow Leopard	83
GRIO on Mac OS X	84
XVM Failover V2 on Mac OS X	85
Troubleshooting for Mac OS X	85
The <code>cxfs_client</code> Daemon is Not Started on Mac OS X	86
XVM Volume Name is Too Long on Mac OS X	86
Large Log Files on Mac OS X	86
Reporting Mac OS X Problems	87
5. Windows Platforms	89
CXFS on Windows	90
Requirements for Windows	91
CXFS Commands on Windows	94
Log Files and Cluster Status for Windows	94

Viewing the Log Files	95
Tuning the Verbosity of CXFS Messages to the System Event Log	95
Using the CXFS Info Window	96
Functional Limitations and Considerations for Windows	101
<i>Warning: DiskManager for Windows Vista, Windows Server 2008, and Windows 7 Destroys Data</i>	102
Windows Server 2008 Marks Newly Discovered Disks Offline	102
UNIX Perspective of CXFS for Windows	103
Windows Perspective of CXFS for Windows	104
Forced Unmount on Windows	105
Define LUN 0 on All Storage Devices for Windows XP and Windows Server 2003	105
Memory-Mapping Large Files for Windows	106
CXFS Mount Scripts for Windows	106
Norton Ghost Prevents Mounting Filesystems	106
Mapping Network and CXFS Drives	106
Windows Filesystem Limitations	106
XFS Filesystem Limitations	107
User Account Control for Windows Vista, Windows Server 2008, and Windows 7	107
Windows Disks Using DDN RAID	107
Windows Time Service Default Synchronization	108
Performance Considerations for Windows	109
Access Controls for Windows	110
User Identification for Windows	111
User Identification Mapping Methods for Windows	112
Matching Windows Users and Groups with CXFS Users and Groups	115
Enforcing Access to Files and Directories for Windows	115
Viewing and Changing File Attributes with Windows Explorer	116
Viewing and Changing File Permissions with Windows Explorer	117

Viewing and Changing File Access Control Lists (ACLs) for Windows	119
Effective Access for Windows	120
Restrictions with file ACLs for Windows	120
Inheritance and Default ACLs for Windows	121
System Tunables for Windows	123
Registry Modification	123
Default Umask for Windows	124
Maximum DMA Size for Windows	124
Memory-Mapping Coherency for Windows	125
DNLC Size for Windows	125
Mandatory Locks for Windows	126
User Identification Map Updates for Windows	127
I/O Size Issues Within the QLogic HBA	127
Command Tag Queueing (CTQ)	128
HBA Installation for Windows	129
Preinstallation Steps for Windows	130
Adding a Private Network for Windows	130
Verifying the Private and Public Networks for Windows	130
Configuring the Windows Firewall for Windows	131
Client Software Installation for Windows	132
Postinstallation Steps for Windows	140
Checking Permissions on the Password and Group Files for Windows	141
Performing User Configuration for Windows	141
I/O Fencing for Windows	142
Determining the WWPN for a QLogic Switch	143
Determining the WWPN for a Brocade Switch	145
Start/Stop the CXFS Client Service for Windows	146

Maintenance for Windows	147
Modifying the CXFS Software for Windows	147
Updating the CXFS Software for Windows	149
Removing the CXFS Software for Windows	151
Downgrading the CXFS Software for Windows	151
GRIO on Windows	152
XVM Failover V2 on Windows	153
Configuring the failover2.conf File for Windows	153
Windows XP SP2 and Windows Server 2003 R2 SP1 failover2 Example	156
Windows Server 2003 R2 SP2, Windows Vista, Windows Server 2008, and Windows 7 failover2 Example	157
Mapping XVM Volumes to Storage Targets on Windows	158
Troubleshooting for Windows	160
Verification that the CXFS Software is Running Correctly for Windows	161
Inability to Mount Filesystems on Windows	161
Access-Denied Error when Accessing Filesystem on Windows	163
Application Works with NTFS but not CXFS for Windows	164
Delayed-Write Error Dialog is Generated by the Windows Kernel	164
CXFS Client Service Does Not Start on Windows	165
CXFS Client Service Cannot Map Users other than Administrator for Windows	165
Filesystems Are Not Displayed on Windows	166
Large Log Files on Windows	167
Windows Failure on Reboot	167
NO_MORE_SYSTEM_PTES Error Message	168
Application Cannot Create File Under CXFS Drive Letter	168
Installation File Not Found Errors	169
Problems Specific to Windows Vista, Windows Server 2008, and Windows 7	169
Node Loses Membership Due to Hibernation	169

Node Appears to be in Membership But Is Not	170
Node Unable to cd to a Mounted Filesystem	170
Slow Installation	170
Reporting Windows Problems	171
Retaining Windows Information	171
Saving Crash Dumps for Windows	172
Saving Application Crash Dumps for Windows Vista, Windows Server 2008, and Windows 7	172
Generating a Crash Dump on a Hung Windows Node	173
6. Configuring Client-Only Nodes	175
Defining the Client-Only Nodes	176
Adding the Client-Only Nodes to the Cluster (GUI)	177
Defining the Switch for I/O Fencing	177
Starting CXFS Services (GUI)	179
Verifying LUN Masking	180
Mounting Filesystems	180
Unmounting Filesystems	181
Forced Unmount of CXFS Filesystems	181
Restarting the Windows Node	181
Verifying the Cluster Configuration	182
Verifying Connectivity in a Multicast Environment (Linux and Mac OS X Nodes)	182
Verifying the Cluster Status	183
Verifying the I/O Fencing Configuration	186
Verifying Access to XVM Volumes	187
7. General Troubleshooting	191
Identifying Problems	191

Is the Node Configured Correctly?	192
Is the Node in Membership?	192
Is the Node Mounting All Filesystems?	192
Can the Node Access All Filesystems?	193
Are There Error Messages?	193
What Is the Network Status?	194
What is the Status of XVM Mirror Licenses?	194
Typical Problems and Solutions	195
cdb Error in the <code>cxfs_client</code> Log	195
Unable to Achieve Membership	195
Filesystem Appears to Be Hung	196
Determining If a Client-Only Node Is Fenced	198
No HBA WWPNS are Detected	199
Membership Is Prevented by Firewalls	199
Devices are Unknown	199
Clients Cannot Join the Cluster After Relocation	200
Verifying the XVM Mirror Licenses	200
Using SGI Knowledgebase	201
Reporting Problems to SGI	201
Appendix A. Operating System Path Differences	203
Appendix B. Filesystem and Logical Unit Specifications	207
Appendix C. Mount Options Support	209
Appendix D. Error Messages	213
Could Not Start CXFS Client Error Messages	213
CMS Error Messages	213

Mount Messages	214
Network Connectivity Messages	214
Device Busy Message	214
Windows Messages	215
Glossary	217
Index	233

Figures

Figure 5-1	CXFS Info Window — Nodes Tab Display	97
Figure 5-2	CXFS Info Window — Filesystems Tab	98
Figure 5-3	CXFS Info Window — User Map Tab	99
Figure 5-4	CXFS Info Window — Group Map Tab	100
Figure 5-5	CXFS Info Window — CXFS Client Log Tab	101
Figure 5-6	Choose Destination Location	134
Figure 5-7	Enter CXFS Details	135
Figure 5-8	Active Directory Details	136
Figure 5-9	Generic LDAP Details	137
Figure 5-10	Review the Settings	138
Figure 5-11	Start CXFS Driver	139
Figure 5-12	Restart the System	140
Figure 5-13	Modify CXFS for Windows	148
Figure 5-14	Upgrading the Windows Software	150
Figure 5-15	CXFS Info Display for GRIO for Windows	152
Figure 5-16	QLogic SANsurfer (Copyright QLogic® Corporation, all rights reserved)	159

Tables

Table 1-1	CXFS Commands Available on All CXFS Client-Only Nodes	3
Table 3-1	Processor Architecture and Package Extensions	42
Table 4-1	Mac OS X Permissions Compared with POSIX Access Permissions	66
Table 5-1	Permission Flags that May Be Edited	118
Table A-1	Linux Paths	203
Table A-2	Mac OS X Paths	204
Table A-3	Windows Paths	205
Table B-1	Filesystem and Logical Unit Specifications	208
Table C-1	Mount Options Support for Client-Only Platforms	210

About This Guide

This publication documents the CXFS 6.0 release. For additional details, see the platform-specific release notes.

Prerequisites

This guide assumes the following:

- Server-capable administration nodes running the supported SGI Foundation Software and CXFS software are operational.
- The CXFS client-only nodes have the appropriate platform-specific operating system software installed.
- The reader is familiar with the information presented in the *CXFS 6 Administration Guide for SGI InfiniteStorage* and the platform's operating system and installation documentation.

Related Publications

For information about this release, see the following release notes:

- SGI InfiniteStorage Software Platform (ISSP): `README.txt`
- CXFS:
 - `README_CXFS_GENERAL.txt`
 - `README_CXFS_LINUX.txt`
 - `README_CXFS_MACOSX.html`
 - `README_CXFS_WINDOWS.html`

The following documents contain additional information:

- CXFS documentation:
 - Platform-specific release notes
 - *CXFS 6 Administration Guide for SGI InfiniteStorage*
- *SGI Foundation Software 2 Start Here*
- QLogic HBA card and driver documentation:
<http://www.qlogic.com>
- Red Hat Linux documentation:
<http://www.redhat.com/docs/manuals/enterprise>
- Novell SUSE Linux Enterprise Software (SLES) documentation:
<http://www.novell.com/documentation/suse.html>
- Apple Mac OS X documentation:
<http://support.apple.com/manuals/#macos>
- Microsoft Windows documentation:
<http://www.microsoft.com>

Note: The external websites referred to in this guide were correct at the time of publication, but are subject to change.

The following man pages are provided on CXFS Linux and Mac OS X client-only nodes:

Client-Only Man Page	Linux RPM ¹
<code>cxfs_client(8)</code>	<code>cxfs_client</code>
<code>cxfs_info(8)</code>	<code>cxfs_client</code>
<code>cxfs-config(8)</code>	<code>cxfs_util</code>
<code>cxfs scp(1)</code>	<code>cxfs_util</code>
<code>cxfsdump(8)</code>	<code>cxfs_util</code>

Obtaining Publications

You can obtain SGI documentation as follows:

- 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, man pages, and other information.
- On all but Windows systems, you can view man pages by typing `man title` at a command line.
- The `/docs` directory on the ISSP DVD or in the Supportfolio download directory contains the following:
 - The ISSP release note: `/docs/README.txt`
 - Other release notes: `/docs/README_NAME.txt`
 - A complete list of the packages and their location on the media:
`/docs/RPMS.txt`
 - The packages and their respective licenses: `/docs/PACKAGE_LICENSES.txt`

¹ For Mac OS X platforms, man pages are provided in the CXFS package.

- The release notes and manuals are provided in the `noarch/sgi-isspdocs` RPM and will be installed on the system into the following location:

`/usr/share/doc/packages/sgi-issp-ISSPVERSION-TITLE`

Conventions

This guide uses the following terminology abbreviations:

- *Linux* refers to the supported Red Hat Enterprise Linux (RHEL) or SUSE Linux Enterprise Server (SLES) as defined in the CXFS Linux release note
- *Mac OS X* refers to the supported Leopard and Snow Leopard releases as defined in the CXFS Mac OS X release note
- *Windows* refers to any of the supported levels of Microsoft Windows operating systems as defined in the CXFS Windows release note

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.
<i>variable</i>	Italic typeface denotes variable entries and words or concepts being defined.
user input	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
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.
[]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.
GUI element	This bold font denotes the names of graphical user interface (GUI) elements, such as windows, screens,

	dialog boxes, menus, toolbars, icons, buttons, boxes, and fields.
<TAB>	Represents pressing the specified key in an interactive session
server-admin#	In an example, this prompt indicates that the command is executed on a server-capable administration node
client#	In an example, this prompt indicates that the command is executed on a client-only node
MDS#	In an example, this prompt indicates that the command is executed on an active metadata server
#	In an example, this prompt indicates that the command is executed on an any node
<i>specificnode</i> #	In an example, this prompt indicates that the command is executed on a node named <i>specificnode</i> or of node type <i>specificnode</i>

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

- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.
- Send mail to the following address:

SGI
Technical Publications
46600 Landing Parkway
Fremont, CA 94538

S&I values your comments and will respond to them promptly.

Introduction

This guide provides an overview of the installation and configuration procedures for CXFS client-only nodes running SGI CXFS clustered filesystems. A *CXFS client-only node* has a minimal implementation of CXFS services that run a single daemon, the CXFS client daemon (`cxfs_client`). A cluster running multiple operating systems is known as a *multiOS cluster*.

For more information about CXFS terminology, concepts, and configuration, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.



Caution: CXFS is a complex product. To ensure that CXFS is installed and configured in an optimal manner, it is **mandatory** that you purchase SGI managed services developed for CXFS. Some of the procedures mentioned in this guide may be performed by SGI personnel or other qualified service personnel; details for these procedures are provided in other documents. Contact your local SGI sales representative for details.

This chapter discusses the following:

- "CXFS on Client-Only Nodes" on page 1
- "License Keys" on page 7
- "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7
- "XVM Failover and CXFS" on page 8
- "Monitoring CXFS" on page 9

Also see Chapter 2, "Best Practices for Client-Only Nodes" on page 11.

CXFS on Client-Only Nodes

This section contains the following:

- "Client-Only Commands" on page 2
- "Client-Only Installation and Configuration Overview" on page 3

- "Cluster Administration" on page 4
- "CXFS Client Processes" on page 5
- "User Administration for CXFS" on page 5
- "User and Group Quotas " on page 5
- "Requirements" on page 6

Client-Only Platforms

CXFS supports client-only nodes running any mixture of the following operating systems:

- Apple Mac OS X
- Red Hat Enterprise Linux (RHEL)
- SUSE Linux Enterprise Server (SLES)
- SGI Foundation Software on RHEL
- SGI Foundation Software on SLES
- Microsoft Windows

See the CXFS release notes for the supported kernels, update levels, and service pack levels.

For details, see the following:

- Chapter 3, "Linux Platform" on page 27
- Chapter 4, "Mac OS X Platform" on page 59
- Chapter 5, "Windows Platforms" on page 89

Client-Only Commands

Table 1-1 lists the CXFS commands that are installed on all client-only nodes.

Table 1-1 CXFS Commands Available on All CXFS Client-Only Nodes

Command	Description
<code>cxfs_client(8)</code>	Controls the CXFS client control daemon
<code>cxfs_info(8)</code>	Provides status information
<code>cxfsdump(8)</code>	Gathers configuration information in a CXFS cluster for diagnostic purposes
<code>xvm(8)</code>	Invokes the XVM command line interface

Also see:

- "CXFS Commands on Linux" on page 30
- "CXFS Commands on Mac OS X" on page 60
- "CXFS Commands on Windows" on page 94

Client-Only Installation and Configuration Overview

Following is the order of installation and configuration steps for a CXFS client-only node. See the *SGI InfiniteStorage Software Platform* release note and the specific operating system (OS) chapters in this guide for details:

1. Read the ISSP and CXFS release notes to learn about any late-breaking changes in the installation procedure.
2. Install the supported OS software according to the directions in the OS documentation.
3. Install and verify the RAID. See the *CXFS 6 Administration Guide for SGI InfiniteStorage* and the release notes.
4. Install and verify the switch. See the *CXFS 6 Administration Guide for SGI InfiniteStorage* and the release notes.
5. Obtain the supported CXFS server-side license keys. For more information about licensing, see "License Keys" on page 7 and *CXFS 6 Administration Guide for SGI InfiniteStorage*.

If you want to access an XVM cluster mirror volume from client-only nodes in the cluster, you must have a valid XVM cluster mirror license installed on the

server-capable administration nodes. No additional license key is needed on the client-only nodes. The client-only node will automatically acquire a mirror license key when the CXFS client service is started on the node.

6. Install and verify the host bus adapter (HBA) and driver.
7. Prepare the node, including adding a private network. See "Preinstallation Steps for Windows" on page 130.
8. Install the **SGI CXFS Clients** YaST pattern containing the CXFS client packages onto one server-capable administration node and transfer the appropriate client packages to the corresponding client-only nodes, as described in the ISSP release note.
9. Perform any required post-installation configuration steps.
10. Configure the cluster to define the new client-only node, add it to the cluster, start CXFS services, and mount filesystems. See Chapter 6, "Configuring Client-Only Nodes" on page 175.

If you run into problems, see the OS-specific troubleshooting section, Chapter 7, "General Troubleshooting" on page 191, and the troubleshooting chapter in *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Cluster Administration

A CXFS cluster must contain at least one server-capable administration node that is responsible for updating that filesystem's metadata. This node is referred to as the *CXFS metadata server*. (Client-only nodes cannot be metadata servers.) Metadata servers store information in the CXFS cluster database. The CXFS cluster database is not stored on client-only nodes; only server-capable administration nodes contain the cluster database.

To modify the cluster database, you will use the CXFS graphical user interface (GUI) or the `cxfs_admin` command from a node with the correct permissions (usually a server-capable administration node) and with `root` access. For more information about using these tools, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

CXFS Client Processes

When CXFS is started on a client-only node, a user-space daemon is started that provides the required processes. This is a subset of the processes needed on a CXFS server-capable administration node.

The `cxfs_client` daemon controls CXFS services on a client-only node. It does the following:

- Obtains the cluster configuration from a remote `fs2d` daemon and manages the local client-only node's CXFS kernel membership services and filesystems accordingly
- Obtains membership and filesystem status from the kernel

The path to the `cxfs_client` command varies among the platforms supported. See Appendix A, "Operating System Path Differences" on page 203

Note: The `cxfs_client` daemon may still be running when CXFS services are disabled.

User Administration for CXFS

A CXFS cluster requires a consistent user identification scheme across all hosts in the cluster so that one person using different cluster nodes has the same access to the files on the cluster. The following must be observed to achieve this consistency:

- Users must have the same usernames on all nodes in the cluster. An individual user identifier (UID) should not be used by two different people anywhere in the cluster. Ideally, group names and group identifiers (GIDs) should also be consistent on all nodes in the cluster.
- Each CXFS client and server node must have access to the same UID and GID information. The simplest way to achieve this is to maintain the same `/etc/passwd` and `/etc/group` files on all CXFS nodes, but other mechanisms may be supported.

User and Group Quotas

Only Linux nodes can view or edit user and group quotas. Quotas are effective on all nodes because they are enforced by the metadata server.

To view or edit quota information on a Linux node, use the `xfstool` command. This is provided by the `xfstools` RPM.

Requirements

Using a CXFS client-only node requires the following:

- A supported storage area network (SAN) hardware configuration.

Note: For details about supported hardware, see the Entitlement Sheet that accompanies the base CXFS release materials. (Using unsupported hardware constitutes a breach of the CXFS license.)

- A private 100baseT (or greater) TCP/IP network connected to each node, to be dedicated to the CXFS private heartbeat and control network. This network must not be a virtual local area network (VLAN) and the Ethernet switch must not connect to other networks. All nodes must be configured to use the same subnet.
- The appropriate license keys. See "License Keys" on page 7.
- A switch, which is required to protect data integrity on nodes without system controllers. See the release notes for supported switches.

Nodes must use I/O fencing (or system reset if available) to protect the data integrity of the filesystems in the cluster. Server-capable administration nodes should use system reset. See "Protect Data Integrity" on page 17.

- There must be at least one server-capable administration node to act as the metadata server and from which to perform cluster administration tasks. You should install CXFS software on the server-capable administration nodes first.
- Nodes that are not potential metadata servers should be CXFS client-only nodes. A cluster may contain as many as 64 nodes, of which as many as 16 can be server-capable administration nodes; the rest must be client-only nodes. See "Make Most Nodes Client-Only Nodes" on page 16.
- Set the `mtcp_nodelay` system tunable parameter to 1 on server-capable administration nodes in order to provide adequate performance on file deletes.

Also see "Requirements for Windows" on page 91, and Chapter 2, "Best Practices for Client-Only Nodes" on page 11.

License Keys

CXFS requires the following license keys:

- CXFS license keys using server-side licensing. Server-side licensing is required on all nodes.

To obtain server-side CXFS and XVM license keys, see information provided in your customer letter and the following web page:

<http://www.sgi.com/support/licensing>

The licensing used for server-capable administration nodes is based the SGI License Key (LK) software. See the general release notes and the *CXFS 6 Administration Guide for SGI InfiniteStorage* for more information.

- XVM cluster mirroring requires a license key on server-capable administration nodes in order for cluster nodes to access the cluster mirror. On CXFS client-only nodes, the user feature where applicable is honored after the `cxfs_client` service is started. XVM cluster mirroring on clients is also honored if it is enabled on the server. All CXFS client nodes need an appropriate mirror license key in order to access local mirrors.
- Guaranteed rate I/O version 2 (GRIOv2) if enabled requires a license key on the server-capable administration nodes.
- Fibre Channel switch license key. See the release notes.

Guaranteed-Rate I/O (GRIO) and CXFS

CXFS supports guaranteed-rate I/O (GRIO) version 2 clients on all platforms, and GRIO servers on server-capable administration nodes. However, GRIO is disabled by default on server-capable administration nodes and Linux client-only nodes. See "GRIO on Linux" on page 51.

Note: GRIO application reservations are functional only for Windows 32-bit nodes; they are not functional on any other nodes (Mac OS X, Windows 64-bit, or Linux, including Linux server-capable administration nodes).

Once GRIO is enabled, the superuser can run the following commands from any node in the cluster:

- `grioadmin`, which provides stream and bandwidth management
- `griogps`, which is the comprehensive stream quality-of-service monitoring tool

Run the above tools with the `-h` (help) option for a full description of all available options. See Appendix A, "Operating System Path Differences" on page 203, for the platform-specific locations of these tools.

See the platform-specific chapters in this guide for GRIO limitations and considerations:

- "GRIO on Linux" on page 51
- "GRIO on Mac OS X" on page 84
- "GRIO on Windows" on page 152

See the *Guaranteed-Rate I/O Version 2 for Linux Guide* for details about GRIO installation, configuration, and use.

XVM Failover and CXFS

XVM failover version 2 (v2) requires that the RAID be configured in AVT mode.

To configure failover v2, you must create and edit the `failover2.conf` file. For more information, see the following:

- The comments in the `failover2.conf` file on a server-capable administration node
- *CXFS 6 Administration Guide for SGI InfiniteStorage*
- *XVM Volume Manager Administrator's Guide*

This guide contains platform-specific examples of `failover2.conf` for the following:

- "XVM Failover V2 on Linux" on page 52
- "XVM Failover V2 on Mac OS X" on page 85
- "XVM Failover V2 on Windows" on page 153

Monitoring CXFS

To monitor CXFS, you can use the following:

- The `cxfs_info` command on the client
- The view area of the CXFS GUI
- The `cxfs_admin` command
- The `clconf_info` command on a CXFS server-capable administration node

For more information, see "Verifying the Cluster Status" on page 183.

Best Practices for Client-Only Nodes

This chapter discusses best-practices for client-only nodes:

- "Configuration Best Practices" on page 11
- "Administration Best Practices" on page 19

Also see the best practices information in the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Configuration Best Practices

This section discusses the following:

- "Use CXFS when Appropriate" on page 12
- "Understand Hostname Resolution and Network Configuration Rules" on page 14
- "Fix Network Issues First" on page 15
- "Use a Private Network" on page 15
- "Make Most Nodes Client-Only Nodes" on page 16
- "Use the Correct Mix of Software Releases" on page 16
- "Protect Data Integrity" on page 17
- "Use a Client-Only Tiebreaker" on page 17
- "Enable Forced Unmount When Appropriate" on page 18
- "Configure Firewalls for CXFS Use" on page 19
- "Use the LSI Drivers that Ship with the Linux OS" on page 19

Use CXFS when Appropriate

This section discusses:

- "When to Use CXFS" on page 12
- "When CXFS May Be Inappropriate" on page 13

When to Use CXFS

You should use CXFS when you have multiple hosts running applications that require high-bandwidth access to common filesystems.

CXFS performs best under the following conditions:

- Data I/O operations are greater than 16 KB
- All processes that perform read/write operations for a given file reside on the same host
- Multiple processes on multiple hosts read the same file
- Direct-access I/O is used for read/write operations for multiple processes on multiple hosts
- Large files and file accesses are being used

Applications that perform well on a client typically do the following:

- Issue large I/O requests, rather than several smaller requests
- Use asynchronous or multithreaded I/O to have several I/O requests in flight at the same time
- Minimize the number of metadata operations they perform (*metadata* is information that describes a file, such as the file's name, size, location, and permissions)

For most filesystem loads, the preceding scenarios represent the bulk of the file accesses. Thus, CXFS delivers fast local-file performance. CXFS is also useful when the amount of data I/O is larger than the amount of metadata I/O. CXFS is faster than NFS because the data does not go through the network.

When CXFS May Be Inappropriate

CXFS may not give optimal performance under the following circumstances:

- When distributed applications write to shared files that are memory-mapped.
- When extending large highly fragmented files. The metadata traffic when growing files with a large number of extents will increase as more extents are added to the file. The following I/O patterns will cause highly fragmented files:

- Random writes to sparse files
- Files generated with memory-mapped I/O
- Writing files in an order other than linearly from beginning to end

Do the following to prevent highly fragmented files:

- Create files with linear I/O from beginning to end
- Use file preallocation to allocate space for a file before writing
- When access would be as slow with CXFS as with network filesystems, such as with the following:
 - Small files.
 - Low bandwidth.
 - Lots of metadata transfer. Metadata operations can take longer to complete through CXFS than on local filesystems. Metadata transaction examples include the following:
 - Opening and closing a file
 - Changing file size (usually extending a file)
 - Creating, renaming, and deleting files
 - Searching a directory

In addition, multiple processes on multiple hosts that are reading and writing the same file using buffered I/O can be slower when using CXFS than when using a local filesystem. This performance difference comes from maintaining coherency among the distributed file buffers; a write into a shared, buffered file will invalidate data (pertaining to that file) that is buffered in other hosts.

For a high-performance NFS edge serving configuration (in which CXFS client nodes can export data with NFS), contact SGI Professional Services.

Also see "Functional Limitations and Considerations for Windows" on page 101.

Understand Hostname Resolution and Network Configuration Rules



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

The following hostname resolution rules and recommendations apply to all nodes:

- You must ensure that the hostname and IP address for each network interface in the cluster is properly configured on each client-only node and server-capable administration node.
- The first node you define must be a server-capable administration node.
- Hostnames cannot begin with an underscore (_) or include any whitespace characters.
- The private network IP addresses on a running node in the cluster cannot be changed while CXFS services are active.
- You must be able to communicate directly between every node in the cluster (including client-only nodes) using IP addresses and logical names, without routing.
- A private network must be dedicated to be the heartbeat and control network. No other load is supported on this network.
- The heartbeat and control network must be connected to all nodes, and all nodes must be configured to use the same subnet for that network.

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

To confirm network connectivity, use the following command line on a server-capable administration node:

```
server-admin# /usr/cluster/bin/cxfs-config -check -ping
```

For more information, see *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Fix Network Issues First

If there are any network issues on the private network, fix them before trying to use CXFS. A stable private network is important for a stable CXFS cluster network. Ensure that you understand the information in "Understand Hostname Resolution and Network Configuration Rules" on page 14.

When you install the CXFS software on the client-only node, you must modify certain system files. **The network configuration is critical.** Each node in the cluster must be able to communicate with every other node in the cluster by both logical name and IP address without going through any other network routing; proper name resolution is key. SGI recommends static routing.

Use a Private Network

You are required to use a private network for CXFS metadata traffic:

- The private network is used for metadata traffic and should not be used for other kinds of traffic.
- A stable private network is important for a stable CXFS cluster environment.
- Two or more clusters should not share the same private network. A separate private network switch is required for each cluster.
- The private network should contain at least a 100-Mbit network switch. A network hub is not supported and should not be used.
- All cluster nodes should be on the same physical network segment (that is, no routers between hosts and the switch).
- Use private (10.x.x.x, 176.16.x.x, or 192.168.x.x) network addresses (RFC 1918).
- The private network must be configured as the highest priority network for the cluster. The public network may be configured as a lower priority network to be used by CXFS network failover in case of a failure in the private network.
- When administering more than one CXFS cluster, use unique private network addresses for each cluster. If you have multiple clusters connected to the same public network, use unique cluster names and cluster IDs.

- A virtual local area network (VLAN) is not supported for a private network.
- When NFS or Samba serving from a CXFS cluster, the network used for remote fileserving cannot be a backup private network for CXFS. Using the fileserving network as a backup private network for CXFS private network may result in heartbeat timeouts, which will cause a severe drop in CXFS and fileserving performance.

Make Most Nodes Client-Only Nodes

You should define most nodes as client-only nodes and define just the nodes that may be used for CXFS metadata as server-capable administration nodes.

The advantage to using client-only nodes is that they do not keep a copy of the cluster database; they contact a server-capable administration node to get configuration information. It is easier and faster to keep the database synchronized on a small set of nodes, rather than on every node in the cluster. In addition, if there are issues, there will be a smaller set of nodes on which you must look for problems.

Use the Correct Mix of Software Releases

All nodes should run the same level of CXFS and the same level of operating system software, according to platform type. To support upgrading without having to take the whole cluster down, nodes can run different CXFS releases during the upgrade process.



Caution: You must upgrade all server-capable administration nodes before upgrading any client-only nodes (servers must run the same release as client-only nodes or a later release.) Operating a cluster with clients running a mixture of older and newer CXFS versions will result in a performance loss. Relocation to a server-capable administration node that is running an older CXFS version is not supported.

For details, see the platform-specific release notes and the information about rolling upgrades in *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Protect Data Integrity

All nodes must be configured to protect data integrity in case of failure. System reset or I/O fencing is required to ensure data integrity for all nodes. I/O fencing (or system reset when available) must be used on client-only nodes. I/O

You should use the `admin` account when configuring I/O fencing. On a Brocade switch running 4.x.x.x or later firmware, modify the `admin` account to restrict it to a single `telnet` session. For details, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

You must keep the `telnet` port on the switch free at all times; **do not** perform a `telnet` to the switch and leave the session connected.

SGI recommends that you use a switched network of at least 100baseT.

You should isolate the power supply for the switch from the power supply for a node and its system controller. You should avoid any possible situation in which a node can continue running while both the switch and the system controller lose power. Avoiding this situation will prevent the possibility a split-brain scenario.

You must put switches used for I/O fencing on a network other than the primary CXFS private network so that problems on the CXFS private network can be dealt with by the fencing process and thereby avoid data corruption issues. The network to which the switch is connected must be accessible by all server-capable administration nodes in the cluster.

See the following:

- "I/O Fencing for Linux" on page 44
- "I/O Fencing for Mac OS X" on page 79
- "I/O Fencing for Windows" on page 142

Use a Client-Only Tiebreaker

SGI recommends that you always define a stable client-only node as the CXFS tiebreaker for all clusters with more than one server-capable administration node and at least one client-only node.

Having a tiebreaker is critical when there are an even number of server-capable administration nodes. A tiebreaker avoids the problem of multiple-clusters being

formed (a split cluster) while still allowing the cluster to continue if one of the metadata servers fails.

As long as there is a reliable client-only node in the cluster, a client-only node should be used as tiebreaker. Server-capable administration nodes are not recommended as tiebreaker nodes because these nodes always affect CXFS kernel membership.

The tiebreaker is of benefit in a cluster even with an odd number of server-capable administration nodes because when one of the server-capable administration nodes is removed from the cluster, it effectively becomes a cluster with an even-number of server-capable administration nodes.

Note the following:

- If exactly two server-capable administration nodes are configured and there are no client-only nodes, **neither** server-capable administration node should be set as the tiebreaker. (If one node was set as the tiebreaker and it failed, the other node would also shut down.)
- If exactly two server-capable administration nodes are configured and there is at least one client-only node, you should specify the client-only node as a tiebreaker.

If one of the server-capable administration nodes is the CXFS tiebreaker in a two-server-capable-node cluster, failure of that node or stopping the CXFS services on that node will result in a cluster-wide forced shutdown. If you use a client-only node as the tiebreaker, either server-capable administration node could fail but the cluster would remain operational via the other server-capable administration node.

- If there are an even number of server-capable administration nodes and there is no tiebreaker set, the fail policy must not contain the `shutdown` option because there is no notification that a shutdown has occurred.

SGI recommends that you start CXFS services on the tiebreaker client after the server-capable administration nodes are all up and running, but before CXFS services are started on any other clients.

Enable Forced Unmount When Appropriate

Normally, an unmount operation will fail if any process has an open file on the filesystem. The *forced unmount* feature allows the unmount to proceed regardless of whether the filesystem is still in use.

If you enable the forced unmount feature for CXFS filesystems (which is turned off by default), you may be able to improve the stability of the CXFS cluster, particularly in situations where the filesystem must be unmounted. However, be aware that a forced unmount will kill running processes to unmount a filesystem, which is potentially destructive.

For more information, see "Forced Unmount of CXFS Filesystems" on page 181 and the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Configure Firewalls for CXFS Use

Do one of the following:

- Configure firewalls to allow CXFS traffic. See *CXFS 6 Administration Guide for SGI InfiniteStorage* for CXFS port usage. (Preferred for most platforms.)
- Configure firewalls to allow all traffic on the CXFS private interfaces. This assumes that the public interface is not a backup metadata network.
- Disable firewalls. (Preferred for Windows. For Vista and Windows 2008, you should check firewall settings after each reboot.)

For more information, see your firewall documentation.

Use the LSI Drivers that Ship with the Linux OS

You should use the LSI drivers that ship with the Linux operating systems. The newer drivers that are available on the LSI web site may not work and are not supported by SGI or the kernels CXFS uses in this release.

Administration Best Practices

This section discusses the following:

- "Upgrade the Software Properly" on page 20
- "Understand the Platform-Specific Limitations and Considerations" on page 21
- "Shut Down Client-Only Nodes Properly" on page 21
- "Do Not Run Backups on a Client Node" on page 21

- "Use cron Jobs Properly" on page 21
- "Repair Filesystems with Care" on page 22
- "Disable CXFS Before Maintenance" on page 22
- "Do Not Run Power Management Software" on page 23
- "Use Fast Copying for Large CXFS Files" on page 23
- "Appropriately Map Physical Device Names to XVM Physvols" on page 23
- "Do Not Overfill CXFS Filesystems" on page 24
- "Limit Client Accounts to 32 Groups" on page 24
- "Turn Off Local XVM on Linux Nodes if Unused" on page 24
- "Access the Correct Cluster at a Multiple-Cluster Site" on page 25

Upgrade the Software Properly

Do the following when upgrading the software:

- Save the current CXFS configuration as a precaution before you start an upgrade and acquire new CXFS server-side licenses (if required). See the information in the *CXFS 6 Administration Guide for SGI InfiniteStorage*.
- Read the release notes and any late-breaking caveats on Supportfolio before installing or upgrading CXFS. These notes contain useful information and caveats needed for a stable install/upgrade.
- Do not make any other configuration changes to the cluster (such as adding new nodes or filesystems) until the upgrade of all nodes is complete and the cluster is running normally.

See the following:

- "Updating the CXFS Software for Mac OS X" on page 82
- "Updating the CXFS Software for Windows" on page 149

Understand the Platform-Specific Limitations and Considerations

Each platform in a CXFS cluster has different issues. See the following:

- "Limitations and Considerations for Linux" on page 33
- "Limitations and Considerations on Mac OS X" on page 62
- "Functional Limitations and Considerations for Windows" on page 101 and "Performance Considerations for Windows" on page 109

Shut Down Client-Only Nodes Properly

When shutting down, resetting, or restarting a CXFS client-only node, do not stop CXFS services on the node. (Stopping CXFS services is more intrusive on other nodes in the cluster because it updates the cluster database. Stopping CXFS services is appropriate only for a CXFS server-capable administration node.) Rather, let the CXFS shutdown scripts on the node stop CXFS when the client-only node is shut down or restarted.

Do Not Run Backups on a Client Node

SGI recommends that you perform backups on the CXFS metadata server.

Do not run backups on a client node, because it causes heavy use of non-swappable kernel memory on the metadata server. During a backup, every inode on the filesystem is visited; if done from a client, it imposes a huge load on the metadata server. The metadata server may experience typical out-of-memory symptoms, and in the worst case can even become unresponsive or crash.

Use `cron` Jobs Properly

Jobs scheduled with `cron` can cause severe stress on a CXFS filesystem if multiple nodes in a cluster start the same filesystem-intensive task simultaneously.

Because CXFS filesystems are considered as local on all nodes in the cluster, the nodes may generate excessive filesystem activity if they try to access the same filesystems simultaneously while running commands such as `find` or `ls`. You should build databases for `rfind` and GNU `locate` only on the active metadata server.

Any task initiated using `cron` on a CXFS filesystem should be launched from a single node in the cluster, preferably from the active metadata server. Edit the nodes' `crontab` file to only execute the `find` command on one metadata server of the cluster.

Repair Filesystems with Care

Always contact SGI technical support before using `xfs_repair` on CXFS filesystems. You must first ensure that you have an actual case of filesystem corruption and retain valuable metadata information by replaying the XFS logs before running `xfs_repair`.



Caution: If you run `xfs_repair` without first replaying the XFS logs, you may introduce data corruption. You should run `xfs_ncheck` and capture the output to a file before running `xfs_repair`. If running `xfs_repair` results in files being placed in the `lost+found` directory, the saved output from `xfs_ncheck` may help you to identify the original names of the files.

Only use `xfs_repair` on server-capable administration nodes and only when you have verified that all other cluster nodes have unmounted the filesystem. Make sure that `xfs_repair` is run only on a cleanly unmounted filesystem. If your filesystem has not been cleanly unmounted, there will be uncommitted metadata transactions in the log, which `xfs_repair` will erase. This usually causes loss of some data and messages from `xfs_repair` that make the filesystem appear to be corrupted.

If you are running `xfs_repair` right after a system crash or a filesystem shutdown, your filesystem is likely to have a dirty log. To avoid data loss, you **MUST** mount and unmount the filesystem before running `xfs_repair`. It does not hurt anything to mount and unmount the filesystem locally, after CXFS has unmounted it, before `xfs_repair` is run.

Disable CXFS Before Maintenance

You should disable CXFS before maintenance as follows:

1. Perform a forced CXFS shutdown.
2. Stop the `cxfs_client` daemon.
3. Disable `cxfs_client` from automatically restarting.

Do Not Run Power Management Software

Do not run power management software, which may interfere with the CXFS cluster.

Use Fast Copying for Large CXFS Files

You can use the `cxfsccp(1)` command to quickly copy large files (64 KB or larger) to and from a CXFS filesystem. It can be significantly faster than `cp(1)` on CXFS filesystems because it uses multiple threads and large direct I/Os to fully use the bandwidth to the storage hardware.

Files smaller than 64 KB do not benefit from large direct I/Os. For these files, `cxfsccp` uses a separate thread using buffered I/O, similar to `cp(1)`.

The `cxfsccp` command is available on Linux and Windows platforms. However, some options are platform-specific, and other limitations apply. For more information and a complete list of options, see the `cxfsccp(1)` man page.

Appropriately Map Physical Device Names to XVM Physvols

To match up physical device names to their corresponding XVM physical volumes (*physvols*), use the following command:

```
# xvm show -v -top -ext vol/volname
```

In the output for this command, the information within the parentheses matches up the XVM pieces with the device name. For example (line breaks shown for readability):

```
# xvm show -v -top -ext vol/test
vol/test                0 online,open
  subvol/test/data      1142792192 online,open
    stripe/stripe0      1142792192 online,tempname,open (unit size:128)
      slice/cc_is4500-lun0-gpts0 142849024 online,open
(cc_is4500-lun0-gpt:/dev/xscsi/pci08.03.0/node200400a0b8119204/port4/lun0/disc)
      slice/cc_is4500-lun1-gpts0 142849024 online,open
(cc_is4500-lun1-gpt:/dev/xscsi/pci08.03.1/node200500a0b8119204/port1/lun1/disc)
      slice/cc_is4500-lun0-gpts1 142849024 online,open
(cc_is4500-lun0-gpt:/dev/xscsi/pci08.03.0/node200400a0b8119204/port4/lun0/disc)
      slice/cc_is4500-lun1-gpts1 142849024 online,open
(cc_is4500-lun1-gpt:/dev/xscsi/pci08.03.1/node200500a0b8119204/port1/lun1/disc)
      slice/cc_is4500-lun0-gpts2 142849024 online,open
```

```
(cc_is4500-lun0-gpt:/dev/xscsi/pci08.03.0/node200400a0b8119204/port4/lun0/disc)
  slice/cc_is4500-lun1-gpts2 142849024 online,open
(cc_is4500-lun1-gpt:/dev/xscsi/pci08.03.1/node200500a0b8119204/port1/lun1/disc)
  slice/cc_is4500-lun0-gpts3 142849024 online,open
(cc_is4500-lun0-gpt:/dev/xscsi/pci08.03.0/node200400a0b8119204/port4/lun0/disc)
  slice/cc_is4500-lun1-gpts3 142849024 online,open
(cc_is4500-lun1-gpt:/dev/xscsi/pci08.03.1/node200500a0b8119204/port1/lun1/disc)
```

Note: The `xvm` command on the Windows platform does not display the worldwide name (WWN). For more information about WWNs and Windows, see "XVM Failover V2 on Windows" on page 153.

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

Do Not Overfill CXFS Filesystems

For best performance, keep your CXFS filesystems under 98% full. This is also a best practice for a local filesystem, but is even more important for a CXFS filesystem because of fragmented files and increased metadata traffic.

Limit Client Accounts to 32 Groups

The CXFS metadata server is only capable of managing permissions for users with 32 or fewer group memberships. Therefore, all accounts (including `root`) on CXFS clients must be limited to 32 or fewer groups.

Turn Off Local XVM on Linux Nodes if Unused

If you do not have a local XVM volume on your Linux system, you should turn off the `boot.lvm` script to avoid unnecessarily probing all of the disks and `lun0` LUNs to which the machine has access. Do the following:

```
# chkconfig boot.lvm off
```

Access the Correct Cluster at a Multiple-Cluster Site

If you have multiple clusters connected to the same public network, you should add `-i clustername` to the `cxfs_client.options` file.

Note: CXFS does not support multiple clusters on the same **private** network.

Linux Platform

CXFS supports a client-only node running the Red Hat Enterprise Linux (RHEL) or SUSE Linux Enterprise Server (SLES) operating system, as defined in the CXFS Linux release notes.

Note: Nodes that you intend to run as metadata servers must be installed as server-capable administration nodes; all other nodes should be client-only nodes. For information about server-capable administration nodes, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

This chapter contains the following sections:

- "CXFS on Linux" on page 28
- "HBA Installation for Linux" on page 35
- "Preinstallation Steps for Linux" on page 36
- "Client Software Installation for Linux" on page 41
- "Start/Stop `cxfs_client` for Linux" on page 45
- "Maintenance for Linux" on page 47
- "Using `cxfs-reprobe` with RHEL" on page 50
- "GRIO on Linux" on page 51
- "XVM Failover V2 on Linux" on page 52
- "Troubleshooting for Linux" on page 53
- "Reporting Linux Problems" on page 56

For information about system tunable parameters, see *CXFS 6 Administration Guide for SGI InfiniteStorage*.

CXFS on Linux

This section contains the following information about CXFS on Linux systems:

- "Requirements for Linux" on page 28
- "CXFS Commands on Linux" on page 30
- "Log Files on Linux" on page 30
- "CXFS Scripts on Linux" on page 31
- "Mount Scripts" on page 31
- "Limitations and Considerations for Linux" on page 33
- "Using the `dm` Mount Option on a SLES Node" on page 34
- "Access Control Lists and Linux" on page 35

Requirements for Linux

In addition to the items listed in "Requirements" on page 6, using a Linux node to support CXFS requires the following:

- One of the following distributions:
 - RHEL
 - SLES

See the release notes for the supported kernels, update levels, and service pack levels, plus information about SGI ProPack and SGI Foundation Software.

- On Altix or Altix XE systems, supported Fibre Channel switches. For supported switches, see the release notes. Either system reset or I/O fencing is required for all nodes.
- A choice of at least one Fibre Channel host bus adapter (HBA), depending upon hardware type:
 - Altix or Altix XE hardware:
 - QLogic QLA2310, QLA2342, or QLA2344
 - LSI Logic LSI7104XP-LC, LSI7204XP-LC, or LSI7204EP-LC

Note: The LSI HBA requires the 01030600 firmware.

- Third-party hardware:
 - QLogic QLA2200, QLA2200F, QLA2310, QLA2342, QLA2344
 - LSI Logic LS17202XP-LC, LS17402XP-LC, LS17104XP-LC, LS17204XP-LC, LS17404XP-LC
-

Note: The LSI HBA requires the 01030600 firmware or newer.

- A CPU of the following class:
 - x86-64 architecture, such as:
 - AMD Opteron
 - Intel Xeon EM64T
 - ia64 architecture, such as Intel Itanium 2

The machine must have at least the following **minimum** requirements:

- 256 MB of RAM memory
- Two Ethernet 100baseT interfaces
- One empty PCI slot (to receive the HBA)

For the latest information, see the CXFS Linux release notes.

Note: If you use I/O fencing and `ipfilterd` on a node, the `ipfilterd` configuration must allow communication between the node and the `telnet` port on the switch. Also see "Configure Firewalls for CXFS Use" on page 19.

CXFS Commands on Linux

The following commands are shipped as part of the CXFS Linux package:

```
/usr/cluster/bin/cxfs_config  
/usr/cluster/bin/cxfs_client  
/usr/cluster/bin/cxfs_info  
/usr/cluster/bin/cxfscp  
/usr/cluster/bin/cxfsdump  
/usr/sbin/grioadmin  
/usr/sbin/griomon  
/usr/sbin/griogqs  
/sbin/xvm
```

The `cxfs_client` and `xvm` commands are needed to include a client-only node in a CXFS cluster. The `cxfs_info` command reports the current status of this node in the CXFS cluster.

The `rpm` command output lists all software added; see "Installing CXFS Software for Linux" on page 41.

For more information, see the man pages.

Log Files on Linux

The `cxfs_client` command creates a `/var/log/cxfs_client` log file. You should monitor the `/var/log/cxfs_client` and `/var/log/messages` log files for problems. Look for a `Membership delivered` message to indicate that a cluster was formed.

The Linux platform uses the `logrotate` system utility to rotate the CXFS logs (as opposed to other multiOS platforms, which use the `-z` option to `cxfs_client`):

- The `/etc/logrotate.conf` file specifies how often system logs are rotated
- The `/etc/logrotate.d/cxfs_client` file specifies the manner in which `cxfs_client` logs are rotated

For information about the log files created on server-capable administration nodes, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

CXFS Scripts on Linux

The following CXFS scripts are provided for execution by the `cxfs_client` daemon:

- "Mount Scripts" on page 31
- "cxfs-reprobe Script" on page 31
- "cxfs-enumerate-wwns Script" on page 32

Mount Scripts

The `cxfs_client` executes the CXFS mount scripts before a CXFS filesystem is mounted and after a CXFS filesystem is unmounted on a Linux client-only node. You can customize these scripts to suit a particular environment. For example, an application could be started when a CXFS filesystem is mounted by extending the `cxfs-post-mount` script. The application could be terminated by changing the `cxfs-pre-umount` script. The mount scripts are installed in the following locations:

```
/var/cluster/cxfs_client-scripts/cxfs-pre-mount  
/var/cluster/cxfs_client-scripts/cxfs-post-mount  
/var/cluster/cxfs_client-scripts/cxfs-pre-umount  
/var/cluster/cxfs_client-scripts/cxfs-post-umount
```

For more details about using these scripts, and for information about the mount scripts on server-capable administration nodes, see *CXFS 6 Administration Guide for SGI InfiniteStorage*.

cxfs-reprobe Script

CXFS uses the `cxfs-reprobe` script to ensure that LUN path failover works after fencing. The `cxfs_client` daemon runs `cxfs-reprobe` to reprobe the Fibre Channel controllers on client-only nodes when they join or rejoin membership. The script is installed in the following location:

```
/var/cluster/cxfs_client-scripts/cxfs-reprobe
```

Note: In order for `cxfs-reprobe` to appropriately probe all of the targets on the SCSI bus, you must define a group of environment variables in the `/etc/cluster/config/cxfs_client.options` file.

cxfs-enumerate-wwns Script

The `cxfs-enumerate-wwns` script enumerates the host's world wide names (WWNs) that are known to CXFS. For example:

```
linux# /var/cluster/cxfs_client-scripts/cxfs-enumerate-wwns
# cxfs-enumerate-wwns
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# lsi @ /proc/mpt/ioc2/info
100000062b0f4ff1
# lsi @ /proc/mpt/ioc1/info
100000062b0f4ff0
# lsi @ /proc/mpt/ioc0/info
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
```

Limitations and Considerations for Linux

Note the following:

- On Linux systems, the use of XVM is supported only with CXFS; XVM does not support local Linux disk volumes.
- On systems running SUSE Linux Enterprise Server 10 (SLES 10) that are greater than 64 CPUs, there are issues with using the md driver and CXFS. The md driver holds the BKL (Big Kernel Lock), which is a single, system-wide spin lock. Attempting to acquire this lock can add substantial latency to a driver's operation, which in turn holds off other processes such as CXFS. The delay causes CXFS to lose membership. This problem has been observed specifically when an md pair RAID split is done, such as the following:

```
raidsetfaulty /dev/md1 /dev/path/to/partition
```

- Although it is possible to mount other filesystems on top of a Linux CXFS filesystem, this is not recommended.
- CXFS filesystems with XFS version 1 directory format cannot be mounted on Linux nodes.
- The implementation of file creates using `O_EXCL` is not complete. Multiple applications running on the same node using `O_EXCL` creates as a synchronization mechanism will see the expected behavior (only one of the creates will succeed). However, applications running between nodes may not get the `O_EXCL` behavior they requested (creates of the same file from two or more separate nodes may all succeed).
- The Fibre Channel HBA driver must be loaded before CXFS services are started. The HBA driver could be loaded early in the initialization scripts or be added to the initial RAM disk for the kernel. See the `mkinitrd` man page for more information.
- RHEL 5 x86-64 nodes have a severely limited kernel stack size. To use CXFS on these nodes requires the following to avoid a stack overflow panic:
 - You must fully disable SELinux on x86-64 RHEL 5 client-only nodes (you cannot simply set it to `permissive` mode). For more information, see:

http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Deployment_Guide/sec-sel-enable-disable.html

Note: This caveat does not apply to RHEL 5 nodes with ia64 architectures.

- Case-insensitive CXFS filesystems are not supported on SLES 10 and RHEL client-only nodes. These nodes will fail to mount the filesystem with messages such as the following:

```
Preparing to mount CXFS file system "/dev/cxvm/tp91"  
XFS: bad version  
XFS: SB validate failed
```

Note: Nodes that use enhanced XFS (SLES 10 nodes that are installed with the CXFS Edge Server software and SLES 11 nodes) do support case-insensitive filesystems.

- An export option called `no_sendfile` has been added to the enhanced NFS server for SLES 10 systems. If you are having an issue with a SLES 10 client serving NFS, SGI recommends that you use `no_sendfile`. For more information, see the `exports(5)` man page.

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 207.

Using the `dmf` Mount Option on a SLES Node

By default, DMAPAPI is turned off on SLES 10 and SLES 11 systems. If you want to mount CXFS filesystems on a SLES 10 or SLES 11 client-only node with the `dmf` mount option, you must set `DMAPI_PROBE="yes"` in the `/etc/sysconfig/sysctl` file on the node. Changes to the file will be processed on the next reboot.

After setting that system configuration file, you can immediately enable DMAPAPI for the current boot session by executing the following:

```
# sysctl -w fs.xfs.probe_dmapapi=1
```

Note: These steps are not required on the DMF server or DMF parallel data mover nodes because these steps are done automatically when installing the `dmf` or `dmf-mover` packages.

Access Control Lists and Linux

All CXFS files have UNIX mode bits (read, write, and execute) and optionally an access control list (ACL). For more information about POSIX ACLs, see the `chmod` and `setfacl` man pages.

HBA Installation for Linux

This section provides an overview of the Fibre Channel host bus adapter (HBA) installation information for Linux nodes.

The installation may be performed by you or by a qualified service representative for your hardware. See the Linux operating system documentation and the documentation for your hardware platform.

The driver requirements are as follows:

- LSI Logic card: the drivers are supplied with the Linux kernel. The module names are `mptscsih` and `mptfc`. The LSI `lsiutil` command displays the number of LSI HBAs installed, the model numbers, and firmware versions.
- QLogic card: the drivers are supplied with the Linux kernel.

You must ensure that the HBA driver is loaded prior to CXFS initialization by building the module into the initial RAM disk automatically or manually. For example, using the QLogic card and the `qla2200` driver:

- **Automatic method:**

For RHEL, add a new line such as the following to the `/etc/modprobe.conf` file:

```
alias scsi_hostadapter1 qla2200
```

For SLES, add the driver name to the `INITRD_MODULES` variable in the `/etc/sysconfig/kernel` file. After adding the HBA driver into `INITRD_MODULES`, you must rebuild `initrd` with `mkinitrd`.

Note: If the host adapter is installed in the box when the operating system is installed, this may not be necessary. Or hardware may be detected at boot time.

When the new kernel is installed, the driver will be automatically included in the corresponding `initrd` image.

- **Manual method:**

Recreate your `initrd` to include the appropriate HBA driver module. For more information, see the operating system documentation for the `mkinitrd` command.

You should then verify the appropriate `initrd` information:

- If using the GRUB loader, verify that the following line appears in the `/boot/grub/grub.conf` file:

```
initrd /initrd-version.img
```

- If using the LILO loader, do the following:

1. Verify that the following line appears in the appropriate stanza of `/etc/lilo.conf`:

```
/boot/initrd-version.img
```

2. Rerun LILO.

The system must be rebooted (and when using LILO, LILO must be rerun) for the new `initrd` image to take effect.

Instead of this procedure, you could also modify the `/etc/rc.sysinit` script to load the `qla2200` driver early in the `initscript` sequence.

Preinstallation Steps for Linux

This section provides an overview of the steps that you will perform on your Linux nodes prior to installing the CXFS software. It contains the following sections:

- "Adding a Private Network for Linux" on page 36
- "Using CXFS GUI Connectivity Diagnostics for Linux" on page 39
- "Verifying the Private and Public Networks for Linux" on page 40

Adding a Private Network for Linux

The following procedure provides an overview of the steps required to add a private network to the Linux system. A private network is required for use with CXFS. See "Use a Private Network" on page 15.

You may skip some steps, depending upon the starting conditions at your site. For details about any of these steps, see the Linux operating system documentation.

1. Edit the `/etc/hosts` file so that it contains entries for every node in the cluster and their private interfaces as well. The `/etc/hosts` file has the following format, where *primary_hostname* can be the simple hostname or the fully qualified domain name:

```
IP_address    primary_hostname    aliases
```

You should be consistent when using fully qualified domain names in the `/etc/hosts` file. If you use fully qualified domain names 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 node 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.

Even if you are using the domain name service (DNS) or the network information service (NIS), you must add every IP address and hostname for the nodes to `/etc/hosts` on all nodes. For example:

```
190.0.2.1 server1.company.com server1
190.0.2.3 stocks
190.0.3.1 priv-server1
190.0.2.2 server2.company.com server2
190.0.2.4 bonds
190.0.3.2 priv-server2
```

You should then add all of these IP addresses to `/etc/hosts` on the other nodes in the cluster.

For more information, see the `hosts` and `resolver` man pages.

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

For more information, see "Understand Hostname Resolution and Network Configuration Rules" on page 14.

2. Edit the `/etc/nsswitch.conf` file so that local files are accessed before either NIS or 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 CXFS, but `files` must be first.)

3. Configure your private interface according to the instructions in the network configuration section of your Linux distribution manual. To verify that the private interface is operational, issue the following command:

```
linux# ifconfig -a
```

For example:

```
linux# ifconfig -a
```

```
eth0      Link encap:Ethernet  HWaddr 00:50:81:A4:75:6A
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13782788  errors:0  dropped:0  overruns:0  frame:0
          TX packets:60846  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:100
          RX bytes:826016878 (787.7 Mb)  TX bytes:5745933 (5.4 Mb)
          Interrupt:19  Base address:0xb880  Memory:fe0fe000-fe0fe038

eth1      Link encap:Ethernet  HWaddr 00:81:8A:10:5C:34
          inet addr:10.0.0.10  Bcast:10.0.0.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0  errors:0  dropped:0  overruns:0  frame:0
          TX packets:0  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:19  Base address:0xef00  Memory:febfd000-febfd038

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:162  errors:0  dropped:0  overruns:0  frame:0
          TX packets:162  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:0
          RX bytes:11692 (11.4 Kb)  TX bytes:11692 (11.4 Kb)
```

This example shows that two ethernet interfaces (`eth0` and `eth1`) are present and running (as indicated by `UP` in the third line of each interface description).

If the second network does not appear, it may be that a network interface card must be installed in order to provide a second network, or it may be that the network is not yet initialized.

Using CXFS GUI Connectivity Diagnostics for Linux

In order to test node connectivity by using the GUI, the `root` user on the node running the CXFS diagnostics must be able to access a remote shell using the `rsh` command (as `root`) on all other nodes in the cluster. (This test is not required when using `cxfs_admin` because it verifies the connectivity of each node as it is added to the cluster.)

There are several ways of accomplishing this, depending on the existing settings in the pluggable authentication modules (PAMs) and other security configuration files.

The following method works with default settings. Do the following on all nodes in the cluster:

1. Install the `rsh-server` RPM.
2. Enable `rsh`.
3. Restart `xinted`.
4. Add `rsh` to the `/etc/securetty` file.
5. Add the hostname of the node from which you will be running the diagnostics into the `/root/.rhosts` file. Make sure that the mode of the `.rhosts` file is set to `600` (read and write access for the owner only).

After you have completed running the connectivity tests, you may wish to disable `rsh` on all cluster nodes.

For more information, see the Linux operating system documentation about PAM and the `hosts.equiv` man page.

Verifying the Private and Public Networks for Linux

For each private network on each Linux node in the pool, verify access with the `ping` command:

1. Enable multicast `ping` using one or more of the following methods (the permanent method will not take affect until after a reboot):

- Immediate but temporary method:

```
linux# echo "0" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

For more information, see <http://kerneltrap.org/node/16225>

- Immediate but temporary method:

```
linux# sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=0"
```

- Permanent method upon reboot (survives across reboots):

1. Remove the following line (if it exists) from the `/etc/sysctl.conf` file:

```
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

2. Add the following line to the `/etc/sysctl.conf` file:

```
net.ipv4.icmp_echo_ignore_broadcasts = 0
```

2. Execute a `ping` using the private network. Enter the following, where *nodeIPAddress* is the IP address of the node:

```
# ping nodeIPAddress
```

For example:

```
linux# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) from 128.162.240.141 : 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.310 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.122 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.127 ms
```

3. Execute a `ping` using the public network.

4. If the ping fails, repeat the following procedure on each node:
 - a. Verify that the network interface was configured up using `ifconfig`. For example:

```
linux# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:81:8A:10:5C:34
          inet addr:10.0.0.10  Bcast:10.0.0.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:19 Base address:0xef00 Memory:febfd000-febfd038
```

In the third output line above, UP indicates that the interface was configured up.

- b. Verify that the cables are correctly seated.
5. Repeat this procedure on each node.

Client Software Installation for Linux

This section discusses the following:

- "Installing CXFS Software for Linux" on page 41
- "Installing the Performance Co-Pilot Agent" on page 44
- "Verifying the Linux Installation" on page 44

Installing CXFS Software for Linux

Table 3-1 provides examples of the differences in package extensions among the various processor architectures supported by CXFS.

Note: The package extensions vary by architecture. Ensure that you install the appropriate package for your processor architecture.

Table 3-1 Processor Architecture and Package Extensions

Class	Processor Architecture	Package Architecture Extension
x86-64	AMD Opteron	.x86_64.rpm
	Intel Xeon EM64T	.x86_64.rpm
ia64	Intel Itanium 2	.ia64.rpm

Installing the CXFS client software for Linux requires approximately 50–200 MB of space, depending upon the packages installed at your site.

To install the required software on a Linux node, do the following:

1. Read the *SGI InfiniteStorage Software Platform* release notes, CXFS general release notes, and CXFS Linux release notes in the `/docs` directory on the ISSP DVD and any late-breaking caveats on Supportfolio.
2. Verify that the node is running a supported Linux distribution and kernel, according to the CXFS for Linux release notes. See the Red Hat `/etc/redhat-release` or SLES `/etc/SuSE-release` files and enter the following:

```
linux_cxfsclient# uname -r
```

3. (Optional) Verify that the node is running the supported level of SGI Foundation Software and (optionally) SGI ProPack, according to the CXFS for Linux release notes. For more information, see the *Start Here* for the supported versions of SGI Foundation Software and SGI ProPack. Also install any required patches. See the `releasenotes/README` file for more information.
4. If you had to install software in one of the above steps, reboot the system:

```
linux_cxfsclient# /sbin/reboot
```

5. Transfer the client-only software (that was downloaded onto a CXFS server-capable administration node during its installation procedure) from the server to the client using `ftp`, `rcp`, or `scp`.

The location of the tarball on the server will be as follows:

```
/usr/cluster/client-dist/CXFS_VERSION/linux/CLIENT_LINUX_VERSION/ARCHITECTURE/cxfs-client.tar.gz
```

For example, for an Altix ia64 client running SLES 10 SP2, the location of the CXFS 6.0 tarball on the server would be:

```
/usr/cluster/client-dist/6.0.0.3/linux/sles10sp2/ia64/cxfs-client.tar.gz
```

Note: Specific packages listed here are examples and may not match the released product.

In this case, you could do the following:

```
server_admin# cd /usr/cluster/client-dist/6.0.0.3/linux/sles10sp2/ia64
server_admin# scp cxfs-client.tar.gz linux_cxfsclient:/tmp/cxfs/
```

6. Disassemble the downloaded tarball on the Linux client-only node. For example:

```
linux_cxfsclient# cd /tmp/cxfs
linux_cxfsclient# tar -zxvf tarball
```

After you extract the information using `tar`, the RPMs will be in the following directory:

```
/tmp/cxfs/sgi-install/SGI/RPMS
```

7. Install the CXFS software:

- For RHEL:

- Including GRIOv2:

```
rhel_cxfsclient# rpm -Uvh *.rpm
```

- Without GRIOv2:

```
rhel_cxfsclient# rpm -Uvh cxfs*rpm sgi-*-kmp-*rpm sgi*rpm
```

- For SLES, where *Kernelvariant* is either `smf` (SLES 10) or `default` (SLES 10 or SLES 11):

- Including GRIOv2:

```
sles_cxfsclient# rpm -Uvh cxfs*rpm grio2*rpm sgi-*-kmp-Kernelvariant-*rpm
```

- Without GRIOv2:

```
sles_cxfsclient# rpm -Uvh cxfs*rpm sgi-*-kmp-Kernelvariant-*rpm
```

8. Edit the `/etc/cluster/config/cxfs_client.options` file as necessary. See the "Maintenance for Linux" on page 47 and the `cxfs_client(8)` man page.

9. Reboot the system:

```
linux_cxfsclient# reboot
```

Installing the Performance Co-Pilot Agent

The `cxfs_utils` package includes a Performance Co-Pilot (PCP) agent for monitoring CXFS heartbeat, CMS status and other statistics. If you want to use this feature, you must also install the following PCP packages:

- `pcp` from the SGI Foundation Software release
- `pcp-sgi` from the SGI ProPack release

Verifying the Linux Installation

Use the `uname -r` command to ensure the kernel installed above is running.

To verify that the CXFS software has been installed properly, use the `rpm -qa` command to display all of the installed packages. You can filter the output by searching for particular package name.

I/O Fencing for Linux

I/O fencing is required on Linux nodes in order to protect data integrity of the filesystems in the cluster. The `cxfs_client` software automatically detects the world wide port names (WWPNs) of any supported host bus adapters (HBAs) for Linux nodes that are connected to a switch that is configured in the cluster database. These HBAs are available for fencing.

However, if no WWPNs are detected, the following message will be logged to the `/var/log/cxfs_client` file:

```
cis_get_hbas no local HBAs found - falling back to /etc/fencing.conf
```

If no WWPNs are detected, you can manually specify the WWPNs in the fencing file.

Note: This method does not work if the WWPNs are partially discovered.

The `/etc/fencing.conf` file enumerates the WWPNs for all of the HBAs that will be used to mount a CXFS filesystem. There must be a line for each HBA WWPN as a 64-bit hexadecimal number.

Note: The WWPN is that of the HBA itself, **not** any of the devices that are visible to that HBA in the fabric.

You must update the `/etc/fencing.conf` file whenever the HBA configuration changes, including the replacement of an HBA.

For dual-ported HBAs, the file must include the WWPNs of any ports that are used to access cluster disks. This may result in multiple WWPNs per HBA in the file; the numbers will probably differ by a single digit. For example, if you determined that port 0 is the port connected to the switch, your fencing file should contain the following (comment lines begin with #):

```
# WWPN of the HBA installed on this system
#
2000000173002c0b
```

To configure fencing, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Start/Stop `cxfs_client` for Linux

The `cxfs_client` service will be invoked automatically during normal system startup and shutdown procedures. This script starts and stops the `cxfs_client` daemon.

To start up `cxfs_client` manually, enter the following:

```
linux# service cxfs_client start
Loading CXFS modules: done
Starting cxfs client: cxfs_client daemon started
done
```

To stop `cxfs_client` manually, enter the following:

```
linux# service cxfs_client stop
Stopping cxfs client:
Waiting for cxfs client to stop... done
```

To stop and then start `cxfs_client` manually, enter the following:

```
linux# service cxfs_client restart
Stopping cxfs client:
Waiting for cxfs client to stop... done
Loading CXFS modules: done
Starting cxfs client: cxfs_client daemon started done
```

To see the current status, use the `status` argument. For example:

```
linux# service cxfs_client status
cxfs_client status [timestamp Apr 20 14:54:30 / generation 4364]

CXFS client:
  state: stable (5), cms: up, xvm: up, fs: up
Cluster:
  connies_cluster (23) - enabled
Local:
  ceara (7) - enabled
Nodes:
  aiden      enabled up    12
  brenna     enabled DOWN 10
  brigid     enabled up    11
  ceara      enabled up    7
  chili      enabled up    4
  cxfsibm2   enabled up    9
  cxfssun4   enabled up    5
  daghada    enabled up    8
  flynn      enabled up    2
  gaeth      enabled up    0
  minnesota  enabled up    6
  rowan      enabled up    3
  rylie      enabled up    1
Filesystems:
  concatfs   enabled mounted      concatfs           /concatfs
  stripefs   enabled mounted      stripefs           /stripefs
  tp9300_stripefs enabled forced mounted  tp9300_stripefs  /tp9300_stripefs
cxfs_client is running.
```

For example, if `cxfs_client` is stopped:

```
linux# service cxfs_client status
cxfs_client is stopped
```

Maintenance for Linux

This section contains information about maintenance procedures for CXFS on Linux:

- "Modifying the CXFS Software for Linux" on page 47
- "Recognizing Storage Changes for Linux" on page 48

Modifying the CXFS Software for Linux

You can modify the behavior of the CXFS client daemon (`cxfs_client`) by placing options in the `/etc/cluster/config/cxfs_client.options` file. The available options are documented in the `cxfs_client` man page.



Caution: Some of the options are intended to be used internally by SGI only for testing purposes and do not represent supported configurations. Consult your SGI service representative before making any changes.

To see if `cxfs_client` is using the options in `cxfs_client.options`, enter the following:

```
linux# ps -ax | grep cxfs_client
3612 ?          S          0:00 /usr/cluster/bin/cxfs_client -i cxfs3-5
3841 pts/0      S          0:00 grep cxfs_client
```

To be sure that `cxfs_client` is configured to start up on boot, view the `chkconfig` output, which should appear similar to the following:

```
linux# chkconfig --list | grep cxfs_client
cxfs_client          0:off 1:off 2:off 3:on  4:off 5:on  6:off
```

Recognizing Storage Changes for Linux

On Linux nodes, the `cxfs-enumerate-wwns` script enumerates the world wide names (WWNs) on the host that are known to CXFS. See "Mount Scripts" on page 31.

The following script is run by `cxfs_client` when it reprobes the Fibre Channel controllers upon joining or rejoining membership:

```
/var/cluster/cxfs_client-scripts/cxfs-reprobe
```

For RHEL nodes, you can define a group of environment variables in the `/etc/cluster/config/cxfs_client.options` file in order for `cxfs-reprobe` to probe specific targets on the SCSI bus.

The script detects the presence of the SCSI and/or XSCSI layers on the system and defaults to probing whichever layers are detected. You can override this decision by setting `CXFS_PROBE_SCSI` and/or `CXFS_PROBE_XSCSI` to one of the following on the appropriate bus:

- 0 to disable the probe
- 1 to force the probe

When an XSCSI scan is performed, all buses are scanned by default. You can override this decision by specifying a space-separated list of buses in `CXFS_PROBE_XSCSI_BUSES`. (If you include space, you must enclose the list within single quotation marks.) For example:

```
export CXFS_PROBE_XSCSI_BUSES='/dev/xscsi/pci0001:00:03.0-1/bus /dev/xscsi/pci0002:00:01.0-2/bus'
```

When a SCSI scan is performed, a fixed range of buses/channels/IDs and LUNs are scanned; these ranges may need to be changed to ensure that all devices are found. The ranges can also be reduced to increase scanning speed if a smaller space is sufficient.

The following summarizes the environment variables (separate multiple values by white space and enclose within single quotation marks):

```
CXFS_PROBE_SCSI=0/1
```

Stops (0) or forces (1) a SCSI probe. Default: 1 if SCSI

```
CXFS_PROBE_SCSI_BUSES=BusList
```

Scans the buses listed. Default: 0 1 2

`CXFS_PROBE_SCSI_CHANNELS=ChannelList`

Scans the channels listed. Default: 0

`CXFS_PROBE_SCSI_IDS=IDList`

Scans the IDS listed. Default: 0 1 2 3

`CXFS_PROBE_SCSI_LUNS=LunList`

Scans the LUNs listed. Default: 0 1 2 3 4 5 6 7 8 9 10 11 12
13 14 15

`CXFS_PROBE_XSCSI=0/1`

Stops (1) or forces (1) an XSCSI probe. Default: 1 if XSCSI

`CXFS_PROBE_XSCSI_BUSES=BusList`

Scans the buses listed. Default: all XSCSI buses

For example, the following would only scan the first two SCSI buses:

```
export CXFS_PROBE_SCSI_BUSES='0 1'
```

The following would scan 16 LUNs on each bus, channel, and ID combination (all on one line):

```
export CXFS_PROBE_SCSI_LUNS='0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15'
```

Other options within the `/etc/cluster/config/cxfs_client.options` file begin with a `-` character. Following is an example `cxfs_client.options` file:

```
# Example cxfs_client.options file
#
-Dnormal -serror
export CXFS_PROBE_SCSI_BUSES=1
export CXFS_PROBE_SCSI_LUNS='0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20'
```

Note: The `-` character or the term `export` must start in the first position of each line in the `cxfs_client.options` file; otherwise, they are ignored by the `cxfs_client` service.

Using `cxfs-reprobe` with RHEL

When `cxfs_client` rescans disk buses, it executes the `/var/cluster/cxfs_client-scripts/cxfs-reprobe` script. This requires the use of parameters in RHEL due to limitations in the SCSI layer. You can export these parameters from the `/etc/cluster/config/cxfs_client.options` file.

The script detects the presence of the SCSI and/or XSCSI layers on the system and defaults to probing whichever layers are detected. You can override this decision by setting `CXFS_PROBE_SCSI` (for Linux SCSI) or `CXFS_PROBE_XSCSI` (for Linux XSCSI) to one of the following:

- 0 to disable the probe
- 1 to force the probe

When an XSCSI scan is performed, all buses are scanned by default. You can override this by specifying a space-separated list of buses in `CXFS_PROBE_XSCSI_BUSES`. (If you include space, you must enclose the list within single quotation marks.) For example:

```
export CXFS_PROBE_XSCSI_BUSES='/dev/xscsi/pci01.03.0-1/bus /dev/xscsi/pci02.01.0-2/bus'
```

When a SCSI scan is performed, a fixed range of buses/channels/IDs and LUNs are scanned; these ranges may need to be changed to ensure that all devices are found. The ranges can also be reduced to increase scanning speed if a smaller space is sufficient.

The following summarizes the environment variables (separate multiple values by white space and enclose within single quotation marks):

`CXFS_PROBE_SCSI=0/1`

Stops (0) or forces (1) a SCSI probe. Default: 1 if SCSI

`CXFS_PROBE_SCSI_BUSES=BusList`

Scans the buses listed. Default: 0 1 2

`CXFS_PROBE_SCSI_CHANNELS=ChannelList`

Scans the channels listed. Default: 0

`CXFS_PROBE_SCSI_IDS=IDList`

Scans the IDS listed. Default: 0 1 2 3

`CXFS_PROBE_SCSI_LUNS=LunList`

Scans the LUNs listed. Default: 0 1 2 3 4 5 6 7 8 9 10 11 12
13 14 15

`CXFS_PROBE_XSCSI=0/1`

Stops (1) or forces (1) an XSCSI probe. Default: 1 if XSCSI

`CXFS_PROBE_XSCSI_BUSES=BusList`

Scans the buses listed. Default: all XSCSI buses

For example, the following would only scan the first two SCSI buses:

```
export CXFS_PROBE_SCSI_BUSES='0 1'
```

The following would scan 16 LUNs on each bus, channel, and ID combination (all on one line):

```
export CXFS_PROBE_SCSI_LUNS='0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15'
```

Other options within the `/etc/cluster/config/cxfs_client.options` file begin with a `-` character. Following is an example `cxfs_client.options` file:

```
# Example cxfs_client.options file
#
-Dnormal -serror
export CXFS_PROBE_SCSI_BUSES=1
export CXFS_PROBE_SCSI_LUNS='0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20'
```

Note: The `-` character or the term `export` must start in the first position of each line in the `cxfs_client.options` file; otherwise, they are ignored by the `cxfs_client` service.

GRIO on Linux

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the Linux platform if GRIO is enabled on the server-capable administration node. However, GRIO is disabled by default on Linux client-only nodes. To enable GRIO on a Linux client-only node, you must install the GRIO software as documented in "Installing CXFS Software for Linux" on page 41 and do the following:

1. Change the following line in `/etc/cluster/config/cxfs_client.options` from:

```
export GRIO2=off
```

to:

```
export GRIO2=on
```

2. Reboot the system.

A Linux node can mount a GRIO-managed filesystem and supports node-level static reservations. A Linux node will interoperate with the dynamic bandwidth allocator for all I/O outside of any reservation.

Application bandwidth reservations must be explicitly released by the application before exit. If the application terminates unexpectedly or is killed, its bandwidth reservations are not automatically released and will cause a bandwidth leak. If this happens, the lost bandwidth could be recovered by rebooting the node.

For more information, see:

- "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7
- *Guaranteed-Rate I/O Version 2 for Linux Guide*

XVM Failover V2 on Linux

Following is an example of the `/etc/failover2.conf` file on a Linux system (this could be RHEL, SLES, or SGI Foundation Software):

```
/dev/disk/by-path/pci-0000:06:02.1-fc-0x200800a0b8184c8e:0x0000000000000000 affinity=0 preferred  
/dev/disk/by-path/pci-0000:06:02.1-fc-0x200900a0b8184c8d:0x0000000000000000 affinity=1
```

For more information, see:

- The comments in the `/etc/failover2.conf.example` file
- "XVM Failover and CXFS" on page 8
- *CXFS 6 Administration Guide for SGI InfiniteStorage*
- *XVM Volume Manager Administrator's Guide*

Troubleshooting for Linux

This section discusses the following:

- "Device Filesystem Enabled for Linux" on page 53
- "The `cxfs_client` Daemon is Not Started on Linux" on page 53
- "Filesystems Do Not Mount on Linux" on page 54
- "Unable to use the `dmi` Mount Option" on page 55
- "Large Log Files on Linux" on page 55
- "xfs off Output from `chkconfig`" on page 55
- "crash Dumps" on page 56

For general troubleshooting information, see Chapter 7, "General Troubleshooting" on page 191 and Appendix D, "Error Messages" on page 213.

Device Filesystem Enabled for Linux

The kernels provided for the Linux node have the Device File System (`devfs`) enabled. This can cause problems with locating system devices in some circumstances. See the `devfs` FAQ at the following location:

<http://www.atnf.csiro.au/people/rgooch/linux/docs/devfs.html>

The `cxfs_client` Daemon is Not Started on Linux

Confirm that the `cxfs_client` is not running. The following command would list the `cxfs_client` process if it were running:

```
linux# ps -ax | grep cxfs_client
```

Check the `cxfs_client` log file for errors.

Restart `cxfs_client` as described in "Start/Stop `cxfs_client` for Linux" on page 45 and watch the `cxfs_client` log file for errors.

To be sure that `cxfs_client` is configured to start up on boot, view the `chkconfig` output, which should appear similar to the following:

```
linux# chkconfig --list | grep cxfs_client
cxfs_client          0:off 1:off 2:off 3:on  4:off 5:on  6:off
```

Filesystems Do Not Mount on Linux

If `cxfs_info` reports that `cms` is up but `XVM` or the filesystem is in another state, then one or more mounts is still in the process of mounting or has failed to mount.

The CXFS node might not mount filesystems for the following reasons:

- The node may not be able to see all of the LUNs. This is usually caused by misconfiguration of the HBA or the SAN fabric:
 - Check that the ports on the Fibre Channel switch connected to the HBA are active. Physically look at the switch to confirm the light next to the port is green, or remotely check by using the `switchShow` command.
 - Check that the HBA configuration is correct.
 - Check that the HBA can see all the LUNs for the filesystems it is mounting.
 - Check that the operating system kernel can see all the LUN devices.
 - If the RAID device has more than one LUN mapped to different controllers, ensure the node has a Fibre Channel path to all relevant controllers.
- The `cxfs_client` daemon may not be running. See "The `cxfs_client` Daemon is Not Started on Linux" on page 53.
- The filesystem may have an unsupported mount option. Check the `cxfs_client.log` for mount option errors or any other errors that are reported when attempting to mount the filesystem.
- The cluster membership (`cms`), `XVM`, or the filesystems may not be up on the node. Execute the `cxfs_info` command to determine the current state of `cms`, `XVM`, and the filesystems. If the node is not up for each of these, then check the `/var/log/cxfs_client` log to see what actions have failed.

Do the following:

- If `cms` is not up, check the following:
 - Is the node is configured on the server-capable administration node with the correct hostname?
 - Has the node been added to the cluster and enabled? See "Verifying the Cluster Status" on page 183.
- If XVM is not up, check that the HBA is active and can see the LUNs.
- If the filesystem is not up, check that one or more filesystems are configured to be mounted on this node and check the `/var/log/cxfs_client` file for mount errors.

Unable to use the `dmi` Mount Option

By default, DMAPI is turned off on SLES 10 and SLES 11 systems. If you try to mount with the `dmi` mount option, you will see errors such as the following:

```
kernel: XFS: unknown mount option [dmi]."
```

See "Using the `dmi` Mount Option on a SLES Node" on page 34.

Large Log Files on Linux

The `/var/log/cxfs_client` log file may become quite large over a period of time if the verbosity level is increased. See the `cxfs_client.options` man page and "Log Files on Linux" on page 30.

`xfs off` Output from `chkconfig`

The following output from `chkconfig --list` refers to the X Font Server, not the XFS filesystem, and has no association with CXFS:

```
xfs                                0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

crash Dumps

To enable the collection of `crash` dumps on a Linux client-only node, consult your operating system documentation. The *CXFS 6 Administration Guide for SGI InfiniteStorage* contains a procedure for enabling `crash` dump collection on a server-capable administration node.

Reporting Linux Problems

Before reporting a problem to SGI, you should run the `cxfsdump` command:

```
linux# /usr/cluster/bin/cxfsdump
```

This will collect the following information:

- System information
- CXFS registry settings
- CXFS client logs
- CXFS version information
- Network settings
- Event log

The `cxfsdump -help` command displays a help message.

Send SGI the `tar.gz` file that is created in the following directory:

```
/var/cluster/cxfsdump-data/date_time
```

Gather the following information:

- Obtain information about the entire cluster by running the `cxfsdump` utility on a server-capable administration node. See the information in the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

- Number of LSI HBAs installed, the model numbers, and firmware versions:

```
linux# lsiutil
```

- Any messages that appeared in the system logs immediately before the system exhibited the problem.

- The debugger information from the `kdb` built-in kernel debugger for SGI Foundation Software systems on an SGI Altix ia64 system after a system kernel panic.



Caution: When the system enters the debugger after a panic, it will render the system unresponsive until the user exits from the debugger. Also, if `kdb` is entered while the system is in graphical (X) mode, the debugger prompt cannot be seen. For these reasons, `kdb` is turned off by default.

You can temporarily enable `kdb` by entering the following:

```
linux# echo 1 > /proc/sys/kernel/kdb
```

To enable `kdb` at every boot, place the following entry in the `/etc/sysctl.conf` file:

```
# Turn on KDB
kernel.kdb = 1
```

For more information, see the `sysctl` man page.

When `kdb` is enabled, a system panic will cause the debugger to be invoked and the keyboard LEDs will blink. The `kdb` prompt will display basic information. To obtain a stack trace, enter the `bt` command at the `kdb` prompt:

```
kdb> bt
```

To get a list of current processes, enter the following:

```
kdb> ps
```

To backtrace a particular process, enter the following, where *PID* is the process ID:

```
kdb> btp PID
```

To exit the debugger, enter the following:

```
kdb> go
```

If the system will be run in graphical mode with `kdb` enabled, SGI highly recommends that you use `kdb` on a serial console so that the `kdb` prompt can be seen.

- Fibre Channel HBA World Wide name mapping:

```
cat /sys/class/fc_transport/bus_ID/node_name
```

For example:

```
cat /sys/class/fc_transport/11:0:0:0/node_name
```

The *bus_ID* value is the output of `hwinfo --disk` in the SysFS BusID field.

Mac OS X Platform

CXFS supports a client-only node running the Mac OS X operating systems as defined in the CXFS Mac OS X release notes. This chapter contains the following sections:

- "CXFS on Mac OS X" on page 59
- "HBA Installation for Mac OS X" on page 73
- "Preinstallation Steps for Mac OS X" on page 75
- "Client Software Installation for Mac OS X" on page 77
- "I/O Fencing for Mac OS X" on page 79
- "Start/Stop `cxfs_client` for Mac OS X" on page 81
- "Maintenance for Mac OS X" on page 82
- "GRIO on Mac OS X" on page 84
- "XVM Failover V2 on Mac OS X" on page 85
- "Troubleshooting for Mac OS X" on page 85
- "Reporting Mac OS X Problems" on page 87

CXFS on Mac OS X

This section contains the following information about CXFS on Mac OS X:

- "Requirements for Mac OS X" on page 60
- "CXFS Commands on Mac OS X" on page 60
- "Log Files on Mac OS X" on page 61
- "Limitations and Considerations on Mac OS X" on page 62
- "Configuring Hostnames on Mac OS X" on page 63
- "Mapping User and Group Identifiers for Mac OS X" on page 63
- "Access Control Lists and Mac OS X" on page 64

Requirements for Mac OS X

In addition to the items listed in "Requirements" on page 6, using a Mac OS X node to support CXFS requires the following:

- Operating system:
 - Mac OS X Leopard (10.5.8 or later)
 - Mac OS X Snow Leopard (10.6.2 or later)
- One of the following single- or multi-processor Apple Computer hardware platforms:
 - Leopard:
 - Mac Pro
 - Power Mac G4
 - Power Mac G5
 - Xserve
 - Xserve G4
 - Xserve G5
 - Snow Leopard:
 - Mac Pro
 - Xserve
- Apple Fibre Channel PCI and PCI-X host bus adapter (HBA) or Apple PCI Express HBA

For the latest information, see the CXFS Mac OS X release notes.

CXFS Commands on Mac OS X

The following commands are shipped as part of the CXFS Mac OS X package:

```
/usr/cluster/bin/autopsy
/usr/cluster/bin/cxfs_client
/usr/cluster/bin/cxfs_info
/usr/cluster/bin/cxfsdump
/usr/cluster/bin/fabric_dump
/usr/cluster/bin/install-cxfs
/usr/cluster/bin/uninstall-cxfs
```



```
/Library/StartupItems/cxfs/cxfs
/usr/sbin/grioadmin
/usr/sbin/griomon
/usr/sbin/griogps
/usr/cluster/bin/xvm
```

If a Mac OS X node panics, the OS will write details of the panic to `/Library/Logs/panic.log`. Running `autopsy` parses this file and adds symbolic backtraces where possible to make it easier to determine the cause of the panic. The `autopsy` script is automatically run as part of the `cxfsdump` script, so the recommended steps for gathering data from a problematic node are still the same. Run `autopsy` with the `-man` option to display the man page.

To display details of all visible devices on the Fibre Channel fabric, run the `fabric_dump` script. The output is useful for diagnosing issues related to mount problems due to missing LUNs. Run `fabric_dump` with the `-man` option to display the man page.

The `cxfs_client` and `xvm` commands are needed to include a client-only node in a CXFS cluster. The `cxfs_info` command reports the current status of this node in the CXFS cluster.

The installation package uses `install-cxfs` to install or update all of the CXFS files. You can use the `uninstall-cxfs` command to uninstall all CXFS files; `uninstall` is not an installation package option.

The `/Library/StartupItems/cxfs/cxfs` command is run by the operating system to start and stop CXFS on the Mac OS X node.

For more information on these commands, see the man pages. For additional information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7 and "GRIO on Mac OS X" on page 84.

Log Files on Mac OS X

The `cxfs_client` command creates a `/var/log/cxfs_client` log file. To rotate this log file, use the `-z` option in the `/usr/cluster/bin/cxfs_client.options` file; see the `cxfs_client` man page for details.

The CXFS installation process (`install-cxfs` and `uninstall-cxfs`) appends to `/var/log/cxfs_inst.log`.

For information about the log files created on server-capable administration nodes, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Also see the Mac OS X `/var/log/system.log` file.

Limitations and Considerations on Mac OS X

CXFS for Mac OS X has the following limitations and considerations:

- XVM volume names are limited to 31 characters and subvolumes are limited to 26 characters. For more information about XVM, see *XVM Volume Manager Administrator's Guide*.
- Mac OS X does not support the CXFS mount scripts.
- Using a RAID mode of RDAC will output numerous error messages. RDAC mode does not permit XVM path failover. SGI supports SGI AVT mode for XVM failover version 2. For more information, see the information about XVM failover in *CXFS 6 Administration Guide for SGI InfiniteStorage*.
- CXFS does not support the Spotlight indexing facility or the Time Machine backup facility, because these activities are applicable only to a local filesystem.
- To show CXFS filesystems on the desktop in the **Finder**, enable the following:

Finder

- > **Preferences**
- > **General**
- > **Connected Servers**

To see CXFS filesystems in **Finder** and **Open/Save** dialogs, enable the following:

Finder

- > **Preferences**
- > **Sidebar**
- > **Devices**
- > ***Your_computer_name***

To access the CXFS filesystems, click on your computer in the side bar to show the CXFS filesystems.

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 207.

Configuring Hostnames on Mac OS X

Normally, you specify the hostname by using the following menu selection:

```
System Preferences
  > Sharing
    > Computer Name
```

Although the `HOSTNAME=-AUTOMATIC-` entry does not exist in the `/etc/hostconfig` file, you can specify a hostname by using the `HOSTNAME` parameter in this file. The hostname specified for the machine will have the following domain by default:

```
.local
```

For example, if the hostname was specified as `cxfsmacl`, then you would see the following when requesting the hostname:

```
macosx# /bin/hostname
cxfsmacl.local
```

The full hostname including `.local` is the hostname that the CXFS software will use to determine its identity in the cluster, not `cxfsmacl`.

Therefore, you must configure the node as `cxfsmacl.local` or specify the fully qualified hostname in `/etc/hostconfig`. For example:

```
HOSTNAME=cxfsmacl.sgi.com
```

Specifying the hostname in this way may impact some applications, most notably Bonjour, and should be researched and tested carefully. There are also known issues with the hostname being reported as `localhost` on some reboots after making such a change.

SGI recommends that you specify other hosts in the cluster by editing `/etc/hosts`.

Mapping User and Group Identifiers for Mac OS X

To ensure that the correct access controls are applied to users on Mac OS X nodes when accessing CXFS filesystems, you must ensure that the user IDs (UIDs) and group IDs (GIDs) are the same on the Mac OS X node as on all other nodes in the cluster, particularly any server-capable administration nodes.

Note: A user does not have to have user accounts on all nodes in the cluster. However, all access control checks are performed by server-capable administration nodes, so any server-capable administration nodes must be configured with the superset of all users in the cluster.

Users can quickly check that their UID and GID settings are correct by using the `id` command on both the Mac OS X node and the server-capable administration node. For example:

```
macosx% id
uid=1113(fred) gid=999(users) groups=999(users), 20(staff)

admin% id
uid=1113(fred) gid=999(users) groups=999(users), 20(staff)
```

If the UID and/or GID do not match, or if the user is not a member of the same groups, then the user may unexpectedly fail to access some files.

The **Accounts Preference Pane** hides a set of advanced options that you can use to customize user account settings. Do the following:

1. Control-click a name in the **Accounts Preference Pane**.
2. Choose **Advanced** from the pop-up menu.
3. Select the item you want to change.

Access Control Lists and Mac OS X

All CXFS files have POSIX mode bits (read, write, and execute) and optionally an access control list (ACL). For more information, see the `chmod` and `chacl` man pages on a server-capable administration node.

CXFS on Mac OS X supports both the enforcement of POSIX ACLs and the editing of POSIX ACLs from the Mac OS X node.

This section discusses the following:

- "Displaying ACLs" on page 65
- "Comparing POSIX ACLs with Mac OS X ACLs" on page 65

- "Editing POSIX ACLs on Mac OS X" on page 68
- "Default or Inherited ACLs on Mac OS X" on page 70

Note: In the following examples, line breaks are shown here for readability.

Displaying ACLs

To display ACLs on a Mac OS X node, use the `ls -l` command. For example, the `+` character after the file permissions indicates that there are ACLs for `newfile`:

```
macosx# ls -l newfile
-rw-r--r--+ 1 userA ptg 4 Jan 18 09:49 newfile
```

To list the ACLs in detail, use the `-le` options (line breaks shown here for readability):

```
macosx# ls -le newfile
-rw--wxr--+ 1 userA ptg 4 Jan 18 09:49 newfile
0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
  readsecurity,writesecurity,chmod
1: user:userA deny execute
2: group:everyone deny read,readattr,readextattr,readsecurity
3: group:ptg allow read,execute,readattr,readextattr,readsecurity
4: group:ptg deny write,delete,append,writeattr,writeextattr,writesecurity,chmod
5: group:everyone allow read,readattr,readextattr,readsecurity
6: group:everyone deny write,execute,delete,append,writeattr,writeextattr,writesecurity,chmod
```

Comparing POSIX ACLs with Mac OS X ACLs

POSIX ACLs (implemented by CXFS) are very different from those available on Mac OS X. Therefore a translation occurs, which places some limitations on what can be achieved with Mac OS X ACLs. As shown in Table 4-1, POSIX supports only three types of access permissions; in contrast, Mac OS X supports many variations. This means that some granularity is lost when converting between the two systems.

Table 4-1 Mac OS X Permissions Compared with POSIX Access Permissions

POSIX	Mac OS X
Read	Read data, read attributes, read extended attributes, read security
Write	Write data, append data, delete, delete child, write attributes, write extended attributes, write security, add file, add subdirectory, take ownership, linktarget, check immutable
Execute	Execute

POSIX ACLs and the file permissions have a particular relationship that must be translated to work with Mac OS X ACLs. For example, the minimum ACL for a file is user, group, and other, as follows:

```
server-admin# chacl -l newfile
newfile [u::rw-,g::r-x,o::r--]
```

The ACL (user, group, and other) exactly matches the file permissions. Further, any changes to the file permissions will be reflected in the ACL, and vice versa. For example:

```
server-admin# chmod 167 newfile
admin# chacl -l newfile
newfile [u::--x,g::rw-,o::rwx]
```

This is slightly complicated by the mask ACL, which if it exists takes the file's group permissions instead. For example:

```
server-admin# chacl -l newfile
newfile [u::rw-,g::r-x,o::r--,m::rwx]
```

With POSIX, it is not possible to have fewer than three ACL entries, which ensures the rules always match with the file permissions. On Mac OS X, ACLs and file permissions are treated differently. ACLs are processed first; if there is no matching rule, the file permissions are used. Further, each entry can either be an `allow` entry or a `deny` entry. Given these differences, some restrictions are enforced to allow translation between these systems. For example, the simplest possible Linux ACL:

```
server-admin# chacl -l newfile
newfile [u::rw-,g::r-x,o::r--]
```

And the comparative Mac OS X ACL:

```
macosx# ls -le newfile
-rw-r-xr--+ 1 userA ptg 4 Jan 18 09:49 newfile
0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,
  writeextattr,readsecurity,writesecurity,chmod
1: user:userA deny execute
2: group:ptg allow read,execute,readattr,readextattr,readsecurity
3: group:ptg deny write,delete,append,writeattr,writeextattr,writesecurity,chmod
4: group:everyone allow read,readattr,readextattr,readsecurity
5: group:everyone deny write,execute,delete,append,writeattr,writeextattr,writesecurity,chmod
```

Each POSIX rule is translated into two Mac OS X rules. For example, the following user rules are equivalent:

- Linux:

```
u::rw-
```

- Mac OS X:

```
0: user:userA allow read,write,delete,append,readattr,writeattr,
  readextattr,writeextattr,readsecurity,writesecurity,chmod
1: user:userA deny execute
```

However, because the mask rule limits the access that can be assigned to anyone except the owner, the mask is represented by a single deny rule. For example, the following are equivalent:

- Linux:

```
linux# chacl -l newfile
newfile [u::rw-,g::r-x,o::r--,m::-wx]
```

- Mac OS X:

```
macosx# ls -le newfile
-rw--wxr--+ 1 userA ptg 4 Jan 18 09:49 newfile
0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,
  writeextattr,readsecurity,writesecurity,chmod
1: user:userA deny execute
2: group:everyone deny read,readattr,readextattr,readsecurity
3: group:ptg allow read,execute,readattr,readextattr,readsecurity
4: group:ptg deny write,delete,append,writeattr,writeextattr,writesecurity,chmod
5: group:everyone allow read,readattr,readextattr,readsecurity
```

```
6: group:everyone deny write,execute,delete,append,writeattr,writeextattr,
  writesecurity,chmod
```

The mask rule (m: :-wx) is inverted into a simple deny rule (group:everyone deny read,readattr,readextattr,readsecurity). If a mask rule exists, it is always rule number 2 because it applies to everyone except for the file owner.

Editing POSIX ACLs on Mac OS X

To add, remove, or edit a POSIX ACL on a file or directory, use the `chmod` command, which allows you to change only a single rule at a time.

However, it is not valid in POSIX to have a single entry in an ACL. Therefore the basic rules are created based on the file permissions. For example:

```
macosx# ls -le newfile
-rw-rw-rw-  1 userA  ptg  0 Jan 18 15:40 newfile
macosx# chmod +a "cxfs allow read,execute" newfile
macosx# ls -le newfile
-rw-rw-rw-+ 1 userA  ptg  0 Jan 18 15:40 newfile
 0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,
  writeextattr,readsecurity,writesecurity,chmod
 1: user:userA deny execute
 2: group:everyone deny execute
 3: user:cxfs allow read,execute,readattr,readextattr,readsecurity
 4: user:cxfs deny write,delete,append,writeattr,writeextattr,writesecurity,chmod
 5: group:ptg allow read,write,delete,append,readattr,writeattr,readextattr,
  writeextattr,readsecurity,writesecurity,chmod
 6: group:ptg deny execute
 7: group:everyone allow read,write,delete,append,readattr,writeattr,readextattr,
  writeextattr,readsecurity,writesecurity,chmod
 8: group:everyone deny execute
```

You should only ever add, modify, or remove the `allow` rules. The corresponding `deny` rule will be created, modified, or removed as necessary. The mask rule is the only `deny` rule that you should specify directly.

For example, to remove a rule by using `chmod`:

```
macosx# chmod -a# 3 newfile
macosx# ls -le newfile
-rw-rw-rw-+ 1 userA  ptg  0 Jan 18 15:40 newfile
 0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,
```



```

writeextattr,readsecurity,writesecurity,chown
1: user:userA deny execute
2: group:everyone deny execute
3: group:ptg allow read,write,delete,append,readattr,writeattr,readextattr,
writeextattr,readsecurity,writesecurity,chown
4: group:ptg deny execute
5: group:everyone allow read,write,delete,append,readattr,writeattr,readextattr,
writeextattr,readsecurity,writesecurity,chown
6: group:everyone deny execute

```

If you remove rules leaving only the user, group, and other rules, ACLs will be removed completely. For example:

```

macosx# chmod -a# 2 newfile
macosx# ls -le newfile
-rw-rw-rw- 1 userA ptg 0 Jan 18 15:40 newfile

```

Adding rules to an existing ACL is complicated slightly because the ordering required by CXFS is different from the order used on Mac OS X. You may see the following error:

```

macosx# chmod +a "cxfs allow execute" newfile
chmod: The specified file newfile does not have an ACL in canonical order, please
specify a position with +a# : Invalid argument

```

However, because an order will be enforced regardless of where the rule is placed, insert at any position and the rules will be sorted appropriately. For example:

```

macosx# chmod +a# 6 "sshd allow execute" newfile
macosx# ls -le newfile
-rw-rw-rw-+ 1 userA ptg 0 Jan 18 15:40 newfile
0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
readsecurity,writesecurity,chown
1: user:userA deny execute
2: group:everyone deny execute
3: user:cxfs allow execute
4: user:cxfs deny read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
readsecurity,writesecurity,chown
5: user:sshd allow execute
6: user:sshd deny read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
readsecurity,writesecurity,chown
7: group:ptg allow read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
readsecurity,writesecurity,chown

```

```
8: group:ptg deny execute
9: group:everyone allow read,write,delete,append,readattr,writeattr,readextattr,
  writeextattr,readsecurity,writesecurity,chmod
10: group:everyone deny execute
```

You can also edit an existing rule by using `chmod`. Assuming the above file and permissions, you could allow the user to read files with the following command:

```
macosx# chmod =a# 3 "cxfs allow execute,read" newfile
macosx# ls -le newfile
-rw-rw-rw-+ 1 userA ptg 0 Jan 18 15:40 newfile
0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
  readsecurity,writesecurity,chmod
1: user:userA deny execute
2: group:everyone deny execute
3: user:cxfs allow read,execute,readattr,readextattr,readsecurity
4: user:cxfs deny write,delete,append,writeattr,writeextattr,writesecurity,chmod
5: user:sshd allow execute
6: user:sshd deny read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
  readsecurity,writesecurity,chmod
7: group:ptg allow read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
  readsecurity,writesecurity,chmod
8: group:ptg deny execute
9: group:everyone allow read,write,delete,append,readattr,writeattr,readextattr,
  writeextattr,readsecurity,writesecurity,chmod
10: group:everyone deny execute
```

Adding a second rule for the same user or group is not permitted with POSIX ACLs. If you attempt to do this, the permissions will be merged. It is important to get the rule number correct when editing a rule.

Default or Inherited ACLs on Mac OS X

It is possible to define default ACLs to a directory, so that all new files or directories created below are assigned a set of ACLs automatically. The semantics are handled differently between Linux and Mac OS X, so the functionality is limited to mimic what is available in POSIX. In POSIX, the default ACL is applied at creation time only; if the default rule subsequently changes, it is not applied to a directory's children. The equivalent behavior on Mac OS X is achieved by the `only_inherit` and `limit_inherit` flags.

For example, a default ACL might look like this on Linux:

```
admin# chacl -l test
test [u::rwx,g::r--,o::---/u::rw-,g::rw-,o::r--,u:501:r--,m::rwx]
```

On Mac OS X, a default ACL might look like the following:

```
macosx# ls -lde test
drwxr-----+ 2 userA ptg 78 Jan 18 15:39 test
0: user:userA allow list,add_file,search,delete,add_subdirectory,delete_child,
  readattr,writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown
1: user:userA deny
2: group:ptg allow list,readattr,readextattr,readsecurity
3: group:ptg deny add_file,search,delete,add_subdirectory,delete_child,writeattr,
  writeextattr,writesecurity,chown
4: group:everyone allow
5: group:everyone deny list,add_file,search,delete,add_subdirectory,delete_child,
  readattr,writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown
6: user:userA allow list,add_file,delete,add_subdirectory,delete_child,readattr,
  writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown,file_inherit,
  directory_inherit,only_inherit
7: user:userA deny search,file_inherit,directory_inherit,only_inherit
8: group:everyone deny file_inherit,directory_inherit,only_inherit
9: user:cxfs allow list,readattr,readextattr,readsecurity,file_inherit,
  directory_inherit,only_inherit
10: user:cxfs deny add_file,search,delete,add_subdirectory,delete_child,writeattr,
  writeextattr,writesecurity,chown,file_inherit,directory_inherit,only_inherit
11: group:ptg allow list,add_file,delete,add_subdirectory,delete_child,readattr,
  writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown,file_inherit,
  directory_inherit,only_inherit
12: group:ptg deny search,file_inherit,directory_inherit,only_inherit
13: group:everyone allow list,readattr,readextattr,readsecurity,file_inherit,
  directory_inherit,only_inherit
14: group:everyone deny add_file,search,delete,add_subdirectory,delete_child,writeattr,
  writeextattr,writesecurity,chown,file_inherit,directory_inherit,only_inherit
```

The default rules are flagged with the inheritance flags (file_inherit,directory_inherit,only_inherit). Editing these rules is similar to editing an access rule, except the inherit flag is included. For example:

```
macosx# mkdir newdir
macosx# chmod +a "cxfs allow read,only_inherit" newdir
macosx# ls -led newdir
```

```
drwxr-xr-x+ 2 userA ptg 6 Jan 20 11:20 newdir
0: user:userA allow list,add_file,search,delete,add_subdirectory,delete_child,
  readattr,writeattr,readextattr,writeextattr,readsecurity,writesecurity,chmod
1: user:userA deny
2: group:ptg allow list,search,readattr,readextattr,readsecurity
3: group:ptg deny add_file,delete,add_subdirectory,delete_child,writeattr,
  writeextattr,writesecurity,chmod
4: group:everyone allow list,search,readattr,readextattr,readsecurity
5: group:everyone deny add_file,delete,add_subdirectory,delete_child,writeattr,
  writeextattr,writesecurity,chmod
6: user:userA allow list,add_file,search,delete,add_subdirectory,delete_child,
  readattr,writeattr,readextattr,writeextattr,readsecurity,writesecurity,chmod,
  file_inherit,directory_inherit,only_inherit
7: user:userA deny file_inherit,directory_inherit,only_inherit
8: group:everyone deny add_file,delete,add_subdirectory,delete_child,writeattr,
  writeextattr,writesecurity,chmod,file_inherit,directory_inherit,only_inherit
9: user:cxfs allow list,readattr,readextattr,readsecurity,file_inherit,
  directory_inherit,only_inherit
10: user:cxfs deny add_file,search,delete,add_subdirectory,delete_child,writeattr,
  writeextattr,writesecurity,chmod,file_inherit,directory_inherit,only_inherit
11: group:ptg allow list,search,readattr,readextattr,readsecurity,file_inherit,
  directory_inherit,only_inherit
12: group:ptg deny add_file,delete,add_subdirectory,delete_child,writeattr,
  writeextattr,writesecurity,chmod,file_inherit,directory_inherit,only_inherit
13: group:everyone allow list,search,readattr,readextattr,readsecurity,
  file_inherit,directory_inherit,only_inherit
14: group:everyone deny add_file,delete,add_subdirectory,delete_child,writeattr,
  writeextattr,writesecurity,chmod,file_inherit,directory_inherit,only_inherit
```

The base ACL is created if its not specified and removing the default ACL is a matter of removing rules until only the base rules are present, at which point the ACL will be removed.

HBA Installation for Mac OS X

CXFS for Mac OS X supports Apple Computer, Inc. host bus adapters (HBAs).

Note: The procedures in this section may be performed by you or by a qualified service representative. You must be logged in as `root` to perform the steps listed in this section.

This section discusses the following:

- "Installing the Apple HBA" on page 73
- "Installing the Fibre Channel Utility for Mac OS X" on page 73
- "Configuring Two or More Apple HBA Ports" on page 74
- "Using `point-to-point` Fabric Setting for Apple HBAs" on page 74

Installing the Apple HBA

Do the following:

1. Install the Apple HBA into a spare PCI, PCI-X, or PCI Express slot in the Mac OS X node, according to the manufacturer's instructions. Do not connect the HBA to the Fibre Channel switch at this time.
-

Note: Apple HBAs are normally shipped with copper SFPs and copper cables, so additional optic SFPs and optic cables may be required.

2. Reboot the node.

Installing the Fibre Channel Utility for Mac OS X

Do the following:

1. Install the configuration utility from the CD distributed with the Apple HBA. To do this, copy **Mac OS X Utilities/Fibre Channel Utility** from the CD to your **Application** directory.
2. Run the Fibre Channel Utility after it is copied to the node. The tool will list the HBA on the left-hand side of the window. Select the **Apple FC card** item to

display the status of the ports via a pull-down menu. Initially, each port will report that it is up (even though it is not connected to the switch), and the speed and port topology will configure automatically.

3. Connect one of the HBA ports to the switch via a Fibre Channel cable. After a few seconds, close and relaunch the Fibre Channel Utility. Select the **Apple FC card** item and then the connected port from the drop-down list to display the speed of the link.

Repeat these steps for the second HBA port if required.

4. *(Optional)* If necessary, use Apple's `/sbin/fibreconfig` tool to modify port speed and topology. See the man page for details.

The CXFS `fabric_dump` tool can also be of use in verifying Fibre Channel fabric configuration. See "CXFS Commands on Mac OS X" on page 60.

Configuring Two or More Apple HBA Ports

The Mac OS X node does its own path management for paths that go to the same RAID controller and thus only presents one `/dev` device to userspace per RAID controller. Even if multiple paths exist to a RAID controller, you will only see one `/dev` device.

Therefore, the Fibre Channel Utility does not support masking logical units (LUNs) on specific ports. However, if the first port can see all of the LUNs, the default is that all I/O will go through a single port. To avoid this, configure the switch so that each port can see a different set of LUNs. You can achieve this by zoning the switch or by using multiple switches, with different controllers and HBA ports to each switch.

Using point-to-point Fabric Setting for Apple HBAs

SGI recommends that you use the manual `point-to-point` fabric setting rather than rely on automatic detection, which can prove unreliable after a reboot.

Preinstallation Steps for Mac OS X

This section provides an overview of the steps that you or a qualified Apple service representative will perform on your Mac OS X nodes prior to installing the CXFS software. It contains the following sections:

- "Adding a Private Network for Mac OS X Nodes" on page 75
- "Verifying the Private and Public Networks for Mac OS X" on page 76
- "Disabling Power Saving Modes for Mac OS X" on page 77

Adding a Private Network for Mac OS X Nodes

The following procedure provides an overview of the steps required to add a private network to the Mac OS X system. A private network is required for use with CXFS. See "Use a Private Network" on page 15.

You may skip some steps, depending upon the starting conditions at your site. For details about any of these steps, see the Mac OS X system documentation.

1. Install Mac OS X and configure the machine's hostname (see "Configuring Hostnames on Mac OS X" on page 63) and IP address on its public network interface.
2. Add the IP addresses and hostnames of other machines in the cluster to the `/etc/hosts` file. You should be consistent about specifying the hostname or the fully qualified domain name for each host. A common convention is to name the CXFS private network address for each host as `hostname-priv`.
3. Install a second network interface card if necessary as per the manufacturer's instructions.
4. Configure the second network interface by using the following menu selection:

System Preferences

> **Network**

> *(select the device for the second network and specify its information)*

Select the second network interface (most likely PCI Ethernet Slot 1), and specify the IP address, subnet mask, and router. The private network interface should not require a DNS server because the private network address of other cluster nodes should be explicitly listed in the `/etc/hosts` file. Relying on a

DNS server for private network addresses introduces another point of failure into the cluster and must be avoided.

5. Confirm the configuration using `ifconfig` to list the network interfaces that are up:

```
macosx# ifconfig -u
```

In general, this should include `en0` (the onboard Ethernet) and `en1` (the additional PCI interface), but the names of these interfaces may vary.

For more information, see the `ifconfig` man page.

Verifying the Private and Public Networks for Mac OS X

Verify each interface by using the `ping` command to connect to the public and private network addresses of the other nodes that are in the CXFS pool.

For example:

```
macosx# grep cxfsmac2 /etc/hosts
134.14.55.115 cxfsmac2
macosx# ping -c 3 134.14.55.115
PING 134.14.55.115 (134.14.55.115): 56 data bytes
64 bytes from 134.14.55.115: icmp_seq=0 ttl=64 time=0.247 ms
64 bytes from 134.14.55.115: icmp_seq=1 ttl=64 time=0.205 ms
64 bytes from 134.14.55.115: icmp_seq=2 ttl=64 time=0.197 ms

--- 134.14.55.115 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.197/0.216/0.247 ms
```


Disabling Power Saving Modes for Mac OS X

Note the following:

- CXFS does not support the energy-saving mode on Mac OS X. If this mode is enabled, the Mac OS X node will lose CXFS membership and unmount the CXFS filesystem whenever it is activated.

Select the following to disable the energy-saving mode:

System Preferences

> **Energy Saver**

> **Put the computer to sleep when it is inactive for**

> **Never**

- Clients connected to a DDN RAID should have disk sleep disabled. Uncheck the following selection:

System Preferences

> **Energy Saver**

> **Put the hard disk(s) to sleep when possible**

- Never put CXFS clients to sleep. Select the following:

System Preferences

> **Energy Saver**

> **Put the computer to sleep when it is inactive**

> **NEVER**

Client Software Installation for Mac OS X

Installing the CXFS client software for Mac OS X requires approximately 30 MB of space.

To install the required software on a Mac OS X node, SGI personnel will do the following:

1. Read the *SGI InfiniteStorage Software Platform* release notes and CXFS release notes in the `/docs` directory on the ISSP DVD and late-breaking caveats on Supportfolio.
2. Verify that the node is running a supported Mac OS X operating system according to the Mac OS X installation guide. Use the following command to display the currently installed system:

```
macosx# uname -r
```

This command should return a Darwin kernel value of 9.8.0 or later for Leopard and 10.2.0 or later for Snow Leopard.

3. As `root` or a user with administrative privileges, transfer the client software that was downloaded onto a server-capable node during its installation procedure using `ftp`, `rcp`, or `scp`. The location of the disk image on the server will be as follows:

```
/usr/cluster/client-dist/CXFS_VERSION/macosx/MAC_VERSION/noarch/cxfs.dmg
```

Note: You must transfer the disk image to the `root` home directory (`/`) or your own home directory in order to make it visible with the **Finder** tool.

4. Double-click the transferred **cxfs.dmg** file to mount the disk image.
5. Double click **cxfs.pkg** to begin the installation.
6. Click **continue** when you see the following message:

```
message : This package contains a program that determines  
if the software can be installed. Are you sure you want to continue
```

7. Click **continue** when you see the following message:

```
The installer will guide you through the steps necessary to  
install CXFS for Mac OS X. To get started, click Continue
```

This will launch the installation application, which will do the following:

- a. Display the CXFS Mac OS X release note. Read the release note and click **continue**.

- b. Display the license agreement. Read the agreement and click **agree** if you accept the terms.
- c. Perform a standard installation of the software on the root drive volume.



Caution: Do not choose **Change install location**.

8. Choose **Continue Installation** at the following message:

Installation of this software requires you to restart your computer when the installation is done. Are you sure you want to install the software now?

9. After the install succeeds, click the highlighted **Restart** button to reboot your machine.

I/O Fencing for Mac OS X

I/O fencing is required on Mac OS X nodes in order to protect data integrity of the filesystems in the cluster. The `cxfs_client` software automatically detects the world wide port names (WWPNs) of any supported host bus adapters (HBAs) for Mac OS X nodes that are connected to a switch that is configured in the cluster database. These HBAs are available for fencing.

However, if no WWPNs are detected, the following messages will be logged to the `/var/log/cxfs_client` file:

```
hba_wwpn_list warning: No WWPN found from IO Registry
cis_get_hbas warning: Not able to find WWN (err=Device not
configured). Falling back to "/etc/fencing.conf".
cis_config_swports_set error fetching hbas
```

If no WWPNs are detected, you can manually specify the WWPNs in the fencing file.

Note: This method does not work if the WWPNs are partially discovered.

The `/etc/fencing.conf` file enumerates the WWPNs for all of the HBAs that will be used to mount a CXFS filesystem. There must be a line for the HBA WWPN as a 64-bit hexadecimal number.

Note: The WWPN is that of the HBA itself, **not** any of the devices that are visible to that HBA in the fabric.

If used, `/etc/fencing.conf` must contain a simple list of WWPNs, one per line. You must update it whenever the HBA configuration changes, including the replacement of an HBA.

Do the following:

1. Set up the switch and HBA. See the release notes for supported hardware.
2. Follow the Fibre Channel cable on the back of the node to determine the port to which it is connected in the switch. Ports are numbered beginning with 0. (For example, if there are 8 ports, they will be numbered 0 through 7.)
3. Use the `telnet` command to connect to the switch and log in as user `admin`. (On Brocade switches, the password is `password` by default).
4. Execute the `switchshow` command to display the switches and their WWPN numbers. For example:

```
brocade04:admin> switchshow
switchName:      brocade04
switchType:      2.4
switchState:     Online
switchRole:      Principal
switchDomain:    6
switchId:        fffc06
switchWwn:       10:00:00:60:69:12:11:9e
switchBeacon:    OFF
port 0: sw Online      F-Port 20:00:00:01:73:00:2c:0b
port 1: cu Online      F-Port 21:00:00:e0:8b:02:36:49
port 2: cu Online      F-Port 21:00:00:e0:8b:02:12:49
port 3: sw Online      F-Port 20:00:00:01:73:00:2d:3e
port 4: cu Online      F-Port 21:00:00:e0:8b:02:18:96
port 5: cu Online      F-Port 21:00:00:e0:8b:00:90:8e
port 6: sw Online      F-Port 20:00:00:01:73:00:3b:5f
port 7: sw Online      F-Port 20:00:00:01:73:00:33:76
port 8: sw Online      F-Port 21:00:00:e0:8b:01:d2:57
port 9: sw Online      F-Port 21:00:00:e0:8b:01:0c:57
port 10: sw Online     F-Port 20:08:00:a0:b8:0c:13:c9
```

```
port 11: sw Online      F-Port  20:0a:00:a0:b8:0c:04:5a
port 12: sw Online      F-Port  20:0c:00:a0:b8:0c:24:76
port 13: sw Online      L-Port  1 public
port 14: sw No_Light
port 15: cu Online      F-Port  21:00:00:e0:8b:00:42:d8
```

The WWPN is the hexadecimal string to the right of the port number. For example, the WWPN for port 0 is 2000000173002c0b (you must remove the colons from the WWPN reported in the `switchshow` output to produce the string to be used in the fencing file).

5. Edit or create `/etc/fencing.conf` and add the WWPN for the port determined in step 2. (Comment lines begin with #.)

For dual-ported HBAs, you must include the WWPNs of any ports that are used to access cluster disks. This may result in multiple WWPNs per HBA in the file; the numbers will probably differ by a single digit.

For example, if you determined that port 0 is the port connected to the switch, your fencing file should contain the following:

```
# WWPN of the HBA installed on this system
#
2000000173002c0b
```

6. To configure fencing, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Start/Stop `cxfs_client` for Mac OS X

The `/Library/StartupItems/cxfs/cxfs` script will be invoked automatically during normal system startup and shutdown procedures. This script starts and stops the `cxfs_client` daemon.

To start `cxfs_client` manually, enter the following:

```
macosx# sudo /Library/StartupItems/cxfs/cxfs start
```

To stop `cxfs_client` manually, enter the following:

```
macosx# sudo /Library/StartupItems/cxfs/cxfs stop
```

To stop and start `cxfs_client` manually, enter the following:

```
macosx# sudo /Library/StartupItems/cxfs/cxfs restart
```

To prevent the automatic startup of `cxfs_client` on boot, move the `/Library/StartupItems/cxfs` directory out of `/Library/StartupItems`.

Maintenance for Mac OS X

This section contains the following:

- "Updating the CXFS Software for Mac OS X" on page 82
- "Modifying the CXFS Software for Mac OS X" on page 82
- "Removing the CXFS Software for Mac OS X" on page 83
- "Recognizing Storage Changes for Mac OS X" on page 83
- "Switching Between 64-bit Kernel and 32-bit Kernel on Snow Leopard" on page 83

Updating the CXFS Software for Mac OS X

Do the following:

1. Ensure that no applications on the node are accessing files on a CXFS filesystem
2. Run the new CXFS software package, which will update all CXFS software.
3. Reboot.

Modifying the CXFS Software for Mac OS X

You can modify the behavior of the CXFS client daemon (`cxfs_client`) by placing options in the `/usr/cluster/bin/cxfs_client.options` file. The available options are documented in the `cxfs_client` man page.



Caution: Some of the options are intended to be used internally by SGI only for testing purposes and do not represent supported configurations. Consult your SGI service representative before making any changes.

To see if `cxfs_client` is using the options in `cxfs_client.options`, enter the following:

```
macosx# ps -axwww | grep cxfs
```

For example:

```
macosx# ps -axwww | grep cxfs
611 ??          0:06.17 /usr/cluster/bin/cxfs_client -D trace -z
```

Removing the CXFS Software for Mac OS X

After terminating any applications that access CXFS filesystems on the Mac OS X node, execute the following:

```
macosx# sudo /usr/cluster/bin/uninstall-cxfs
```

Restart the system to unload the CXFS module from the Mac OS X kernel.

Recognizing Storage Changes for Mac OS X

If you make changes to your storage configuration, you may have to reboot your machine because there is currently no mechanism in Mac OS X to reprobe the storage.

Switching Between 64-bit Kernel and 32-bit Kernel on Snow Leopard

To determine whether your Snow Leopard machine boots with the 32-bit kernel or the 64-bit kernel, you can examine the output from the `system_profiler` command.

For example, the following output indicates 32-bit:

```
snow$ system_profiler | grep -i kernel
Kernel Version: Darwin 10.2.0
64-bit Kernel and Extensions: No
```

The following output indicates 64-bit:

```
snow$ system_profiler | grep -i kernel
Kernel Version: Darwin 10.2.0
64-bit Kernel and Extensions: Yes
```

To use the System Profiler GUI:

1. Select the following menu:

Apple Menu
 > **About This Mac**
 > **More Info**

2. In the left pane, click **Software**
3. Examine the **64-bit Kernel and Extensions** field:
 - **No** indicates 32-bit
 - **Yes** indicates 64-bit

If you need to switch between 64-bit and 32-bit, see the following website:

<http://support.apple.com/kb/HT3773>

GRIO on Mac OS X

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the Mac OS X platform if GRIO is enabled on the server-capable administration node. Application bandwidth reservations must be explicitly released by the application before exit. If the application terminates unexpectedly or is killed, its bandwidth reservations are not automatically released and will cause a bandwidth leak. If this happens, the lost bandwidth could be recovered by rebooting the client node.

A Mac OS X node can mount a GRIO-managed filesystem and supports node-level reservations. A Mac OS X node will interoperate with the dynamic bandwidth allocator for all I/O outside of any reservation.

For more information, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7 and the *Guaranteed-Rate I/O Version 2 for Linux Guide*.

XVM Failover V2 on Mac OS X

Following is an example of the `/etc/failover2.conf` file on Mac OS X:

```
/dev/rxvm-200400a0b80cd5fe-000 affinity=1 preferred
/dev/rxvm-200500a0b80cd5fe-000 affinity=2

/dev/rxvm-200400a0b80cd5fe-001 affinity=2
/dev/rxvm-200500a0b80cd5fe-001 affinity=1 preferred
```

The device is the node's WWN plus the LUN number.

Note: Even if multiple paths exist to a RAID controller, you will only see one `/dev` device. The Mac OS X node does its own path management for paths that go to the same RAID controller and thus only presents one `/dev` device to userspace per RAID controller. See "Configuring Two or More Apple HBA Ports" on page 74.

For more information, see:

- "XVM Failover and CXFS" on page 8
- The comments in the `/etc/failover2.conf` file,
- *CXFS 6 Administration Guide for SGI InfiniteStorage*
- *XVM Volume Manager Administrator's Guide*

Troubleshooting for Mac OS X

This section discusses the following:

- "The `cxfs_client` Daemon is Not Started on Mac OS X" on page 86
- "XVM Volume Name is Too Long on Mac OS X" on page 86
- "Large Log Files on Mac OS X" on page 86

For general troubleshooting information, see:

- Chapter 7, "General Troubleshooting" on page 191
- Appendix D, "Error Messages" on page 213

The `cxfs_client` Daemon is Not Started on Mac OS X

Confirm that the `cxfs_client` is not running. The following command would list the `cxfs_client` process if it were running:

```
macosx# ps -auxww | grep cxfs_client
```

Check the `cxfs_client` log file for errors.

Restart `cxfs_client` as described in "Start/Stop `cxfs_client` for Mac OS X" on page 81 and watch the `cxfs_client` log file for errors.

XVM Volume Name is Too Long on Mac OS X

On Mac OS X nodes, the following error message in the `system.log` file indicates that the volume name is too long and must be shortened so that the Mac OS X node can recognize it:

```
devfs: volumename name slot allocation failed (Errno=63)
```

See "Limitations and Considerations on Mac OS X" on page 62.

Large Log Files on Mac OS X

The `/var/log/cxfs_client` log file may become quite large over a period of time if the verbosity level is increased.

To manually rotate this log file, use the `-z` option in the `/usr/cluster/bin/cxfs_client.options` file.

See the `cxfs_client.options` man page and "Log Files on Mac OS X" on page 61.

Reporting Mac OS X Problems

Before reporting a problem about to SGI, you should run the `cxfsdump` command:

```
macosx# /usr/cluster/bin/cxfsdump
```

This will collect the following information:

- System information
- CXFS registry settings
- CXFS client logs
- CXFS version information
- Network settings
- Event log

The `cxfsdump -help` command displays a help message.

Send the `tar.gz` file that is created in the `/var/cluster/cxfsdump-data/date_time` directory to SGI.

You should also obtain information about the entire cluster by running the `cxfsdump` utility on a server-capable administration node. See the information in the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Windows Platforms

CXFS supports a client-only node running the Windows operating systems defined in the CXFS Windows release notes. The information in this chapter applies to all of these versions of Windows unless otherwise noted.

This chapter contains the following sections:

- "CXFS on Windows" on page 90
- "HBA Installation for Windows" on page 129
- "Preinstallation Steps for Windows" on page 130
- "Client Software Installation for Windows" on page 132
- "Postinstallation Steps for Windows" on page 140
- "I/O Fencing for Windows" on page 142
- "Start/Stop the CXFS Client Service for Windows" on page 146
- "Maintenance for Windows" on page 147
- "GRIO on Windows" on page 152
- "XVM Failover V2 on Windows" on page 153
- "Mapping XVM Volumes to Storage Targets on Windows" on page 158
- "Troubleshooting for Windows" on page 160
- "Reporting Windows Problems" on page 171

Note: Your **Start** menu may differ from the examples shown in this guide, depending upon your start menu preferences. For example, this guide describes selecting the control panel as follows:

Start
 > **Control Panel**

However, on your system this menu could be as follows:

Start
 > **Settings**
 > **Control Panel**

CXFS on Windows

This section contains the following information about CXFS on Windows:

- "Requirements for Windows" on page 91
- "CXFS Commands on Windows" on page 94
- "Log Files and Cluster Status for Windows" on page 94
- "Functional Limitations and Considerations for Windows" on page 101
- "Performance Considerations for Windows" on page 109
- "Access Controls for Windows" on page 110
- "System Tunables for Windows" on page 123

Requirements for Windows

In addition to the items listed in "Requirements" on page 6, CXFS requires at least the following:

- A supported Windows operating system.

Note: SGI has fully tested the latest Service Packs of the platforms in the first list below. Some earlier Service Packs of these platforms do not have significant differences that affect CXFS and SGI therefore expects them to be fully functional, although SGI has not tested them.

- Supported and fully tested:
 - Windows XP SP3
 - Windows XP/64 SP2
 - Windows Server 2003 SP2
 - Windows Server 2003/64 SP2
 - Windows Server 2008 Service Pack 2
 - Windows Server 2008/64 Service Pack 2
 - Windows Vista SP2
 - Windows Vista/64 SP2
 - Windows 7
- Supported but not tested (expected to work):
 - Windows XP SP2
 - Windows XP/64
 - Windows Server 2003 R2
 - Windows Server 2003/64 R2
 - Windows Server 2008
 - Windows Server 2008/64

- Windows Vista SP1
- Windows Vista/64 SP1

Note: Earlier versions of Windows XP and Windows Vista do have significant differences that cause problems with CXFS and are therefore not supported. For more details about Windows platform support, see the CXFS Windows release note.

- One of the following:
 - An Intel Pentium or compatible processor
 - Xeon family with Intel Extended Memory 64 Technology (EM64T) processor architecture, or AMD Opteron family, AMD Athlon family, or compatible processor
- Minimum RAM requirements (more will improve performance): at least 1 GB of physical RAM
- A minimum of 10 MB of free disk space
- Host bus adapter (HBA):
 - LSI Logic LSI 2Gb/4Gb, single/dual/quad-port , PCI-X/PCI-E HBAs
 - QLogic QLA2200, QLA2310, QLA2342, or QLA2344 HBAs
 - ATTO Celerity Fibre Channel HBAs:
 - CTFC-41XS-0R0 FC/HBA single PCI X
 - CTFC-42XS-BRK FC/HBA dual PCI X
 - CTFC-41ES-0R0 FC/HBA single PCI e
 - CTFC-42ES-BRK FC/HBA dual PCI e
 - CTFC-44ES-0R0 FC/HBA quad PCI e
- The following LSI Logic software from the <http://www.lsilogic.com> website:
 - Windows Server 2003: 1.26.01
 - Windows XP: 1.26.01

- Windows Vista, Windows Server 2008, and Windows 7: 1.26.01
- The following QLogic software from the <http://www.qlogic.com> website:
 - QLA2200:
 - Windows Server 2003: v8.1.5.15
 - Windows XP: v8.1.5.12

Note: Windows Vista, Windows Server 2008, and Windows 7 do not support the QLA2200.

- QLA2310, QLA2342 and QLA2344:
 - Windows XP, Windows Server 2003: v9.1.4.10 SCSI Miniport Driver
 - Windows XP, Windows Server 2003: v9.1.4.15 STOR Miniport Driver
 - Windows Vista, Windows Server 2008, and Windows 7: v9.1.7.15 STOR Miniport for both 32-bit and 64-bit
- SANsurfer FC HBA Manager 5.0.0 build 17

You should install the documentation associated with the software. See the SANsurfer `README` for the default password. Follow the QLogic instructions to install the driver, the SANsurfer NT Agent, and the SANsurfer Manager software. See the SANsurfer help for information on target persistent binding.

- The following ATTO software from the <http://www.attotech.com> website:
 - Windows XP, Windows Server 2003, Windows Vista x86 (32 and 64 bit) version 2.62, Windows Server 2008, and Windows 7
 - Windows Flash Bundle version 2007_11_13
 - Windows Configuration Tool version 3.17

For the latest information, see the CXFS Windows release notes.

CXFS Commands on Windows

The following commands are shipped as part of the CXFS Windows package:

```
%windir%\system32\cxfs_client.exe  
%ProgramFiles%\CXFS\cxfs_info.exe  
%ProgramFiles%\CXFS\cxfsbmap.exe  
%ProgramFiles%\CXFS\cxfsdp.exe  
%ProgramFiles%\CXFS\cxfsdump.exe  
%ProgramFiles%\CXFS\grioadmin.exe  
%ProgramFiles%\CXFS\griomon.exe  
%ProgramFiles%\CXFS\griooqs.exe  
%ProgramFiles%\CXFS\idbg.exe  
%ProgramFiles%\CXFS\xvm.exe
```

The CXFS software for Windows also includes the `grio2lib` library.

A single CXFS Client service and two CXFS filesystem drivers are installed as part of the Windows installation. The service and the CXFS filesystem drivers can be configured to run automatically when the first user logs into the node.

The command `%ProgramFiles%\CXFS\cxfs_info.exe` displays the current state of the node in the cluster in a graphical user interface. See "Log Files and Cluster Status for Windows" and "Verifying the Cluster Status" on page 183.

For information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7 and "GRIO on Windows" on page 152.

Log Files and Cluster Status for Windows

This section discusses the following:

- "Viewing the Log Files" on page 95
- "Tuning the Verbosity of CXFS Messages to the System Event Log" on page 95
- "Using the CXFS Info Window" on page 96

Viewing the Log Files

The Windows node will log important events in the system event log. You can view these events by selecting the following:

```
Start
  > Control Panel
      > Administrative Tools
          > Event Viewer
```

For information about the log files created on server-capable administration nodes, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*. The CXFS Client service will also log important information to the following file:

```
%ProgramFiles%\CXFS\log\cxfs_client.log
```

When CXFS is first installed, the log file is automatically rotated when it grows to 10 MB. This is set by the `-z` option in the CXFS Client service **Additional arguments** window during installation (see Figure 5-7 on page 135) and may be adjusted by following the steps described in "Modifying the CXFS Software for Windows" on page 147.

Tuning the Verbosity of CXFS Messages to the System Event Log

You can specify the level of verbosity for CXFS messages that are logged to the System Event log by editing the Registry Editor:

```
Start
  > Run
      > regedit
```

Navigate to the following value:

```
HKEY_LOCAL_MACHINE
  > SYSTEM
      > CurrentControlSet
          > Services
              > CXFS
                  > Parameters
                      > LogVerbosity
```

The data type for **LogVerbosity** is **REG_DWORD**. By default, it is set to 4, which is fairly verbose. The higher the number, the more messages that are logged. You can reset the value to one of the following, as appropriate for your site:

Value	Events Logged
0	None (disables the logging of all events from the CXFS driver)
1	Panic events only
2	Alert and panic events
3	Warning, alert, and panic events
4	Notice, warning, alert, and panic events (default)
5	Informational, notice, warning, alert, and panic events
6	Debug, informational, notice, warning, alert, and panic events events

Note: If you enter a value that is not in the range 0 through 6, it will be rejected and the CXFS driver will then use the default value of 4 instead.

Using the CXFS Info Window

You may wish to keep the **CXFS Info** window open to check the cluster status and view the log file. To open this informational window on any Windows system, select the following:

```
Start
  > Programs
    > CXFS
      > CXFS Info
```

The top of **CXFS Info** window displays the overall state of the cluster environment:

- Number of stable nodes
- Status of the `cms` cluster membership daemon
- Status of XVM
- Status of filesystems
- Status of the cluster
- Status of the local node

The **CXFS Info** window provides the following tabs:

- **Nodes** displays each node in the cluster, its state, and its cell ID number. For more information, see "Verifying the Cluster Status" on page 183.

Figure 5-1 shows an example of the **CXFS Info** window **Nodes** tab.

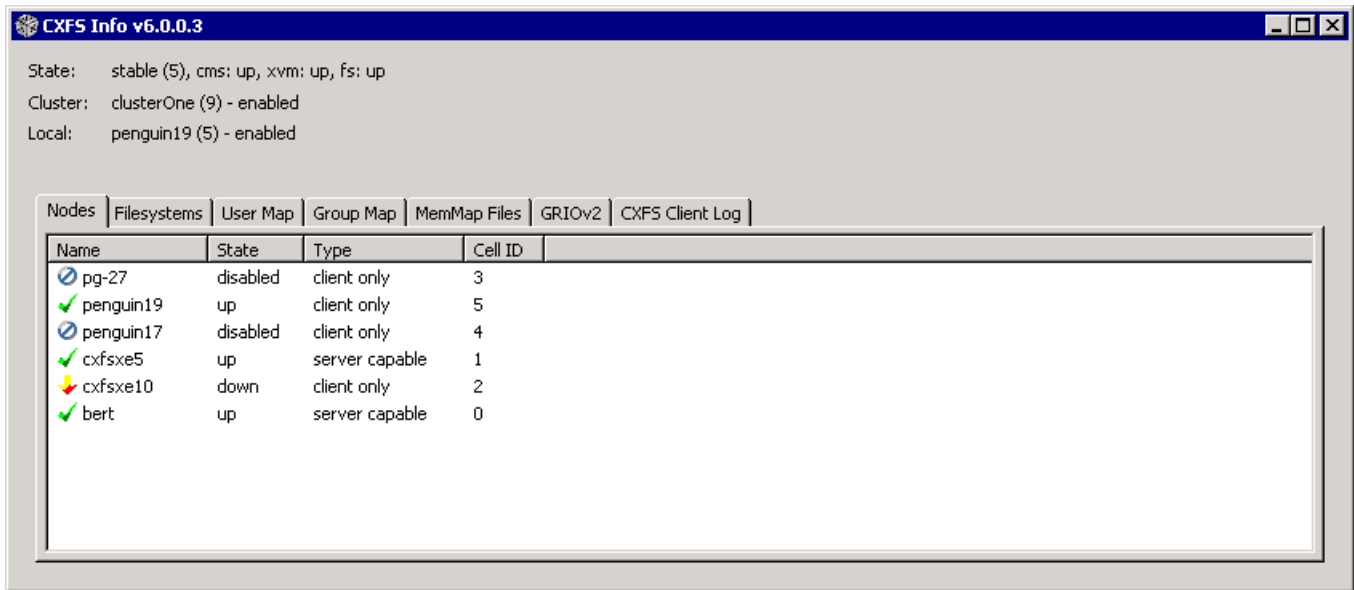


Figure 5-1 CXFS Info Window — Nodes Tab Display

- **Filesystems** displays each CXFS filesystem, its state, size, and other statistics. Figure 5-2 shows an example.

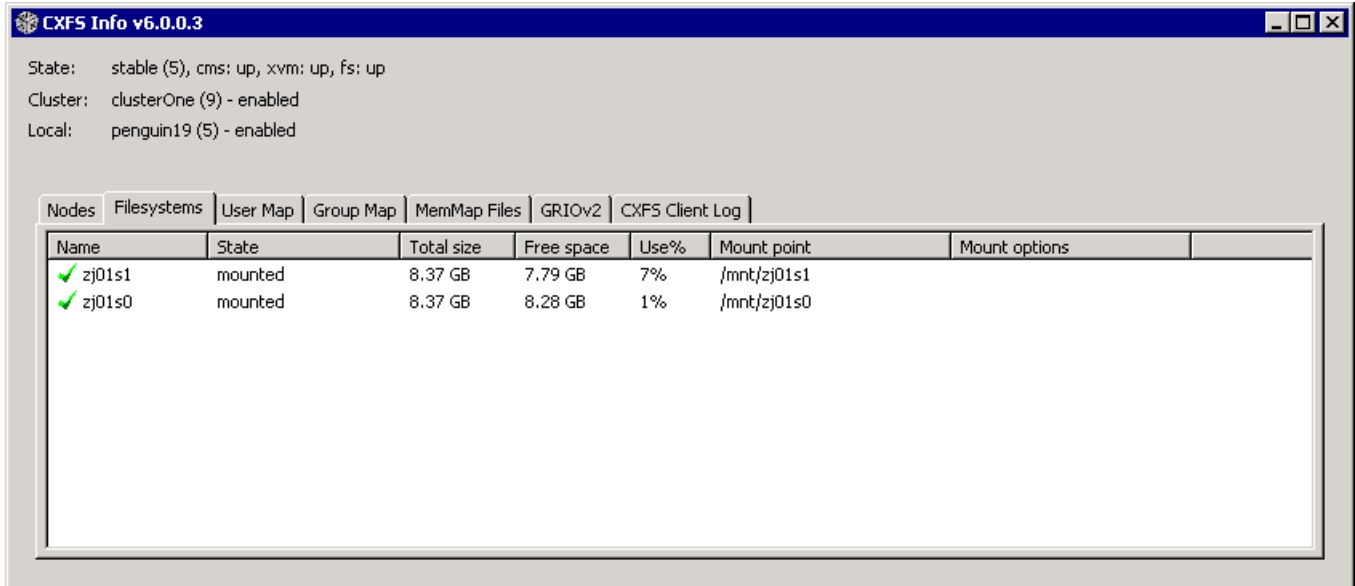


Figure 5-2 CXFS Info Window — Filesystems Tab

- **User Map** displays the usernames that are mapped to UNIX user identifiers. Figure 5-3 shows an example.

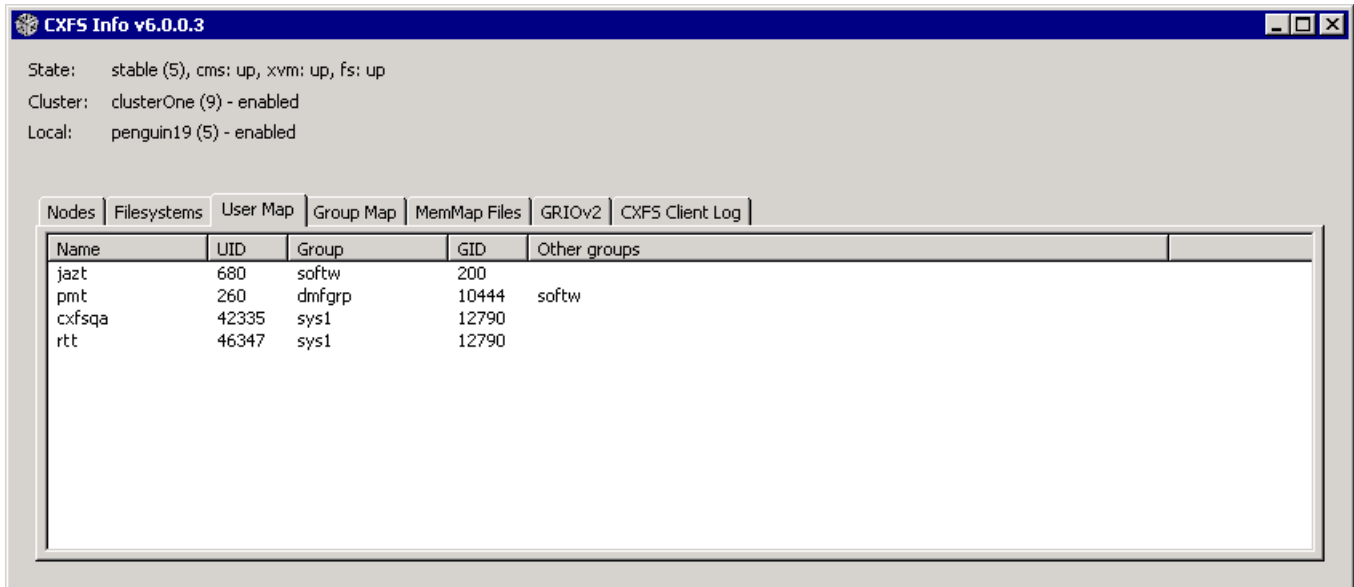


Figure 5-3 CXFS Info Window — User Map Tab

- **Group Map** displays the groups that are mapped to UNIX group identifiers. Figure 5-4 shows an example.

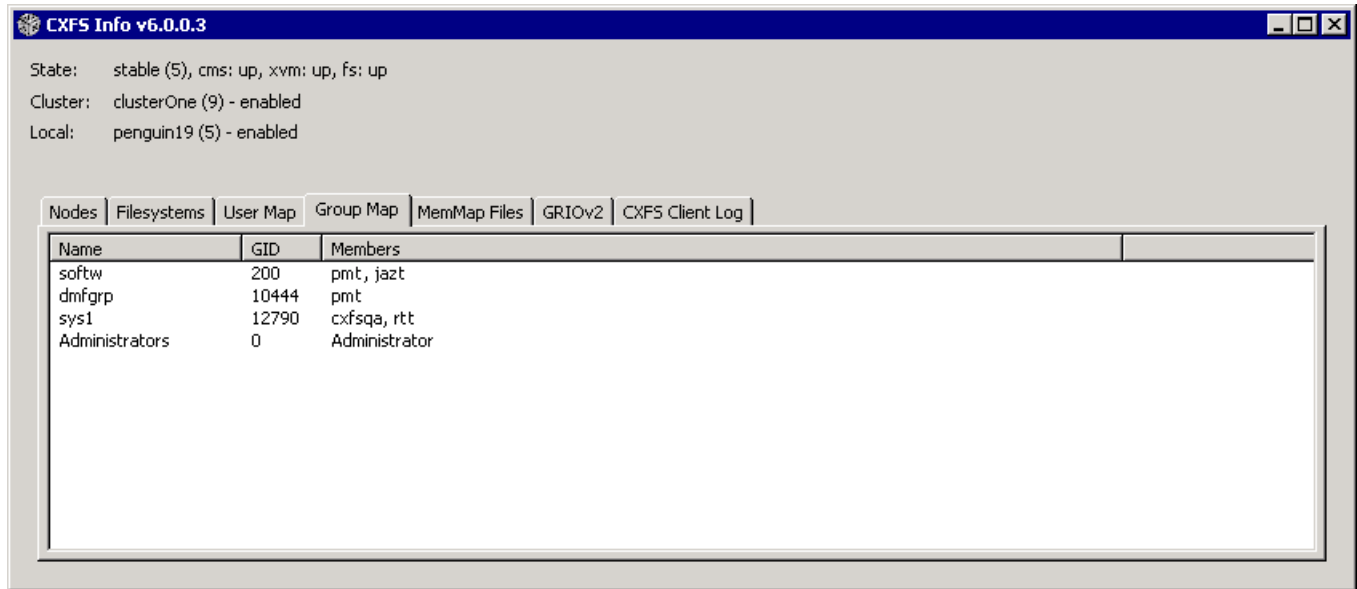


Figure 5-4 CXFS Info Window — Group Map Tab

- **GRIOv2** displays each guaranteed-rate I/O (GRIO) stream, its reservation size, and other statistics. See "GRIO on Windows" on page 152.
 - **CXFS Client log** displays the log since the CXFS Client service last rebooted. It highlights the text in different colors based on the severity of the output:
 - Red indicates an error, which is a situation that will cause a problem and must be fixed
 - Orange indicates a warning, which is a situation that might cause a problem and should be examined
 - Black indicates general log information that can provide a frame of reference
 - Green indicates good progress in joining membership and mounting filesystems
- Figure 5-5 shows an example.

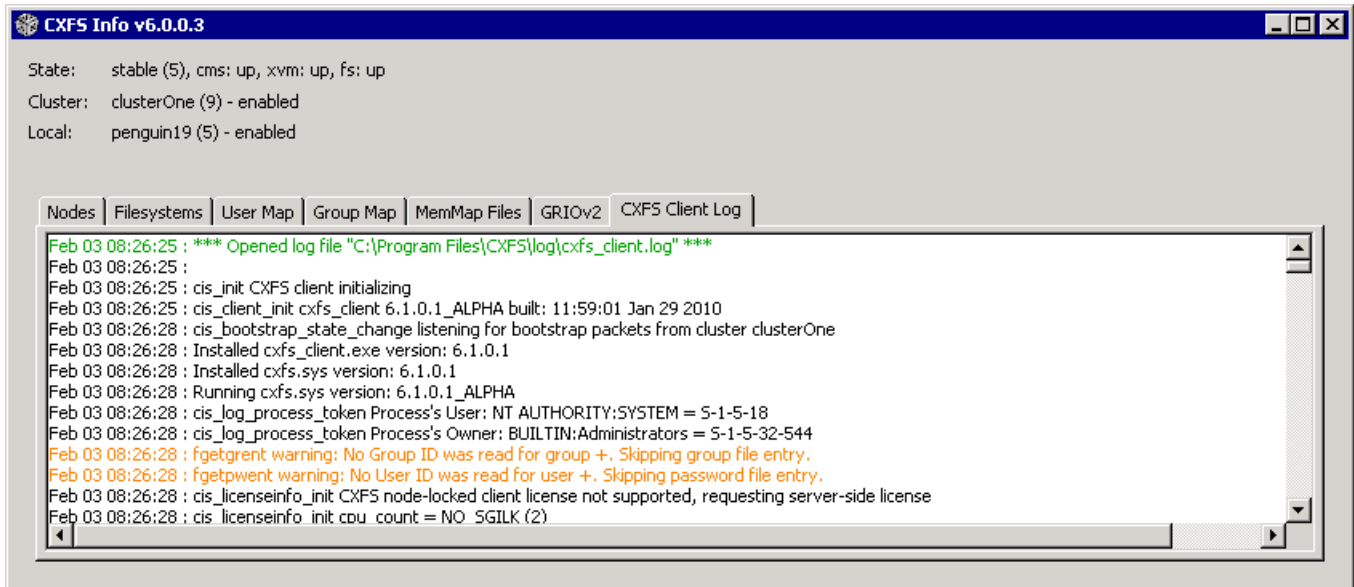


Figure 5-5 CXFS Info Window — CXFS Client Log Tab

The **CXFS Info** icon in the task bar will change from green to yellow or red depending on the state of the node in the cluster:

- Green indicates that the node is in the membership, everything is fully functional, and all enabled filesystems are mounted
- Yellow indicates an in-between state (neither inactive nor stable state)
- Red indicates that CXFS is not running (inactive state)

Also see Figure 5-15 on page 152.

Functional Limitations and Considerations for Windows

This section contains the following:

- "*Warning: DiskManager for Windows Vista, Windows Server 2008, and Windows 7 Destroys Data*" on page 102
- "*Windows Server 2008 Marks Newly Discovered Disks Offline*" on page 102

- "UNIX Perspective of CXFS for Windows" on page 103
- "Windows Perspective of CXFS for Windows" on page 104
- "Forced Unmount on Windows" on page 105
- "Define LUN 0 on All Storage Devices for Windows XP and Windows Server 2003" on page 105
- "Memory-Mapping Large Files for Windows" on page 106
- "CXFS Mount Scripts for Windows" on page 106
- "Norton Ghost Prevents Mounting Filesystems" on page 106
- "Mapping Network and CXFS Drives" on page 106
- "Windows Filesystem Limitations" on page 106
- "XFS Filesystem Limitations" on page 107
- "User Account Control for Windows Vista, Windows Server 2008, and Windows 7" on page 107
- "Windows Disks Using DDN RAID" on page 107
- "Windows Time Service Default Synchronization" on page 108

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 207.

Warning: DiskManager for Windows Vista, Windows Server 2008, and Windows 7 Destroys Data

After CXFS is installed on the Windows Vista, Windows Server 2008, or Windows 7 platform, you **must not** use DiskManager to format the disks that are exported from the RAID.



Warning: Using DiskManager to format the disks will destroy the XVM labels and therefore the data on the RAID. To make the XVM volumes functional again, you would have to rebuild the XVM labels.

Windows Server 2008 Marks Newly Discovered Disks Offline

On Windows Server 2008, newly discovered Fibre Channel disks are marked *Offline* the first time they are discovered. This can result in write-permission errors

being returned to the user or application from CXFS. To resolve this problem, use the DiskManager to mark the disks `Online`.

UNIX Perspective of CXFS for Windows

This section describes the differences and limitations of a CXFS filesystem on a Windows node from a UNIX perspective:

- Windows nodes can support multiple CXFS filesystems mounted under a single drive letter. Only one CXFS drive letter may be configured on a Windows node.

The top-level file structure under the CXFS drive letter consists of an in-memory directory structure that mimics the mount points on the server-capable administration node. The CXFS software creates these directories before mounting the CXFS filesystems. For example, a CXFS filesystem with a mount point of `/mnt/cxfs` on a CXFS Windows node configured to use drive letter `X` will create `X:\mnt\cxfs` during filesystem mount process.

This file structure supports only creating and deleting directories; there is no support for creating and deleting regular files, renaming directories, and so on. Attempts to perform unsupported actions will generally result in an invalid parameter error. You can perform normal filesystem operations on files and directories beneath the mount points, but an application that must write to the directory directly under the CXFS drive letter will fail.

Note: A CXFS mount point or directory beneath a mount point can be mapped to another drive letter by using the `subst` command from a command shell to which the application can write. See "Application Cannot Create File Under CXFS Drive Letter" on page 168.

- A Windows node can support regular files, directories, and links. However, it does not support other XFS file types.
- Symbolic links cannot be distinguished from normal files or directories on a Windows node. Opening a symbolic link will open the target of the link, or will report `file not found` if it is a dangling link.
- By default, copying a symbolic link will result in copying the file or directory that the link refers to, rather than the normal UNIX behavior that copies the link itself. To copy the link itself, you must use the `cp -a` option.

For example, on a normal Linux platform:

```
linux# touch file; mkdir dir; ln -sf file file_link; \  
ln -sf dir dir_link; cp -a file_link file_link_copy; cp -a dir_link dir_link_copy  
  
linux# file *  
dir/:          directory  
dir_link:     symbolic link to 'dir'  
dir_link_copy: symbolic link to 'dir'  
file:         empty  
file_link:    symbolic link to 'file'  
file_link_copy: symbolic link to 'file'
```

On a Windows platform using a cygwin shell:

```
user@host /cygdrive/x/mnt/lun0  
$ touch file; mkdir dir; ln -sf file file_link; ln -sf dir dir_link; \  
cp -a file_link file_link_copy; cp -a dir_link dir_link_copy  
  
user@host /cygdrive/x/mnt/lun0  
$ file *  
dir:          directory  
dir_link.lnk: symbolic link to 'dir'  
dir_link_copy.lnk: MS Windows shortcut  
file:         empty  
file_link.lnk: symbolic link to 'file'  
file_link_copy.lnk: MS Windows shortcut
```

Windows Perspective of CXFS for Windows

This section describes the differences and limitations of a CXFS filesystem on a Windows node in comparison to other Windows filesystems from a Windows perspective:

- Avoid using duplicate filenames in the same directory that vary only in case. CXFS is case-sensitive, but some Windows applications may not maintain the case of all filenames, which may result in unexpected behavior.
- CXFS software does not export 8.3 alternative filenames. Older Windows applications that only support 8.3 filenames may be unable to open files with longer filenames and may fail with file not found errors.
- Avoid using completely uppercase 8.3 filenames. If you use completely uppercase 8.3 filenames, some applications (including Windows Explorer) may incorrectly

assume that only 8.3 filenames are supported by the filesystem and will not preserve case.

- Install the CXFS software components onto a NTFS partition rather than a FAT partition. The security of the following files cannot be guaranteed if these files are installed onto a FAT filesystem:

`%ProgramFiles%\CXFS\passwd`

`%ProgramFiles%\CXFS\group`

- There is no recycle bin; deleted files are permanently deleted.
- There is no automatic notification of directory changes performed by other nodes in the cluster. Applications (such as Windows Explorer) will not automatically update their display if another node adds or removes files from the directory currently displayed.
- A CXFS filesystem cannot be used as the boot partition of a Windows node.
- The volume properties window in Windows Explorer for the CXFS drive letter will display the total capacity of all mounted filesystems and the largest free space on any one of those filesystems.

Forced Unmount on Windows

A forced unmount causes all processes that have open files on the specified filesystem to be unconditionally killed and therefore permit the filesystem to be unmounted without delay. SGI recommends that you enable the forced unmount feature on CXFS filesystems. See:

- "Enable Forced Unmount When Appropriate" on page 18
- "Forced Unmount of CXFS Filesystems" on page 181

Define LUN 0 on All Storage Devices for Windows XP and Windows Server 2003

Windows XP and Windows Server 2003 (and therefore CXFS) might not detect any LUNs on a storage device if LUN 0 is not defined on the storage device. This problem may occur when **CXFS Info** reports that XVM is up, but one or more filesystems are not mounted and CXFS therefore retries the mount continuously. For more information about this issue, see the following (the problem exists for all supported Windows XP and Windows Server 2003 platforms):

<http://support.microsoft.com/kb/821666/en-us>

Memory-Mapping Large Files for Windows

You can memory-map a file much larger than 2 GB under Windows, but only up to 2 GB of that file in one or more parts can be mapped into a process at any one time on a 32-bit platform. See the Windows Platform Software Development Kit for more details.

CXFS Mount Scripts for Windows

Windows does not support the CXFS mount scripts.

Norton Ghost Prevents Mounting Filesystems

If Norton Ghost is installed on a node, CXFS cannot mount filesystems on the mount-point driver letter. You must uninstall Norton Ghost in order to use CXFS.

Mapping Network and CXFS Drives

Under Windows XP, users may define their own local set of drive letter mappings that can override the global settings for the host. When identifying the filesystem mapped to a drive letter, Windows XP will check the local mappings and may hide CXFS from the user. Users and administrators of CXFS Windows nodes must avoid mapping network and CXFS drives to the same drive letter.

Windows Filesystem Limitations

A Windows node running CXFS has the following filesystem limitations:

- Does not support shutdown of the CXFS driver via the device manager. If restarting the CXFS Client service fails to achieve membership, you must reboot the Windows node.
- Does not support opportunistic locking, also known as *oplocks*. Hosts that are using a CXFS Windows node as an SMB server will not be able to cache data locally. The workaround is to use NFS or Samba to export the filesystem on one of the server-capable administration nodes.
- Enforces the Windows file sharing options when opening a file on the same node, but does not enforce it on other nodes in the cluster.

XFS Filesystem Limitations

Support for unwritten extents is limited on Windows nodes. However, reading and writing unwritten extents will work correctly in the absence of concurrent reading and writing of the same file extent elsewhere in the cluster.

User Account Control for Windows Vista, Windows Server 2008, and Windows 7

By default, User Account Control is enabled for Windows Vista, Windows Server 2008, and Windows 7, but it is not appropriate for use with CXFS. You must therefore disable user account control. See step 4 in "Client Software Installation for Windows" on page 132.

Windows Disks Using DDN RAID

For Windows disks using DDN RAID (versions prior to rm6700), you should set the disk spin-down value so that disks never spin down. (Spinning down a disk could issue a `STOP LUN` command to the storage.)

On Windows XP and Windows Server 2003, do the following:

1. Select the following:

Start
 > **Control Panel**
 > **Power Options**

2. In the **Plugged in** scheme, select **Never** for **Turn off hard disks**.

On Windows Vista and Windows Server 2008, do the following:

1. Select the following:

Start
 > **Control Panel**
 > **Power Options**

2. Select the **High performance** preferred plans.
3. Click the **Change plan settings** link.
4. Click the **Change advanced power settings** link. This will pop-up the **Advanced settings** dialog.

5. Locate the **Hard disk** entry in the tree and expand it.
6. Change the **Turn off hard disk after : Setting: 20 Minutes** setting to **Never**.
7. Click **OK** to save the changes.

On Windows 7, do the following:

1. Select the following:

Start
 > **Control Panel**
 > **Power Options**

2. Click the down-arrow on the right side opposite from **Show additional plans** to reveal the **High performance** plan.
3. Select the **High performance** preferred plans.
4. Click the **Change plan settings** link.
5. Locate the **Hard disk** entry in the tree and expand it.
6. Verify that the setting is **Turn off hard disk after : Setting: Never**.
7. Click **OK** to exit and save any changes.

Windows Time Service Default Synchronization

The Windows Time Service is capable of synchronizing with NTP servers, but the default configuration synchronizes only once a week. SGI recommends modifying the default configuration to keep Windows nodes more closely synchronized. See the Microsoft documentation for the Windows Time Service for details, including the following:

<http://technet.microsoft.com/en-us/library/bb490605.aspx>

Performance Considerations for Windows

The following are performance considerations on a CXFS Windows node, in addition to the limitations described in "Use CXFS when Appropriate" on page 12:

- Using CIFS to share a CXFS filesystem from a CXFS Windows node to another Windows host is not recommended for the following reasons:
 - Metadata operations sent to the Windows node must also be sent to the CXFS metadata server causing additional latency
 - CXFS Windows does not support opportunistic locking, which CIFS uses to improve performance (see "Windows Filesystem Limitations" on page 106)

For optimal performance, SGI recommends that you use Samba on the CXFS active metadata server to export CXFS filesystems to other nodes that are not running CXFS.

- Windows supplies autonotification APIs for informing applications when files or directories have changed on the local client. With each notification, Windows Explorer will do a full directory lookup. Under CXFS, directory lookups can require multiple RPCs to the server (about 1 per 30 files in the directory), resulting in a linear increase in network traffic. This can grow to megabytes per second for directories with large numbers of files.

For better performance, do one of the following:

- Select the destination folder itself
- Close the drive tree or mount point folder by clicking on the | + | on the drive icon or mount point folder
- If you open the Windows Explorer **Properties** window on a directory, it will attempt to traverse the filesystem in order to count the number and size of all subdirectories and files; this action is the equivalent of running the UNIX `du` command. This can be an expensive operation, especially if performed on directories between the drive letter and the mount points, because it will traverse all mounted filesystems.
- Virus scanners, Microsoft Find Fast, and similar tools that traverse a filesystem are very expensive on a CXFS filesystem. Such tools should be configured so that they do not automatically traverse the CXFS drive letter.
- The mapping from Windows user and group names to UNIX identifiers occurs as the CXFS software starts up. In a Windows domain environment, this process can

take a number of seconds per user for usernames that do not have accounts within the domain. If you are using a `passwd` file for user identification and the file contains a number of unknown users on the Windows node, you should remove users who do not have accounts on the Windows nodes from the `passwd` file that is installed on the Windows nodes.

This issue has less impact on Windows nodes in a workgroup than on those in a domain because the usernames can be quickly resolved on the node itself, rather than across the network to the domain controller.

- With 1-GB fabric to a single RAID controller, it is possible for one 32-bit 33-MHz QLogic card to reach the bandwidth limitations of the fabric, and therefore there will be no benefit from load balancing two HBAs in the same PCI bus. This can be avoided by using 2-GB fabric and/or multiple RAID controllers.
- For load balancing of two HBAs to be truly beneficial, the host must have at least one of the following three attributes:
 - A 64-bit PCI bus
 - A 66-MHz PCI bus
 - Multiple PCI buses
- Applications running on a Windows node should perform well when their I/O access patterns are similar to those described in "When to Use CXFS" on page 12.
- The maximum I/O size issued by the QLogic HBA to a storage target and the command tag queue length the HBA maintains to each target can be configured in the registry. See "System Tunables for Windows" on page 123.

Access Controls for Windows

The XFS filesystem used by CXFS implements and enforces UNIX mode bits and POSIX access control lists (ACLs), which are quite different from Windows file attributes and access control lists. The CXFS software attempts to map Windows access controls to the UNIX access controls for display and manipulation, but there are a number of features that are not supported (or may result in unexpected behavior) that are described here.

This section contains the following:

- "User Identification for Windows" on page 111
- "User Identification Mapping Methods for Windows" on page 112
- "Matching Windows Users and Groups with CXFS Users and Groups" on page 115
- "Enforcing Access to Files and Directories for Windows" on page 115
- "Viewing and Changing File Attributes with Windows Explorer" on page 116
- "Viewing and Changing File Permissions with Windows Explorer" on page 117
- "Viewing and Changing File Access Control Lists (ACLs) for Windows" on page 119
- "Effective Access for Windows" on page 120
- "Restrictions with file ACLs for Windows" on page 120
- "Inheritance and Default ACLs for Windows" on page 121

User Identification for Windows

The CXFS software supports several user identification mechanisms, which are described in "User Identification Mapping Methods for Windows" on page 112. Only Windows user and group names that exactly match entries in the configured user list will be mapped to those user IDs (UIDs) and group IDs (GIDs). Windows users and groups that do not have a match in the mapping list will be mapped to `nobody`. Users and groups in the mapping list that do not match a Windows user or group are ignored. To avoid confusion and improve performance, you should remove unused users and groups from the mapping list.

The following additional mappings are automatically applied:

- User **Administrator** is mapped to `root` (UID = 0)
- Group **Administrators** is mapped to `sys` (GID = 0)

A user's default UNIX GID is the default GID in the `passwd` listing for the user and is not based on a Windows group mapped to a UNIX group name.

You can display the users and groups that have been successfully mapped by looking at the tables for the **User Map** and **Group Map** tabs in the **CXFS Info** window.

The following sections assume that a CXFS Windows node was configured with the following `passwd` and `group` files:

```
C:\> type %ProgramFiles%\CXFS\passwd
root::0:0:Super-User:/root:/bin/tcsh
guest::998:998:Guest Account:/usr/people/guest:/bin/csh
fred::1040:402:Fred Costello:/users/fred:/bin/tcsh
diane::1052:402:Diane Green:/users/diane:/bin/tcsh

C:\> type %ProgramFiles%\CXFS\group
sys::0:root,bin,sys,adm
root::0:root
guest:*:998:
video::402:fred,diane
audio::403:fred
```

User Identification Mapping Methods for Windows

User identification can be performed by choosing one of the following methods for the **User ID mapping lookup sequence** item of the **Enter CXFS Details** window:

- **files:** `/etc/passwd` and `/etc/group` files from the metadata server copied onto the clients. The format of the `passwd` and `group` files for CXFS Windows is the same as on the metadata server. In the `passwd` file, only the user name, `uid` and `gid` fields are used. In the `group` file, only the group name, `gid`, and member list are used. Other fields may be removed to make the file more readable. If you select this method, you must install the `passwd` and `group` files immediately after installing the CXFS software, as described in "Performing User Configuration for Windows" on page 141.
- **ldap_actedir:** Windows Active Directory server with Services for UNIX (SFU) installed, which uses lightweight directory access protocol (LDAP).

The **ldap_actedir** method configures the CXFS Windows software to communicate with the Active Directory for the CXFS node's domain. With the Windows Services for UNIX (SFU) extensions, the Active Directory User Manager lets you define UNIX identifiers for each user and export these identifiers as an LDAP database.

Permissions on the Active Directory server must allow Authenticated Users to read the SFU attributes from the server. Depending on the installation and configuration of the server, LDAP clients may or may not be able to access the

SFU attributes. For more information, see "CXFS Client Service Cannot Map Users other than Administrator for Windows" on page 165.

This configuration requires a domain controller that is installed with the following:

- Windows Server 2003 with Active Directory.
- Windows Services for UNIX (SFU) version 2 or later with the NFS server component installed. SGI recommends SFU version 3.5.

Note: The domain controller does not have to be a CXFS node.

- **ldap_generic:** Generic LDAP lookup for UNIX users and groups from another LDAP server.

The **ldap_generic** method configures the CXFS software to communicate with an LDAP database that maps user names and group names to UNIX identifiers.

Following is an example of a user record:

```
# ldap2, people, example.com
dn: uid=ldap2,ou=people,dc=example,dc=com
cn: Ldap Tu User
givenName: Ldap
homeDirectory: /home/ldap2
loginShell: /bin/bash
objectClass: top
objectClass: posixAccount
objectClass: inetOrgPerson
sn: User
uid: ldap2
uidNumber: 1102
gidNumber: 1100
```

Following is an example of a group record:

```
# ldapgroup, group, example.com
dn: cn=ldapgroup,ou=group,dc=example,dc=com
cn: ldapgroup
gidNumber: 1100
memberUid: ldap1,ldap2
objectClass: top
objectClass: posixGroup
objectClass: groupOfNames
```

Note: For the group mapping, you must use `memberUid`, not `member`. You should also use the simple `uid` (such as `myname`) rather than Descriptive Notation (`uid=myname,ou=people,dc=mycompany,dc=com`).

For an example of the window, see Figure 5-7 on page 135.

You must select one of these as the primary mapping method during installation, but you can change the method at a later time, as described in "Modifying the CXFS Software for Windows" on page 147.

Optionally, you can select a secondary mapping method that will be applied to users that are not covered by the first method. If you choose a primary and a secondary mapping method, one of them must be **files**.

For example, suppose the user has selected **ldap_generic** as the primary method and **files** as the secondary method. A user mapping will be created for all suitable **ldap_generic** users and this mapping will be extended with any additional users found in the secondary method (**files**). The primary method will be used to resolve any duplicate entries.

Suppose the primary method (**ldap_generic**) has users for UIDs 1, 2 and 3, and the secondary method (**files**) has users for UIDs 2 and 4. The username for UIDs 1, 2 and 3 will be determined by the **ldap_generic** method and the username for UID 4 will be determined by the **files** method. If the LDAP lookup failed (such as if the LDAP server was down), a user mapping for UIDs 2 and 4 would be generated using the **files** method.

The default behavior is to use the **files** method to map Windows usernames to UNIX UIDs and GIDs, with no secondary method selected.

Regardless of the method used, the consistent mapping of usernames is a requirement to ensure consistent behavior on all CXFS nodes. Most platforms can be configured to use an LDAP database for user identification.

Matching Windows Users and Groups with CXFS Users and Groups

If a file (or a component of the path to the file) has an owner or group that does not exist on the Windows node, Windows may assume that there is a significant security vulnerability and may not allow access to the file or path. This may be true even if the file and every component of the path is world readable/writable.

To avoid this problem, do the following:

1. Create Windows Users and Groups for every user and group likely to be found on the CXFS filesystems.
2. Configure CXFS user and group mapping so that the above Windows Users and Groups are mapped to the CXFS users and groups.

Enforcing Access to Files and Directories for Windows

Access controls are enforced on the CXFS metadata server by using the mapped UID and GID of the user attempting to access the file. Therefore, a user can expect the same access on a Windows node as any other node in the cluster when mounting a given filesystem. Access is determined using the file's ACL (if one is defined) or by using the file's mode bits.

ACLs that are set on any files or directories are also enforced as they would be on any Linux node. The presentation of ACLs is customized to the interfaces of Windows Explorer, so the enforcement of the ACL may vary from an NTFS ACL that is presented in the same way. A new file will inherit the parent directory default ACL, if one is defined.

The user `Administrator` has read and write access to all files on a CXFS filesystem, in the same way that `root` has superuser privileges on a UNIX node.

The following example is a directory listing on the metadata server:

```
MDS# ls -l
drwxr-x---  2 fred  video          6 Nov 20 13:33 dir1
-rw-r----- 1 fred  audio          0 Nov 20 12:59 file1
-rw-rw-r--  1 fred  video          0 Nov 20 12:59 file2
```

Users will have the following access to the contents of this directory:

- `dir1` will be readable, writable, and searchable by user `fred` and Administrator. It will be readable and searchable by other users in group `video`, and not accessible by all other users.
- `file1` will be readable and writable to user `fred` and Administrator on a CXFS Windows node. It can also be read by other users in group `audio`. No other users, including `diane` and `guest`, will be able to access this file.
- `file2` will be readable by all users, and writable by user `fred`, `diane` (because she is in group `video`), and Administrator.

Viewing and Changing File Attributes with Windows Explorer

File permissions may be viewed and manipulated in two different ways when using Windows Explorer:

- By displaying the list of attributes in a detailed directory listing; this is the most limited approach
- By selecting properties on a file

The only file attribute that is supported by CXFS is the read-only attribute; other attributes will not be set by CXFS and changes to those attributes will be ignored.

If the user is not permitted to write to the file, the read-only attribute will be set. The owner of the file may change this attribute and modify the mode bits. Other users, including the user Administrator, will receive an error message if they attempt to change this attribute.

Marking a file read-only will remove the write bit from the user, group, and other mode bits on the file. Unsetting the read-only attribute will make the file writable by the owner only.

For example, selecting file properties on `file1` using Windows Explorer on a CXFS Windows node will display the read-only attribute unset if logged in as Administrator or `fred`, and it will be set for `diane` and `guest`.

Only user `fred` will be able to change the attribute on these files, which will change the files under UNIX to the following:

```
-r--r----- 1 fred  audio          0 Nov 20 12:59 file1
-r--r--r--  1 fred  video          0 Nov 20 12:59 file2
```


If fred then unset these flags, only he could write to both files:

```
-rw-r----- 1 fred  audio          0 Nov 20 12:59 file1
-rw-r--r--  1 fred  video          0 Nov 20 12:59 file2
```

Viewing and Changing File Permissions with Windows Explorer

By selecting the **Security** tab in the **File Properties** window of a file, a user may view and change a file's permissions with a high level of granularity.

Windows Explorer will list the permissions of the file's owner and the file's group. The **Everyone** group, which represents the mode bits for other users, will also be displayed if other users have any access to the file. Not all Windows permission flags are supported.

The permissions on file1 are displayed as follows:

```
audio (cxfs1\audio)          Allow: Read
Fred Costello (cxfs1\fred)   Allow: Read, Write
```

Using the **Advanced** button, file1 is displayed as follows:

```
Allow   Fred Costello (cxfs1\fred)   Special
Allow   audio (cxfs1\audio)         Read
```

User fred is listed as having **Special** access because the permission flags in the next example do not exactly match the standard Windows permissions for read and write access to a file. Select **Fred Costello** and then click **View/Edit** to display the permission flags listed in Table 5-1. (The table displays the permissions in the order in which they appear in the **View/Edit** window). You can choose to allow or deny each flag, but some flags will be ignored as described in Table 5-1.

Table 5-1 Permission Flags that May Be Edited

Permission	Description
Traverse Folder / Execute File	Used to display and change the execute mode bit on the file or directory
List Folder / Read Data	Used to display and change the read mode bit on the file or directory
Read Attributes	Set if the read mode bit is set; changing this flag has no effect
Read Extended Attributes	Set if the read mode bit is set; changing this flag has no effect
Create Files / Write Data	Used to display and change the write mode bit on the file or directory
Create Folders / Append Data	Set if the write mode bit is set; changing this flag has no effect
Write Attributes	Set if the write mode bit is set; changing this flag has no effect
Write Extended Attributes	Set if the write mode bit is set; changing this flag has no effect
Delete Subfolders and Files	Set for directories if you have write and execute permission on the directory; changing this flag has no effect
Delete	Never set (because delete depends on the parent directory permissions); changing the flag has no effect
Read Permissions	Always set; changing the flag has no effect
Change Permissions	Always set for the owner of the file and the user Administrator; changing this flag has no effect
Take Ownership	Always set for the owner of the file and the user Administrator; changing this flag has no effect

The permissions for file2 are displayed as follows:

```

Everyone                Allow: Read
video (cxfs1\video)    Allow: Read, Write
Fred Costello (cxfs1\fred) Allow: Read, Write

```

The permissions for dir1 are displayed as follows:

```

Fred Costello (cxfs1\fred) Allow:
Video (cxfs1\video)       Allow:

```

Note: In this example, the permission flags for directories do not match any of the standard permission sets, therefore no Allow flags are set.

In general, you must click the **Advanced** button to see the actual permissions of directories. For example:

Allow	Fred Costello	Special	This folder only
Allow	video	Read & Execute	This folder only

The `dir1` directory does not have a default ACL, so none of these permissions are inherited, as indicated by the `This folder only` tag, when a new subdirectory or file is created.

Viewing and Changing File Access Control Lists (ACLs) for Windows

If the file or directory has an ACL, the list may include other users and groups, and the `CXFS ACL Mask` group that represents the Linux ACL mask. See the `chacl(1)` man page on the server-capable administration node for an explanation of Linux ACLs and the mask bits. The effective permissions of all entries except for the owner will be the intersection of the listed permissions for that user or group and the mask permissions. Therefore, changing the `CXFS ACL Mask` permissions will set the maximum permissions that other listed users and groups may have. Their access may be further constrained in the specific entries for those users and groups.

By default, files and directories do not have an ACL, only mode bits, but an ACL will be created if changes to the permissions require an ACL to be defined. For example, granting or denying permissions to another user or group will force an ACL to be created. Once an ACL has been created for a file, the file will continue to have an ACL even if the permissions are reduced back to only the owner or group of the file. The `chacl(1)` command under Linux can be used to remove an ACL from a file.

For example, fred grants diane read access to `file1` by adding user `diane` using the file properties dialogs, and then deselecting `Read & Execute` so that only `Read` is selected. The access list now appears as follows:

audio (cxfs1\audio)	Allow: Read
Diane Green (cxfs1\diane)	Allow: Read
Fred Costello (cxfs1\fred)	Allow: Read, Write

After clicking **OK**, the properties for `file1` will also include the `CXFS ACL Mask` displayed as follows:

```
audio (cxfs1\audio)           Allow: Read
CXFS ACL Mask (cxfs1\CXFS...) Allow: Read
Diane Green (cxfs1\diane)     Allow: Read
Fred Costello (cxfs1\fred)    Allow: Read, Write
```

Note: You should select and deselect entries in the `Allow` column only, because UNIX ACLs do not have the concept of `Deny`. Using the `Deny` column will result in an ACL that allows everything that is not denied, even if it is not specifically selected in the `Allow` column, which is usually not what the user intended.

Effective Access for Windows

The effective access of user `diane` and group `audio` is read-only. Granting write access to user `diane` as in the following example does not give `diane` write access because the mask remains read-only. However, because user `fred` is the owner of the file, the mask does not apply to his access to `file1`.

For example:

```
audio (cxfs1\audio)           Allow: Read
CXFS ACL Mask (cxfs1\CXFS...) Allow: Read
Diane Green (cxfs1\diane)     Allow: Read, Write
Fred Costello (cxfs1\fred)    Allow: Read, Write
```

Restrictions with file ACLs for Windows

If the users and groups listed in a file's permissions (whether mode bits and/or ACL entries) cannot be mapped to users and groups on the Windows node, attempts to display the file permissions in a file properties window will fail with an unknown user or group error. This prevents the display of an incomplete view, which could be misleading.

Both the owner of the file and the user `Administrator` may change the permissions of a file or directory using Windows Explorer. All other users will get a `permission denied` error message.

Note: A user must use a node that is **not** running Windows to change the ownership of a file because a Windows user takes ownership of a file with Windows Explorer, rather than the owner giving ownership to another user (which is supported by the UNIX access controls).

Inheritance and Default ACLs for Windows

When a new file or directory is created, normally the mode bits are set using a umask of 022. Therefore, a new file has a mode of 644 and a new directory of 755, which means that only the user has write access to the file or directory.

You can change this umask during CXFS installation or later by modifying the installation. For more information, see "Client Software Installation for Windows" on page 132 and "Inheritance and Default ACLs for Windows" on page 121.

The four umask options available during installation or modification correspond to the following umask values:

000	Everyone can write
002	User and group can write
022	User only can write (default)
222	Read only (no one can write)

Therefore, creating a file on a UNIX CXFS client results in a mode of 644 for a umask of 022:

```
admin% ls -lda .
drwxr-xr-x  3 fred      video           41 Nov 21 18:01 ./

admin% umask
0022

admin% touch file3
admin% ls -l file3
-rw-r--r--  1 fred      video           0 Nov 21 18:23 file3
```

For more information, see the `umask` man page on the server-capable administration node.

Creating a file in Windows Explorer on a Windows node will have the same result.

A Linux directory ACL may include a default ACL that is inherited by new files and directories, instead of applying the umask. Default ACLs are displayed in the Windows Explorer file permission window if they have been set on a directory. Unlike a Windows inheritable ACL on an NTFS filesystem, a Linux default ACL applies to both new files and subdirectories; there is no support for an inheritable ACL for new files and another ACL for new subdirectories.

The following example applies an ACL and a default ACL to `dir1` and then creates a file and a directory in `dir1`:

```
admin% chacl -b "u::rwx,g::r-x,u:diane:r-x,o::---,m::r-x" \  
          "u::rwx,g::r-x,u:diane:rwx,o::---,m::rwx" dir1  
admin% touch dir1/newfile  
admin% mkdir dir1/newdir  
admin% ls -D dir1  
newdir [u::rwx,g::r-x,u:diane:rwx,o::---,m::r-x/  
        u::rwx,g::r-x,u:diane:rwx,o::---,m::rwx]  
newfile [u::rw-,g::r-x,u:diane:rwx,o::---,m::r--]
```

The permissions for `dir1` will be as follows:

```
CXFS ACL Mask (cxfs1\CXFS...) Allow:  
Diane Green (cxfs1\diane) Allow:  
Fred Costello (cxfs1\fred) Allow: Read & Exec, List, Read, Write  
Video (cxfs1\video) Allow: Read & Exec, List, Read
```

After clicking on **Advanced**, the permissions displayed are as follows.:

Allow	Fred Costello	Special	This folder, subfolders and files
Allow	video	Read & Execute	This folder, subfolders and files
Allow	Diane Green	Read, Write & Exec	Subfolders and files
Allow	CXFS ACL Mask	Read, Write & Exec	Subfolders and files
Allow	Diane Green	Read & Exec	This folder only
Allow	CXFS ACL Mask	Read & Exec	This folder only

If an ACL entry is the same in the default ACL, a single entry is generated for the `This folder, subfolders and files` entry. Any entries that are different will have both `Subfolders and files` and `This folder only` entries.

Adding the first inheritable entry to a directory will cause CXFS to generate any missing ACL entries like the owner, group, and other users. The mode bits for these entries will be generated from the umask.

Adding different `Subfolders Only` and `Files Only` entries will result in only the first entry being used because a Linux ACL cannot differentiate between the two.

System Tunables for Windows

This section discusses the following topics:

- "Registry Modification" on page 123
- "Default Umask for Windows" on page 124
- "Maximum DMA Size for Windows" on page 124
- "Memory-Mapping Coherency for Windows" on page 125
- "DNLC Size for Windows" on page 125
- "Mandatory Locks for Windows" on page 126
- "User Identification Map Updates for Windows" on page 127
- "I/O Size Issues Within the QLogic HBA" on page 127
- "Command Tag Queueing (CTQ)" on page 128

Note: These system tunables are removed when the software is removed. They may need to be reset when downgrading the CXFS for Windows software.

You should only change system tunable parameters if you are fully aware of their consequences or when directed to do so by SGI support.

Registry Modification

In order to configure system tuning settings, you must modify the registry. Do the following:

1. Back up the registry before making any changes.
2. Click **Start**, select **Run**, and open the `regedit.exe` program.
3. Select **HKEY_LOCAL_MACHINE** and follow the tree structure down to the parameter you wish to change.
4. Reboot the system to apply the changes.



Caution: Only the parameters documented here may be changed to modify the behavior of CXFS. All other registry entries for CXFS must not be modified or else the software may no longer function.

Default Umask for Windows

The default umask that is set up during installation can be configured to a value not supported by the installer. For more information on the umask, see "Inheritance and Default ACLs for Windows" on page 121.

In **regedit**, navigate and edit the following value:

```
HKEY_LOCAL_MACHINE
> SYSTEM
  > CurrentControlSet
    > Services
      > CXFS
        > Parameters
          > DefaultUMask
```

This value specifies the umask in hexadecimal (and decimal), not its normal octal representation used on UNIX platforms.

Maximum DMA Size for Windows

By default, CXFS for Windows breaks down large direct I/O requests into requests no larger than 16 MB. You can change the size of these requests.

In **regedit**, navigate to the following value:

```
HKEY_LOCAL_MACHINE
> SYSTEM
  > CurrentControlSet
    > Services
      > CXFS
        > Parameters
```

Create a new **DWORD** key called `MaxDMASize` and specify the maximum I/O request size in bytes.

Memory-Mapping Coherency for Windows

By default, a CXFS Windows node enforces memory-mapping coherency by preventing other clients and the CXFS metadata server access to the file while it is mapped. This can cause problems for some applications that do not expect this behavior.

Microsoft Office applications and `Notepad.exe` use memory-mapped I/O to read and write files, but use byte-range locks to prevent two people from accessing the same file at the same time. The CXFS behavior causes the second Office application to hang until the file is closed by the first application, without displaying a dialog that the file is in use.

Backup applications that search the filesystem for modified files will stall when they attempt to back up a file that has been memory-mapped on a CXFS Windows node.

In **regedit**, navigate to the following value:

```
HKEY_LOCAL_MACHINE
> SYSTEM
  > CurrentControlSet
    > Services
      > CXFS
        > Parameters
```

You can disable this behavior in CXFS by changing the `DisableMemMapCoherency` parameter from 0 to 1 to avoid these problems. However, CXFS can no longer ensure data coherency if two applications memory-map the same file at the same time on different nodes in the cluster.



Caution: Use this option with extreme caution with multiple clients concurrently accessing the same files.

DNLC Size for Windows

The Directory Name Lookup Cache (DNLC) in a CXFS Windows node allows repetitive lookups to be performed without going to the metadata server for each component in a file path. This can provide a significant performance boost for applications that perform several opens in a deep directory structure.

In **regedit**, navigate to the following value:

```
HKEY_LOCAL_MACHINE
> SYSTEM
  > CurrentControlSet
    > Services
      > CXFS
        > Parameters
```

The `DnLcSize` parameter is set to 4096 by default. You can change it to a value from 0 (which disables the DNLC) to 100000. Values outside this range will be reset to 4096.

Note: Increasing the DNLC size can have a significant memory impact on the Windows node and the metadata server because they maintain data structures for every actively opened file on the CXFS clients. You should monitor the memory usage on these nodes before and after changing this parameter because placing nodes under memory pressure is counter-productive to increasing the DNLC size.

Mandatory Locks for Windows

By default, byte-range locks across the cluster are advisory locks, which do not prevent a rogue application from reading and writing to locked regions of a file.

Note: Windows filesystems (NTFS and FAT) implement a mandatory locking system that prevents applications from reading and writing to locked regions of a file. Mandatory locks are enabled within a Windows node.

In **regedit**, navigate to the following value:

```
HKEY_LOCAL_MACHINE
> SYSTEM
  > CurrentControlSet
    > Services
      > CXFS
        > Parameters
```

To enable mandatory byte-range locks across the cluster, set the `ForceMandatoryLocks` parameter to 1. Setting this parameter will adversely affect performance of applications using these locks.

User Identification Map Updates for Windows

User identification maps are updated automatically by the following triggers:

- An unmapped user logs into the system
- The `passwd` and/or `group` file is modified when the primary mapping method is **files**
- An LDAP database change is detected when the primary mapping method is **ldap_activedir** or **ldap_generic**

The most common trigger in a typical environment is when an unmapped user logs into the system; the other two triggers are generally static in nature.

Updating the map can be a resource-intensive operation in a domain environment. Therefore, by default, an update is triggered only when an unmapped user logs in and not more often than every 5 minutes.

To configure the minimum update interval using **regedit**, navigate to the following value:

```
HKEY_LOCAL_MACHINE
  > SYSTEM
    > CurrentControlSet
      > Services
        > CXFS_Client
          > Parameters
```

In the **regedit** menu, select:

```
Edit
  > New
    > DWORD Value
```

Enter `MinMapGenTime` for the name. Press **Enter** to edit the value, which is the minimum time between updates in minutes. The minimum time is 1 minute.

I/O Size Issues Within the QLogic HBA

The maximum size of I/O issued by the QLogic HBA defaults to only 256 KB. Many applications are capable of generating much larger requests, so you may want to increase this I/O size to the HBA's maximum of 1 MB.

To increase the size of the I/O, do the following:

1. In **regedit**, navigate to the following QLogic driver value:

```
HKEY_LOCAL_MACHINE
> SYSTEM
  > CurrentControlSet
    > Services
      > ql2xxx
        > Parameters
          > Device
```

2. Double-click **MaximumSGList:REG_DWORD:0x21**.
3. Enter a value from 16 through 255 (0x10 hexadecimal to 0xFF). A value of 255 (0xFF) enables the maximum 1-MB transfer size. Setting a value higher than 255 results in 64-KB transfers. The default value is 33 (0x21).
4. Exit **regedit**.
5. Reboot the system to apply the changes.

Command Tag Queueing (CTQ)

Command Tag Queueing (CTQ) is used by HBAs to manage the number of outstanding requests each adapter port has to each target. Adjusting this value (up or down) can improve the performance of applications, depending on the number of clients in the cluster and the number of I/O requests they require to meet the required quality of service.

You should only modify this setting for HBA ports that are to be used by CXFS. Do not modify ports used for local storage.

While it is possible to change this value with the volume mounted, I/O will halt momentarily and there may be problems if the node is under a heavy load.

Note: The Windows HBA may not recognize the CTQ setting placed on the disk by Linux nodes.

To configure the CTQ, use the management tool provided by the HBA. You may also be able to set the execution throttle in the HBA BIOS during boot-up by pressing a key combination when you see the HBA's BIOS message. You should use an execution throttle value (that is, how many commands will be queued by the HBA) in

the range 1 through 256. You must reboot the system to apply the changes. For more information, see the HBA card and driver documentation.

Note: Unlike CTQ, you cannot have separate depths per LUN. Execution throttle limits the number of simultaneous requests for **all** targets in the specified port.

HBA Installation for Windows

Note: SGI recommends that you use XVM failover V2 and disable any failover capability provided by Windows or the HBA. See "XVM Failover V2 on Windows" on page 153.

The Fibre Channel host bus adapter (HBA) should be installed according to the HBA vendor's hardware and driver installation instructions.

For information regarding large logical unit (LUN) support under Windows, see the HBA vendor's documentation and Microsoft's support database:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q310072>
<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q245637>

To confirm that the HBA and driver are correctly installed, select the following to display all of the LUNs visible to the HBA and listed within the Device Manager:

- Start**
 - > **Control Panel**
 - > **Administrative Tools**
 - > **Computer Management**
 - > **Device Manager**
 - > **View**
 - > **Devices by connection**

The Windows Device Manager hardware tree will differ from one configuration to another, so the actual location of the HBA within the Device Manager may differ. After it is located, any LUNs attached will be listed beneath it.

Preinstallation Steps for Windows

This section provides an overview of the steps that you or a qualified Windows service representative will perform on your Windows nodes prior to installing the CXFS software. It contains the following:

- "Adding a Private Network for Windows" on page 130
- "Verifying the Private and Public Networks for Windows" on page 130
- "Configuring the Windows Firewall for Windows" on page 131

Adding a Private Network for Windows

A private network is required for use with CXFS. See "Use a Private Network" on page 15.

Verifying the Private and Public Networks for Windows

You can confirm that the previous procedures to add private networks were performed correctly by using the `ipconfig` command in a DOS command shell.

Create a DOS command shell with the following sequence:

```
Start
  > Programs
    > Accessories
      > Command Prompt
```

In the following example, the 10 network is the private network and the 192.168.0 network is the public network on a Windows system:

```
C:\> ipconfig /all
Windows IP Configuration

Host Name . . . . . : cxfs1
Primary Dns Suffix . . . . . : cxfs-domain.sgi.com
Node Type . . . . . : Unknown
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : cxfs-domain.sgi.com
                                     sgi.com
```

Ethernet adapter Public:

```

Connection-specific DNS Suffix . : cxfs-domain.sgi.com
Description . . . . . : 3Com EtherLink PCI
Physical Address. . . . . : 00-01-03-46-2E-09
Dhcp Enabled. . . . . : No
IP Address. . . . . : 192.168.0.101
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.0.1
DNS Servers . . . . . : 192.168.0.x
    
```

Ethernet adapter Private:

```

Connection-specific DNS Suffix . :
Description . . . . . : 3Com EtherLink PCI
Physical Address. . . . . : 00-B0-D0-31-22-7C
Dhcp Enabled. . . . . : No
IP Address. . . . . : 10.0.0.101
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
    
```

Configuring the Windows Firewall for Windows

The Windows firewall will prevent a CXFS Windows node from achieving membership unless several ports are opened using the following applet:

```

Start
  > Control Panel
    > Windows Firewall
    
```

In the **Exceptions** tab, add the following **Ports**:

```

UDP on port 5449
TCP on port 5450
TCP on port 5451
UDP on port 5453
    
```

Client Software Installation for Windows

Note: This procedure assumes that the CXFS software is installed under the default path %ProgramFiles%\CXFS. If a different path is selected, then that path should be used in its place in the following instructions.

To install the CXFS client software on a Windows node, do the following:

1. Read the *SGI InfiniteStorage Software Platform* release notes CXFS release notes in the /docs directory on the ISSP DVD and any late-breaking caveats on Supportfolio.
2. Log onto the Windows node as Administrator.
3. Verify that the node has been updated to the correct service pack:

Start

- > **Programs**
 - > **Accessories**
 - > **System Tools**
 - > **System Information**
-

Note: If you must reinstall the operating system, disconnect the system from the fabric first.

4. (*Windows Vista, Windows Server 2008, and Windows 7 Only*) Disable User Account Control (requires administrator privileges). By default, User Account Control is enabled for Windows Vista, Windows Server 2008, and Windows 7, but it is not appropriate for use with CXFS. Do the following to disable it, according to OS type:
 - Windows Vista or Windows Server 2008:
 - a. Using the **User Accounts** control panel, click the **Turn User Account Control on or off** link.
 - b. Uncheck the **Use User Account Control (UAC) to help protect your computer** check box. Press the **OK** button to confirm your selection.
 - c. Reboot the system to apply the changes.

- Windows 7:
 - a. Using the **User Accounts** control panel, select **Change User Account Control Settings**.
 - b. In the **User Account Control Settings** window, move the slider to the bottom **Never notify** setting. Click **OK**.
 - c. Reboot the system to apply the changes
- 5. Transfer the client software that was downloaded onto a server-capable administration node during its installation procedure using `ftp`, `rcp`, or `scp`. The location of the Windows installation program on the server will be as follows:

```
/usr/cluster/client-dist/CXFS_VERSION/windows/all/noarch/setup.exe
```

6. Double-click the **setup.exe** installation program to execute it.
7. Acknowledge the software license agreement when prompted and read the CXFS Windows release notes, which may contain corrections to this guide.
8. Install the CXFS software, as shown in Figure 5-6. If the software is to be installed in a nondefault directory, click **Browse** to select another directory. Click **Next** when finished.

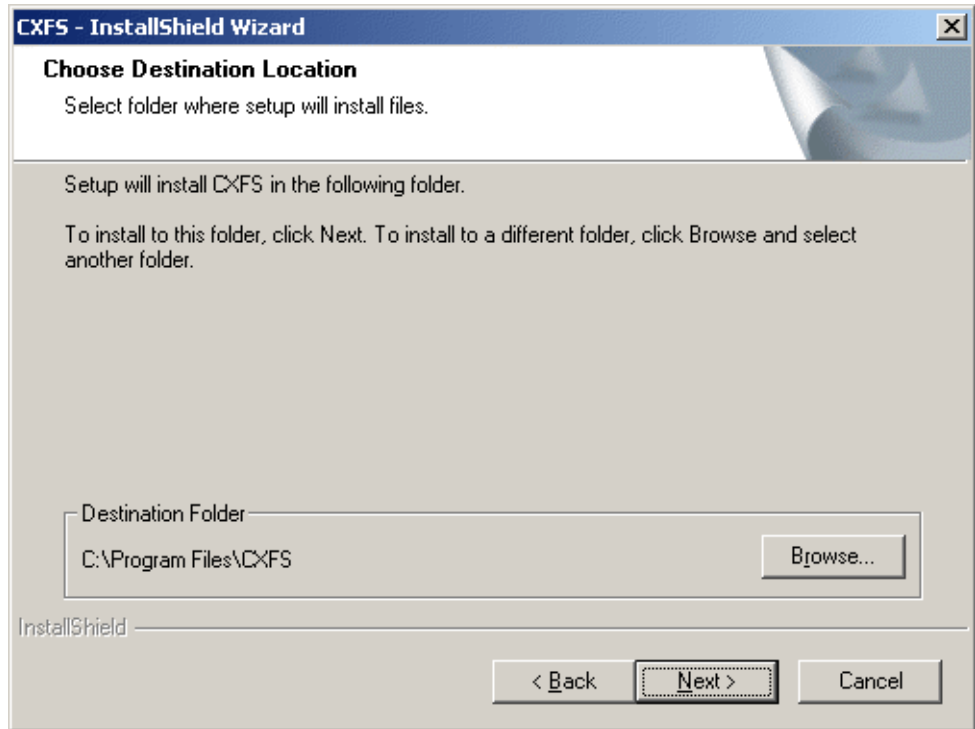


Figure 5-6 Choose Destination Location

9. Enter details for the following fields as shown in Figure 5-7 and click **Next** when finished:
 - **Drive letter for CXFS:** specify the drive letter under which all CXFS filesystems will be mounted. You cannot select a drive letter that is currently in use.
 - **Default Umask:** choose the default umask. For more information on the umask, see "Inheritance and Default ACLs for Windows" on page 121.
 - **User ID mapping lookup sequence:** choose the appropriate primary and (optionally) secondary method. See "User Identification Mapping Methods for Windows" on page 112.
 - **Location of fencing, UNIX /etc/passwd and /etc/group files:** specify the path where the configuration files will be installed and accessed by the CXFS

software if required. The default is the same location as the software under %ProgramFiles%\CXFS.

- **IP address of the heartbeat network adapter:** specify the IP address of the private network adapter on the Windows node.
- **Additional arguments:** contains parameters that are used by the CXFS Client service when it starts up. For most configurations, this should be left alone. To get a list of options, type `cxfs_client -h` in a command shell (cmd) window. .

CXFS - InstallShield Wizard

Enter CXFS Details

Select drive letter for CXFS volume access and also enter the locations for the UNIX passwd and group file for volume access.

Drive letter for CXFS: X: Default Umask: Everyone can write

User ID mapping lookup sequence: files (none)

Location of fencing, UNIX /etc/passwd and /etc/group files:
C:\Program Files\CXFS\ Browse...

IP address of the heartbeat network adapter: 10.0.0.2

Additional arguments: -I"C:\Program Files\CXFS\log\cxfs_client.log" -z 10000000

InstallShield

< Back Next > Cancel

Figure 5-7 Enter CXFS Details

10. If you select **ldap_activedir** as the user ID mapping method, the dialog in Figure 5-8 is displayed after you click **Next**.

CIFS for Windows Setup

Enter LDAP Details
Enter details for creating Windows/UNIX user ID mappings from an LDAP server.

Server Details: Host name: Port:

Bind details: Simple Auth. User name: Password:

Base DN to search from:

Search Settings: Services for UNIX defaults:

User filter: Group filter:

Attributes: User Name: Windows SID: Unix UID: Unix GID: Grp Members:

InstallShield

< Back

Figure 5-8 Active Directory Details

If you have a standard Active Directory configuration with Windows Services for UNIX (SFU), you need only to select the version of SFU and **Auth** (authenticated) for **Bind details**; doing so will then define the correct Active Directory defaults. The other server details can normally remain blank.

11. If you select **ldap_generic** as the user ID mapping method, the dialog in Figure 5-9 is displayed after you click **Next**. You must provide entries for the **Host name** and the **Base DN to search from** fields. For a standard OpenLDAP server, you can select a simple anonymous bind (default settings with the **User name** and **Password** fields left blank) and select the standard search settings by clicking **Posix**.

The screenshot shows a dialog box titled "CXFS for Windows Setup" with a sub-header "Enter LDAP Details". Below the sub-header is a brief instruction: "Enter details for creating Windows/UNIX user ID mappings from an LDAP server." The dialog is divided into several sections:

- Server Details:** Includes a "Host name:" text box and a "Port:" dropdown menu set to "389".
- Bind details:** Features two radio buttons, "Simple" (selected) and "Auth.". It also includes "User name:" and "Password:" text boxes.
- Base DN to search from:** A single-line text box.
- Search Settings:** Contains a "Generic LDAP defaults:" dropdown menu set to "Posix".
- User filter:** A text box containing "(OBJECTCLASS=POSIXACCO".
- Group filter:** A text box containing "(OBJECTCLASS=POSIXGROU".
- Attributes:** A table with five columns: "User Name:", "Unix UID:", "Group Name:", "Unix GID:", and "Grp Members:". Each column has a corresponding text box below it containing "UID", "UIDNUMBER", "CN", "GIDNUMBER", and "MEMBERUID" respectively.

At the bottom left, the text "InstallShield" is visible. At the bottom right, there are three buttons: "< Back", "Next >" (highlighted with a dashed border), and "Cancel".

Figure 5-9 Generic LDAP Details

12. Review the settings, as shown in Figure 5-10. If they appear as you intended, click **Next**. If you need to make corrections, click **Back**.

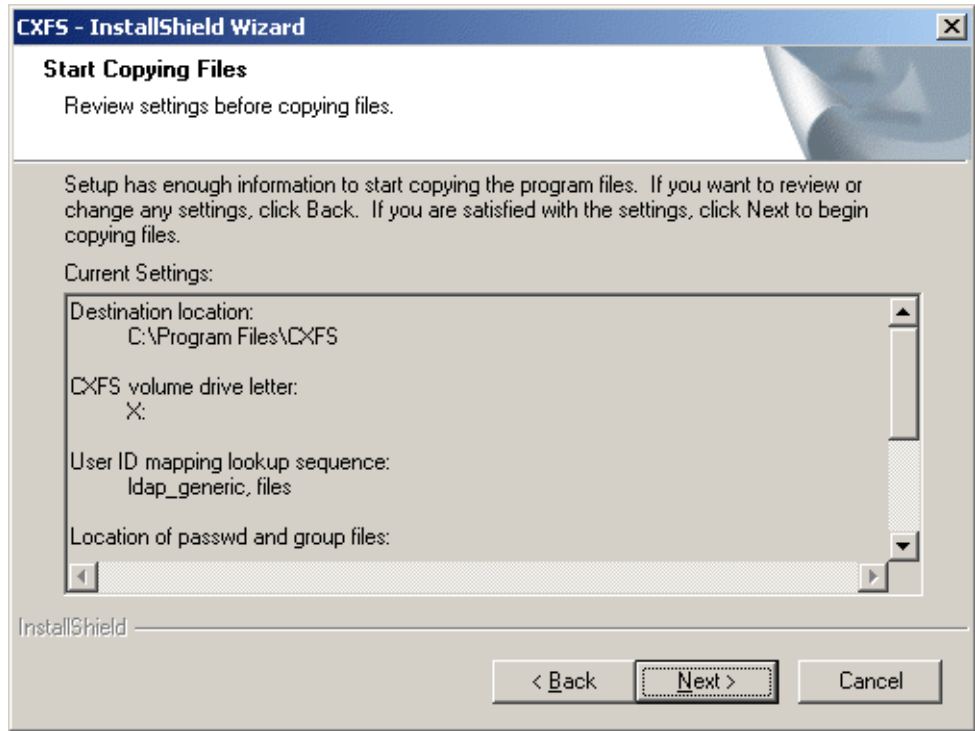


Figure 5-10 Review the Settings

After you click **Next**, the CXFS software will be installed.

13. You will be given the option to start the driver at system start-up, as shown in Figure 5-11. By checking the boxes, you will start the driver automatically when the system starts up and invoke the **CXFS Info** window minimized to an icon. If you choose not to start the CXFS driver automatically, you must start it manually through the **Services** control panel before you can access CXFS filesystems. To manually start the **CXFS Info** window, select the following:

Start
 > Programs
 > CXFS
 > CXFS Info

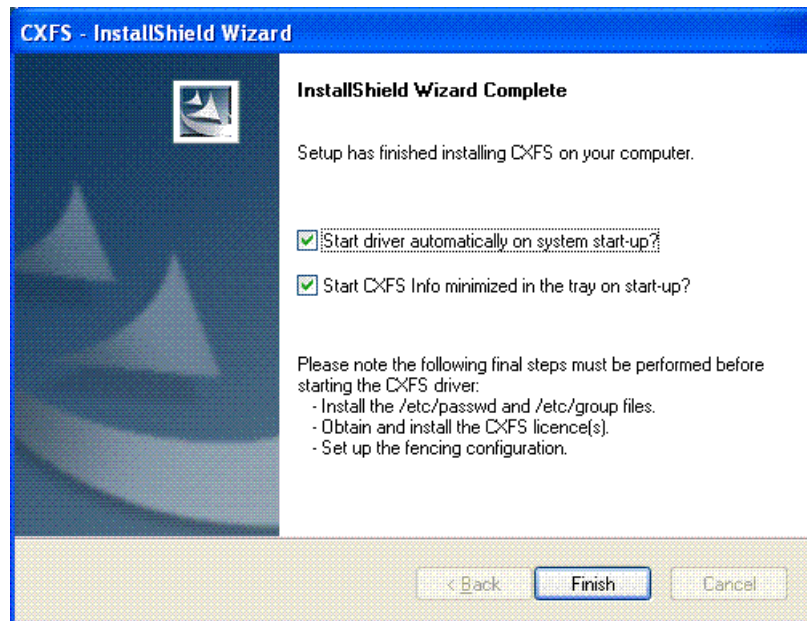


Figure 5-11 Start CXFS Driver

14. Choose to restart your computer later if you need to install `passwd` and `group` files or set up fencing; see "Postinstallation Steps for Windows" on page 140. Otherwise, choose to restart your computer now. The default is to restart later, as shown in Figure 5-12. (CXFS will not run until a restart has occurred.)

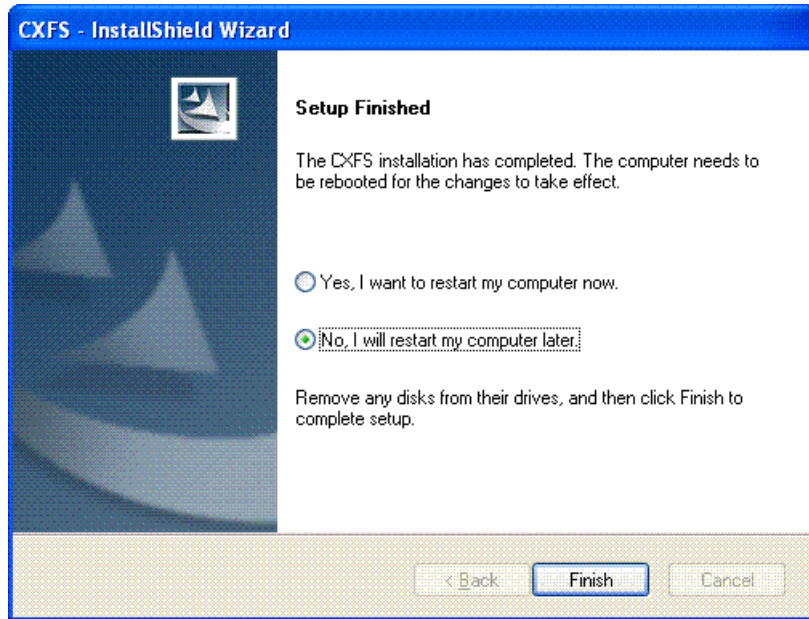


Figure 5-12 Restart the System

Postinstallation Steps for Windows

This section discusses the configuration steps that you should perform after installing CXFS software but before rebooting a Windows node.

The following postinstallation steps are required to ensure the correct operation of the CXFS software:

- "Checking Permissions on the Password and Group Files for Windows" on page 141
- "Performing User Configuration for Windows" on page 141

Checking Permissions on the Password and Group Files for Windows

The permissions on the `passwd` and `group` files must restrict access so that only the system administrator can modify these files. This can be done by right-clicking on the filenames in Windows Explorer and selecting the following:

Properties
 > **Security**

Verify that the permissions are Read for Everyone and Full Control for Administrators.



Caution: Failure to set permissions on the `passwd` and `group` files would allow users to change their UID/GID at will and even gain superuser access to the files on the CXFS filesystem.

Performing User Configuration for Windows

If the user mapping is not correctly configured, all filesystem operations will be as user `nobody`.

If you selected the **passwd and group files** user ID mapping method, you must install the `passwd` and `group` files. The default `passwd` and `group` files that are installed are invalid files containing comments; these invalid files will cause the CXFS Client service to generate warnings in its log file and users may not be correctly configured. You must remove the comments in these files when you install the `passwd` and `group` files.

After installing the CXFS software onto the Windows node but before rebooting it, you must install the `/etc/passwd` and `/etc/group` files from the metadata server to the location on the Windows node specified during installation.

The defaults are as follows:

- `/etc/passwd` as `%ProgramFiles%\CXFS\passwd`
- `/etc/group` as `%ProgramFiles%\CXFS\group`

Do the following:

1. Verify that permissions are set as described in "Checking Permissions on the Password and Group Files for Windows" on page 141.

2. If you selected the **Active Directory** method, you must specify the UNIX identifiers for all users of the CXFS node. On the domain controller, run the following to specify the UNIX UID and GID of a given user:

Start

- > **Program Files**
- > **Administrative Tools**
- > **Active Directory Users and Computers**
- > **Users**

3. Select a user and then select:

Properties

- > **UNIX Attributes**

The CXFS software will check for changes to the LDAP database every 5 minutes.

4. After the CXFS software has started, you can use **CXFS Info** to confirm the user configuration, regardless of the user ID mapping method chosen. See "User Identification for Windows" on page 111.

If only the Administrator user is mapped, see "CXFS Client Service Cannot Map Users other than Administrator for Windows" on page 165.

I/O Fencing for Windows

Note: For all 64-bit platforms on Windows and for 32-bit Windows Vista, you must manually configure the `fencing.conf` file due to the absence of 64-bit SNIA runtime libraries.

I/O fencing is required on Windows nodes in order to protect data integrity of the filesystems in the cluster. The CXFS client software automatically detects the worldwide port names (WWPNs) of any supported host bus adapters (HBAs) for Windows nodes that are connected to a switch that is configured in the cluster database. These HBAs are available for fencing.

However, if no WWPNs are detected, there will be messages about loading the HBA/SNIA library logged to the `%ProgramFiles%\CXFS\log\cxfs_client.log` file.

If no WWPNs are detected, you must manually specify the WWPNs in the fencing file.

Note: This method does not work if the WWPNs are partially discovered.

The `%ProgramFiles%\CXFS\fencing.conf` file enumerates the WWPN for all of the HBAs that will be used to mount a CXFS filesystem. There must be a line for the HBA WWPN as a 64-bit hexadecimal number.

Note: The WWPN is that of the HBA itself, **not** any of the devices that are visible to that HBA in the fabric.

If used, `%ProgramFiles%\CXFS\fencing.conf` must contain a simple list of WWPNs, one per line. You must update it whenever the HBA configuration changes, including the replacement of an HBA.

This section discusses the following:

- "Determining the WWPN for a QLogic Switch" on page 143
- "Determining the WWPN for a Brocade Switch" on page 145

Determining the WWPN for a QLogic Switch

Do the following to determine the WWPN for a QLogic switch:

1. Set up the switch and HBA. See the release notes for supported hardware.
2. Use the `telnet` command to connect to the switch and log in as user `admin`. (The password is `password` by default).
3. Enter the `show topology` command to retrieve the WWPN numbers.

For example:

SANbox #> **show topology**

Unique ID Key

A = ALPA, D = Domain ID, P = Port ID

Port Number	Loc Type	Local PortWWN	Rem Type	Remote NodeWWN	Unique ID	
-----	-----	-----	-----	-----	-----	
0	F	20:00:00:c0:dd:06:ff:7f	N	20:00:00:01:ff:03:05:b2	020000	P
2	F	20:02:00:c0:dd:06:ff:7f	N	20:01:00:e0:8b:32:ba:14	020200	P
4	F	20:04:00:c0:dd:06:ff:7f	N	20:00:00:01:ff:03:05:b2	020400	P
5	F	20:05:00:c0:dd:06:ff:7f	N	20:00:00:e0:8b:0b:81:24	020500	P
6	F	20:06:00:c0:dd:06:ff:7f	N	20:01:00:e0:8b:32:06:c8	020600	P
8	F	20:08:00:c0:dd:06:ff:7f	N	20:00:00:01:ff:03:05:b2	020800	P
12	F	20:0c:00:c0:dd:06:ff:7f	N	20:00:00:01:ff:03:05:b2	020c00	P
15	F	20:0f:00:c0:dd:06:ff:7f	N	20:00:00:e0:8b:10:04:13	020f00	P
17	E	20:11:00:c0:dd:06:ff:7f	E	10:00:00:c0:dd:06:fb:04	1(0x1)	D
19	E	20:13:00:c0:dd:06:ff:7f	E	10:00:00:c0:dd:06:fb:04	1(0x1)	D

The WWPN is the hexadecimal string in the Remote NodeWWN column are the numbers that you copy for the fencing.conf file. For example, the WWPN for port 0 is 20000001ff0305b2 (you must remove the colons from the WWPN reported in the show topology output in order to produce the string to be used in the fencing file).

4. Edit or create %ProgramFiles%\CXFS\fencing.conf and add the WWPN for the port. (Comment lines begin with #.)

For dual-ported HBAs, you must include the WWPNs of any ports that are used to access cluster disks. This may result in multiple WWPNs per HBA in the file; the numbers will probably differ by a single digit.

For example, if you determined that port 0 is the port connected to the switch, your fencing file should contain the following:

```
# WWPN of the HBA installed on this system
#
2000000173002c0b
```

5. To enable fencing, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Determining the WWPN for a Brocade Switch

Do the following to determine the WWPN for a Brocade switch:

1. Set up the switch and HBA. See the release notes for supported hardware.
2. Use the `telnet` command to connect to the switch and log in as user `admin`. (The password is `password` by default).
3. Execute the `switchshow` command to display the switches and their WWPN numbers.

For example:

```
brocade04:admin> switchshow
switchName:      brocade04
switchType:      2.4
switchState:     Online
switchRole:      Principal
switchDomain:    6
switchId:        fffc06
switchWwn:       10:00:00:60:69:12:11:9e
switchBeacon:    OFF
port 0: sw Online      F-Port 20:00:00:01:73:00:2c:0b
port 1: cu Online      F-Port 21:00:00:e0:8b:02:36:49
port 2: cu Online      F-Port 21:00:00:e0:8b:02:12:49
port 3: sw Online      F-Port 20:00:00:01:73:00:2d:3e
port 4: cu Online      F-Port 21:00:00:e0:8b:02:18:96
port 5: cu Online      F-Port 21:00:00:e0:8b:00:90:8e
port 6: sw Online      F-Port 20:00:00:01:73:00:3b:5f
port 7: sw Online      F-Port 20:00:00:01:73:00:33:76
port 8: sw Online      F-Port 21:00:00:e0:8b:01:d2:57
port 9: sw Online      F-Port 21:00:00:e0:8b:01:0c:57
port 10: sw Online     F-Port 20:08:00:a0:b8:0c:13:c9
port 11: sw Online     F-Port 20:0a:00:a0:b8:0c:04:5a
port 12: sw Online     F-Port 20:0c:00:a0:b8:0c:24:76
port 13: sw Online     L-Port 1 public
port 14: sw No_Light
port 15: cu Online     F-Port 21:00:00:e0:8b:00:42:d8
```

The WWPN is the hexadecimal string to the right of the port number. For example, the WWPN for port 0 is 2000000173002c0b (you must remove the colons from the WWPN reported in the `switchshow` output in order to produce the string to be used in the fencing file).

4. Edit or create `%ProgramFiles%\CXFS\fencing.conf` and add the WWPN for the port. (Comment lines begin with #.)

For dual-ported HBAs, you must include the WWPNs of any ports that are used to access cluster disks. This may result in multiple WWPNs per HBA in the file; the numbers will probably differ by a single digit.

For example, if you determined that port 0 is the port connected to the switch, your fencing file should contain the following:

```
# WWPN of the HBA installed on this system
#
2000000173002c0b
```

5. To enable fencing, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Note: You may be able to use an HBA vendor-provided utility to determine the WWPN.

Start/Stop the CXFS Client Service for Windows

The CXFS Client service is automatically started when a Windows node is rebooted. This behavior may be altered by changing the configuration of the CXFS filesystem driver and the CXFS Client service.

By default, the driver is configured to start manually and the Client service is configured to start automatically. Because the CXFS Client service depends on the CXFS filesystem driver, the driver will be started by the service.

SGI recommends that the CXFS driver configuration remains manual.

You can change the CXFS Client service configuration to start manually, meaning that CXFS does not automatically start, by selecting the following:

Start
 > Control Panel
 > Administrative Tools
 > Services

Change **CXFS Client** to manual rather than automatic. CXFS can then be started and stopped manually by the Administrator using the same selection sequence.

Maintenance for Windows

This section contains the following:

- "Modifying the CXFS Software for Windows" on page 147
- "Updating the CXFS Software for Windows" on page 149
- "Removing the CXFS Software for Windows" on page 151
- "Downgrading the CXFS Software for Windows" on page 151

Modifying the CXFS Software for Windows

To change the location of the software and other configuration settings that were requested in "Client Software Installation for Windows" on page 132, perform the following steps:

1. Select the following:
 - Windows XP and Windows 2003:

Start
 > Control Panel
 > Add or Remove Programs
 > CXFS
 > Add/Remove
 > Modify

- Windows Vista, Windows Server 2008, and Windows 7:

Start
 > Control Panel
 > Programs and Features
 > CXFS
 > Modify

Figure 5-13 shows the screen that lets you modify the software.

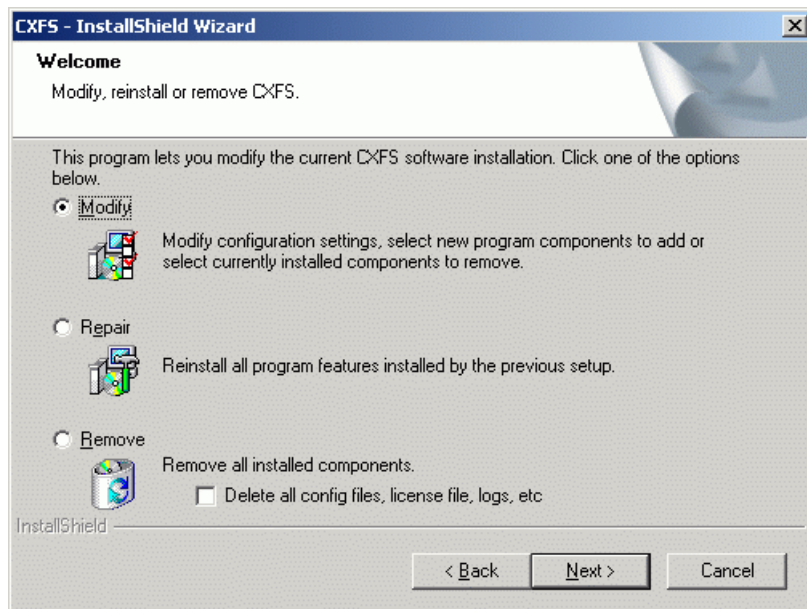


Figure 5-13 Modify CXFS for Windows

2. Make the necessary configuration changes.

You can display the list of possible command line arguments supported by the CXFS Client service by running the service from a command line as follows:

```
C:\> %SystemRoot%\system32\cxfs_client.exe -h
```

3. Reboot the system to apply the changes.

Updating the CXFS Software for Windows

To upgrade the CXFS for Windows software, perform the following steps:

1. Obtain the CXFS update software according to the directions in the *CXFS 6 Administration Guide for SGI InfiniteStorage* and the *SGI InfiniteStorage Software Platform* release notes.
2. Transfer the client software (which was downloaded onto a server-capable administration node during its installation procedure) using `ftp`, `rsh`, or `scp`. The location of the Windows installation program will be as follows:

```
/usr/cluster/client-dist/CXFS_VERSION/windows/all/noarch/setup.exe
```

3. Double-click the **setup.exe** installation program to execute it.
4. A welcome screen will appear that displays the version you are upgrading from and the version you are upgrading to. Figure 5-14 shows an example of the screen that appears when you are upgrading the software (the actual versions displayed by your system will vary based upon the release that is currently installed and the release that will be installed. All the configuration options are available to update as discussed in "Client Software Installation for Windows" on page 132.

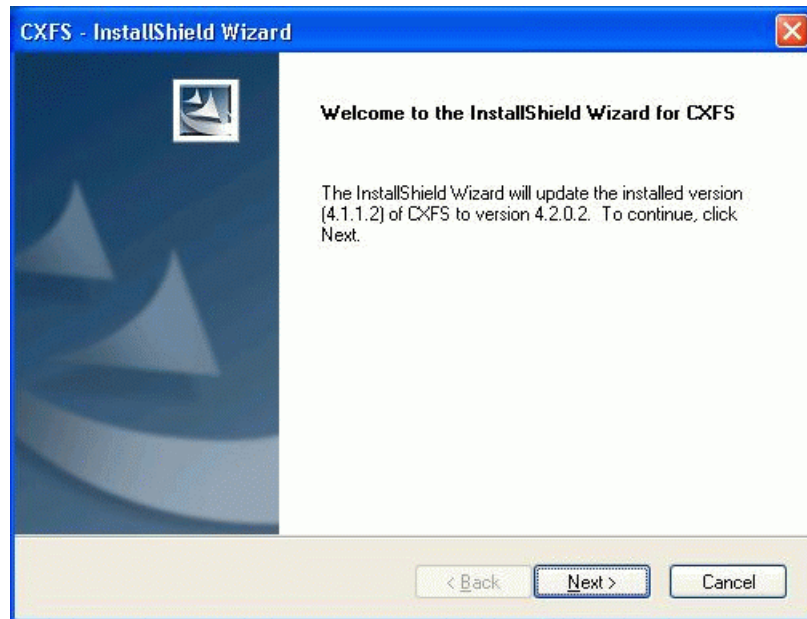


Figure 5-14 Upgrading the Windows Software

5. Reboot the system to apply the changes.

Removing the CXFS Software for Windows

To remove the CXFS for Windows software, do the following:

1. Ensure that no applications on this node are accessing files on a CXFS filesystem.
2. Select the following sequence to remove all installed files and registry entries:

- Windows XP and Windows 2003:

Start

> **Control Panel**
> **Add or Remove Programs**
> **CXFS**
> **Add/Remove**
> **Remove**

- Windows Vista, Windows Server 2008, and Windows 7:

Start

> **Control Panel**
> **Programs and Features**
> **CXFS**
> **Remove**

Figure 5-13 on page 148 shows the screen that lets you remove the software.

Note: By default, the `passwd`, `group`, and `log` files will not be removed. To remove these other files, check the following box:

Delete all config files, license file, logs, etc

Then click **Next**.

3. Reboot the system to apply the changes.

Downgrading the CXFS Software for Windows

To downgrade the CXFS software, do the following:

1. Back up the configuration file.

Note: The removal process may remove the configuration file. You should back up the configuration file before removing the CXFS software so that you can easily restore it after installing the downgrade.

2. Follow the instructions to remove the software in "Removing the CXFS Software for Windows" on page 151.
3. Install the older version of the software as directed in "Client Software Installation for Windows" on page 132.

GRIO on Windows

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the Windows platform if GRIO is enabled on the server-capable administration node.

Figure 5-15 shows an example of the **CXFS Info** display for GRIO.

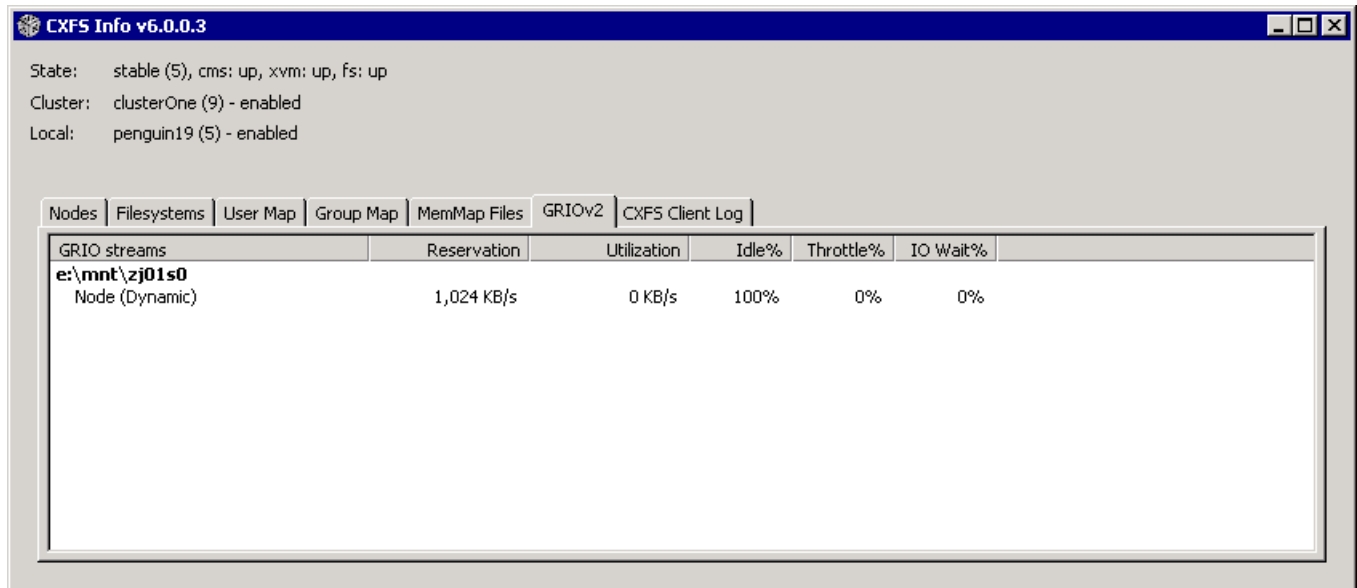


Figure 5-15 CXFS Info Display for GRIO for Windows

A Windows node can mount a GRIO-managed filesystem and supports application- and node-level reservations. A Windows node will interoperate with the dynamic bandwidth allocator for all I/O outside of any reservation.

For more information, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7 and the *Guaranteed-Rate I/O Version 2 for Linux Guide*.

XVM Failover V2 on Windows

This section discusses the following:

- "Configuring the `failover2.conf` File for Windows" on page 153
- "Windows XP SP2 and Windows Server 2003 R2 SP1 `failover2` Example " on page 156
- "Windows Server 2003 R2 SP2, Windows Vista, Windows Server 2008, and Windows 7 `failover2` Example" on page 157

Configuring the `failover2.conf` File for Windows

Note: You must not install RDAC pseudo/virtual LUNs onto the Windows client.

To configure the `failover2.conf` file for a Windows node, do the following:

1. Run the HBA utility (SanSurfer for QLogic, LSIUtil for LSI HBA) and set the persistent binding to bind the target (node and port's WWN) to the target ID. For more information, see "Mapping XVM Volumes to Storage Targets on Windows" on page 158.

Note: For the `failover2.conf` file to work properly, persistent bindings must be enabled in the HBA driver.

When you bind a persistent target ID to a specific LUN, you can find the WWN of the corresponding port and node (controller) on the storage array. As a result, a target ID corresponds to a controller and a port on the controller. You must make sure that the `failover2.conf` setting is consistent across the cluster.

In the persistent binding, there are normally the following fields:

- Type
- Target's node WWN (the controller's WWN)
- Target's port WWN (the port on the controller)
- A configurable target ID

Note the controller and port to which the target ID corresponds.

2. Reboot the Windows node.
3. Run the following command:

```
C:\> xvm show -v phys | find "affinity" > failover2.conf
```

4. Modify the failover2.conf file so that it has affinity values that are consistent across the cluster. The affinity value for the target ID corresponding to controller A should be different from the affinity value for the target ID corresponding to controller B.
5. Copy the failover2.conf file to the CXFS folder.
6. Set the preferred path for each target depending on the storage array's setting.
7. Run xvm commands to read in the new configuration and change to the preferred path:

```
xvm foconfig -init  
xvm foswitch -preferred phys
```

For example, assume there are two controllers in a storage array. Controller A has a WWN of 200400a0b82925e2; it has two ports connecting to the host or the fabric. Port 1 has a WWN of 201400A0B82925E2, port 2 has a WWN of 202400A0B82925E2. Controller B has a WWN of 200500a0b82925e2; it also has two ports with WWNs of 201500A0B82925E2 and 202500A0B82925E2, respectively. There are therefore four paths to LUN 0.

The metadata server in this cluster would have entries like the following in its failover2.conf file (where information within angle brackets is an embedded comment):

```
/dev/xscsi/pci08.03.1/node200500a0b82925e2/port2/lun0/disc affinity=2  
/dev/xscsi/pci08.03.1/node200500a0b82925e2/port1/lun0/disc affinity=2  
/dev/xscsi/pci08.03.1/node200400a0b82925e2/port2/lun0/disc affinity=1  
/dev/xscsi/pci08.03.1/node200400a0b82925e2/port1/lun0/disc affinity=1 preferred <current path>
```

In this configuration, controller A (node200400a0b82925e2) has an affinity of 1, controller B has an affinity of 2. Controller A's port 1 is the preferred path.

To create the corresponding `failover2.conf` file on the Windows node, you must first define the persistent-binding targets. Use SANSurfer (for Qlogic HBA) or LSIUtil (for LSI HBA) to define four possible targets:

Binding type	World Wide Node Name	World Wide port Name	Target ID
WWN	200500a0b82925e2	202500A0B82925E2	0
WWN	200500a0b82925e2	201500A0B82925E2	1
WWN	200400a0b82925e2	202400A0B82925E2	2
WWN	200400a0b82925e2	201400A0B82925E2	3

As a result, target 0 corresponds to the first path on the metadata server. Targets 1, 2, and 3 correspond to the 2nd, 3rd, and 4th path, respectively. To be consistent, target 2 or 3 (on controller A) should be the preferred path on Windows.

Then you would run the following command:

```
C:\> xvm show -v phys | find "affinity" > failover2.conf
```

Assuming that there are two HBA ports on the Windows node, you would end up with eight paths for the two HBA ports. The `failover2.conf` file would contain something like the examples shown in the following sections (the format varies by the Windows OS version):

- "Windows XP SP2 and Windows Server 2003 R2 SP1 failover2 Example " on page 156
- "Windows Server 2003 R2 SP2, Windows Vista, Windows Server 2008, and Windows 7 failover2 Example" on page 157

For more information, see:

- "XVM Failover and CXFS" on page 8
- The comments in the `failover2.conf` file
- *CXFS 6 Administration Guide for SGI InfiniteStorage*
- *XVM Volume Manager Administrator's Guide*

Windows XP SP2 and Windows Server 2003 R2 SP1 failover2 Example

Windows XP SP 2 and Windows Server 2003 R2 SP1 failover2.conf example:

```
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&030 <dev 321> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&020 <dev 301> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&010 <dev 281> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000 <dev 261> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&030 <dev 236> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&020 <dev 216> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&010 <dev 196> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000 <dev 176> affinity=0
#
# Where
# SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&030 <dev 321> affinity=0
#           ^^^^^^^^   ^^^
#           |           |||-- Lun = 0
#           |           ||--- Target = 1 (1-2 hex digits)
#           |           |---- Bus ID = 0
#           |----- Host HBA port ID = 67032E4
```

You would set the proper affinity values and add the preferred tag to target 2 or 3:

```
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&030 <dev 321> affinity=1 preferred
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&020 <dev 301> affinity=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&010 <dev 281> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000 <dev 261> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&030 <dev 236> affinity=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&020 <dev 216> affinity=1 preferred
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&010 <dev 196> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000 <dev 176> affinity=2
```

In this setting, the access to LUN 0 from one HBA (with its ID of 67032E4) goes to controller A, port 1. From another HBA (with ID of 1F095A8E), it goes to controller A, port 2. Controller A (to which targets 2 and 3 belong) has an affinity of 1; controller B has an affinity of 2.

Windows Server 2003 R2 SP2, Windows Vista, Windows Server 2008, and Windows 7 failover2 Example

Windows Server 2003 R2 SP2, Windows Vista, and Windows Server 2008 failover2 example:

```

SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000300 <dev 321> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000200 <dev 301> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000100 <dev 281> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000000 <dev 261> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000300 <dev 236> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000200 <dev 216> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000100 <dev 196> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000000 <dev 176> affinity=0
#
# Where
#   SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000300 <dev 321> affinity=0
#                                     ^^^^^^^^  ^^^^^^
#                                     |          || |- Lun = 0   (2 hex digits)
#                                     |          ||--- Target = 3 (2 hex digits)
#                                     |          |---- Bus ID = 0
#                                     |----- Host HBA port ID = 67032E4

```

You would set the proper affinity values and add the preferred tag to target 2 or 3:

```

SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000300 <dev 321> affinity=1 preferred
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000200 <dev 301> affinity=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000100 <dev 281> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000000 <dev 261> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000300 <dev 236> affinity=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000200 <dev 216> affinity=1 preferred
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000100 <dev 196> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000000 <dev 176> affinity=2

```

In this setting, the access to LUN 0 from one HBA (with its ID of 67032E4) goes to controller A, port 1. From another HBA (with ID of 1F095A8E), it goes to controller A, port 2. Controller A (to which targets 2 and 3 belong) has an affinity of 1; controller B has an affinity of 2.

Mapping XVM Volumes to Storage Targets on Windows

You must configure the host bus adapter (HBA) on each node to use persistent bindings for all ports used for CXFS filesystems. The method for configuration varies depending on your HBA vendor. For more information, see the following:

- Information about binding target devices is in the QLogic SANsurfer help. You must select a port number and then select **Bind** and the appropriate **Target ID** for each disk. For example, see Figure 5-16.
- Information about persistent bindings is in the LSI Logic MPT Configuration Utility (`LSIUtil.exe`). `LSIUtil` is a command line tool. It has a submenu for displaying and changing persistent mapping. Do the following:
 1. Choose the HBA port.
 2. Select **e** to enable expert mode.
 3. Select **15** to manipulate persistent binding.
 4. Choose one of the following:
 - **2** to automatically add persistent mappings for all targets
 - **3** to automatically add persistent mappings for some targets
 - **6** to manually add persistent mappings

Note: You should disable any failover functionality provided by the HBA.

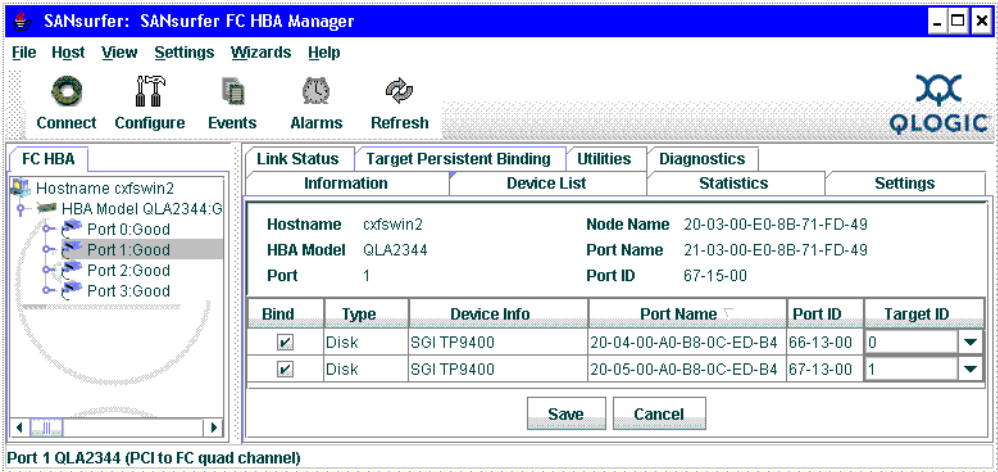


Figure 5-16 QLogic SANsurfer (Copyright QLogic® Corporation, all rights reserved)

Troubleshooting for Windows

This section discusses the following:

- "Verification that the CXFS Software is Running Correctly for Windows" on page 161
- "Inability to Mount Filesystems on Windows" on page 161
- "Access-Denied Error when Accessing Filesystem on Windows" on page 163
- "Application Works with NTFS but not CXFS for Windows" on page 164
- "Delayed-Write Error Dialog is Generated by the Windows Kernel" on page 164
- "CXFS Client Service Does Not Start on Windows" on page 165
- "CXFS Client Service Cannot Map Users other than Administrator for Windows" on page 165
- "Filesystems Are Not Displayed on Windows" on page 166
- "Large Log Files on Windows" on page 167
- "Windows Failure on Reboot" on page 167
- "NO_MORE_SYSTEM_PTES Error Message" on page 168
- "Application Cannot Create File Under CXFS Drive Letter" on page 168
- "Installation File Not Found Errors" on page 169
- "Problems Specific to Windows Vista, Windows Server 2008, and Windows 7" on page 169

Also see:

- The Windows `cxfsdump` documentation located at `%ProgramFiles%\CXFS\cxfsdump.html`
- Chapter 7, "General Troubleshooting" on page 191

Verification that the CXFS Software is Running Correctly for Windows

To verify that the CXFS Client service has started, select the following:

Start
 > **Control Panel**
 > **Administrative Tools**
 > **Services**

Inability to Mount Filesystems on Windows

If **CXFS Info** reports that `cms` is up but `XVM` or the filesystem is in another state, then one or more mounts is still in the process of mounting or has failed to mount.

The CXFS node might not mount filesystems for many reasons, including the following:

- The metadata server is unable to mount the filesystem. In this case, no clients will be able to mount the filesystem.
- The metadata server is processing a recovery or relocation that is not progressing. In this case, clients cannot mount the filesystem until that state is cleared. Clearing the state may require you to take action on one or more of the other nodes in the cluster.
- The node may not be able to see all the LUNs. This is usually caused by misconfiguration of the HBA or the SAN fabric:
 - Check that the ports on the Fibre Channel switch connected to the HBA are active. Physically look at the switch to confirm the light next to the port is green, or remotely check by using the `switchShow` command.
 - Check that the HBA configuration is correct.
 - Check that the HBA can see all the LUNs for the filesystems it is mounting.
 - Check that the operating system kernel can see all the LUN devices. For example:

Start

- > **Control Panel**
- > **Administrative Tools**
- > **ComputerManagement**
- > **Device Manager**
- > **View**
- > **Devices by connection**

- Use `debugview` to monitor the CXFS driver when it probes the disk devices. You should see it successfully probe each of the LUN devices.

Note: For Windows Vista, Windows Server 2008, and Windows 7: By default, Debug messages are turned off. To view Debug messages using `debugview`, you must enable them in the registry. Do the following:

1. Using `regedit`, navigate to the following value:

- HKEY_LOCAL_MACHINE**
- > **SYSTEM**
- > **CurrentControlSet**
- > **Control**
- > **Session Manager**
- > **Debug Print Filter**

2. Add the following value:

```
"DEFAULT" REG_DWORD : 0xF
```

3. Reboot the system to apply the changes.

-
- If the RAID device has more than one LUN mapped to different controllers, ensure the node has a Fibre Channel path to all relevant controllers.
 - The CXFS Client service may not be running. To verify that it is running, open the **Task Manager** by pressing the `Ctrl+Shift+Esc`, or right-mouse click an empty area of the taskbar and select **Task Manager** from the popup menu. In the **Processes** tab, search for `cxfs_client.exe` in the **Image Name** column. You can sort the processes by name by clicking the heading of the column.
 - The filesystem may have an unsupported mount option. Check the `cxfs_client.log` for mount option errors or any other errors that are reported when attempting to mount the filesystem.

- The cluster membership (cms), XVM, or the filesystems may not be up on the node. Use **CXFS Info** to determine the current state of cms, XVM, and the filesystems. Do the following:
 - If cms is not up, check the following:
 - Is the node is configured on the server-capable administration node with the correct hostname or IP address?
 - Has the node been added to the cluster and enabled? See "Verifying the Cluster Status" on page 183.
 - If XVM is not up, check that the HBA is active and can see the LUNs.
 - If the filesystem is not up, check that one or more filesystems are configured to be mounted on this node and check the **CXFS Client Log** tab in **CXFS Info** for mount errors. They will be highlighted.
- The LUN may be too large. Windows XP does not support LUNs greater than 2 TB in size. Filesystem corruption will occur if you attempt to write to the LUN above the 2-TB boundary. On a Windows XP node, CXFS will not allow a filesystem to be mounted if any part of it resides on a LUN that is greater than 2-TB in size.

Also, check the **CXFS Client Log** tab in **CXFS Info** for mount errors.

Access-Denied Error when Accessing Filesystem on Windows

If an application reports an access-denied error, do the following:

- Check the list of users and groups that **CXFS Info** has mapped to a UNIX UID and GID. If the current user is not listed as one of those users, check that the user mapping method that was selected is configured correctly, that there is an LDAP server running (if you are using LDAP), and that the user is correctly configured.
- Increase the verbosity of output from the CXFS Client service so that it shows each user as it is parsed and mapped.
- Use Process Monitor to monitor the application and verify that there is no file that has been created below a mount point under the CXFS drive letter. An error may be caused by attempting to create a file below the drive letter but above the mount point. For more information, see:

<http://technet.microsoft.com/en-us/sysinternals/bb896642.aspx>

Application Works with NTFS but not CXFS for Windows

The Windows filesystem APIs are far more extensive than the UNIX POSIX APIs and there are some limitations in mapping the native APIs to POSIX APIs (see "Functional Limitations and Considerations for Windows" on page 101). Sometimes these limitations may affect applications, other times the applications that have only ever been tested on NTFS make assumptions about the underlying filesystem without querying the filesystem first.

If an application does not behave as expected, and retrying the same actions on an NTFS filesystem causes it to behave as was expected, then third-party tools like Process Monitor can be used to capture a log of the application when using both NTFS and CXFS. Look for differences in the output and try to determine the action and/or result that is different. Using the same filenames in both places will make this easier. For more information about Process Monitor, see:

<http://technet.microsoft.com/en-us/sysinternals/bb896642.aspx>

Note: There are some problems that may not be visible in a Process Monitor log. For example, some older applications use only a 32-bit number when computing filesystem or file size. Such applications may report out of disk space errors when trying to save a file to a large (greater than 1 TB) filesystem.

Delayed-Write Error Dialog is Generated by the Windows Kernel

A delayed-write error is generated by the Windows kernel when it attempts to write file data that is in the cache and has been written to disk, but the I/O failed. The write call made by the application that wrote the data may have completed successfully some time ago (the application may have even exited by now), so there is no way for the Windows kernel to notify the application that the I/O failed.

This error can occur on a CXFS filesystem if CXFS has lost access to the disk due to the following:

- Loss of membership resulting in the Windows node being fenced and the filesystem being unmounted. Check that the Windows node is still in membership and that there are no unmount messages in the `cxfs_client.log` file.
- Loss of Fibre Channel connection to the Fibre Channel switch or RAID. Check the Fibre Channel connections and use the SanManager tool to verify that the HBA can still see all of the LUNs. Make sure the filesystems are still mounted.

- The metadata server returned an I/O error. Check the system log on the metadata server for any I/O errors on the filesystem and take corrective action on the server if required.

CXFS Client Service Does Not Start on Windows

The following error may be seen when the CXFS Client service attempts to start:

```
Error 10038: An operation was attempted on something that is not a socket.
```

Check the **CXFS Client Log** in **CXFS Info** for information on why the CXFS node failed to start.

CXFS Client Service Cannot Map Users other than Administrator for Windows

If the CXFS Client service cannot map any users other than Administrator and there are no LDAP errors in the `cxfs_client` log file (and you are using LDAP), you must change the configuration to allow reading of the attributes.

Do the following:

1. Select the following:

Start

> **Control Panel**

> **Administrative Tools**

> **Active Directory Users and Computers**

2. Select the following:

View

> **Advanced Features**

3. Right-mouse click the **Users** folder under the domain controller you are using and select the following:

Properties

> **Security**

> **Advanced**

> **Add**

4. Select **Authenticated Users** from the list and click **OK**.
5. Select **Child Objects Only** from the **Apply onto** drop-down list and check **Read All Properties** from the list of permissions.
6. Click **OK** to complete the operation.

If the above configuration is too broad security-wise, you can enable the individual attributes for each user to be mapped.

Filesystems Are Not Displayed on Windows

If the CXFS drive letter is visible in Windows Explorer but no filesystems are mounted, do the following:

- Run `%ProgramFiles%\CXFS\cxfs_info` to ensure that the filesystems have been configured for this node.
- Verify the filesystems that should be mounted. For more information, see "Mounting Filesystems" on page 180.

- Ensure that the CXFS metadata server is up and that the Windows node is in the cluster membership; see "Verifying the Cluster Status" on page 183.
- Check that the CXFS Client service has started. See "Start/Stop the CXFS Client Service for Windows" on page 146 and "Verification that the CXFS Software is Running Correctly for Windows" on page 161.
- Check the **CXFS Client Log** in **CXFS Info** for warnings and errors regarding mounting filesystems.
- Check the cluster configuration to ensure that this node is configured to mount one or more filesystems.

Large Log Files on Windows

The CXFS Client service creates the following log file:

```
%ProgramFiles%\CXFS\log\cxfs_client.log
```

On an upgraded system, this log file may become quite large over a period of time if the verbosity level is increased. (New installations perform automatic log rotation when the file grows to 10 MB.)

To verify that log rotation is enabled, check the **Addition** arguments by modifying the installation (see "Modifying the CXFS Software for Windows" on page 147) and append the following if the `-z` option is not present:

```
-z 10000000
```

You must restart the CXFS Client service for the new settings to take effect. See "Start/Stop the CXFS Client Service for Windows" on page 146.

Windows Failure on Reboot

If the CXFS Windows node fails to start and terminates in a blue screen, reboot your computer and select the backup hardware profile with CXFS disabled. Alternatively, pressing `L` at the **Hardware Profile** menu will select the last configuration that was successfully started and shut down. If the node has only one hardware profile, press the spacebar after selecting the boot partition to get to the **Hardware Profile** menu.

NO_MORE_SYSTEM_PTES Error Message

A Windows problem may affect Windows CXFS nodes that are performing large asynchronous I/O operations. If the Windows node crashes with a NO_MORE_SYSTEM_PTES message, following these steps may help:



Caution: You should only try this if you are familiar with editing the registry and the risks involved in making these modifications. Doing so without proper experience may cause damage to your system. The value of **SystemPages** should only be increased above 110000 after consulting with a Microsoft Technical Support Engineer.

1. In **regedit**, navigate to the following value:

```
HKEY_LOCAL_MACHINE
> SYSTEM
  > CurrentControlSet
    > Control
      > Session Manager
        > Memory Management
```

2. Set the value of **PagePoolSize** to 0.
3. Set the value of **SystemPages** to 110000.
4. Close the registry editor.
5. Reboot the system to apply the changes.

Application Cannot Create File Under CXFS Drive Letter

If an application requires that it be able to create files and/or directories in the root of the CXFS drive, you must create a virtual drive for the system that maps to a mounted filesystem directory.

This can be performed using the `subst` command from the command prompt. For example, to use the CXFS filesystem `X:\mnt\tp9500_0` to the free drive letter `V`, you would enter the following:

```
C:\> subst V: X:\mnt\tp9500_0
```

To remove the mapping, run:

```
C:\> subst v: /D
```

Installation File Not Found Errors

Some installation programs are known to use old Windows APIs for file operations so that they work on older versions of Windows. These APIs use 8.3 filenames rather than the full filename, so the installation may fail with `file not found` or similar errors. In general, SGI recommends that you install software to a local disk and use CXFS filesystems primarily for data storage.

Problems Specific to Windows Vista, Windows Server 2008, and Windows 7

This section discusses problems specific to Windows Vista and Windows Server 2008:

- "Node Loses Membership Due to Hibernation" on page 169
- "Node Appears to be in Membership But Is Not" on page 170
- "Node Unable to cd to a Mounted Filesystem" on page 170
- "Slow Installation" on page 170

Node Loses Membership Due to Hibernation

If the Windows Vista, Windows Server 2008, or Windows 7 node hibernates, it will lose membership in the CXFS cluster. Hibernation is turned on by default for Windows Vista, Windows Server 2008, and Windows 7 and must be modified.

Do the following:

1. Select the following:

```
Start
  > Control Panel
    > Power Options
```

2. Verify that **Put the computer to sleep** is set to **Never**.

Alternatively, you can use the following command:

```
C:\> powercfg -S SCHEME_MIN
```

Node Appears to be in Membership But Is Not

If the Windows Vista, Windows Server 2008, or Windows 7 node appears to be in membership when the `cxfs_info` command is run from the node but is not in membership according to administration tools run on a server-capable administration node, it may be that User Account Control is still enabled (it is enabled by default).

User Account Control is not appropriate for use with CXFS, and you must disable it. See step 4 in "Client Software Installation for Windows" on page 132.

Node Unable to `cd` to a Mounted Filesystem

If you are unable to use the `cd` command on a Windows Vista, Windows Server 2008, or Windows 7 node for a filesystem that appears to be mounted, it may be that User Account Control is still enabled (it is enabled by default). For example, using a cygwin shell:

```
user@host /home/user
$ cd /cygdrive/x/mnt/stripefs
-bash: cd: /cygdrive/x/mnt/stripefs: Input/Output error**
```

User Account Control is not appropriate for use with CXFS, and you must disable it. See step 4 in "Client Software Installation for Windows" on page 132.

Slow Installation

If the installation of the Windows Vista, Windows Server 2008, or Windows 7 operating system seems to take a long time or does not complete, it may be caused by the HBAs or SAN fabric.

To resolve this problem, do the following:

1. Disconnect the system from the SAN fabric.
2. Remove the HBAs from the system or disable them in the BIOS.
3. Install the operating system.
4. Reinstall or reenable the HBA.
5. Install CXFS.
6. Reconnect the SAN fabric.

Reporting Windows Problems

This section discusses the following:

- "Retaining Windows Information" on page 171
- "Saving Crash Dumps for Windows" on page 172
- "Saving Application Crash Dumps for Windows Vista, Windows Server 2008, and Windows 7" on page 172
- "Generating a Crash Dump on a Hung Windows Node" on page 173

Retaining Windows Information

To report problems about a Windows node, you should retain platform-specific information and save crash dumps.

When reporting a problem about a CXFS Windows node to SGI, run the following:

```
Start
  > Program Files
    > CXFS
      > CXFS Dump
```

This will collect the following information:

- System information
- CXFS registry settings
- CXFS client logs
- CXFS version information
- Network settings
- Event log
- *(optionally)* Windows crash dump, as described in "Saving Crash Dumps for Windows" on page 172

In the dialog window, you will specify the location of the folder in which the `cxfsdump` output will be placed. The output will be placed beneath this folder, in a new folder whose name is of the form `CxfsDump_date_time`, where *date* is the

numeric date (such as 20080925 for September 25, 2008) and *time* is in military notation to the nearest second (such as 214456 for 9:44pm, 56 seconds).

Inside the `CxfsDump_date_time` folder will be a collection of `log` and `txt` files. You should compress the folder and files (using `zip` or `tar`) and send them to SGI.

The `cxfsdump /?` command displays a help message.

You should also obtain information about the entire cluster by running the `cxfsdump` utility on a server-capable administration node. See the information in the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Saving Crash Dumps for Windows

If you are experiencing crashes or if the Windows node hangs, you should configure the Windows node to save crash dumps to a filesystem that is not a CXFS filesystem. This crash dump can then be analyzed by SGI.

Do the following:

1. Click the right mouse button on the **My Computer** icon and select the following:

- Properties**
- > **Advanced**
- > **Startup and Recovery**
- > **Write debugging information to**

2. Enter a path on a filesystem other than a CXFS filesystem. You may also select a **Kernel Memory Dump**, which is a smaller dump that typically contains enough information regarding CXFS problems.
3. Reboot the system to apply the changes.

Saving Application Crash Dumps for Windows Vista, Windows Server 2008, and Windows 7

When a user space application crashes, it will remain in the TaskManager. In the dialog pop up that appears, detailing the crash information, you should right-click the application that caused the crash and select the crash dump option. This will save the dump to the current **User** directory so that the dump can then be analyzed.

Note: If you close the dialog without saving, the process will be removed from the TaskManager and the dump information will be lost.

For more information, see the following Microsoft article:

<http://support.microsoft.com/kb/931673>

Generating a Crash Dump on a Hung Windows Node

If user applications on a Windows node are no longer responsive and cannot be killed, you should attempt to generate a crash dump by forcing the node to crash.

After configuring the crash dump location (see "Saving Crash Dumps for Windows" on page 172), you can modify the registry so that a combination of key strokes will cause the Windows node to crash.

Note: This will only work on machines with a PS/2 keyboard.

Do the following:

1. In **regedit**, navigate to the following value:

```
HKEY_LOCAL_MACHINE
> SYSTEM
  > CurrentControlSet
    > Services
      > i8042prt
        > Parameters
```

2. Add a new entry by selecting the following:

```
Edit
  > Add Value
```

3. Enter the following information:

- **Value Name:** `CrashOnCtrlScroll`
- **Data Type:** `REG_DWORD`

- **Value:** 1

4. Reboot the system to apply the changes.

To generate a crash on the node after applying these changes, hold the right CTRL key and press SCROLL LOCK twice. See the following for more information:

<http://support.microsoft.com/?kbid=244139>

Configuring Client-Only Nodes

This chapter provides an overview of the procedures to add the client-only nodes to an established cluster. It assumes that you already have a cluster of server-capable administration nodes installed and running with mounted filesystems. These procedures will be performed by you or by SGI service personnel.

All CXFS administrative tasks other than restarting the Windows node must be performed using the CXFS GUI (invoked by the `cxfsmgr` command and connected to a server-capable administration node) or the `cxfs_admin` command on any host that has access permission to the cluster. The GUI and `cxfs_admin` provide a guided configuration and setup help for defining a cluster.

This section discusses the following tasks in cluster configuration:

- "Defining the Client-Only Nodes" on page 176
- "Adding the Client-Only Nodes to the Cluster (GUI)" on page 177
- "Defining the Switch for I/O Fencing" on page 177
- "Starting CXFS Services (GUI)" on page 179
- "Verifying LUN Masking" on page 180
- "Mounting Filesystems" on page 180
- "Unmounting Filesystems" on page 181
- "Forced Unmount of CXFS Filesystems" on page 181
- "Restarting the Windows Node" on page 181
- "Verifying the Cluster Configuration" on page 182
- "Verifying Connectivity in a Multicast Environment (Linux and Mac OS X Nodes)" on page 182
- "Verifying the Cluster Status" on page 183
- "Verifying the I/O Fencing Configuration" on page 186
- "Verifying Access to XVM Volumes" on page 187

For detailed configuration instructions, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Defining the Client-Only Nodes

To add a client-only node to a CXFS cluster, you must define it as a node in the pool.

Do the following to determine the value for the hostname field in the GUI:

- Linux: use the value displayed by `/bin/hostname`
- Mac OS X: use the value displayed by `/bin/hostname`
- Windows: select the following:

Start

> Settings

> Network and Dial-up Connections

> Advanced

> Network Identification

For example, the following shows the entries used to define a node named `mac1` in the `mycluster` cluster:

```
# /usr/cluster/bin/cxfs_admin -A -i mycluster
cxfs_admin:mycluster> create node name=mac1 os=macosx private_net=192.168.0.178
Event at [ Jan 21 15:58:02 ]
Node "mac1" has been created, waiting for it to join the cluster...
Waiting for node mac1, current status: Inactive
Waiting for node mac1, current status: Establishing membership
Waiting for node mac1, current status: Probing XVM volumes
Operation completed successfully
```

Or, in prompting mode:

```
# /usr/cluster/bin/cxfs_admin -i mycluster
Event at [ Jan 21 15:59:02 ]
cxfs_admin:mycluster> create node
Specify the attributes for create node:
  name? mac1
  type? client_only
  os? macosx
  private_net? 192.168.0.178
Event at [ Jan 21 15:59:10 ]
Node "man1" has been created, waiting for it to join the cluster...
Waiting for node mac1, current status: Inactive
Waiting for node mac1, current status: Establishing membership
Waiting for node mac1, current status: Probing XVM volumes
Operation completed successfully
```

For client-only nodes, you must specify a unique node ID if you use the GUI; `cxfs_admin` provides a default node ID.

For details about these commands, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Adding the Client-Only Nodes to the Cluster (GUI)

If you are using the GUI, you must add the defined nodes to the cluster. This happens by default if you are using `cxfs_admin`.

Depending upon your filesystem configuration, you may also need to add the node to the list of clients that have access to the volume. See "Mounting Filesystems" on page 180.

Defining the Switch for I/O Fencing

In order to protect data integrity, you must use I/O fencing on client-only nodes (or reset for those nodes with system controllers). I/O fencing requires a switch; see the release notes for supported switches.

For example, for a QLogic switch named `myswitch`:

```
cxfs_admin:mycluster> create switch name=myswitch vendor=qlogic
```

After you have defined the switch, you must ensure that all of the switch ports that are connected to the cluster nodes are enabled. To determine port status, enter the following on a server-capable administration node:

```
server-admin# /usr/cluster/bin/hafence -v
```

If there are disabled ports that are connected to cluster nodes, you must enable them. Log into the switch as user `admin` and use the following command:

```
switch# portEnable portnumber
```

You must then update the switch port information

For example, suppose that you have a cluster with port 0 connected to the node `blue`, port 1 connected to the node `green`, and port 5 connected to the node `yellow`, all of which are defined in cluster `colors`. The following output shows that the status of port 0 and port 1 is `disabled` and that the host is `UNKNOWN` (as opposed to port 5, which has a status of `enabled` and a host of `yellow`). Ports 2, 3, 4, 6, and 7 are not connected to nodes in the cluster and therefore their status does not matter.

```
server-admin# /usr/cluster/bin/hafence -v
Switch[0] "ptg-brocade" has 8 ports
Port 0 type=FABRIC status=disabled hba=0000000000000000 on host UNKNOWN
Port 1 type=FABRIC status=disabled hba=0000000000000000 on host UNKNOWN
Port 2 type=FABRIC status=enabled hba=210000e08b05fecf on host UNKNOWN
Port 3 type=FABRIC status=enabled hba=210000e08b01fec5 on host UNKNOWN
Port 4 type=FABRIC status=enabled hba=210000e08b01fec3 on host UNKNOWN
Port 5 type=FABRIC status=enabled hba=210000e08b019ef0 on host yellow
Port 6 type=FABRIC status=enabled hba=210000e08b0113ce on host UNKNOWN
Port 7 type=FABRIC status=enabled hba=210000e08b027795 on host UNKNOWN
```

In this case, you would need to enable ports 0 and 1:

Logged in to the switch:

```
switch# portEnable 0
switch# portEnable 1
```

Logged in to a server-capable administration node:

```
server-admin# /usr/cluster/bin/hafence -v
Switch[0] "ptg-brocade" has 8 ports
Port 0 type=FABRIC status=disabled hba=210000e08b0103b8 on host UNKNOWN
Port 1 type=FABRIC status=disabled hba=210000e08b0102c6 on host UNKNOWN
Port 2 type=FABRIC status=enabled hba=210000e08b05fecf on host UNKNOWN
Port 3 type=FABRIC status=enabled hba=210000e08b01fec5 on host UNKNOWN
Port 4 type=FABRIC status=enabled hba=210000e08b01fec3 on host UNKNOWN
Port 5 type=FABRIC status=enabled hba=210000e08b019ef0 on host yellow
Port 6 type=FABRIC status=enabled hba=210000e08b0113ce on host UNKNOWN
Port 7 type=FABRIC status=enabled hba=210000e08b027795 on host UNKNOWN
```

```
server-admin# /usr/cluster/bin/hafence -v
Switch[0] "ptg-brocade" has 8 ports
Port 0 type=FABRIC status=disabled hba=210000e08b0103b8 on host blue
Port 1 type=FABRIC status=disabled hba=210000e08b0102c6 on host green
Port 2 type=FABRIC status=enabled hba=210000e08b05fecf on host UNKNOWN
Port 3 type=FABRIC status=enabled hba=210000e08b01fec5 on host UNKNOWN
Port 4 type=FABRIC status=enabled hba=210000e08b01fec3 on host UNKNOWN
Port 5 type=FABRIC status=enabled hba=210000e08b019ef0 on host yellow
Port 6 type=FABRIC status=enabled hba=210000e08b0113ce on host UNKNOWN
Port 7 type=FABRIC status=enabled hba=210000e08b027795 on host UNKNOWN
```

Starting CXFS Services (GUI)

After adding the client-only nodes to the cluster with the GUI, you must start CXFS services for them, which enables the node by setting a flag for the node in the cluster database. This happens by default with `cxfs_admin`.

Verifying LUN Masking

You should verify that the HBA has logical unit (LUN) masking configured such that the LUNs are visible to all the nodes in the cluster after you connect the HBA to the switch and before configuring the filesystems with XVM. For more information, see the RAID documentation.

Mounting Filesystems

If you have specified that the filesystems are to be automatically mounted on any newly added nodes (such as setting `mount_new_nodes=true` for a filesystem in `cxfs_admin`), you do not need to specifically mount the filesystems on the new client-only nodes that you added to the cluster.

If you have specified that filesystems **will not be automatically mounted** (for example, by setting the advanced-mode `mount_new_nodes=false` for a filesystem in `cxfs_admin`), you can do the following to mount the new filesystem:

- With `cxfs_admin`, use the following command to mount the specified filesystem:

```
mount filesystemname nodes=nodename
```

For example:

```
cxfs_admin:mycluster> mount fs1 nodes=mac2
```

You can leave `mount_new_nodes=false`. You do not have to unmount the entire filesystem.

- With the GUI, you can mount the filesystems on the new client-only nodes by unmounting the currently active filesystems, enabling the mount on the required nodes, and then performing the actual mount.

Note: SGI recommends that you enable the *forced unmount* feature for CXFS filesystems, which is turned off by default; see:

- "Enable Forced Unmount When Appropriate" on page 18
 - "Forced Unmount of CXFS Filesystems" on page 181
-

Unmounting Filesystems

You can unmount a filesystem from all nodes in the cluster or from just the node you specify.

For example, to unmount the filesystem `fs1` from all nodes:

```
cxfs_admin:mycluster> unmount fs1
```

To unmount the filesystem only from the node `mynode`:

```
cxfs_admin:mycluster> unmount fs1 nodes=mynode
```

Forced Unmount of CXFS Filesystems

Normally, an unmount operation will fail if any process has an open file on the filesystem. However, a *forced unmount* allows the unmount to proceed regardless of whether the filesystem is still in use.

For example:

```
cxfs_admin:mycluster> create filesystem name=myfs forced_unmount=true
```

Using the CXFS GUI, define or modify the filesystem to unmount with force and then unmount the filesystem.

For details, see the “CXFS Filesystems Tasks with the GUI” sections of the GUI chapter in the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Restarting the Windows Node

After completing the steps in “Postinstallation Steps for Windows” on page 140 and this chapter, you should restart the Windows node. This will automatically start the driver and the CXFS Client service.

When you log into the node after restarting it, Windows Explorer will list the CXFS drive letter, which will contain the CXFS filesystems configured for this node.

Verifying the Cluster Configuration

To verify that the client-only nodes have been properly added to the cluster, run the `cxfs-config` command on the metadata server. For example:

```
/usr/cluster/bin/cxfs-config -all -check
```

This command will dump the current cluster nodes, private network configuration, filesystems, XVM volumes, failover hierarchy, and switches. It will check the configuration and report any common errors. You should rectify these error before starting CXFS services.

Verifying Connectivity in a Multicast Environment (Linux and Mac OS X Nodes)

To verify general connectivity in a multicast environment (other than on a Windows node), you can execute a `ping` command on the `224.0.0.1` IP address.

To verify the CXFS heartbeat, use the `224.0.0.250` IP address. The `224.0.0.250` address is the default CXFS heartbeat multicast address (because it is the default, this address does not have to appear in the `/etc/hosts` file).

Note: A node is capable of responding only when the administration daemons (`fs2d`, `cmdond`, `cad`, and `crsd`) or the `cxfs_client` daemon is running.

For example, to see the response for two packets sent from Linux IP address `10.0.0.27` to the multicast address for CXFS heartbeat and ignore loopback, enter the following:

```
linux# ping -I 10.0.0.27 -L -c 2 224.0.0.250
```

Note: By default, most Linux nodes will not respond to a multicast `ping`. To enable multicast `ping` for Linux nodes, see "Verifying the Private and Public Networks for Linux" on page 40.

To override the default address, you can use the `-c` and `-m` options or make the name `cluster_mcast` resolvable on all nodes (such as in the `/etc/hosts` file). For more information, see the `cxfs_client` man page.

Verifying the Cluster Status

To verify that the client-only nodes have been properly added to the cluster and that filesystems have been mounted, use the view area of the CXFS GUI, the `cxfs_admin status` command, or the `clconf_info` command (on a server-capable administration node) and the `cxfs_info` command (on a client-only node).

For example, using `cxfs_admin`:

```
cxfs_admin:clusterOne> status
Event at [ Oct 22 13:40:51 ]
Cluster      : clusterOne
Tiebreaker   :
Client Licenses : enterprise  allocated 0 of 256
                  workstation allocated 2 of 50
-----
```

Node	Cell ID	Age	Status
bert *	0	8	Mounted 1 of 2 filesystems
cxfsxe5 *	1	7	Stable
twig *	2	-	Inactive
cxfsxe10	3	-	Disabled
penguin17	4	0	Establishing membership
pg-27	5	0	Establishing membership

```
-----
```

Filesystem	Server	Status
zj01s0	cxfsxe5	1 of 6 nodes mounted, bert trying to mount
zj01s1	bert	Mounted [2 of 6 nodes]

```
-----
```

Switch	Port Count	Known Fenced Ports
brocade26cp1	192	4, 20, 21, 132, 223

```
-----
```

The following example for a different cluster shows `clconf_info` output:

```
cxfsxe5:~ # /usr/cluster/bin/clconf_info
```

```
Event at [2009-10-22 13:41:24]
```

```
Membership since Thu Oct 22 13:39:22 2009
```

Node	NodeID	Status	Age	CellID
cxfsxe5	1	up	7	1
twig	2	DOWN	-	2
bert	3	DOWN	-	0
pg-27	4	DOWN	-	5
penguin17	5	DOWN	-	4
cxfsxe10	6	inactive	-	3

```
2 CXFS FileSystems
```

```
/dev/cxvm/zj01s1 on /mnt/zj01s1 enabled server=(cxfsxe5) 0 client(s)=() status=UP  
/dev/cxvm/zj01s0 on /mnt/zj01s0 enabled server=(cxfsxe5) 0 client(s)=() status=UP
```

On client-only nodes, the `cxfs_info` command serves a similar purpose. The command path is as follows:

- Linux and Mac OS X: `/usr/cluster/bin/cxfs_info`
- Windows: `%ProgramFiles%\CXFS\cxfs_info.exe`

On Linux and Mac OS X, you can use the `-e` option to wait for events, which keeps the command running until you kill the process and the `-c` option to clear the screen between updates.

For example, on a Linux node named `pg-27`:

```
pg-27% /usr/cluster/bin/cxfs_info
```

```
cxfs_client status [timestamp Oct 22 13:40:46 / generation 5648504]
```

```
CXFS client:
```

```
state: reconfigure (5), cms: quiesce, xvm: down, fs: down
```

```
Cluster:
```

```
clusterOne (9) - enabled
```

```
Local:
```

```
pg-27 (5) - enabled
```

```

Servers:
  bert      enabled  DOWN  0
  cxfsxe5   enabled  DOWN  1
  twig      enabled  DOWN  2
Nodes:
  cxfsxe10  disabled DOWN  3
  penguin17 enabled  DOWN  4
  pg-27     enabled  DOWN  5
Filesystems:
  zj01s0
  zj01s1

```

The `CXFS client` line shows the state of the client in the cluster, which can be one of the following states:

<code>bootstrap</code>	Initial state after starting <code>cxfs_client</code> , while listening for bootstrap packets from the cluster.
<code>connect</code>	Connecting to the CXFS metadata server.
<code>query</code>	The client is downloading the cluster database from the metadata server.
<code>reconfigure</code>	The cluster database has changed, so the client is reconfiguring itself to match the cluster database.
<code>stable</code>	The client has been configured according to what is in the cluster database.
<code>stuck</code>	The client is unable to proceed, usually due to a configuration error. Because the problem may be transient, the client periodically reevaluates the situation. The number in parenthesis indicates the number of seconds the client will wait before retrying the operation. With each retry, the number of seconds to wait is increased; therefore, the higher the number the longer it has been stuck. See the log file for more information.
<code>terminate</code>	The client is shutting down.

The `cms` field has the following states:

<code>unknown</code>	Initial state before connecting to the metadata server.
<code>down</code>	The client is not in membership.
<code>fetal</code>	The client is joining membership.
<code>up</code>	The client is in membership.
<code>quiesce</code>	The client is dropping out of membership.

The `xvm` field has the following states:

<code>unknown</code>	Initial state before connecting to the metadata server.
<code>down</code>	After membership, but before any XVM information has been gathered.
<code>fetal</code>	Gathering XVM information.
<code>up</code>	XVM volumes have been retrieved.

The `fs` field has the following states:

<code>unknown</code>	Initial state before connecting to the metadata server.
<code>down</code>	One or more filesystems are not in the desired state.
<code>up</code>	All filesystems are in the desired state.
<code>retry</code>	One or more filesystems cannot be mounted/unmounted, and will retry. See the "Filesystem" section of <code>cxfs_info</code> output to see the affected filesystems.

Verifying the I/O Fencing Configuration

To determine if a node is correctly configured for I/O fencing, log in to a server-capable administration node and use the `cxfs-config(8)` command. For example:

```
server-admin# /usr/cluster/bin/cxfs-config
```

The failure hierarchy for a client-only node should be listed as Fence, Shutdown, as in the following example:

```
Machines:
node cxfswin2: node 102 cell 1 enabled Windows client_only
hostname: cxfswin2.melbourne.sgi.com
```

```
fail policy: Fence, Shutdown
nic 0: address: 192.168.0.102 priority: 1
```

See "Defining the Client-Only Nodes" on page 176 to change the failure hierarchy for the node if required.

The HBA ports should also be listed in the switch configuration:

Switches:

```
switch 1: 16 port brocade admin@asg-fcsw7 <no ports masked>
  port 5: 210200e08b51fd49 cxfswin2
  port 15: 210100e08b32d914 admin1
switch 2: 16 port brocade admin@asg-fcsw8 <no ports masked>
  port 5: 210300e08b71fd49 cxfswin2
  port 14: 210000e08b12d914 admin1
```

No warnings or errors should be displayed regarding the failure hierarchy or switch configuration.

If the HBA ports for the client node are not listed, see the following:

- "I/O Fencing for Linux" on page 44
- "I/O Fencing for Mac OS X" on page 79
- "I/O Fencing for Windows" on page 142

Verifying Access to XVM Volumes

To verify that a client node has access to all XVM volumes that are required to mount the configured filesystems, log on to a server-capable administration node and run:

```
server-admin# /usr/cluster/bin/cxfs-config -xvm
```

This will display the list of filesystems and the XVM volume and volume elements used to construct those filesystems. For example:

```
fs stripe1: /mnt/stripel          enabled
  device = /dev/cxvm/stripel
  force = false
  options = []
  servers = cxfs5 (0), cxfs4 (1)
```

```
clients = cxfs4, cxfs5, cxfs6, cxfsmac4, cxfssun1
xvm:
  vol/stripel                               0 online,open
    subvol/stripel/data                     2292668416 online,open
      stripe/stripel                       2292668416 online,open
        slice/d9400_0s0                    1146334816 online,open
        slice/d9400_1s0                    1146334816 online,open

data size: 1.07 TB
```

You can then run the `xvm` command to identify the XVM volumes and disk devices. This provides enough information to identify the device's WWN, LUN, and controller. In the following example, the `slice/d9400_0s0` from `phys/d9400_0` is LUN 0 located on a RAID controller with WWN 200500a0b80cedb3.

```
server-admin# /sbin/xvm show -e -t vol
vol/stripel                               0 online,open
  subvol/stripel/data                     2292668416 online,open
    stripe/stripel                       2292668416 online,open (unit size: 1024)
      slice/d9400_0s0                    1146334816 online,open (d9400_0:/dev/rdisk/200500a0b80cedb3/lun0vol/c2p1)
      slice/d9400_1s0                    1146334816 online,open (d9400_1:/dev/rdisk/200400a0b80cedb3/lun1vol/c3p1)
```


On Linux and Mac OS X platforms, you can then run the `xvm` command on the client to identify the matching disk devices on the client. For example:

```
linux# /sbin/xvm show -e -t vol
```

Note: The `xvm` command on the Windows does not display WWNs.

If a disk device has not been found for a particular volume element, the following message will be displayed instead of the device name:

```
no direct attachment on this cell
```

Using the device information from the server-capable administration node, it should then be possible to determine if the client can see the same devices using the client HBA tools and the RAID configuration tool.

To see the complete list of volumes and devices mappings, especially when XVM failover V2 is configured, run:

```
linux# /sbin/xvm show -v phys
```

For more information about `xvm`, see the *XVM Volume Manager Administrator's Guide*.

General Troubleshooting

This chapter contains the following:

- "Identifying Problems" on page 191
- "Typical Problems and Solutions" on page 195
- "Verifying the XVM Mirror Licenses" on page 200
- "Using SGI Knowledgebase" on page 201
- "Reporting Problems to SGI" on page 201

Also see:

- "Troubleshooting for Linux" on page 53
- "Troubleshooting for Mac OS X" on page 85
- "Troubleshooting for Windows" on page 160

For more advanced cluster troubleshooting, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Identifying Problems

This section provides tips about identifying problems:

- "Is the Node Configured Correctly? " on page 192
- "Is the Node in Membership?" on page 192
- "Is the Node Mounting All Filesystems?" on page 192
- "Can the Node Access All Filesystems?" on page 193
- "Are There Error Messages?" on page 193
- "What Is the Network Status?" on page 194
- "What is the Status of XVM Mirror Licenses?" on page 194

Is the Node Configured Correctly?

To determine the current configuration of a node in a cluster, run the following command on a CXFS server-capable administration node:

```
server-admin# /usr/cluster/bin/cxfs-config -all
```

For more information, see "Verifying the Cluster Status" on page 183.

Confirm that the host type, private network, and failure hierarchy are configured correctly, and that no warnings or errors are reported. You should rectify any warnings or errors before proceeding with further troubleshooting.

Is the Node in Membership?

To determine if the node is in the cluster membership, use the tools described in "Verifying the Cluster Status" on page 183.

If the client is not in membership, see:

- "Verifying the Cluster Configuration" on page 182
- "Verifying Connectivity in a Multicast Environment (Linux and Mac OS X Nodes)" on page 182
- "Unable to Achieve Membership" on page 195

Is the Node Mounting All Filesystems?

To determine if the node has mounted all configured filesystems, use the tools described in "Verifying the Cluster Status" on page 183.

If the client has not mounted all filesystems, see:

- "Verifying the Cluster Configuration" on page 182
- "Verifying Access to XVM Volumes" on page 187
- "Determining If a Client-Only Node Is Fenced" on page 198
- Appendix C, "Mount Options Support" on page 209

Can the Node Access All Filesystems?

To determine if the client-only node can access a filesystem, navigate the filesystem and attempt to create a file.

If the filesystem appears to be empty, the mount may have failed or been lost. See:

- "Determining If a Client-Only Node Is Fenced" on page 198
- "Verifying Access to XVM Volumes" on page 187

If accessing the filesystem hangs the viewing process, see "Filesystem Appears to Be Hung" on page 196.

Are There Error Messages?

When determining the state of the client-only node, you should check error message logs to help identify any problems.

Appendix A, "Operating System Path Differences" on page 203 lists the location of the `cxfs_client` log file for each platform. This log is also displayed in the Windows version of `cxfs_info`.

Each platform also has its own system log for kernel error messages that may also capture CXFS messages. See:

- "Log Files on Linux" on page 30
- "Log Files on Mac OS X" on page 61
- "Log Files and Cluster Status for Windows" on page 94

There are various logs located on the CXFS server-capable administration nodes. For more information, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Note: The `syslog` file or Linux `/var/log/messages` file may contain spurious error messages. This problem occurs in clusters with multiple private networks when one of the active network interfaces is downed using the `ifconfig` command. This problem may also happen when the network is interrupted for other reasons.

Some of the transport failure messages generated may have cell IDs that are different from the cell ID of the node with the downed interface. The spurious error messages do not appear to affect the continued operation of the cluster.

What Is the Network Status?

Use the `netstat` command on a client-only node to determine the network status.

For example, to determine if you have a bad connection, you could enter the following from a DOS console on the Windows platform:

```
C:\Documents and Settings\cxfsqa>netstat -e -s
```

The `-s` option shows per-protocol statistics. The Linux and Windows systems support the `-e` option, which shows Ethernet statistics. See the `netstat(1)` man page for information about options.

What is the Status of XVM Mirror Licenses?

To view the current status of XVM mirror licenses, use the following command and search for the line containing the keyword `mirrors`:

```
xvm show -subsystem
```

For example:

```
# xvm show -subsystem
XVM Subsystem Information:
-----
apivers:                26
config gen:             33
privileged:             1
clustered:              1
cluster initialized:    1
user license enabled:   1
local mirrors enabled:  1
cluster mirrors enabled: 1
snapshot enabled:       1
snapshot max blocks:    -1
snapshot blocks used:   0
```

Typical Problems and Solutions

This section contains the following typical problems that apply to any platform:

- "cdb Error in the `cxfs_client` Log" on page 195
- "Unable to Achieve Membership" on page 195
- "Filesystem Appears to Be Hung" on page 196
- "Determining If a Client-Only Node Is Fenced" on page 198
- "No HBA WWPNs are Detected" on page 199
- "Membership Is Prevented by Firewalls" on page 199
- "Devices are Unknown" on page 199
- "Clients Cannot Join the Cluster After Relocation" on page 200

`cdb` Error in the `cxfs_client` Log

The following errors in the `cxfs_client` may log indicate that the client is not found in the cluster database:

```
cxfs_client: cis_client_run querying CIS server
cxfs_client: cis_cdb_go ERROR: Error returned from server: cdb error (6)
```

Run the `cxfs-config` command on the metadata server and verify that the client's hostname appears in the cluster database. For additional information about the error, review the `/var/cluster/ha/log/fs2d_log` file on the metadata server.

Unable to Achieve Membership

If `cxfs_info` does not report that CMS is UP, do the following:

1. Check that `cxfs_client` is running. See:
 - "Start/Stop `cxfs_client` for Linux" on page 45
 - "Start/Stop `cxfs_client` for Mac OS X" on page 81
 - "Start/Stop the CXFS Client Service for Windows" on page 146

2. Look for other warnings and error messages in the `cxfs_client` log file. For the location of the log file on different platforms, see Appendix A, "Operating System Path Differences" on page 203.
3. Check `cxfs-config` output on the CXFS server-capable administration node to ensure that the client is correctly configured and is reachable via the configured CXFS private network. For example:

```
server-admin# /usr/cluster/bin/cxfs-config -all
```
4. Check that the client is enabled into the cluster by running `clconf_info` on a CXFS server-capable administration node.
5. Look in the system log on the CXFS metadata server to ensure that the server detected the client that is attempting to join membership and check for any other CXFS warnings or errors.
6. Check that the metadata server has the node correctly configured in its hostname lookup scheme (`/etc/host` file or DNS).
7. If you are still unable to resolve the problem, reboot the client node.
8. If rebooting the client node in step 7 did not resolve the problem, restart the cluster administration daemons (`fs2d`, `cad`, `cmond`, and `crsd`) on the metadata server. This step may result in a temporary delay in access to the filesystem from all nodes.
9. If restarting cluster administration daemons in step 8 did not solve the problem, reboot the metadata server. This step may result in the filesystems being unmounted on all nodes.

Filesystem Appears to Be Hung

If any CXFS filesystem activity appears to be hung, do the following:

1. Check that the client is still in membership and that the filesystem is mounted according to `cxfs_info`.
2. Check on the metadata server to see if any messages are more than a few seconds in age (known as a *stuck message*).
3. If there is a stuck message, gather information for SGI support:

- Find the stack trace for the stuck thread. For example:

```
crash> bt 0xe00000305f2a0000
#0 [BSP:e00000305f2a1e90] schedule at a0000001006e3d30
#1 [BSP:e00000305f2a1e60] schedule_timeout at a0000001006e4460
#2 [BSP:e00000305f2a1e20] __down at a0000001006e5d60
#3 [BSP:e00000305f2a1df0] down at a0000001000d5db0
#4 [BSP:e00000305f2a1dd0] xfs_buf_lock at a000000203e15030
#5 [BSP:e00000305f2a1d80] _xfs_buf_find at a000000203e18cf0
#6 [BSP:e00000305f2a1d20] xfs_buf_get_flags at a000000203e18f60
#7 [BSP:e00000305f2a1ce8] xfs_buf_read_flags at a000000203e19220
#8 [BSP:e00000305f2a1c90] xfs_trans_read_buf at a000000203df6750
#9 [BSP:e00000305f2a1c48] xfs_btree_read_bufs at a000000203d88690
#10 [BSP:e00000305f2a1ba8] xfs_inobt_lookup at a000000203db79b0
#11 [BSP:e00000305f2a1b68] xfs_inobt_lookup_eq at a000000203db8010
#12 [BSP:e00000305f2a1ab0] xfs_dialloc at a000000203db4ca0
#13 [BSP:e00000305f2a1a38] xfs_ialloc at a000000203dc6660
#14 [BSP:e00000305f2a1990] xfs_dir_ialloc at a000000203df90e0
#15 [BSP:e00000305f2a18f8] xfs_mkdir at a000000203e076e0
#16 [BSP:e00000305f2a18a8] cxfs_mkdir at a000000203c0a460
#17 [BSP:e00000305f2a1858] dmapi_bnc_mkdir at a000000203bb6b60
#18 [BSP:e00000305f2a17f0] bhvlock_vop_mkdir at a000000203bb0330
#19 [BSP:e00000305f2a1788] xfs_vn_mknod at a000000203e23110
#20 [BSP:e00000305f2a1758] xfs_vn_mkdir at a000000203e23460
#21 [BSP:e00000305f2a1710] vfs_mkdir at a0000001001cb320
#22 [BSP:e00000305f2a16b8] cxfs_server_lock_mkdir at a000000203c21980
#23 [BSP:e00000305f2a1438] I_dsvn_create_0 at a000000203bea810
#24 [BSP:e00000305f2a1398] dsvn_msg_dispatcher at a000000203b1e4a0
#25 [BSP:e00000305f2a1330] msg_demux at a00000020393e0a0
#26 [BSP:e00000305f2a11f0] mtcp_notify at a000000203954200
#27 [BSP:e00000305f2a1148] tsv_thread_setup at a000000203a2b8e0
#28 [BSP:e00000305f2a10c8] kthread_init at a000000203a22050
#29 [BSP:e00000305f2a10a0] kernel_thread_helper at a000000100014a30
#30 [BSP:e00000305f2a10a0] start_kernel_thread at a00000010000a4c0
```

- Run `cxfsdump` on the metadata server.
 - Run `cxfsdump` on the client that has the stuck message.
 - If possible, force the client that has the stuck message to generate a crash dump.
4. Reboot the client that has the stuck message. This is required for CXFS to recover.

Determining If a Client-Only Node Is Fenced

To determine if a client-only node is fenced, log in to a CXFS server-capable administration node and use the `hafence(1m)` command. A fenced port is displayed as `status=disabled`.

In the following example, all ports that have been registered as CXFS host ports are not fenced:

```
admin# /usr/cluster/bin/hafence -q
Switch[0] "brocade04" has 16 ports
Port 4 type=FABRIC status=enabled hba=210000e08b0042d8 on host o200c
Port 5 type=FABRIC status=enabled hba=210000e08b00908e on host cxfs30
Port 9 type=FABRIC status=enabled hba=2000000173002d3e on host cxfssun3
```

All switch ports can also be shown with `hafence`:

```
admin# /usr/cluster/bin/hafence -v
Switch[0] "brocade04" has 16 ports
Port 0 type=FABRIC status=enabled hba=2000000173003b5f on host UNKNOWN
Port 1 type=FABRIC status=enabled hba=2000000173003adf on host UNKNOWN
Port 2 type=FABRIC status=enabled hba=210000e08b023649 on host UNKNOWN
Port 3 type=FABRIC status=enabled hba=210000e08b021249 on host UNKNOWN
Port 4 type=FABRIC status=enabled hba=210000e08b0042d8 on host o200c
Port 5 type=FABRIC status=enabled hba=210000e08b00908e on host cxfs30
Port 6 type=FABRIC status=enabled hba=2000000173002d2a on host UNKNOWN
Port 7 type=FABRIC status=enabled hba=2000000173003376 on host UNKNOWN
Port 8 type=FABRIC status=enabled hba=2000000173002c0b on host UNKNOWN
Port 9 type=FABRIC status=enabled hba=2000000173002d3e on host cxfssun3
Port 10 type=FABRIC status=enabled hba=2000000173003430 on host UNKNOWN
Port 11 type=FABRIC status=enabled hba=200900a0b80c13c9 on host UNKNOWN
Port 12 type=FABRIC status=disabled hba=0000000000000000 on host UNKNOWN
Port 13 type=FABRIC status=enabled hba=200d00a0b80c2476 on host UNKNOWN
Port 14 type=FABRIC status=enabled hba=1000006069201e5b on host UNKNOWN
Port 15 type=FABRIC status=enabled hba=1000006069201e5b on host UNKNOWN
```

When the client-only node joins membership, any fences on any switch ports connected to that node should be lowered and the status changed to `enabled`.

However, if the node still does not have access to the storage, do the following:

- Check that the HBA WWPNs were correctly identified. See "Verifying the I/O Fencing Configuration" on page 186.

- Check the `cxfs_client` log file for warnings or errors while trying to determine the HBA WWPNs. See "No HBA WWPNs are Detected" on page 199.
- Log into the Fibre Channel switch. Check the status of the switch ports and confirm that the WWPNs match those identified by `cxfs_client`.

No HBA WWPNs are Detected

On most platforms, the `cxfs_client` software automatically detects the world wide port names (WWPNs) of any supported host bus adapters (HBAs) in the system that are connected to a switch that is configured in the cluster database. These HBAs will then be available for fencing.

However, if no WWPNs are detected, there will be messages about loading the HBA/SNIA library.

See:

- "I/O Fencing for Linux" on page 44
- "I/O Fencing for Mac OS X" on page 79
- "I/O Fencing for Windows" on page 142

Membership Is Prevented by Firewalls

If a client has trouble obtaining membership, verify that the system firewall is configured for CXFS use. See "Configure Firewalls for CXFS Use" on page 19.

Devices are Unknown

You can run the `cxfs-reprobe` script on a client-only node (other than Windows) to look for devices and perform a SCSI bus reset if necessary. `cxfs-reprobe` will also issue an XVM probe to tell XVM that there may be new devices available:

```
client# /var/cluster/cxfs_client-scripts/cxfs-reprobe
```

Clients Cannot Join the Cluster After Relocation

If a CXFS client fails or exits the cluster during the metadata server relocation process, the relocation process and the client recovery are likely to hang. This prevents any clients, including the failed client, from joining the cluster.

Once in this state, it may be possible to resolve the deadlock by resetting or power-cycling the `fs2d` quorum master. To determine the quorum master, see the instructions in *CXFS 6 Administration Guide for SGI InfiniteStorage*.

Verifying the XVM Mirror Licenses

To view the current status of XVM mirror licenses on client-only nodes, use the following command and search for the line containing the keyword `mirrors`:

```
xvm show -subsystem
```

For example:

```
client# xvm show -subsystem
XVM Subsystem Information:
-----
apivers:                26
config gen:             33
privileged:             1
clustered:              1
cluster initialized:    1
user license enabled:   1
local mirrors enabled:  1
cluster mirrors enabled: 1
snapshot enabled:       1
snapshot max blocks:    -1
snapshot blocks used:   0
```

Using SGI Knowledgebase

If you encounter problems and have an SGI support contract, you can log on to Supportfolio and access the Knowledgebase tool to help find answers.

To log in to Supportfolio Online, see:

<https://support.sgi.com/login>

Then click on **Search the SGI Knowledgebase** and select the type of search you want to perform.

If you need further assistance, contact SGI Support.

Reporting Problems to SGI

When reporting a problem with a client-only node, it is important to retain the appropriate information; having access to this information will greatly assist SGI in the process of diagnosing and fixing problems. The methods used to collect required information for problem reports are platform-specific:

- "Reporting Linux Problems" on page 56
- "Reporting Mac OS X Problems" on page 87
- "Reporting Windows Problems" on page 171

Operating System Path Differences

This appendix lists the location of CXFS-specific commands and files. For more information, see the `cxfs_client` man page.

Table A-1 Linux Paths

Component	Path
CXFS client service:	<code>/usr/cluster/bin/cxfs_client</code>
Command that normally invokes the client daemon:	<code>/etc/init.d/cxfs_client</code>
Log file:	<code>/var/log/cxfs_client</code>
Options file:	<code>/etc/cluster/config/cxfs_client.options</code>
CXFS status:	<code>/usr/cluster/bin/cxfs_info</code>
Hostname/address information	<code>/etc/hosts</code>
GRIo v2 administration	<code>/usr/sbin/grioadmin</code>
GRIo monitoring	<code>/usr/sbin/griomon</code>
GRIo v2 quality of service	<code>/usr/sbin/griogos</code>
XVM query	<code>/sbin/xvm</code>

Table A-2 Mac OS X Paths

Component	Path
CXFS client daemon:	/usr/cluster/bin/cxfs_client
Command that normally invokes the client daemon:	/Library/StartupItems/cxfs/cxfs
Log file:	/var/log/cxfs_client
Options file:	/usr/cluster/bin/cxfs_client.options
CXFS status:	/usr/cluster/bin/cxfs_info
Hostname/address information	/etc/hosts
GRIo v2 administration	/usr/sbin/grioadmin
GRIo monitoring	/usr/sbin/griomon
GRIo v2 quality of service	/usr/sbin/griogps
XVM query	/usr/cluster/bin/xvm

Table A-3 Windows Paths

Component	Path
CXFS client service:	%SystemRoot%\system32\cxfs_client.exe
Command that normally invokes the client service:	See "Start/Stop the CXFS Client Service for Windows" on page 146
Log file:	%ProgramFiles%\CXFS\log\cxfs_client.log
Options file:	See "Modifying the CXFS Software for Windows" on page 147
CXFS status:	%ProgramFiles%\CXFS\cxfs_info.exe
Hostname and address information:	%SystemRoot%\system32\drivers\etc\hosts
GRIO v2 administration:	%ProgramFiles%\CXFS\grioadmin.exe
GRIO%ProgramFiles%\CXFS\griomon.exe monitoring	
GRIO v2 quality of service:	%ProgramFiles%\CXFS\griooqs.exe
XVM query:	(unsupported)

Filesystem and Logical Unit Specifications

Table B-1 on page 208 summarizes filesystem and logical unit specifications differences among the supported client-only platforms.

Table B-1 Filesystem and Logical Unit Specifications

Item	Linux x86-64	Linux ia64	Mac OS X	Windows
Maximum filesystem size	2^{64} bytes	2^{64} bytes	2^{64} bytes	2^{64} bytes
Maximum file size/offset	$2^{63}-1$ bytes	$2^{63}-1$ bytes	$2^{63}-1$ bytes	$2^{63}-1$ bytes
Filesystem block size (in bytes) ¹	512, 1024, 2048, or 4096	512, 1024, 2048, 4096, 8192, or 16384	4096, 8192, 16384, 32768, or 65536	512, 1024, 2048, 4096, 8192, 16384, 32768, or 65536
XVM device block size (in bytes)	512	512	512	512
Physical LUN limit for DVH-labeled disks	2 TB	2 TB	2 TB	2 TB
Physical LUN limit for GPT-labeled disks ²	2^{63} device blocks	2^{63} device blocks	2^{63} device blocks	2^{63} device blocks
Maximum concatenated slices	65536	65536	65536	65536

¹ If the filesystem is to be accessible by other platforms in a multiOS cluster, its block size must be supported on all platforms in the cluster.

² Note the following about physical LUN limits for GPT-labeled disks:

- Physical LUNs with GPT labels are not constrained by XVM or CXFS to be smaller than the largest possible filesystem.
- Cluster nodes may constrain the LUN size to be smaller due to driver or other operating system constraints. A LUN used in the cluster may not be larger than the maximum size allowed by any node.
- All nodes that mount a filesystem using LUNs larger than 2 TB must be upgraded to CXFS 4.2 or later. However, Windows XP does not support LUNs greater than 2 TB in size. Filesystem corruption will occur if you attempt to write to the LUN above the 2-TB boundary. CXFS for Windows will not allow a filesystem to be mounted if any part of it resides on a LUN that is greater than 2-TB in size.

Mount Options Support

The table in this appendix list the mount options that are supported by CXFS, depending upon the server platform. Some of these mount options affect only server behavior and are ignored by client-only nodes.

The tables also list those options that are not supported, especially where that support varies from one platform to another. The `mount` commands supports many additional options, but these options may be silently ignored by the clients, or cause the mount to fail and should be avoided. For more information, see the `mount(8)` man page.

Note: The following are mandatory, internal CXFS mount options that cannot be modified and are set by `clconfd` and `cxfs_client`:

```
client_timeout
server_list
```

The table uses the following abbreviations:

Y = Yes, client checks for the option and sets flag/fields for the metadata server

N = No, client does not check for the option

S = Supported

n = Not supported

D = Determined by the CXFS administration tools (not user-configurable)

The Linux architectures are (as output by `uname -i`) 64-bit Linux on x86-64 and ia64 architectures.

Table C-1 Mount Options Support for Client-Only Platforms

Option	Checked by Client	Linux 64	Mac OS X	Windows
attr2	N	n	n	n
biosize	Y	S	S	S
client_timeout	Y	D	D	D
dmapi	Y	S	S	S
dmi	Y	S	S	S
filestreams ¹	Y	S	S	S
gnoenforce	Y	S	S	S
gquota	Y	S	S	S
grpuid	Y	S	S	S
grpquota	Y	S	S	S
inode64	Y	S	S	S
largeio	Y	S	S	S
logbsize	Y	S	S	S
logbufs	Y	S	S	S
logdev	N	S	S	S

¹ Do not use the `dmi` and `filestreams` options together. DMF is not able to arrange file extents on disk in a contiguous fashion when restoring offline files. This means that a DMF-managed filesystem most likely will not maintain the file layouts or performance characteristics normally associated with filesystems using the `filestreams` mount option.

Option	Checked by Client	Linux 64	Mac OS X	Windows
mrquota	Y	S	n	n
noalign	Y	S	n	S
noatime	Y	S	S	S
noattr2	N	n	n	n
noauto	N	n	n	n
nobarrier	N	S	S	S
nodev	N	S	S	S
nolargeio	N	S	n	n
noquota	Y	S	S	S
nosuid	Y	S	S	S
osyncisdsync	Y	S	n	n
pqnoenforce	Y	S	S	S
pquota	Y	S	S	S
prjquota	Y	S	S	S
qnoenforce	Y	S	S	S
quota	Y	S	S	S
ro	Y	S	S	S
rtdev	N	n	n	n
rw	N	S	S	S

Option	Checked by Client	Linux 64	Mac OS X	Windows
server_list	Y	D	D	D
server_timeout	Y	D	D	D
sunit	Y	S	S	S
swalloc	Y	S	S	S
swidth	Y	S	S	S
uqnoenforce	Y	S	S	S
uquota	Y	S	S	S
usrquota	Y	S	S	S
wsync	Y	S	S	S

Error Messages

The following are commonly seen error messages:

- "Could Not Start CXFS Client Error Messages" on page 213
- "CMS Error Messages" on page 213
- "Mount Messages" on page 214
- "Network Connectivity Messages" on page 214
- "Device Busy Message" on page 214
- "Windows Messages" on page 215

Could Not Start CXFS Client Error Messages

The following error message indicates that the `cxfs_client` service has failed the license checks:

```
Could not start the CXFS Client service on Local Computer.
```

```
Error 10038: An operation was attempted on something that is not a socket.
```

You must install the license as appropriate. See the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

CMS Error Messages

The following messages may be logged by CMS.

```
CMS excluded cells 0xXXX with incomplete connectivity
```

Generated when CMS delivers a membership that excluded some **new** cells that had not established connections with enough cells yet to be admitted. `0xXXX` is a bitmask of excluded cells.

```
CMS calculation limited to last membership:configuration change incomplete on cells 0xXXX
```

Generated when the leader is attempting to make a configuration change current (that is, actually use the change on all nodes), but some cells in the cluster have not yet received the configuration change staged (uploaded and ready to be made current). 0xXXX is a bitmask of cells that do not yet have the change in their configuration. Changes make their way through the cluster asynchronously, so this situation is expected. It can take a few attempts by the CMS leader before all nodes have the change staged. As long as this situation resolves eventually, there is no problem.

CMS calculation limited to last membership:recovery incomplete

Generated when new members were disallowed due to recovery from the last cell failure that is still being processed.

Mount Messages

cxfs_client:op_failed ERROR: Mount failed for concat0

A filesystem mount has failed and will be retried.

Network Connectivity Messages

```
unable to join multicast group on interface
unable to create multicast socket
unable to allocate interface list
unable query interfaces
failed to configure any interfaces
unable to create multicast socket
unable to bind socket
```

Check the network configuration of the node, ensuring that the private network is working and the Windows node can at least reach the metadata server by using the ping command from a command shell.

Device Busy Message

You may see the following error message repeatedly on a node when you stop services on another node until the shutdown completes:

```
Nov  4 15:35:12 ray : Nov 04 15:35:12 cxfs_client:
cis_cms_exclude_cell ERROR: exclude cellset ffffffff00 failed: Device busy
```

After the other node completes shutdown, the error will cease to be sent. However, if the error message continues to appear even after shutdown is complete, another problem may be present. In this case, contact your SGI support person.

Windows Messages

The following are common Windows CXFS messages.

```
cis_driver_init() failed: could not open handle to driver
cis_driver_init() failed: could not close handle to CXFS driver
```

The CXFS driver may not have successfully started. Check the system event log for errors.

```
cis_generate_userid_map warning: could not open group file
```

The group file could not be found.

Even with `passwd` and `group` warnings above, filesystem mounts should proceed; however, all users will be given `nobody` credentials and will be unable to view or modify files on the CXFS filesystems. For more information about these files, see "Log Files and Cluster Status for Windows" on page 94. Also see the log files on the server-capable administration node; for more information, see the *CXFS 6 Administration Guide for SGI InfiniteStorage*.

```
cis_generate_userid_map warning: could not open passwd file
```

The passwd file could not be found.

```
could not get location of passwd/group files
could not retrieving fencing configuration file name from registry
error retrieving passwd filename
error retrieving group filename
error retrieving fencing filename
```

The registry entries for the location of the `passwd`, `group`, or `fencing.conf` files may be missing, or the path provided on the command line to the CXFS Client service is badly formed. Reset these values by modifying the current installation as described in "Modifying the CXFS Software for Windows" on page 147.

could not open passwd file

could not open group file

fencing configuration file not found

Check that the passwd, group and fencing.conf files are in the configured location and are accessible as described in "Checking Permissions on the Password and Group Files for Windows" on page 141.

no valid users configured in passwd file

No users in the passwd file could be matched to users on the Windows node. All users will be treated as user nobody for the purpose of all access control checks.

no valid groups configured in group file

No groups in the group file could be matched to groups on the Windows node. Attempts to display file permissions will most likely fail with the message Unknown Group Errors.

op_failed ERROR: Mount failed for concat0

A filesystem mount has failed and will be retried.

unable to create mount point

Configured drive letter may already be in use

Check that the configured drive letter is not already in use by a physical or mapped drive.

Unix user is something other than a user on the NT domain/workgroup

Unix group is something other than a group on the NT domain/workgroup

This warning indicates that a username or groupname is not a valid user or group on the Windows node, which may be confusing when examining file permissions.

Glossary

ACL

Access control list.

active metadata server

A server-capable administration node chosen from the list of potential metadata servers. There can be only one active metadata server for any one filesystem. See also *metadata*.

administration node

See *server-capable administration node*.

administrative stop

See *forced CXFS shutdown*.

advanced mode

The `cxfs_admin` complexity mode that provides a list of possible choices when using the <TAB> key, prompts for all possible fields, displays all attributes, and includes debugging information in output.

ARP

Address resolution protocol.

basic mode

The `cxfs_admin` complexity mode that only shows the common options and attributes in `show` output, provides a list of possible choices when using the <TAB> key, and uses prompting.

bandwidth

Maximum capacity for data transfer.

blacklisted

A node that is explicitly not permitted to be automatically configured into the cluster database.

BMC

Baseboard management controller.

cell ID

A number associated with a node that is allocated when a node is added into the cluster definition with the GUI or `cxfs_admin`. The first node in the cluster has cell ID of 0, and each subsequent node added gets the next available (incremental) cell ID. If a node is removed from the cluster definition, its cell ID becomes available. It is not the same thing as the *node ID*.

CLI

Underlying command-line interface commands used by the CXFS Manager graphical user interface (GUI).

client

In CXFS, a node other than the active metadata server that mounts a CXFS filesystem. A *server-capable administration node* can function as either an active metadata server or as a CXFS client, depending upon how it is configured and whether it is chosen to be the active metadata server. A *client-only node* always functions as a client.

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 CXFS metadata. Any node can be client-only node. See also *server-capable administration node*.

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. A cluster running multiple operating systems is known as a *multiOS cluster*.

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. Cluster names must be unique.

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.

cluster administration daemons

The set of daemons on a server-capable administration node that provide the cluster infrastructure: `cad`, `cmond`, `fs2d`, `crsd`.

cluster administration tools

The CXFS graphical interface (GUI) and the `cxfs_admin` command-line tools that let you configure and administer a CXFS cluster, and other tools that let you monitor the state of the cluster.

cluster administrator

The person responsible for managing and maintaining a cluster.

cluster database

The database that contains configuration information about all nodes and the cluster. The database is managed by the cluster administration daemons.

cluster database membership

The group of server-capable administration nodes in the **pool** that are accessible to cluster administration daemons and therefore are able to receive cluster database updates; this may be a subset of the nodes defined in the pool. The cluster administration daemons manage the distribution of the cluster database (CDB) across the server-capable administration nodes in the pool. (Also known as *user-space membership* and *fs2d database membership*.)

cluster domain

The XVM concept in which a filesystem applies to the entire cluster, not just to the local node. See also *local domain*.

cluster ID

A unique number within your network in the range 1 through 255. The cluster ID is used by the operating system 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. Clusters IDs must be unique.

cluster mode

One of two methods of CXFS cluster operation, *Normal* or *Experimental*. In *Normal* mode, CXFS monitors and acts upon CXFS kernel heartbeat or cluster database heartbeat failure; in *Experimental* mode, CXFS ignores heartbeat failure. *Experimental* mode allows you to use the kernel debugger (which stops heartbeat) without causing node failures. You should only use *Experimental* mode during debugging with approval from SGI support.

complexity mode

The manner in which `cxfs_admin` operates. See *basic mode* and *advanced mode*.

control messages

Messages that the cluster software sends between the cluster nodes to request operations on or distribute information about cluster nodes. Control messages, CXFS kernel heartbeat messages, CXFS metadata, and cluster database heartbeat messages are sent through a node's network interfaces that have been attached to a private network.

cluster node

A node that is defined as part of the cluster. See also *node*.

control network

See *private network*.

CXFS

Clustered XFS, a clustered filesystem for high-performance computing environments.

CXFS client daemon

The daemon (`cxfs_client`) that controls CXFS services on a client-only node.

CXFS control daemon

The daemon (`clconfd`) that controls CXFS services on a server-capable administration node.

CXFS database

See *cluster database*.

CXFS kernel membership

The group of CXFS nodes that can share filesystems in the cluster, which may be a subset of the nodes defined in a cluster. During the boot process, a node applies for CXFS kernel membership. Once accepted, the node can share the filesystems of the cluster. (Also known as *kernel-space membership*.) CXFS kernel membership differs from *cluster database membership*.

CXFS services

The enabling/disabling of a node, which changes a flag in the cluster database. This disabling/enabling does not affect the daemons involved. The daemons that control CXFS services are `clconfd` on a server-capable administration node and `cxfs_client` on a client-only node.

CXFS services start

To enable a node, which changes a flag in the cluster database, by using an administrative task in the CXFS GUI or the `cxfs_admin enable` command.

CXFS services stop

To disable a node, which changes a flag in the cluster database, by using the CXFS GUI or the `cxfs_admin disable` command. See also *forced CXFS shutdown*.

CXFS shutdown

See *forced CXFS shutdown* and *shutdown*.

CXFS tiebreaker node

A node identified as a tiebreaker for CXFS to use in the process of computing CXFS kernel membership for the cluster, when exactly half the nodes in the cluster are up and can communicate with each other. There is no default CXFS tiebreaker. SGI recommends that the tiebreaker node be a client-only node.

database

See *cluster database*.

database membership

See *cluster database membership*.

details area

The portion of the GUI window that displays details about a selected component in the view area. See also *view area*.

domain

See *cluster domain* and *local domain*.

dynamic heartbeat monitoring

Starts monitoring CXFS kernel heartbeat only when an operation is pending. Once monitoring initiates, it monitors at 1-second intervals and declares a timeout after 5 consecutive missed seconds, just like *static heartbeat monitoring*.

DVH

Disk volume header.

easy client configuration

Using the `cxfs_admin` command and the `autoconf` object to specify new client-only nodes that are allowed to be automatically configured into the cluster database.

edge serving

See *NFS edge serving*.

fail policy hierarchy

See *fail policy*.

failure policy

The set of instructions that determine what happens to a failed node; the second instruction will be followed only if the first instruction fails; the third instruction will be followed only if the first and second fail. The available actions are: *fence*, *fencerreset*, *reset*, and *shutdown*.

fence

The failure policy method that isolates a problem node so that it cannot access I/O devices, and therefore cannot corrupt data in the shared CXFS filesystem. I/O fencing can be applied to any node in the cluster (CXFS clients and metadata servers). The rest of the cluster can begin immediate recovery.

fencerreset

The failure policy method that fences the node and then, if the node is successfully fenced, performs an asynchronous system reset; recovery begins without waiting for reset acknowledgment. If used, this fail policy method should be specified first. If the fencing action fails, the reset is not performed; therefore, *reset* alone is also highly recommended for all server-capable administration nodes (unless there is a single server-capable administration node in the cluster).

fencing recovery

The process of recovery from fencing, in which the affected node automatically withdraws from the CXFS kernel membership, unmounts all filesystems that are using an I/O path via fenced HBA(s), and then rejoins the cluster.

forced CXFS shutdown

The withdrawal of a node from the CXFS kernel membership, either due to the fact that the node has failed somehow or by issuing an `admin cxfs_stop` command. This disables filesystem and cluster volume access for the node. The node remains enabled in the cluster database. See also *CXFS services stop* and *shutdown*.

fs2d database membership

See *cluster database membership*.

gratuitous ARP

ARP that broadcasts the MAC address to IP address mappings on a specified interface.

GUI

Graphical user interface. The CXFS GUI lets you set up and administer CXFS filesystems and XVM logical volumes. It also provides icons representing status and structure.

GPT

GUID partition table

heartbeat messages

Messages that cluster software sends between the nodes that indicate a node is up and running. CXFS kernel heartbeat messages, cluster database heartbeat messages, CXFS metadata, and control messages are sent through the node's network interfaces that have been attached to a private network.

heartbeat timeout

If no CXFS kernel heartbeat or cluster database heartbeat is received from a node in this period of time, the node is considered to be dead. The heartbeat timeout value must be at least 5 seconds for proper CXFS operation.

I/O fencing

See *fence*.

IPMI

Intelligent Platform Management Interface.

ISSP

SGI InfiniteStorage Software Platform, the distribution method for CXFS software.

kernel-space membership

See *CXFS kernel membership*.

LAN

Local area network.

local domain

XVM concept in which a filesystem applies only to the local node, not to the cluster. See also *cluster domain*.

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 `crsd-nodeA` log only on the node `nodeA` 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.

log group

A set of one or more CXFS 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 CXFS will write into an associated log group's log file. A log level is part of the log configuration for a log group.

logical volume

A logical organization of disk storage in XVM that enables an administrator to combine underlying physical disk storage into a single unit. Logical volumes behave like standard disk partitions. A logical volume allows a filesystem or raw device to be larger than the size of a physical disk. Using logical volumes can also increase disk

I/O performance because a volume can be striped across more than one disk. Logical volumes can also be used to mirror data on different disks. For more information, see the *XVM Volume Manager Administrator's Guide*.

LUN

Logical unit. A logical disk provided by a RAID. A logical unit number (LUN) is a representation of disk space. In a RAID, the disks are not individually visible because they are behind the RAID controller. The RAID controller will divide up the total disk space into multiple LUNs. The operating system sees a LUN as a hard disk. A LUN is what XVM uses as its physical volume (*physvol*). For more information, see the *XVM Volume Manager Administrator's Guide*.

membership

See *cluster database membership* and *CXFS kernel membership*.

membership version

A number associated with a node's cell ID that indicates the number of times the CXFS kernel membership has changed since a node joined the membership.

metadata

Information that describes a file, such as the file's name, size, location, and permissions.

metadata server

The server-capable administration node that coordinates the updating of metadata on behalf of all nodes in a cluster. There can be multiple potential metadata servers, but only one is chosen to be the active metadata server for any one filesystem.

metadata server recovery

The process by which the metadata server moves from one node to another due to an interruption in CXFS services on the first node. See also *recovery*.

multiOS cluster

A cluster that is running multiple operating systems, such Linux and Windows.

multiport serial adapter cable

A device that provides four DB9 serial ports from a 36-pin connector.

NFS edge serving

A configuration in which CXFS client nodes can export data with NFS.

node

A *node* is an operating system (OS) image, usually an individual computer. (This is different from the NUMA definition for a brick/blade on the end of a NUMALink cable.)

A given node can be a member of only one pool and only one cluster. See also *client-only node*, *server-capable administration node*, and *standby node*.

node ID

An integer in the range 1 through 32767 that is unique among the nodes defined in the pool. You must not change the node ID number after the node has been defined. It differs from *cell ID*.

node membership

The list of nodes that are active (have CXFS kernel membership) in a cluster.

notification command

The command used to notify the cluster administrator of changes or failures in the cluster and nodes. The command must exist on every node in the cluster.

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 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.

peer-to-disk

A model of data access in which the shared files are treated as local files by all of the hosts in the cluster. Each host can read and write the disks at near-local disk speeds; the data passes directly from the disks to the host requesting the I/O, without passing through a data server or over a LAN. For the data path, each host is a peer on the SAN; each can have equally fast direct data paths to the shared disks.

physvol

Physical volume. A disk that has been labeled for use by XVM. For more information, see the *XVM Volume Manager Administrator's Guide*.

pool

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. 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.)

potential metadata server

A server-capable administration node that is listed in the metadata server list when defining a filesystem; only one node in the list will be chosen as the active metadata server.

private network

A network that is dedicated to CXFS kernel heartbeat messages, cluster database heartbeat messages, CXFS metadata, and control messages. The private network is accessible by administrators but not by users. Also known as *control network*.

quorum

The number of nodes required to form a cluster, which differs according to membership:

- For CXFS kernel membership:
 - A majority (>50%) of the server-capable administration nodes in the cluster are required to **form** an initial membership
 - Half (50%) of the server-capable administration nodes in the cluster are required to **maintain** an existing membership
- For cluster database membership, 50% of the **nodes in the pool** are required to form and maintain a cluster.

quorum master

The node that is chosen to propagate the cluster database to the other server-capable administration nodes in the pool.

RAID

Redundant array of independent disks.

recovery

The process by which a node is removed from the CXFS kernel membership due to an interruption in CXFS services. It is during this process that the remaining nodes in the CXFS kernel membership resolve their state for cluster resources owned or shared with the removed node. See also *metadata server recovery*.

relocation

The process by which the metadata server moves from one node to another due to an administrative action; other services on the first node are not interrupted.

reset

The failure policy method that performs a system reset via the system controller.

SAN

Storage area network. A high-speed, scalable network of servers and storage devices that provides storage resource consolidation, enhanced data access, and centralized storage management.

server-capable administration node

A node that is installed with the `cluster_admin` product and is also capable of coordinating CXFS metadata.

server-side licensing

Licensing that uses license keys on the CXFS server-capable administration nodes; it does not require node-locked license keys on CXFS client-only nodes. The license keys are node-locked to each server-capable administration node and specify the number and size of client-only nodes that may join the cluster membership. All nodes require server-side licensing.

shutdown

The fail policy that tells the other nodes in the cluster to wait before reforming the CXFS kernel membership. The surviving cluster delays the beginning of recovery to allow the node time to complete the shutdown. See also *forced CXFS shutdown*.

split cluster

A situation in which cluster membership divides into two clusters due to an event (such as a network partition or an unresponsive server-capable administration node) and the lack of reset or CXFS tiebreaker capability. This results in multiple clusters, each claiming ownership of the same filesystems, which can result in filesystem data corruption. Also known as *split-brain syndrome*.

snooping

A security breach involving illicit viewing.

split-brain syndrome

See *split cluster*.

spoofing

A security breach in which one machine on the network masquerades as another.

standby node

A server-capable administration node that is configured as a potential metadata server for a given filesystem, but does not currently run any applications that will use that filesystem.

static heartbeat monitoring

Monitors CXFS kernel heartbeat constantly at 1-second intervals and declares a timeout after 5 consecutive missed seconds (default). See also *dynamic heartbeat monitoring*.

storage area network

See *SAN*.

system controller port

A port sitting on a node that provides a way to power-cycle the node remotely. Enabling or disabling a system controller port in the cluster database tells CXFS whether it can perform operations on the system controller port.

system log file

Log files in which system messages are stored.

tiebreaker node

See *CXFS tiebreaker node*.

transaction rates

I/O per second.

user-space membership

See *cluster database membership*.

view area

The portion of the GUI window that displays components graphically. See also *details area*.

VLAN

Virtual local area network.

whitelisted

A node that is explicitly allowed to be automatically configured into the cluster database.

XFS

A filesystem implementation type for the Linux operating system. It defines the format that is used to store data on disks managed by the filesystem.

Index

32-bit kernel, 83
64-bit kernel, 83
100baseT TCP/IP network, 6

A

ACLs
 Linux, 35
 Mac OS X, 64
 Windows, 110, 119
Active Directory user ID mapping method, 135
address space, 83
admin account, 17
administration best practices, 19
administrative tasks, 5
Apple HBA installation, 73
Apple HBA port configuration, 74
application crash dumps, 172
appropriate use of CXFS, 13
attr2, 210

B

backup private network, 16
backups, 21
bandwidth, 12, 13
best practices
 administration tasks, 19
 configuration tasks, 11
BIOS version, 93
biosize, 210
boot.lvm, 24
buffered I/O, 14

C

case-insensitive filesystems on Linux, 34
cdb error, 195
cis_client_run, 195
client processes, 5
client software installation
 Linux, 41
 Mac OS X, 77
 Windows, 132
client-only commands, 3
client-only installation overview, 3
client-only node
 add to the cluster, 177
 added to cluster, 177
 advantage, 16
 configuration, 176
 define the node, 176
 define the switch, 178
 modify the cluster, 177
 mount filesystems, 180
 permit fencing, 176
 platforms, 2
 start CXFS services, 180
 verify the cluster, 183
client_timeout, 210
clients cannot join the cluster, 200
cluster
 configuration, 175
 verification, 183
cluster administration, 5
CMS, 195
cms error messages, 213
Command Tag Queueing (CTQ), 128
commands installed
 Linux, 30
 Mac OS X, 60

- Windows, 94
- common problems, 195
- concatenated slice limit, 208
- concepts, 1
- configuration best practices, 11
- configuration verification, 182, 192
- connectivity in a multicast environment, 182
- could not start error, 213
- CPU types for Linux, 29
- crash dumps
 - Windows, 172
- cron jobs, 21
- crontab, 22
- CXFS Client log color meanings, 100
- CXFS Client service command line arguments, 135
- CXFS GUI, 175
- CXFS Info icon color meanings, 101
- CXFS software removal on Windows, 151
- CXFS startup/shutdown
 - Linux, 45
 - Mac OS X, 81
 - Windows, 146
- cxfs-config, 192
- cxfs-enumerate-wwns, 32
- cxfs-reprobe, 31, 199
- cxfs-reprobe and RHEL, 50
- cxfs_admin, 175
- cxfs_client, 3
 - daemon is not started
 - Linux, 53
 - Mac OS X, 86
- cxfs_client.options, 25
- cxfs_config, 182, 186, 187
- cxfs_info, 3, 94
 - state information, 185
- cxfsdp, 23
- cxfsdump, 3

D

- data integrity, 17

- define a client-only node, 176
- devfs, 53
- device block size, 208
- device busy message, 214
- devices are unknown, 199
- direct-access I/O, 12
- Directory Name Lookup Cache (DNLC), 125
- DiskManager, 102
- disks marked offline, 102
- display LUNs for QLogic HBA, 129
- distributed applications, 13
- dmapi, 210
- dmi, 210
- dmi mount option
 - Linux, 55
- DNLC size, 125
- DNS
 - Linux, 38
 - Mac OS X, 75
- DOS command shell, 130

E

- edge serving, 14
- enhanced XFS, 34
- Entitlement Sheet, 6
- error messages, 193, 213
 - /etc/hosts, 37, 63, 75
 - /etc/nsswitch.conf file, 14
- examples
 - CXFS software installation
 - Linux, 42
 - Windows, 134
 - define a switch, 178
 - /etc/hosts file
 - Linux, 38
 - /etc/inet/hosts file
 - Linux, 38
 - ifconfig
 - Linux, 38, 41

- Mac OS X, 76
- modifying the CXFS software
 - Windows, 147
- name services
 - Linux, 38
- ping
 - Linux, 40
 - Mac OS X, 76
- private network interface test
 - Linux, 40
 - Mac OS X, 76
- start CXFS services, 180
- verify the cluster configuration, 183
- Windows Client service command line options, 148

F

- failover v2, 8
- failover2.conf for Windows, 153
- failure on reboot, 167
- fast copying, 23
- fencing
 - data integrity protection, 17
- fencing verification, 198
- fencing.conf file, 44, 79, 142
- Fibre Channel HBA
 - See "host bus adapter", 35
- Fibre Channel HBA driver on Linux, 33
- Fibre Channel utility, 73
- file size and CXFS, 13
- file size/offset maximum, 208
- filestreams, 210
- filesystem access, 193
- filesystem does not mount
 - Windows, 166
- filesystem fullness, 24
- filesystem hang, 196
- filesystem mounting, 192
- filesystem network access, 12
- filesystem repair, 22

- filesystem specifications, 208
- filesystem unmounting, 181
- filesystems and logical unit specifications, 208
- filesystems do not mount
 - Linux, 54
- find, 22
- find and crontab, 22
- Finder, 62
- firewalls, 19, 199
- forced unmount, 18, 181
- free disk space required, 92

G

- G5 Xserve, 60
- gqnoenforce, 210
- gquota, 210
- GRIO, 7
 - Linux, 51
 - Mac OS X, 84
 - Windows, 152
- grioadmin, 8
- griooqs, 8
- group quotas, 5
- grpquota, 210
- grpquota, 210
- Guaranteed-rate I/O
 - See "GRIO", 7
- guided configuration, 175

H

- hafence, 198
- hardware requirements
 - all platforms, 6
 - Linux, 28
 - Mac OS X, 60
 - Windows, 91
- HBA

- Linux, 28, 35
- Mac OS X, 73
- Windows, 92, 129

HBA installation, 129

HBA WWPNs not detected, 199

hibernation, 169

host bus adapter

- See "HBA", 129

hostname resolution rules, 14

hung filesystem, 196

I

I/O fencing

- Mac OS X, 79
- See "fencing", 17
- Windows, 142

I/O fencing verification, 186

I/O operations, 12

identifying problems, 193

ifconfig

- Linux, 38, 41
- Mac OS X, 76

ifconfig errors, 193

initial setup services, 1

inode64, 210

Intel Pentium processor, 92

internode communication, 14

introduction, 1

IP address, changing, 14

ipconfig, 130

J

JBOD, 6

K

kdb, 57

kernel and extensions, 83

kernel stack size for RHEL 5 x86-64, 33

Knowledgebase, 201

L

large files, 12

large log files

- Linux, 55
- Mac OS X, 86

largeio, 210

LDAP generic user ID mapping method, 136

Leopard, 60

/Library/StartupItems/cxfs/cxfs, 61

license key, 7

- obtaining, 7

licenses for XVM mirrors, 200, 194

licensing, 6

limit client accounts, 24

Linux

- ACLs, 35
- client software installation, 41
- commands installed by CXFS, 30
- common problems, 53
- cxfs-reprobe, 50
- cxfs_client daemon is not started, 53
- cxfs_client.options, 47
- device filesystem enabled, 53
- dmi mount option, 34, 55
- filesystem do not mount, 54
- GRIO, 51
- GUI connectivity, 39
- HBA installation, 35
- I/O fencing, 44
- identifying problems, 191
- ifconfig, 41
- large log files, 55
- limitations, 33
- log files, 30
- maintenance, 47

- manual CXFS startup/shutdown, 45
 - preinstallation steps, 36
 - private network, 36
 - private network verification, 40
 - problem reporting, 56
 - recognizing storage changes, 48
 - requirements, 28
 - space requirements, 42
 - start/stop `cxfs_client`, 45
 - troubleshooting, 53
 - xfs off output, 55
 - XVM failover v2, 52
 - local XVM, 24
 - locate, 22
 - log files
 - Linux, 30
 - Mac OS X, 61
 - Windows, 95, 167
 - logbsize, 210
 - logbufs, 210
 - logdev, 210
 - ls, 22
 - LSI drivers, 19
 - LUN limit, 208
 - LUN masking, 180
- M**
- Mac OS X
 - access control lists, 64
 - client software installation, 77
 - commands installed, 60
 - common problems, 86
 - `cxfs_client` daemon not started, 86
 - Fibre Channel utility, 73
 - GRIO, 84
 - hardware platforms, 60
 - HBA, 73
 - hostname configuration, 63
 - I/O fencing, 79
 - identifying problems, 191
 - ifconfig, 76
 - large log files, 86
 - limitations and considerations, 62
 - log files, 61
 - manual CXFS startup/shutdown, 81
 - modifying CXFS software, 82
 - multiple Apple HBA ports, 74
 - point-to-point fabric setting, 74
 - power-save mode disabling, 77
 - preinstallation steps, 75
 - private network, 75, 76
 - problem reporting, 87
 - removing CXFS software, 83
 - requirements, 60
 - software maintenance, 82
 - troubleshooting, 85
 - UID and GID mapping, 64
 - upgrading CXFS software, 82
 - user and group identifiers, 63
 - XVM failover V2, 85
 - XVM volume name is too long, 86
 - maintenance and CXFS services, 22
 - manual CXFS startup/shutdown
 - Linux, 45
 - Windows, 146
 - mapping XVM volumes
 - Windows, 158
 - md driver and SGI Altix systems, 33
 - membership, 192
 - membership problem, 195
 - membership problems and firewalls, 199
 - memory-mapped shared files, 13
 - memory-mapping large files
 - Windows, 106
 - messages, 213
 - metadata, 12, 13
 - metadata server, 4
 - mirror licenses, 194, 200
 - mirroring feature and license key, 7
 - monitoring CXFS, 9
 - mount failed, 214

- mount filesystems, 180
- mount messages, 214
- mount options support, 209
- mount scripts, 31
 - Windows, 106
- mounting of filesystems, 192
- mrquota, 211
- multicast environment, 182
- multiOS cluster, 1
- multiple private networks and errors, 193
- multiple-cluster site, 25

N

- name restrictions, 14
- nested mounting on Linux, 33
- netstat, 194
- network
 - interface configuration, 14
 - requirements, 6
- network configuration rules, 14
- network connectivity messages, 214
- network issues, 15
- network size, 17
- network status, 194
- NFS edge serving, 14
- NFS fileserving network and private network, 16
- NIS
 - Linux, 38
- NO_MORE_SYSTEM_PTES, 168
- no_sendfile, 34
- noalign, 211
- noatime, 211
- noattr2, 211
- noauto, 211
- nobarrier, 211
- node membership, 192
- node shutdown, 21
- nodev, 211
- nolargeio, 211
- noquota, 211

- Norton Ghost, 106
- nosuid, 211

O

- O2, 6
- O_EXCL, 33
- oplocks and Windows, 106
- opportunistic locking and Windows, 106
- osyncidsync, 211

P

- partitioned system licensing, 6
- passwd and group files user ID mapping
 - method, 112
- path differences, 203
- pcp package, 44
- pcp-sgi package, 44
- Performance Co-Pilot (PCP), 44
- performance considerations, 13
- physical device names and XVM physvols, 23
- physical LUN limit, 208
- ping, 40, 76
- platform-specific limitations, 21
- point-to-point fabric setting for Apple HBAs, 74
- POSIX ACLs and Mac OS X ACLs, 65
- postinstallation steps
 - Windows, 140
- postmount scripts, 31
- Power Mac, 60
- power management software, 23
- power-save mode for Mac OS X, 77
- pqnoenforce, 211
- pquota, 211
- preinstallation steps
 - Linux, 36
 - Mac OS X, 75
 - Windows, 130

premount scripts, 31
 primary hostname
 Windows, 130
 private network, 15
 heartbeat and control, 14
 interface test
 Linux, 40
 Mac OS X, 76
 Linux, 37
 Mac OS X, 75
 required, 6
 prjquota, 211
 problem reporting
 Linux, 56
 Mac OS X, 87
 Windows, 171
 %ProgramFiles%\CXFS directory , 132
 %ProgramFiles%\CXFS\log\cxfs_client.log
 file, 167
 protect data integrity, 17

Q

QLogic HBA and Windows I/O size issues, 127
 QLogic HBA model numbers and driver
 versions, 92
 qnoenforce, 211
 quota, 211
 quotas, 5

R

RDAC, 62
 relocation error, 200
 remove CXFS software
 Windows, 151
 removing CXFS software
 Mac OS X, 83
 reporting problems to SGI, 201
 requirements

all platforms, 6
 Linux, 28
 Mac OS X, 60
 Windows, 91
 reset, 28
 restart Windows node, 181
 rfind, 22
 RHEL 5 x86-64 nodes kernel stack size, 33
 ro, 211
 rtdev, 211
 rw, 211

S

Samba fileserving network and private network, 16
 /sbin/fibreconfig, 74
 scripts on Linux
 cxfs-enumerate-wwns, 32
 cxfs-reprobe, 31
 mount scripts, 31
 SELinux, 33
 server_list, 212
 server_timeout, 212
 service cxfs_client, 45
 setup services, 1
 SGI Knowledgebase, 201
 SGI/AVT, 62
 Silicon Graphics O2, 6
 small files, 13
 Snow Leopard, 60
 software maintenance
 Linux, 47
 Mac OS X, 82
 Windows, 147
 software release mix, 16
 software requirements
 all platforms, 6
 Linux, 28
 Mac OS X, 60
 Windows, 91

- software upgrades, 20
 - Mac OS X, 82
 - Windows, 149
- space requirements
 - Linux, 42
- Spotlight, 62
- spurious errors, 193
- start
 - CXFS Client service
 - Windows, 146
 - CXFS processes
 - Mac OS X, 81
 - CXFS services, 146, 180
 - cxfs_client
 - Linux, 45
- Start menu differences, 90
- startup/shutdown of CXFS
 - Mac OS X, 81
- stop CXFS Client service
 - Windows, 146
- stop CXFS processes
 - Mac OS X, 81
- stop cxfs_client
 - Linux, 45
- storage changes on Mac OS X, 83
- subnet, 15
- sunit, 212
- Supportfolio, 201
- swalloc, 212
- swidth, 212
- switch definition, 178
- switched network, 17
- switchshow, 80, 145
- sysctl, 57
- system device location problems, 53
- system_profiler, 83

T

- TaskManager, 173
- TCP/IP network requirements, 6

- telnet port
 - fencing and, 17
- tiebreaker
 - client-only, 17
- Time Machine, 62
- troubleshooting
 - general, 191
 - Linux, 53
 - Mac OS X, 85
 - Windows, 160
 - xfs_repair appropriate use, 22

U

- unknown devices, 199
- unmounting filesystems, 181
- upgrade CXFS software
 - Mac OS X, 82
 - Windows, 149
- upgrades, 20
- uqnoenforce, 212
- uquota, 212
- user administration, 5
- User ID mapping methods, 112
 - Active Directory, 135
 - Generic LDAP, 136
- user mapping problems on Windows, 165
- user quotas, 5
- usrquota, 212

V

- /var/cluster/cxfs_client-scripts/cxfs-enumerate-wwns, 32
- /var/cluster/cxfs_client-scripts/cxfs-reprobe, 31
- /var/log/cxfs_client, 30, 62
- /var/log/cxfs_inst.log, 62
- verify
 - cluster, 183

verify the cluster configuration, 182
 verify the configuration, 192
 volume access, 187

W

Windows

access-denied error, 163
 ACLs, 110, 119
 cannot create file under drive letter, 168
 client service does not start, 165
 client software installation steps, 132
 common problems, 160
 crash dumps, 172
 CTQ, 128
 CXFS commands installed, 94
 CXFS from a UNIX perspective, 103
 CXFS from a Windows perspective, 104
 CXFS Info window, 96
 CXFS software removal, 151
 DDN RAID, 107
 debugging information, 172
 default umask, 124
 delayed-write error, 164
 DMA size, 124
 DNLC size, 125
 downgrading CXFS software, 151
 effective access, 120
 enforcing access to files and directories, 115
 failure on reboot, 167
 file attributes, 116
 file mounting problems, 161
 file not found error, 169
 file permissions, 117
 filesystem limitations, 106
 filesystems not displayed, 166
 firewall for Windows, 131
 forced unmount, 105
 GRIO, 152
 HBA installation, 129
 hibernation, 169
 I/O fencing, 142
 I/O size issues, 127
 identifying problems, 191
 ipconfig, 130
 large log files, 167
 Limitations, 101
 log files, 95
 LUN 0, 105
 LUNs, 129
 mandatory locks, 126
 manual CXFS startup/shutdown, 146
 mapping XVM volumes, 158
 membership problems, 170
 memory configuration, 168
 memory-mapping coherency, 125
 memory-mapping large files, 106
 message verbosity, 95
 modify the CXFS software, 147
 mount scripts, 106
 network and CXFS drives, 106
 NO_MORE_SYSTEM_PTES, 168
 Norton Ghost, 106
 NTFS, 164
 passwd and group files permissions, 141
 performance considerations, 109
 postinstallation steps, 140
 preinstallation steps, 130
 private network, 130
 problem reporting, 171
 registry modification, 123
 requirements, 6, 91
 restarting the node, 181
 slow installation, 170
 software maintenance, 147
 software upgrades, 149
 system tunables, 123
 Time Service default synchronization, 108
 troubleshooting, 160
 unable to cd, 170
 user account control, 107
 user configuration, 141

- user identification, 111
- user identification map updates, 127
- user mapping problems, 165
- verify CXFS, 161
- verify networks, 130
- WWPN for Brocade switch, 145
- WWPN for QLogic switch, 143
- XFS filesystem limitations, 107
- XVM failover v2, 153
- Windows 7 problems, 169
- windows messages, 215
- Windows Server 2008 problems, 169
- Windows Vista problems, 169
- worldwide name, 24
- worldwide port name, 44, 79, 142
 - Linux, 44, 199
 - Mac OS X, 79
 - Windows, 142
- wsync, 212
- WWN, 24
- WWPN, 44, 79, 142
 - Linux, 44, 199
 - Mac OS X, 79
 - Windows, 142
- WWPNs not detected, 199

X

- xfs off output
 - Linux, 55
- XFS version 1 directory format on Linux, 33
- xfs_repair, 22
- xfs_repair appropriate use, 22
- Xserve, 60
- xvm, 3
- XVM failover, 8
- XVM failover V2
 - Mac OS X, 85
- XVM failover v2
 - Linux, 52
 - Windows, 153
- XVM in local mode, 24
- XVM mirror licenses, 194, 200
- XVM mirroring license key, 7
- XVM physvols and physical device names, 23
- xvm show, 189, 194
- XVM volume access, 187
- XVM volume name is too long, 86
- XVM volume name size on Mac OS X, 62