CXFS™ 5 Client-Only Guide for SGI® InfiniteStorage

007–4507–019

New Features in this Guide

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

This update includes support for running the following on client-only nodes:

- SLES 11
- Windows Vista Service Pack 2
- Windows Server 2008 Service Pack 2

Some caveats and considerations that were formerly listed in the CXFS general release note have been incorporated into this guide.

Record of Revision

Version	Description
001	March 2002 Original publication with the CXFS MultiOS Clients 2.0 release for IRIX 6.5.16f.
002	May 2002 Revised to support the CXFS MultiOS Clients 2.1 release for IRIX 6.5.16f. This release supports the Sun Microsystems Solaris and Microsoft Windows NT platforms.
003	June 2002 Revised to support the CXFS MultiOS Clients 2.1.1 release for IRIX 6.5.16f. This release supports the Sun Microsystems Solaris and Microsoft Windows NT platforms.
004	August 2002 Revised to support the CXFS MultiOS 2.2 Clients release for IRIX 6.5.17f. This release supports the Sun Microsystems Solaris, Microsoft Windows NT, and Microsoft Windows 2000 platforms.
005	November 2002 Revised to support the CXFS MultiOS Clients 2.3 release for IRIX 6.5.18f. This release supports the Sun Microsystems Solaris, Microsoft Windows NT, and Microsoft Windows 2000 platforms.
006	February 2003 Revised to support the CXFS MultiOS Clients 2.4 release for IRIX 6.5.19f. This release supports the Sun Microsystems Solaris, Microsoft Windows NT, and Microsoft Windows 2000 platforms.
007	May 2003 Revised to support the CXFS MultiOS Clients 2.5 release for IRIX 6.5.20f. This release supports the IBM AIX platform, Linux on supported 32-bit platforms, SGI ProPack for Linux on SGI Altix 3000 family of servers and superclusters, Sun Microsystems Solaris platform, Microsoft Windows NT platform, and Microsoft Windows 2000 platform.

008	September 2003 Revised to support CXFS MultiOS Clients 3.0. This release supports the IBM AIX platform, Linux on supported 32–bit platforms, Sun Microsystems Solaris platform, Microsoft Windows NT platform, Microsoft Windows 2000 platform, and Microsoft Windows XP platform. The documentation for Linux 64–bit nodes supported by the CXFS 3.0 for SGI ProPack release will appear in the next version of the *CXFS 5 Administration Guide for SGI InfiniteStorage*.
009	February 2004 Revised to support CXFS MultiOS Clients 3.1. This release supports the Apple Mac OS X platform, IBM AIX platform, Linux on supported 32–bit platforms, Sun Microsystems Solaris platform, Microsoft Windows 2000 platform, and Microsoft Windows XP platform.
010	November 2004 Revised to support CXFS MultiOS Clients 3.2. This release supports the Apple Mac OS X platform, IBM AIX platform, Linux on supported 32–bit platforms, Sun Microsystems Solaris platform, Microsoft Windows 2000 platform, and Microsoft Windows XP platform.
011	April 2005 Revised to support CXFS MultiOS Clients 3.3. This release supports the Apple Mac OS X platform, IBM AIX platform, Linux on supported third-party platforms (x86, AMD64/EM64T, Intel Itanium 2), Sun Microsystems Solaris platform, Microsoft Windows 2000 platform, Microsoft Windows Server 2003, and Microsoft Windows XP platform.
012	July 2005 Revised to support CXFS MultiOS Clients 3.4. This release supports the Apple Mac OS X platform, IBM AIX platform, Linux on supported third-party platforms (x86, AMD64/EM64T, Intel Itanium 2), Sun Microsystems Solaris platform, Microsoft Windows 2000 platform, Microsoft Windows Server 2003, and Microsoft Windows XP platform.
013	May 2006 Supports CXFS 4.0

014	January 2007 Supports CXFS 4.1
015	September 2007 Supports CXFS 4.2
016	March 2008 Supports CXFS 5.0
017	September 2008 Supports CXFS 5.2
018	March 2009 Supports CXFS 5.4 in ISSP 1.6
019	September 2009 Supports CXFS 5.6 in ISSP 1.8

Contents

Figures

Tables

About This Guide

This publication documents the CXFS 5.6 release. For additional details, see the platform-specific release notes.

Prerequisites

This guide assumes the following:

- Server-capable administration nodes running SGI Foundation Software and CXFS software are installed and 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 5 Administration Guide for SGI InfiniteStorage* and the platform's operating system and installation documentation.

Related Publications

For information about this release, see the SGI InfiniteStorage Software Platform (ISSP) release notes (`/README.txt`) and the CXFS release notes (`README_`*`NAME`*`.txt`).

The following documents contain additional information (if you are viewing this document online, you can click on `TPL Link` below to link to the book on the SGI TechPubs library):

- CXFS documentation:
  - Platform-specific release notes
  - *CXFS 5 Administration Guide for SGI InfiniteStorage* (`TPL link`)
  - *SGI InfiniteStorage High Availability Using Linux-HA Heartbeat*
- QLogic HBA card and driver documentation. See the QLogic website at:

  http://www.qlogic.com

- AIX documentation on the IBM website at:

  http://www.ibm.com

- IRIX documentation:
  - *IRIX 6.5 Installation Instructions*
  - *IRIX Admin: Disks and Filesystems*
  - *IRIX Admin: Networking and Mail*
  - *Personal System Administration Guide*
  - *Performance Co-Pilot for IRIX Advanced User's and Administrator's Guide*
  - *Performance Co-Pilot Programmer's Guide*
- Linux documentation:
  - Red Hat:

    http://www.redhat.com/docs/manuals/enterprise/
  - SLES:

    http://www.novell.com/documentation/sles10/index.html
- Mac OS X software documentation:
  - *Welcome to Mac OS X*
  - *Mac OS X Server Administrator's Guide*
  - *Understanding and Using NetInfo*

  See the Apple website at:

  http://www.apple.com
- Solaris documentation:
  - *Solaris 10 Installation Guide*
  - *Solaris 10 System Administration Collection*

  See the Sun Microsystems website at:

  http://www.sun.com

- Sun Microsystems owner's guide and product notes for the Sun hardware platform
- Windows software documentation: see the Microsoft website at:

  http://www.microsoft.com
- Hardware documentation for the Intel platform

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 client-only nodes (other than Windows nodes):

Client-Only Man Page	Linux RPM or IRIX Subsystem[1]
`cxfs_client`(1M)	`cxfs_client`
`cxfs_info`(1M)	`cxfs_client`
`cxfs-config`(1M)	`cxfs_util`
`cxfscp`(1)	`cxfs_util`
`cxfsdump`(1M)	`cxfs_util`

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.

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

- 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
- 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 systems running Red Hat Enterprise Linux (RHEL) or SUSE Linux Enterprise Server (SLES)
- *Mac OS X* refers to both the Tiger and Leopard releases
- *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
command	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
variable	Italic typeface denotes variable entries and words or concepts being defined.
user input	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)

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.

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

SGI values your comments and will respond to them promptly.

Introduction

This guide provides an overview of the installation and configuration procedures for the following 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 5 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 installation services developed for CXFS. Many of the procedures mentioned in this guide will 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:

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

CXFS on Client-Only Nodes

This section contains the following:

Client-Only Platforms

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

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

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

For details, see the following:

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
`cxfs_client`(1m)	Controls the CXFS client control daemon
`cxfs_info`(1m)	Provides status information.
`cxfsdump`(1M)	Gathers configuration information in a CXFS cluster for diagnostic purposes.
`xvm`(1m)	Invokes the XVM command line interface

Also see:

- "CXFS Commands on AIX" on page 29
- "CXFS Commands on IRIX" on page 55
- "CXFS Commands on Linux" on page 83
- "CXFS Commands on Mac OS X" on page 112
- "CXFS Commands on Solaris" on page 143
- "CXFS Commands on Windows" on page 176

Client-Only Installation and Configuration Overview

Following is the order of installation and configuration steps for a CXFS client-only node. See the specific operating system (OS) chapter for details:

1. Read the CXFS release notes to learn about any late-breaking changes in the installation procedure.
2. Install the OS software according to the directions in the OS documentation (if not already done).
3. Install and verify the RAID. See the *CXFS 5 Administration Guide for SGI InfiniteStorage* and the release notes.

4. Install and verify the switch. See the *CXFS 5 Administration Guide for SGI InfiniteStorage* and the release notes.
5. Obtain the CXFS server-side license key. For more information about licensing, see "License Keys" on page 10 and *CXFS 5 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 212.
8. Install the RPMs containing the CXFS client packages onto the server-capable administration node and transfer the appropriate client packages to the corresponding client-only nodes.
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 9, "Cluster Configuration Tasks to Include Client-Only Nodes" on page 257.
11. Start CXFS services on the client-only node to see the mounted filesystems.

If you run into problems, see the OS-specific troubleshooting section, Chapter 10, "General Troubleshooting" on page 273, and the troubleshooting chapter in *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Cluster Administration

There must be at least one server-capable administration node in the cluster 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.

A server-capable administration node is required to perform administrative tasks, using the `cxfs_admin` command or the CXFS graphical user interface (GUI). For

more information about using these tools, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

CXFS Client Processes

When CXFS is started on a client-only node, a user-space daemon/service 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 287

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 and IRIX 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 `xfs_quota` command. This is provided by the `xfsprogs` RPM. On an IRIX node, use `repquota` and `edquota`. If you want to provide a viewing command on other nodes, you can construct a shell script similar to the following:

```
# ! /bin/sh
#
# Where repquota lives on IRIX
repquota=/usr/etc/repquota

# The name of an IRIX node in the cluster
irixnode=cain

rsh $irixnode "$repquota $*"
exit
```

CXFS Mount Scripts

CXFS mount scripts are provided for execution by the `cxfs_client` daemon prior to and after a CXFS filesystem is mounted or unmounted on the following platforms:

- AIX
- IRIX
- Linux
- Solaris

The CXFS mount scripts are not supported on Mac OS X or Windows.

The CXFS 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
```

The following script is run when needed to reprobe the Fibre Channel controllers on client-only nodes:

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

The CXFS mount scripts are used by CXFS to ensure that LUN path failover works after fencing. 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.

For information about using these scripts, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

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 Linux nodes, you must define a group of environment variables in the `/etc/cluster/config/cxfs_client.options` file in order for `cxfs-reprobe` to appropriately probe all of the targets on the SCSI bus. For more information, see "Using `cxfs-reprobe` on IRIX Nodes" on page 63.

On Linux nodes, the following script enumerates the world wide names (WWNs) on the host that are known to CXFS. The following example is for a Linux node with two single-port HBAs:

```
linux#  /var/cluster/cxfs_client-scripts/cxfs-enumerate-wwns
# cxfs-enumerate-wwns
# xscsi @ /dev/xscsi/pci01.01.0/bus
# xscsi @ /dev/xscsi/pci01.03.01/bus
# xscsi @ /dev/xscsi/pci01.03.02/bus
# xscsi @ /dev/xscsi/pci02.02.0/bus
210000e08b100df1
# xscsi @ /dev/xscsi/pci02.02.1/bus
210100e08b300df1
```

For more details about using these scripts, and for information about the mount scripts on server-capable administration nodes, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Requirements

Using a client-only node in a multiOS CXFS cluster 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. CXFS does **not** support the Silicon Graphics O2 workstation as a CXFS node nor does it support JBOD.

- 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 10.
- A switch, which is required to protect data integrity on nodes without system controllers. See the release notes for supported switches.

  AIX, Linux, Solaris, Mac OS X, and Windows client-only nodes must use I/O fencing to protect the data integrity of the filesystems in the cluster. Server-capable administration nodes should use serial reset lines. 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 17.
- 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 Solaris" on page 142, "Requirements for Windows" on page 173, and Chapter 2, "Best Practices for Client-Only Nodes" on page 13.

License Keys

CXFS requires the following license keys:

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

  Note: As of CXFS 4.2, all server-capable administration nodes running 4.2 and client-only nodes running 4.2 require server-side licensing. If **all** existing client-only nodes are running a prior supported release, they may continue to use client-side license as part of the rolling upgrade policy until they are upgraded to 4.2. All client-only nodes in the cluster must use the same licensing type — if any client-only node in the cluster is upgraded to 4.2 or if a new 4.2 client-only node is added, then all nodes must use server-side licensing.

  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 5 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.
- AIX using XVM failover version 2 also requires a SANshare license for storage partitioning; see "XVM Failover V2 on AIX" on page 45.

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 104.

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

- `grioadmin`, which provides stream and bandwidth management
- `grioqos`, 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 287, for the platform-specific locations of these tools.

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

- "GRIO on AIX" on page 45
- "GRIO on IRIX" on page 66
- "GRIO on Linux" on page 104
- "GRIO on Mac OS X" on page 136
- "GRIO on Solaris" on page 163
- "GRIO on Windows" on page 234

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

XVM Failover and CXFS

XVM failover version 2 (v2) requires that the RAID be configured in AVT mode. AIX also requires a SANshare license; see "XVM Failover V2 on AIX" on page 45.

To configure failover v2, you must create and edit the `failover2.conf` file. For more information, see the comments in the `failover2.conf` file on a CXFS

server-capable administration node, *CXFS 5 Administration Guide for SGI InfiniteStorage*, and the *XVM Volume Manager Administrator's Guide*.

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

- "XVM Failover V2 on AIX" on page 45
- "XVM Failover V2 on IRIX" on page 66
- "XVM Failover V2 on Linux" on page 105
- "XVM Failover V2 on Mac OS X" on page 137
- "XVM Failover V2 on Solaris" on page 163
- "XVM Failover V2 on Windows" on page 235

Monitoring CXFS

To monitor CXFS, you can use the `cxfs_info` command on the client, or view area of the CXFS GUI, the `cxfs_admin` command, or the `clconf_info` command on a CXFS server-capable administration node. For more information, see "Verifying the Cluster Status" on page 265.

Chapter 2

Best Practices for Client-Only Nodes

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

Also see the best practices information in the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Configuration Best Practices

This section discusses the following:

Use CXFS when Appropriate

CXFS may not give optimal performance under the following circumstances:

- When distributed applications write to shared files that are memory-mapped.
- Although SGI supports *NFS edge serving* (in which CXFS client nodes can export data with NFS), there are no performance guarantees. For best performance, SGI recommends that you use the active metadata server. If you require a high-performance solution, contact SGI Professional Services.
- 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.

Also see "Functional Limitations and Considerations for Windows" on page 182.

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.

Use the `cxfs-config -check -ping` command line on a server-capable administration node to confirm network connectivity. For more information, see *CXFS 5 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. Ensure that you understand the information in "Understand Hostname Resolution and Network Configuration Rules" on page 15.

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 must use a private network for CXFS metadata traffic:

- A private network is a requirement.
- 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).
- 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.
- A virtual local area network (VLAN) is not supported for a private network.
- Use private (10.*x.x.x*, 176.16.*x.x*, or 192.168.*x.x*) network addresses (RFC 1918).

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. For details, see the platform-specific release notes and the information about rolling upgrades in *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Protect Data Integrity

I/O fencing is required on client-only nodes without reset capability in order to protect the data integrity of the filesystems in the cluster.

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 5 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 AIX" on page 42
- "I/O Fencing for IRIX Nodes" on page 64
- "I/O Fencing for Linux" on page 97
- "I/O Fencing for Mac OS X" on page 132
- "I/O Fencing for Solaris" on page 159
- "I/O Fencing for Windows" on page 224

Use a Client-Only Tiebreaker

SGI recommends that you always define a client-only node as the CXFS tiebreaker. (Server-capable administration nodes are not recommended as tiebreaker nodes.) This is most important when there are an even number of server-capable administration nodes.

The tiebreaker is of benefit in a cluster with an odd number of server-capable administration nodes when one of the server-capable administration nodes is removed from the cluster for maintenance (via a stop of CXFS services).

The following rules apply:

- 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 cluster, failure of that node or stopping the CXFS services on that node will result in a cluster-wide forced shutdown. Therefore SGI recommends that you use client-only nodes as tiebreakers so that either server could fail but the cluster would remain operational via the other server.

Setting a client-only node as the tiebreaker avoids the problem of multiple-clusters being formed (also known as *split-brain syndrome*) while still allowing the cluster to continue if one of the metadata servers fails.

- Setting a server-capable administration node as tiebreaker is recommended only when there are four or more server-capable administration nodes and no client-only nodes.
- If there are an even number of servers and there is no tiebreaker set, the failure action hierarchy should not contain the `shutdown` option because there is no notification that a shutdown has occurred.

SGI recommends that you start CXFS services on the tie-breaker client after the metadata servers are all up and running, and 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 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 263 and the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Configure Firewalls for CXFS Use

Do one of the following:

- Configure firewalls to allow CXFS traffic. See *CXFS 5 Administration Guide for SGI InfiniteStorage* for CXFS port usage. (Preferred.)
- 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.

For more information, see your firewall documentation.

Administration Best Practices

This section discusses the following:

Upgrade the Software Properly

Do the following when upgrading the software:

- Read the release notes when installing and/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 AIX" on page 44
- "Updating the CXFS Software for Mac OS X" on page 135
- "Updating the CXFS Software for Solaris" on page 161
- "Updating the CXFS Software for Windows" on page 231

Understand the Platform-Specific Limitations and Considerations

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

- "Limitations and Considerations for AIX" on page 30
- "Limitations and Considerations on IRIX" on page 56
- "Limitations and Considerations for Linux" on page 85
- "Limitations and Considerations on Mac OS X" on page 114
- "Limitations and Considerations on Solaris" on page 144
- "Functional Limitations and Considerations for Windows" on page 182 and "Performance Considerations for Windows" on page 189

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 backups are done 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

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 metadata server.

On IRIX systems, the default `root crontab` on some platforms has the following `find` job that should be removed or disabled on all nodes (line breaks added here for readability):

```
    0       5       *       *       *       /sbin/suattr -m -C CAP_MAC_READ,
CAP_MAC_WRITE,CAP_DAC_WRITE,CAP_DAC_READ_SEARCH,CAP_DAC_EXECUTE=eip
-c "find / -local -type f '(' -name core -o -name dead.letter ')' -atime +7
-mtime +7 -exec rm -f '{}' ';'"
```

Repair Filesystems with Care

Do not use any filesystem defragmenter software. You can use Linux `xfs_fsr` command **only** on a metadata server for the filesystem it acts upon.

Always contact SGI technical support before using `xfs_repair` on CXFS filesystems. Only use `xfs_repair` on metadata servers and only when you have verified that all other cluster nodes have unmounted the filesystem.

When using `xfs_repair`, make sure it is run only on a cleanly unmounted filesystem. If your filesystem has not been cleanly unmounted, there will be un-committed 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

Disable CXFS before maintenance (perform a forced CXFS shutdown, stop the `cxfs_client` daemon, and disable `cxfs_client` from automatically restarting).

Running 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 `cxfscp`(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, `cxfscp` uses a separate thread using buffered I/O, similar to `cp`(1).

The `cxfscp` command is available on IRIX, 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 `cxfscp`(1) man page.

Mapping 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 235.

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

See also "Mapping Physical Device Names to Unlabeled LUNs on AIX" on page 50.

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.

AIX Platform

CXFS supports a client-only node running the IBM AIX operating system. This chapter contains the following sections:

CXFS on AIX

This section contains the following information about CXFS on AIX:

Requirements for AIX

In addition to the items listed in "Requirements" on page 9, using an AIX node to support CXFS requires the following:

- IBM AIX 5L: Version 5.3 Maintenance Level 3 (64-bit mode) APAR number IY71011 or its successor

  To verify the operating system level, use the following command:

  ```
  oslevel -r
  ```

- IBM FC5716, FC6228, or FC6239 2-Gbit Fibre Channel host bus adapters (HBAs)
- One or more of the following IBM hardware platforms:

  pSeries 570
  pSeries 575
  pSeries 595
  pSeries 610
  pSeries 620
  pSeries 630
  pSeries 640
  pSeries 650
  pSeries 660
  pSeries 670
  pSeries 680
  pSeries 690

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

CXFS Commands on AIX

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

```
/usr/cxfs_cluster/bin/cxfs_client
/usr/cxfs_cluster/bin/cxfs_cluster
/usr/cxfs_cluster/bin/cxfs_info
/usr/cxfs_cluster/bin/cxfscp
/usr/cxfs_cluster/bin/cxfsdump
/usr/cxfs_cluster/bin/grioadmin
/usr/cxfs_cluster/bin/griomon
/usr/cxfs_cluster/bin/grioqos
/usr/cxfs_cluster/bin/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 `lslpp` output lists all of the software added; see "Installing CXFS Software on AIX" on page 39.

For more information on these commands, see the man pages. For information about the GRIO commands, also "Guaranteed-Rate I/O (GRIO) and CXFS" on page 11 and "GRIO on AIX" on page 45.

Log Files on AIX

The `cxfs_client` command creates a `/var/tmp/cxfs_client` log file. To rotate this log file, use the `-z` option in the following file:

```
/usr/cxfs_cluster/bin/cxfs_client.options
```

See the `cxfs_client` man page for details.

Some daemons related to CXFS output a message in the console log. To see the contents of this log file, use the following command:

```
alog -o -t console
```

The console log is rotated.

For information about the log files created on server-capable administration nodes, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*. Also see the AIX `/etc/syslog.conf` file.

CXFS Mount Scripts on AIX

AIX supports the CXFS mount scripts. See "CXFS Mount Scripts" on page 7 and the *CXFS 5 Administration Guide for SGI InfiniteStorage.*

Limitations and Considerations for AIX

Note the following:

- Although it is possible to mount a JFS or NFS filesystem on top of an AIX CXFS filesystem, this is not recommended.
- There is no default access control list (ACL) in AIX. Therefore, the setup and display of the default ACL cannot be completed using the following commands:

  ```
  aclget
  aclput
  acledit
  ```

  If an IRIX ACL exists, the ACL becomes effective when the default ACL is set up by IRIX and a file and a directory are made under that directory in AIX.
- There is no `MASK` entry in AIX, but the access permissions in AIX follow those established when an ACL set up by Linux contains a `MASK` entry. If the default ACL is set up for a given directory and the `MASK` entry exists, then that `MASK` entry is used when a file or a subdirectory is made by AIX. When the `MASK` entry does not exist, `rwx` is used.
- ACL control of the following, which the AIX JFS filesystem has, cannot be applied to CXFS:
  - The access to a certain user or the group is rejected (`deny`)
  - When a user belongs to the specific group, access is permitted or rejected (`specify`)

  If `deny` or `specify` is used, an error occurs (`EINVAL`) because these features are not in CXFS.
- Socket files cannot be copied. The following error is returned:

  ```
  AIX:The socket does not allow the requested operation.
  ```

- You can use the `fuser` command to extract process information about the mounted filesystem, but you cannot extract process information about the file or the directory.
- The AIX node does not automatically detect the worldwide port number (WWPN). In order to use I/O fencing, you must list the WWPN in the `/etc/fencing.conf` file. See "I/O Fencing for AIX" on page 42.
- If your users want to use a file size/offset maximum greater than 1 GB, you must change their user properties to allow files of unlimited size. To do this, use the `smit` command. For more information, see the `smit` man page.
- By default, the maximum request size for direct I/O is 512 MB (524288 KB). A direct I/O request larger than 512 MB will revert to buffered I/O. However, you can change the maximum XVM direct memory access (DMA) size to improve direct I/O performance. To do this, use the `chdev` command to modify the `xvm_maxdmasz` attribute. The actual maximum limit will always be 4 KB less than any of the supplied or displayed values (for example, the default is actually 512 MB minus 4 KB).

  Note: The XVM module must be loaded if any attribute changes are to be noticed and applied.

  To display the current setting, use the following command:

  ```
  lsattr -E -l xvm -a xvm_maxdmasz
  ```

  To change the current setting, use the following command:

  ```
  chdev [-P|-T] -l xvm -a xvm_maxdmasz=NewValue
  ```

  Legal values for *NewValue* are specified in KB units in the range 256 to 2097152 (that is, 256 KB to 2 GB).

  By default, using `chdev` on a running system makes a permanent change for subsequently mounted filesystems. (Running filesystems will not be changed until they are remounted, either manually or after a reboot.)

  If you use `-P`, the change is deferred until the next boot and after that it is permanent. If you use `-T` (temporary), the change is immediate for subsequently mounted filesystems, but lasts only until the next boot.

For example, to change the DMA size to 2 GB for subsequently mounted filesystems on the currently running device and in the database, enter the following:

```
aix# chdev -l xvm -a xvm_maxdmasz=2097152
```

For more information, see the `lsattr` and `chdev` man pages.

- Due to FC controller limitations, large (> 256K) direct I/O requests may be problematic.
- If you use the `frametest` command on AIX, make sure that the `posix_aio0` device is available. Do the following:

  1. Change the setting of the `posix_aio0` device to `available`:

     ```
     aix# chdev -l posix_aio0 -a autoconfig=available
     ```

  2. Add the device to the system:

     ```
     aix# mkdev -l posix_aio0
     ```

  3. Verify that the device is available. For example:

     ```
     aix# lsdev|grep aio
     aio0       Defined                 Asynchronous I/O (Legacy)
     posix_aio0 Available               Posix Asynchronous I/O
     ```

  For more information about the AIX commands, see their man pages.

  For an overview of `frametest`, see the section about generation of streaming workload for video streams in the *CXFS 5 Administration Guide for SGI InfiniteStorage*. For details about `frametest` and its command-line options, see the `frametest`(1) man page.

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 293.

Maximum CXFS I/O Request Size and AIX

By default, the maximum CXFS I/O request size for normal filesystem I/O is 1 MB (1024 KB). However, depending on filesystem size and internal layout, the actual request size can be smaller or larger:

- Requests that are smaller than 1 MB are unaffected by the limit and proceed normally

- Requests larger than 1 MB are automatically split into multiple smaller requests in order to accommodate the limit

The `cxfs_maxiosz` attribute determines the CXFS maximum I/O size request. To display the current setting, use the `lsattr` command. For example:

```
aix# lsattr -E -l xvm -a cxfs_maxiosz
```

To change the CXFS maximum I/O request size, use the `chdev` command to modify the `cxfs_maxiosz` attribute. For example:

```
aix# chdev [-P|-T] -l xvm -a cxfs_maxiosz=NewValue
```

Note: For attribute changes to be noticed and applied, the XVM module must be loaded.

Legal values for *NewValue* are specified in KB units in the range 64 through 2048 (that is, 64 KB to 2 MB).

By default, using `chdev` on a running system makes a permanent change for subsequently mounted filesystems. (Running filesystems will not be changed until they are remounted, either manually or after a reboot.)

If you use `-P`, the change is deferred until the next boot and after that it is permanent. If you use `-T` (temporary), the change is immediate for subsequently mounted filesystems, but lasts only until the next boot.

For example, to change the CXFS maximum I/O request size to 512 KB for subsequently mounted filesystems on the currently running device `xvm` and in the database, enter the following:

```
aix# chdev -l xvm -a cxfs_maxiosz=512
```

For more information, see the `lsattr` and `chdev` man pages.

There is a possibility that CXFS I/O limits may conflict with AIX's internal disk driver limits. In such cases, you will see console error messages from CXFS that specify an illegal request size error. You can use one of the following ways to correct this problem:

- You can decrease CXFS maximum I/O size to match the limit imposed by the AIX disk driver using a procedure similar to the above. This AIX limit is per physical disk drive and is described by the AIX attribute `max_transfer`. You can display this limit with the `lsattr` command if you know the name of the physical disk

that corresponds to your XVM volume. For example, where hdisk*XX* is the subsystem name that AIX chooses for each physical disk driver it finds at boot time (the *XX* number will vary depending upon controller configuration and number of drives):

```
aix# lsattr -E -l hdiskXX -a max_transfer
max_transfer 0x40000 Maximum TRANSFER Size True
```

The hexadecimal value 0x40000 is 256 KB. From the CXFS error messages on the console, you can find the transfer size that CXFS tried to use; it will likely be hexadecimal 0x80000 (512 KB), which is too large. You can decrease the CXFS maximum I/O size to 256 KB to match AIX's max_transfer limit. This decrease may slightly decrease overall filesystem performance.

- You can increase AIX's per-physical-disk max_transfer attribute to 512 KB to match the CXFS maximum I/O request size. You must perform the following command for each physical disk that is part of the cluster configuration:

  ```
  aix# chdev -l hdiskXX -a max_transfer=0x80000
  ```

  You can verify the change by using lsattr command as described above.

After modifying AIX's disk driver limits, you must reboot the machine to allow the changes to take effect.

Access Control Lists and AIX

All CXFS files have UNIX mode bits (read, write, and execute) and optionally an ACL. For more information about POSIX ACLs, see the AIX chmod, acledit, aclget, and aclput man pages.

If you want to use an AIX node to restore a CXFS file with an ACL, you should use the backup and restore commands. If you use the tar, cpio, or pax command, the ACL will not be used because these tools behave "intelligently" by not calling acl subroutines to set an ACL. These tools will only set the file mode.

When using the ls command to display access permissions for a file with an ACL, the mode reported for a CXFS file follows Linux semantics instead of AIX JFS semantics.

The CXFS model calls for reporting the ACL MASK for the group permission in the mode. Therefore, if the GROUP entry is r-x and the MASK entry is rw-, the group permission will be reported as rw-. Although it appears that the group has write permission, it does not and an attempt to write to the file will be rejected. You can

obtain the real (that is, effective) group permission by using the AIX `aclget` command.

Note: Normally, AIX filesystem ACLs can have up to one memory page (4096 bytes) for a file and a directory. However, CXFS filesystems on AIX nodes in a multiOS cluster must maintain compatibility with the metadata server. The CXFS filesystems on an AIX node are limited to a maximum of 25 ACL entries converted to Linux ACL type for a file and a directory.

HBA Installation for AIX

For information about installing and configuring the host bus adapter (HBA), see the IBM HBA documentation.

Preinstallation Steps for AIX

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

- "Adding a Private Network for AIX" on page 35
- "Verifying the Private and Public Network for AIX" on page 38

Adding a Private Network for AIX

The following procedure provides an overview of the steps required to add a private network to the AIX system. A private network is required for use with CXFS. See "Use a Private Network" on page 16.

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

1. If your system is already operational and on the network, skip to step 2. If the AIX operating system has not been installed, install it in accordance with the AIX documentation.
2. Edit the `/etc/hosts` file so that it contains entries for every node in the cluster and their private interfaces.

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 the /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.

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 15 and the hosts, named, and nis man pages.

3. *(Optional)* Edit the /etc/netsvc.conf file so that local files are accessed before either NIS or DNS. That is, the hosts line in /etc/netsvc.conf must list local first. For example:

```
hosts = local,nis,bind
```

(The order of `nis` and `bind` is not significant to CXFS, but `local` must be first.)

4. Determine the name of the private interface by using the `ifconfig` command as follows, to list the available networks. For example:

```
# ifconfig  -l
en0 en1 lo0
```

However, if the second network interface (`en1`) does not appear, then the network interface must be set up in accordance with the AIX documentation.

You can set up an IP address by using `ifconfig` after restarting the system. If it is set up properly, the following information is output (line breaks added here for readability):

```
#  ifconfig  -a
en0:flags=4e080863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT,PSEG>
   inet 10.208.148.61 netmask 0xffffff00 broadcast 10.208.148.255
en1:flags=7e080863,10<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRPT,64BIT,
   CHECKSUM_OFFLOAD,CHECKSUM_SUPPORT,RSEG>
   inet 192.168.10.61 netmask 0xffffff00 broadcast 192.168.10.255
lo0:flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
   inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
```

To make the IP address you set permanent across reboots, use the `smit` system administration tool.

5. *(Optional)* Edit the `/.rhosts` file if you want to use remote access or if you want to use the connectivity diagnostics with CXFS. Make sure that the mode of the `.rhosts` file is set to `600` (read and write access for the owner only).

Make sure that the `/.rhosts` file on each AIX node allows all of the nodes in the cluster to have access to each other. The connectivity tests execute a `ping` command from the local node to all nodes and from all nodes to the local node. To execute `ping` on a remote node, CXFS uses `rsh` as user `root`.

For example, suppose you have a cluster with three nodes: `linux0`, `aix1`, and `aix2`. The `/.rhosts` files could be as follows (where the prompt denotes the node name):

```
linux0# cat /.rhosts
aix1 root
aix2 root
```

```
aix1# cat /.rhosts
linux0 root
aix2 root

aix2# cat /.rhosts
linux0 root
aix1 root
```

Verifying the Private and Public Network for AIX

For each private network on each AIX node in the pool, verify access with the AIX `ping` command. Enter the following, where *nodeIPaddress* is the IP address of the node:

`/usr/sbin/ping -c 3` *nodeIPaddress*

For example:

```
aix# /usr/sbin/ping -c 3 192.168.10.61
PING  192.168.10.61: (192.168.10.61): 56 data data bytes
64 bytes from 192.168.10.61  icmp_seq=0 ttl=255 time=0 ms
64 bytes from 192.168.10.61  icmp_seq=1 ttl=255 time=0 ms
64 bytes from 192.168.10.61  icmp_seq=2 ttl=255 time=0 ms
----192.168.10.61 PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0/0/00 ms
```

You should also execute a `ping` on the public networks. If that `ping` fails, follow these steps:

1. Verify that the network interface was configured up. For example:

```
aix# /usr/sbin/ifconfig en0
en0: flgs=4e08086<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT,PSEG>
  inet 10.208.148.61 netmask 0xffffff00 broadcast 10.208.148.255
```

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

2. Verify that the cables are correctly seated. Repeat this procedure on each node.

Client Software Installation for AIX

The CXFS software initially will be installed and configured by SGI personnel. This section discusses the following:

Installing CXFS Software on AIX

Installing CXFS for AIX requires approximately 20 MB of space. To install the required software on an AIX node, SGI personnel will do the following:

1. Read the *SGI InfiniteStorage Software Platform* release notes, CXFS general release notes, and CXFS AIX release notes from the `/docs` directory on the ISSP DVD and any late-breaking caveats on Supportfolio.
2. Verify that the node has been upgraded to the supported AIX version according to the AIX documentation. Use the following command to display the currently installed system:

   ```
   oslevel -r
   ```

   For example, the following output indicates AIX version 5, revision 3, maintenance level 03:

   ```
   aix# oslevel -r
   5300-03
   ```

3. 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 tarball on the server will be as follows:

```
/usr/cluster/client-dist/CXFS_VERSION/aix/53ml3/noarch/SGIcxfs-aix5L
```

4. Install the CXFS software (the example output below is truncated):

```
aix# installp -a -d SGIcxfs-aix5L all
+-----------------------------------------------------------------------------+
                    Pre-installation Verification...
+-----------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
```

```
Results...

SUCCESSES
---------
  Filesets listed in this section passed pre-installation verification
  and will be installed.

  Selected Filesets
  -----------------
  SGIcxfs-aix5L 5.6.0.5                       # CXFS CLIENT for AIX

  << End of Success Section >>

FILESET STATISTICS
------------------
    1  Selected to be installed, of which:
        1  Passed pre-installation verification
  ----
    1  Total to be installed

+-----------------------------------------------------------------------------+
                        Installing Software...
+-----------------------------------------------------------------------------+

installp:  APPLYING software for:
        SGIcxfs-aix5L 5.6.0.5


. . . . . << Copyright notice for SGIcxfs-aix5L >> . . . . . .
...

Finished processing all filesets.  (Total time:  4 secs).

+-----------------------------------------------------------------------------+
                                Summaries:
+-----------------------------------------------------------------------------+

Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
```

```
SGIcxfs-aix5L               5.6.0.5         USR          APPLY       SUCCESS
SGIcxfs-aix5L               5.6.0.5        ROOT         APPLY       SUCCESS
```

5. Reboot to start CXFS services automatically.

Verifying the AIX Installation

To verify that the CXFS software has been installed properly, use the `lslpp` command as follows:

```
aix# lslpp -L SGIcxfs-aix5L
```

For example, the following output (showing a state of `C`, for "committed") indicates that the CXFS package installed properly:

```
aix# lslpp -L SGIcxfs-aix5L
Fileset                      Level  State  Type  Description (Uninstaller)
  ----------------------------------------------------------------------------
  SGIcxfs-aix5L              5.6.0.5   C     F    CXFS CLIENT for AIX


State codes:
 A -- Applied.
 B -- Broken.
 C -- Committed.
 E -- EFIX Locked.
 O -- Obsolete.  (partially migrated to newer version)
 ? -- Inconsistent State...Run lppchk -v.

Type codes:
 F -- Installp Fileset
 P -- Product
 C -- Component
 T -- Feature
 R -- RPM Package
```

I/O Fencing for AIX

I/O fencing is required on AIX nodes in order to protect data integrity of the filesystems in the cluster. The `/etc/fencing.conf` file enumerates the worldwide port name (WWPN) for all of the host bus adapters (HBAs) that will be used to mount a CXFS filesystem. The `/etc/fencing.conf` file must contain a simple list of WWPNs as 64-bit hexadecimal numbers, one per line. These HBAs will then be available for fencing.

If you want to use the `/etc/fencing.conf` file, you must update it whenever the HBA configuration changes, including the replacement of an HBA.

Do the following:

1. Follow the Fibre Channel cable on the back of the AIX host 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.)
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 WWPNs. 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 `/etc/fencing.conf` file.)

4. Edit or create the `/etc/fencing.conf` file on the AIX node and add the WWPN for the port determined in step 1. (Comment lines begin with a # character.) For example, if you determined that port 0 is the port connected to the switch, your `/etc/fencing.conf` file should appear as follows:

   ```
   2000000173002c0b
   ```

5. To configure fencing, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Start/Stop `cxfs_client` for AIX

The `/usr/cxfs_cluster/bin/cxfs_cluster` script will be invoked automatically during normal system startup and shutdown procedures. This script starts and stops the processes required to run CXFS.

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

```
aix# /usr/cxfs_cluster/bin/cxfs_cluster start
```

To stop `cxfs_client` manually, enter the following:

```
aix# /usr/cxfs_cluster/bin/cxfs_cluster stop
```

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

```
aix# /usr/cxfs_cluster/bin/cxfs_cluster restart
```

Maintenance for AIX

This section contains the following:

Updating the CXFS Software for AIX

To upgrade the CXFS software on an AIX system, do the following:

1. Make sure that no applications on the node are accessing files on a CXFS filesystem.
2. Determine the name of the CXFS package that is installed. For example:

```
aix# lslpp -L | grep cxfs
  SGIcxfs-aix5L         5.6.0.5   C     F    CXFS CLIENT for AIX
```

3. Uninstall the old version by using the following command:

   `installp -u` *packagename*

   For example, given a package name of `SGIcxfs-aix5L`:

```
aix# installp -u SGIcxfs-aix5L
```

4. Obtain the CXFS update software from Supportfolio according to the directions in *CXFS 5 Administration Guide for SGI InfiniteStorage*.
5. Install the new version. See "Client Software Installation for AIX" on page 39.

Modifying the CXFS Software for AIX

You can modify the behavior of the CXFS client daemon (`cxfs_client`) by placing options in the `/usr/cxfs_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.

Recognizing Storage Changes for AIX

If you make changes to your storage configuration, you must rerun the HBA utilities to reprobe the storage. For more information, see the IBM HBA documentation.

GRIO on AIX

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the AIX 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 node.

An AIX node can mount a GRIO-managed filesystem and supports node-level reservations. An AIX 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 11 and the *Guaranteed-Rate I/O Version 2 for Linux Guide*.

XVM Failover V2 on AIX

Following is an example of the `/etc/failover2.conf` file on AIX:

```
/dev/hdisk199 affinity=1 preferred
/dev/hdisk135 affinity=1
/dev/hdisk231 affinity=2
/dev/hdisk167 affinity=2
```

For more information, see "XVM Failover and CXFS" on page 11, the comments in the `/etc/failover2.conf` file, *CXFS 5 Administration Guide for SGI InfiniteStorage*, and the *XVM Volume Manager Administrator's Guide*.

IBM hosts running the AIX 5L operating system set the QERR mode page bit to 1 for support storage (other than IBM storage), which does not work well with server-capable administration nodes: Linux will leave CTQ enabled but suffer from timeouts.

There is an administrative work-around for this problem. Engenio offers an enhanced feature called *SANshare* for storage partioning. There is an additional licensing cost required to obtain a SANshare license. SANshare allows hosts to be grouped separately and still access the same LUNs, thus allowing the IBM AIX 5L hosts to set the QERR mode page bit to 1 and not affect the other hosts accessing the LUN.

For each RAID unit, create one **Host Group** for all of the AIX 5L systems separate from the other hosts in the CXFS cluster. Set the **Host Type** as follows:

- LINUX for the AIX nodes (do not use the AIX selection)
- SGIAVT for all other nodes in the cluster

Troubleshooting for AIX

This section discusses the following:

Also see Chapter 10, "General Troubleshooting" on page 273 and Appendix D, "Error Messages" on page 301.

Unable to Mount Filesystems on AIX

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 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 AIX" on page 48.
- 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 `/usr/cxfs_cluster/bin/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/tmp/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? See "Understand Hostname Resolution and Network Configuration Rules" on page 15.

  • Has the node been added to the cluster and enabled? See "Verifying the Cluster Status" on page 265.

– 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/tmp/cxfs_client` file for mount errors.

The `cxfs_client` Daemon is Not Started on AIX

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

```
aix# ps -ef | grep cxfs_client
```

The `cxfs_client` daemon might not start for the following reasons:

• The workstation is in 32-bit kernel mode, which is indicated if the following message is output to the console:

```
CXFS works only in the 64 bit kernel mode
```

In this case, you must change to 64-bit mode as follows:

1. Link the following libraries:

```
aix# ln -fs /usr/lib/boot/unix_64 /unix
aix# ln -fs /usr/lib/boot/unix_64 /usr/lib/boot/unix
```

2. Create the boot image:

```
aix# bosboot -ad /dev/ipldevice
```

3. Reboot the system.

Filesystems Do Not Mount on AIX

If the `/var/tmp` filesystem is full, CXFS cannot write logs to it and the CXFS filesystem will not be able to mount on the AIX node. In this case, you should clean out the `/var/tmp` filesystem.

If a disk is read from an AIX node and the following message is output, it means that the Fibre Channel switch has broken down:

```
no such device or address
```

In this case, you should restart the Fibre Channel switch.

Panic Occurs when Executing `cxfs_cluster` on AIX

If the following message is output, then the `genkex` command does not exist:

```
genkex isn't found
```

In this case, you must install the `bos.perf.tools` file set.

A Memory Error Occurs with `cp -p` on AIX

If an error occurs when a file is copied with the `cp -p` command and the following message is output, there is a problem with NFS:

```
There is not enough memory available now
```

In this case, you must use maintenance level 5100-04+IY42428.

For more information, see:

https://techsupport.services.ibm.com/server/aix.fdc

An ACL Problem Occurs with `cp -p` on AIX

If an ACL is not reflected when a file with an ACL is copied from JFS to CXFS using the `cp -p` command, there is a problem with the AIX software. (The ACL information for the file is indicated by the `aclget` command.) In this case, you must use maintenance level 5100-04.

For more information, see:

https://techsupport.services.ibm.com/server/aix.fdc

Large Log Files on AIX

The /var/tmp/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/cxfs_cluster/bin/cxfs_client.options file.

See the cxfs_client man page and "Log Files on AIX" on page 29.

Mapping Physical Device Names to Unlabeled LUNs on AIX

To map physical devices names to unlabeled LUNs, use the AIX lscfg command. For example (line break shown for readability, output truncated):

```
# lscfg | sed -n -e 's/000000000000//' -e 's/.*\(hdisk[0-9]*\).*-\
(P.*\)-W\(.*\)-L\([0-9ABCDEF]*\) .*/\/dev\/\1 # HBA=\2 WWN=\3 LUN=\4/p' | sort -k4
/dev/hdisk34 # HBA=P1-C1-T1 WWN=201700A0B829A930 LUN=0
/dev/hdisk35 # HBA=P1-C1-T1 WWN=201700A0B829A930 LUN=1
/dev/hdisk36 # HBA=P1-C1-T1 WWN=201700A0B829A930 LUN=2
/dev/hdisk37 # HBA=P1-C1-T1 WWN=201700A0B829A930 LUN=3
/dev/hdisk38 # HBA=P1-C1-T1 WWN=201700A0B829A930 LUN=4
/dev/hdisk39 # HBA=P1-C1-T1 WWN=201700A0B829A930 LUN=5
/dev/hdisk40 # HBA=P1-C1-T1 WWN=201700A0B829A930 LUN=6
/dev/hdisk41 # HBA=P1-C1-T1 WWN=201700A0B829A930 LUN=7
/dev/hdisk42 # HBA=P1-C1-T1 WWN=201700A0B829A930 LUN=1F
/dev/hdisk43 # HBA=P1-C1-T1 WWN=201600A0B829A930 LUN=0
/dev/hdisk44 # HBA=P1-C1-T1 WWN=201600A0B829A930 LUN=1
/dev/hdisk45 # HBA=P1-C1-T1 WWN=201600A0B829A930 LUN=2
/dev/hdisk46 # HBA=P1-C1-T1 WWN=201600A0B829A930 LUN=3
/dev/hdisk47 # HBA=P1-C1-T1 WWN=201600A0B829A930 LUN=4
/dev/hdisk48 # HBA=P1-C1-T1 WWN=201600A0B829A930 LUN=5
/dev/hdisk49 # HBA=P1-C1-T1 WWN=201600A0B829A930 LUN=6
/dev/hdisk50 # HBA=P1-C1-T1 WWN=201600A0B829A930 LUN=7
/dev/hdisk51 # HBA=P1-C1-T1 WWN=201600A0B829A930 LUN=1F
```

See also "Mapping Physical Device Names to XVM Physvols" on page 23.

Reporting AIX Problems

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

```
aix# 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.

Also 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 5 Administration Guide for SGI InfiniteStorage*.
- Moduler debugger output from the `kdb` command:
  - For panics or generated dumps, use the following commands and save the output:

    ```
    aix# kdb /var/adm/ras/vmcore.xx[/unix]
    (0)> stat
    ```

  - For dumps from hangs:

    ```
    aix# kdb /var/adm/ras/vmcore.xx[/unix]
    (0)> th* (to find the slot value of the working process or thread)
    (0)> sw slot_value
    (0)> stat
    ```

- A list of the installed CXFS packages. Use the `lslpp` command as follows:

  ```
  aix# lslpp -l SGIcxfs-aix5L
  ```

- The version information of the operating system. Use the following `oslevel` commands:

  ```
  aix# oslevel -r
  aix# oslevel -g
  ```

- A list of the loaded AIX kernel extensions. Use the `genkex` command.

IRIX Platform

CXFS supports a client-only node running the SGI IRIX operating system on supported SGI machines. This chapter discusses the following:

For information about running the CXFS graphical user interface (GUI), system reset, and system tunable parameters, see *CXFS 5 Administration Guide for SGI InfiniteStorage.*

CXFS on IRIX

This section contains the following information about CXFS on IRIX:

Requirements for IRIX

In addition to the items listed in "Requirements" on page 9, using an IRIX node to support CXFS requires the following:

- One of the following operating systems:
  - IRIX 6.5.28
  - IRIX 6.5.29
  - IRIX 6.5.30

  Note: See the release notes for patch requirements.

- SGI server hardware:
  - SGI Origin 200 server
  - SGI Origin 2000 series
  - SGI Origin 300 server
  - SGI Origin 350 server
  - SGI Origin 3000 series
  - Silicon Graphics Onyx2 system
  - Silicon Graphics Fuel visual workstation
  - Silicon Graphics Octane system
  - Silicon Graphics Octane2 system
  - Silicon Graphics Tezro

- The following Fibre Channel HBAs:
  - LSI Logic models:

    LSI7104XP-LC
    LSI7204XP-LC

  - QLogic models:

    QLA2200
    QLA2200F
    QLA2310
    QLA2310F
    QLA2342
    QLA2344

For additional information, see the CXFS IRIX release notes.

CXFS Commands on IRIX

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

```
/usr/cluster/bin/cxfs_client
/usr/cluster/bin/cxfs-config
/usr/cluster/bin/cxfs_info
/usr/cluster/bin/cxfscp
/usr/cluster/bin/cxfsdump
/usr/cluster/bin/frametest
/usr/cluster/bin/framesort
/usr/sbin/grioadmin
/usr/sbin/griomon
/usr/sbin/grioqos
/usr/sbin/xvm
```

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

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.

For more information, see the man pages. For additional information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 11 and "GRIO on IRIX" on page 66.

Log Files on IRIX

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

For information about the log files created on server-capable administration nodes, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

CXFS Mount Scripts on IRIX

IRIX supports the CXFS mount scripts. See "CXFS Mount Scripts" on page 7 and the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Limitations and Considerations on IRIX

Note the following:

- The inode monitor device (`imon`) is not supported on CXFS filesystems.
- Do not use the IRIX `fsr` command; the `bulkstat` system call has been disabled for CXFS client-only nodes.

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 293.

Access Control Lists and IRIX

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.

Preinstallation Steps for IRIX

This section discusses the following:

Adding a Private Network for IRIX

The following procedure provides an overview of the steps required to add a private network.

Note: A private network is required for use with CXFS.

You may skip some steps, depending upon the starting conditions at your site.

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 are going to resolve names for their client/server programs (such as NFS), 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-example.com server1
   190.0.2.3 stocks
   190.0.3.1 priv-server1
   190.0.2.2 server2-example.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`(5) and `resolve.conf`(5) 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.

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 *IRIX Admin: Networking and Mail*. To verify that the private interface is operational, use the `ifconfig -a` command. For example:

```
irix# 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.

4. *(Optional)* Make the modifications required to use CXFS connectivity diagnostics.

Verifying the Private and Public Networks for IRIX

For each private network on each node in the pool, verify access with the `ping` command. Enter the following, where *nodeIPaddress* is the IP address of the node:

`ping` *nodeIPaddress*

For example:

```
irix# 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
```

Also execute a `ping` on the public networks. If `ping` fails, follow these steps:

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

   ```
   irix# 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.

2. Verify that the cables are correctly seated.

Repeat this procedure on each node.

Client Software Installation for IRIX

Note: CXFS does not support a miniroot installation.

You cannot combine the IRIX operating system installation and the CXFS installation. You must install the operating system first.

To install the required IRIX software, do the following on each IRIX node:

1. Read the *SGI InfiniteStorage Software Platform* release notes, and CXFS general release notes in the `/docs` directory on the ISSP DVD and any late-breaking caveats on Supportfolio.
2. Verify that you are running the correct version of IRIX or upgrade to IRIX 6.5.*x* according to the *IRIX 6.5 Installation Instructions*.

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

   ```
   irix# uname -aR
   ```

3. (*For sites with a serial port server*) Install the version of the serial port server driver that is appropriate to the operating system. Use the CD that accompanies the serial port server. Reboot the system after installation.

   For more information, see the documentation provided with the serial port server.
4. Insert *IRIX CD-ROM #1* into the CD drive.
5. Start up `inst` and instruct it to read the CD:

   ```
   # inst
   ...
   Inst> open /CDROM/dist
   ```

Caution: Do not install to an alternate root using the `inst -r` option. Some of the exit operations (exitops) do not use pathnames relative to the alternate root, which can result in problems on both the main and alternate root filesystem if you use the `-r` option. For more information, see the `inst` man page.

6. *(Optional*) If you want to use Performance Co-Pilot to run XVM statistics, install the default `pcp_eoe` subsystems. This installs the Performance Co-Pilot PMDA (the agent to export XVM statistics) as an exit operation (exitop).

```
Inst> keep *
Inst> install pcp_eoe default
Inst> go
...
Inst> quit
```

7. 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 tardist on the server will be as follows:

`/usr/cluster/client-dist/`*CXFS_VERSION*`/irix/`*IRIX_VERSION*`/noarch/cxfs-client.tardist`

   For example, for IRIX 6.5.30:

`/usr/cluster/client-dist/5.0.0.3/irix/6530/noarch/cxfs-client.tardist`

8. Read the IRIX release notes by choosing the following from the desktop **Toolchest** to bring up the **Software Manager** window:

   System
   > Software Manager

   Choose **Customize Installation** by typing the directory into which you downloaded the software into the **Available Software** box. A list of products available for installation will come up. If the product name is highlighted (similar to an HTML link), then there are release notes available. Click on the link to bring up the **Release Notes** window.

9. Change to the directory containing the tardist and install the software. For example:

10. Install the CXFS software:

```
irix# cd download_directory
irix# inst -f cxfs-client.tardist
Inst> install *
```

Caution: Do not install to an alternate root using the `inst -r` option. Some of the exit operations (exitops) do not use pathnames relative to the alternate root, which can result in problems on both the main and alternate root filesystem if you use the `-r` option. For more information, see the `inst` man page.

If you do not install `cxfs_client`, the `inst` utility will not detect a conflict, but the CXFS cluster will not work. You **must** install the `cxfs_client` subsystem.

11. *(Optional)* If you do not want to install GRIO:

```
Inst> keep *.*.grio2*
```

12. Install the chosen software:

```
Inst> go
...
Inst> quit
```

This installs the following packages:

```
cxfs.books.CXFS_AG
cxfs.man.relnotes
cxfs.sw.cxfs
cxfs.sw.grio2_cell    (Optional)
cxfs.sw.xvm_cell
cxfs_client.man.man
cxfs_client.sw.base
cxfs_util.man.man
cxfs_util.sw.base
eoe.sw.grio2          (Optional)
eoe.sw.xvm
patch_cxfs.eoe_sw.base
patch_cxfs.eoe_sw64.lib
```

The process may take a few minutes to complete.

13. Reboot the system.

Using `cxfs-reprobe` on IRIX Nodes

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

The `cxfs_reprobe` script detects the presence of the SCSI layer on the system and probes all SCSI layer devices by default. You can override this decision by setting `CXFS_PROBE_SCSI` to `0` to disable the probe or `1` to force the probe (default).

When a SCSI scan is performed, all buses/channels/IDs and LUNs are scanned by default to ensure that all devices are found. You can override this decision by setting one or more of the environment variables listed below. This may be desired to reduce lengthy probe times.

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`

`CXFS_PROBE_SCSI_BUSES=`*BusList*

Scans the buses listed. Default: All buses (`-`)

`CXFS_PROBE_SCSI_CHANNELS=`*ChannelList*

Scans the channels listed. Default: All channels (`-`)

`CXFS_PROBE_SCSI_IDS=`*IDList*

Scans the IDs listed. Default: All IDs (`-`)

CXFS_PROBE_SCSI_LUNS=*LunList*

Scans the LUNs listed. Default: All LUNs (-)

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 `/etc/init.d/cxfs_client` script.

I/O Fencing for IRIX Nodes

On the IRIX platform, 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 logged to the `/var/adm/cxfs_client` file.

To configure fencing, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Start/Stop `cxfs_client` for IRIX

The `/etc/init.d/cxfs_client` 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:

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

To stop `cxfs_client` manually, enter the following:

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

Automatic Restart for IRIX

If you want nodes to restart automatically when they are reset or when the node is powered on, you must set the boot parameter `AutoLoad` variable on each IRIX node to `yes` as follows:

```
# nvram AutoLoad yes
```

This setting is recommended, but is not required for CXFS.

You can check the setting of this variable with the following command:

```
# nvram AutoLoad
```

CXFS `chkconfig` Arguments for IRIX

The `cxfs_client` argument to `chkconfig` controls whether or not the `cxfs_client` daemon should be started.

Modifying the CXFS Software for IRIX

You can modify the CXFS client daemon (`/usr/cluster/bin/cxfs_client`) by placing options in the `cxfs_client.options` file:

```
/etc/config/cxfs_client.options
```

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.

For example, to see if `cxfs_client` is using the options in `cxfs_client.options`:

```
irix# ps -ef | grep cxfs_client
    root     219311     217552  0 12:03:17 pts/0   0:00 grep cxfs_client
    root        540          1  0   Feb 26 ?      77:04 /usr/cluster/bin/cxfs_client -i cxfs3-5
```

GRIO on IRIX

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the IRIX 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 node.

An IRIX node can mount a GRIO-managed filesystem and supports application- and node-level reservations. An IRIX 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 11 and the *Guaranteed-Rate I/O Version 2 for Linux Guide.*

XVM Failover V2 on IRIX

Following is an example of the `/etc/failover2.conf` file on an IRIX system:

```
/dev/dsk/20000080e5116e2a/lun0vol/c2p1 affinity=0 preferred
/dev/dsk/20000080e511ab60/lun0vol/c2p3 affinity=1

/dev/dsk/20000080e5116e2a/lun1vol/c2p1 affinity=0
/dev/dsk/20000080e511ab60/lun1vol/c2p3 affinity=1 preferred

/dev/dsk/200400a0b80f7ecf/lun0vol/c2p1 affinity=0 preferred
/dev/dsk/200500a0b80f7ecf/lun0vol/c2p1 affinity=1
```

```
/dev/dsk/200400a0b80f7ecf/lun1vol/c2p1 affinity=0
/dev/dsk/200500a0b80f7ecf/lun1vol/c2p1 affinity=1 preferred

/dev/dsk/200400a0b80f7ecf/lun2vol/c2p1 affinity=0 preferred
/dev/dsk/200500a0b80f7ecf/lun2vol/c2p1 affinity=1

/dev/dsk/200400a0b80f7ecf/lun3vol/c2p1 affinity=0
/dev/dsk/200500a0b80f7ecf/lun3vol/c2p1 affinity=1 preferred
```

For more information, see:

- The comments in the `/etc/failover2.conf.example` file
- "XVM Failover and CXFS" on page 11
- *CXFS 5 Administration Guide for SGI InfiniteStorage*
- *XVM Volume Manager Administrator's Guide*

Troubleshooting on IRIX

This section discusses the following:

Identify the Cluster Status

When you encounter a problem, identify the cluster status by answering the following questions:

- Are the cluster daemons running?
- Is the cluster state consistent on each node? Run the `clconf_info` command on each server-capable administration node and compare.
- Which nodes are in the CXFS kernel membership? Check the cluster status and the `/var/adm/SYSLOG` file.
- Which nodes are in the cluster database (`fs2d`) membership? See the `/var/cluster/ha/log/fs2d_log` files on each server-capable administration node.
- Is the database consistent on all server-capable administration nodes? Determine this logging in to each server-capable administration node and examining the `/var/cluster/ha/log/fs2d_log` file and database checksum.
- Log onto the various CXFS client-only nodes or use the GUI view area display with details showing to answer the following:
  - Are the devices available on all nodes? Use the following:
    - The `xvm` command to show the physical volumes:

      ```
      xvm:cluster> show -v phys/
      ```

    - Is the client-only node in the cluster? Use the `cxfs_info` command.
    - List the contents of the `/dev/cxvm` directory with the `ls` command:

      ```
      # ls /dev/cxvm
      ```

    - Use the `hinv` command to display the hardware inventory.
  - Are the filesystems mounted on all nodes? Use `mount` and `clconf_info` commands.
  - Which node is the metadata server for each filesystem? Use the `clconf_info` command.

  On the metadata server, use the `clconf_info` command.

- Is the metadata server in the process of recovery? Look at the following file: `/var/log/messages`

  Messages such as the following indicate that recovery status:

  – In process:

```
Mar 13 11:31:02 1A:p2 unix: ALERT: CXFS Recovery: Cell 1: Client Cell 0 Died, Recovering </scratch/p9/local>
```

  – Completed:

```
Mar 13 11:31:04 5A:p2 unix: NOTICE: Signaling end of recovery cell 1
```

- If filesystems are not mounting, do they appear online in XVM? You can use the following `xvm` command:

  ```
  xvm:cluster> show vol/*
  ```

Physical Storage Tools

Understand the following physical storage tools:

- To display the hardware inventory:

  ```
  irix# /sbin/hinv
  ```

  If the output is not what you expected, do a probe for devices and perform a SCSI bus reset, using the following command:

  ```
  irix# /usr/sbin/scsiha -pr bus_number
  ```

- To configure I/O devices on an IRIX node, use the following command:

  ```
  irix# /sbin/ioconfig -f /hw
  ```

- To show the physical volumes, use the `xvm` command:

  ```
  irix# /sbin/xvm show -v phys/
  ```

  See the *XVM Volume Manager Administrator's Guide*.

Disk Activity

Use the `sar` system activity reporter to show the disks that are active. For example, the following example for IRIX will show the disks that are active, put the disk name at the end of the line, and poll every second for 10 seconds:

```
irix# sar -DF 1 10
```

For more information, see the `sar`(1) man page.

Buffers in Use

Use the IRIX `bufview` filesystem buffer cache activity monitor to view the buffers that are in use. Within `bufview`, you can use the `help` subcommand to learn about available subcommands, such as the `f` subcommand to limit the display to only those with the specified flag. For example, to display the in-use (busy) buffers:

```
# bufview
f
Buffer flags to display bsy
```

For more information, see the `bufview`(1) man page.

Performance Monitoring Tools

Understand the following performance monitoring tools:

- To monitor system activity:

  `/usr/bin/sar`

- To monitor filesystem buffer cache activity :

  `/usr/sbin/bufview`

  Note: Do not use `bufview` interactively on a busy IRIX node; run it in batch mode.

- To monitor operating system activity data on an IRIX node:

  `/usr/sbin/osview`

- To monitor the statistics for an XVM volume, use the `xvm` command:

  `/sbin/xvm change stat on {`*concatname*`|`*stripename*`|`*physname*`}`

  See the *XVM Volume Manager Administrator's Guide.*

- To monitor system performance, use Performance Co-Pilot (PCP). See the PCP documentation and the `pmie`(1) and `pmieconf`(1) man pages.

- To monitor CXFS heartbeat timeouts, use the `icrash` command. For example, the following command prints the CXFS kernel messaging statistics:

```
irix# icrash -e "load -F cxfs; mtcp_stats"
corefile = /dev/mem, namelist = /unix, outfile = stdout

Please wait.............
Loading default Sial macros...........


>> load cxfs

>> mtcp_stats
STATS @ 0xc000000001beebb8
Max delays: discovery 500767 multicast 7486 hb monitor 0
hb generation histogram:(0:0)(1:0)(2:0)(3:0)(4:0)(5:0)
Improperly sized alive mesgs 0 small 0 big 0
Alive mesgs with: invalid cell 0 invalid cluster 0 wrong ipaddr 2
Alive mesgs from: unconfigured cells 100 cells that haven't discovered us 6000
mtcp_config_cell_set 0x0000000000000007
cell 0:starting sequence # 77 skipped 0
hb stats init @ 15919:(0:1)(1:478301)(2:29733)(3:0)(4:0)
cell 1:starting sequence # 0 skipped 0
hb stats init @ 360049:(0:1)(1:483337)(2:21340)(3:0)(4:0)
cell 2:starting sequence # 0 skipped 0
```

  The following fields contain information that is helpful to analyzing CXFS heartbeat timing:

  – `discovery`: The maximum time in HZ that the discovery thread (that is, the thread that processes incoming heartbeats) has slept. Because nodes generate heartbeats once per second, this thread should never sleep substantially longer than 100 HZ.

A value much larger than 100 suggests either that it was not receiving heartbeats or that something on the node prevented this thread from processing the heartbeats.

- `multicast`: The thread that generates heartbeats sleeps for 100 HZ after sending the last heartbeat and before starting on the next. This field contains the maximum time in HZ between the start and end of that sleep. A value substantially larger than 100 indicates a problem getting the thread scheduled; for example, when something else on the node is taking all CPU resources.
- `monitor`: The maximum time in HZ for the heartbeat thread tp sleep and send its heartbeat. That is, it contains the value for `multicast` plus the time it takes to send the heartbeat. If this value is substantially higher than 100 but `multicast` is not, it suggests a problem in acquiring resources to send a heartbeat, such as a memory shortage.
- `gen_hist`: A histogram showing the number of heartbeats generated within each interval. There are 6 buckets tracking each of the first 5 seconds (anything over 5 seconds goes into the 6th bucket).
- `hb_stats`: Histograms for heartbeats received. There is one histogram for each node in the cluster.
- `seq_stats`: Number of consecutive incoming heartbeats that do not have consecutive sequence numbers. There is one field for each node. A nonzero value indicates a lost heartbeat message.
- `overdue`: Time when an overdue heartbeat is noticed. There is one field per node.
- `rescues`: Number of heartbeats from a node that are overdue but CXFS message traffic has been received within the timeout period.
- `alive_small`: Number of times a heartbeat message arrived that was too small, (that is, contained too few bytes).
- `alive_big`: Number of times a heartbeat arrived that was too large.
- `invalid_cell`: Number of heartbeats received from nodes that are not defined in the cluster.
- `invalid_cluster`: Number of heartbeats received with the wrong cluster ID.
- `wrong_ipaddr`: Number of heartbeats received with an IP address that does not match the IP address configured for the node ID.

– not_configured: Number of heartbeats received from nodes that are not defined in the cluster.

– unknown: Number of heartbeats from nodes that have not received the local node's heartbeat.

Kernel Status Tools

Note: You must run the sial scripts version of icrash commands.

Understand the following kernel status tools (this may require help from SGI service personnel):

• To determine IRIX kernel status, use the icrash command:

```
# /usr/bin/icrash
>> load -F cxfs
```

Note: Add the -v option to these commands for more verbose output.

– cfs to list CXFS commands

– dcvn to obtain information on a single client vnode

– dcvnlist to obtain a list of active client vnodes

– dsvn to obtain information on a single server vnode

– dsvnlist to obtain a list of active server vnodes

– mesglist to trace messages to the receiver (you can pass the displayed object address to the dsvn command to get more information about the server vnodes and pass the thread address to the mesgargs command to get more information about the stuck message). For example (line breaks shown here for readability):

```
>> mesglist
 Cell:2
TASK ADDR           MSG ID TYPE CELL MESSAGE                          Time(Secs) Object
================== ======= ==== ==== ================================ ========== ============================
0xe0000030e5ba8000      14  Snt    0                     I_dsvn_fcntl          0 N/A
```

```
0xe0000030e5ba8000      14  Cbk    0                   I_ucopy_copyin          0 N/A
0xa80000000bb77400    1210  Rcv    0               I_dsxvn_allocate_1       1:06 (dsvn_t*)0xa80000000a7f8900
```

```
>> mesgargs 0xa80000000bb77400
(dsvn_t*)0xa80000000a7f8900
    (dsxvn_allocate_1_in_t*)0xa800000001245060
        objid=0xa80000000a7f8910 (dsvn=0xa80000000a7f8900)
        offset=116655
        length=0x1
        total=1
        mode=2
        bmapi_flags=0x7
        wr_ext_count=0
        &state=0xa8000000012450b0 credid=NULLID
        lent_tokens=0xa800000 (DVN_TIMES_NUM(SWR)|DVN_SIZE_NUM(WR)|DVN_EXTENT_NUM(RD))
        reason_lent=0x24800000 (DVN_TIMES_NUM(CLIENT_INITIATED)|DVN_SIZE_NUM(CLIENT_INITIATED)|
DVN_EXTENT_NUM(CLIENT_INITIATED))
        lender_cell_id=0
    (dsxvn_allocate_1_inout_t*)0xa800000001245110
        cxfs_flags=0x200
        cxfs_gen=4661
```

```
>> dsvn 0xa80000000a7f8900
(dsvn_t*)0xa80000000a7f8900:
    flags 0x10
    kq.next 0xc000000001764508 kq.prev 0xc000000001764508
    &tsclient 0xa80000000a7f8a30  &tserver 0xa80000000a7f8a80
    bhv 0xa80000000a7f8910 dsvfs 0xa800000026342b80
    (cfs_frlock_info_t*)0xa80000000bfee280:
        wait: none
        held: none
    vp 0xa8000000224de500 v_count 2 vrgen_flags 0x0
    dmvn 0x0000000000000000
    objid 0xa80000000a7f8910 gen 4 obj_state 0xa80000000a7f8940
    (dsxvn_t*)0xa80000000a7f8900:
        dsvn 0xa80000000a7f8900 bdp 0xa800000010b52d30
        tkclient 0xa80000000a7f8a30 tserver 0xa80000000a7f8a80
        ext gen 4661 io_users 2 exclusive_io_cell -1
```

```
oplock 0 oplock_client -1 &dsx_oplock_lock 0xa80000000a7f8b9
```

- `sinfo` to show clients/servers and filesystems
- `sthread | grep cmsd` to determine the CXFS kernel membership state. You may see the following in the output:
  - `cms_dead()` indicates that the node is dead
  - `cms_follower()` indicates that the node is waiting for another node to create the CXFS kernel membership (the leader)
  - `cms_leader()` indicates that the node is leading the CXFS kernel membership creation
  - `cms_declare_membership()` indicates that the node is ready to declare the CXFS kernel membership but is waiting on resets
  - `cms_nascent()` indicates that the node has not joined the cluster since starting
  - `cms_shutdown()` indicates that the node is shutting down and is not in the CXFS kernel membership
  - `cms_stable()` indicates that the CXFS kernel membership is formed and stable
- `tcp_channels` to determine the status of the connection with other nodes
- `t -a -w` *filename* to trace for CXFS
- `t` *cms_thread* to trace one of the above threads

• To invoke internal kernel routines that provide useful debugging information, use the `idbg` command (available in the IRIX OS software but not installed by default):

```
# /usr/sbin/idbg
```

Are there any long running (>20 seconds) kernel messages? Use the `icrash mesglist` command to examine the situation.

No Cluster Name ID Error

For example:

```
Mar  1 15:06:18 5A:nt-test-07 unix: NOTICE: Physvol (name cip4) has no
CLUSTER name id: set to ""
```

This message means the following:

- The disk labeled as an XVM physvol was probably labeled under IRIX 6.5.6f and the system was subsequently upgraded to a newer version that uses a new version of XVM label format. This does not indicate a problem.
- The cluster name had not yet been set when XVM encountered these disks with an XVM cluster physvol label on them. This is normal output when XVM performs the initial scan of the disk inventory, before node/cluster initialization has completed on this host.

  The message indicates that XVM sees a disk with an XVM cluster physvol label, but that this node has not yet joined a CXFS membership; therefore, the cluster name is empty ("").

  When a node or cluster initializes, XVM rescans the disk inventory, searching for XVM cluster physvol labels. At that point, the cluster name should be set for this host. An empty cluster name after node/cluster initialization indicates a problem with cluster initialization.

  The first time any configuration change is made to any XVM element on this disk, the label will be updated and converted to the new label format, and these notices will go away.

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

System is Hung

The following may cause the system to hang:

- Overrun disk drives.
- CXFS heartbeat was lost. In this case, you will see a message that mentions `withdrawl of node`.
- As a last resort, do a non-maskable interrupt (NMI) of the system and contact SGI. (The NMI tells the kernel to panic the node so that an image of memory is saved and can be analyzed later.) For more information, see the owner's guide for the node.

  Make the following files available:

  – System log file: `/var/adm/SYSLOG`

  – IRIX `vmcore.#.comp`

  – IRIX `unix.#`

`SYSLOG credid` Warnings

Messages such as the following in the `SYSLOG` indicate that groups from another node are being dropped, and you may not be able to access things as expected, based on group permissions (line breaks added here for readability):

```
May  1 18:34:42 4A:nodeA unix: WARNING: credid_bundle_import: received cred for uid 5778 with 23 groups when \
     configured for only 16 groups. Extra groups dropped.
May  1 18:34:59 4A:nodeB unix: WARNING: credid_getcred: received cred for uid 5778 with 23 groups when \
     configured for only 16 groups. Extra groups dropped.
May  1 18:35:44 4A:nodeB unix: WARNING: credid_getcred: received cred for uid 5778 with 23 groups when  \
     configured for only 16 groups. Extra groups dropped.
May  1 18:36:29 4A:nodeA unix: WARNING: credid_bundle_import: received cred for uid 5778 with 23 groups  \
     when configured for only 16 groups. Extra groups dropped.
May  1 18:38:32 4A:nodeA unix: WARNING: credid_bundle_import: received cred for uid 5778 with 23 groups  \
     when configured for only 16 groups. Extra groups dropped.
May  1 18:38:50 4A:nodeB unix: WARNING: credid_getcred: received cred for uid 5778 with 23 groups when  \
     configured for only 16 groups. Extra groups dropped.
May  1 18:39:32 4A:nodeB unix: WARNING: credid_getcred: received cred for uid 5778 with 23 groups when  \
     configured for only 16 groups. Extra groups dropped.
May  1 18:40:13 4A:nodeB unix: WARNING: credid_getcred: received cred for uid 5778 with 23 groups when  \
```

```
     configured for only 16 groups. Extra groups dropped.
May  1 18:40:35 4A:nodeA unix: WARNING: credid_bundle_import: received cred for uid 5778 with 23 groups  \
     when configured for only 16 groups. Extra groups dropped.
May  1 19:04:52 4A:nodeA unix: WARNING: credid_bundle_import: received cred for uid 6595 with 21 groups  \
     when configured for only 16 groups. Extra groups dropped.
May  1 19:38:58 4A:nodeA unix: WARNING: credid_bundle_import: received cred for uid 6595 with 21 groups  \
     when configured for only 16 groups. Extra groups dropped.
```

The IRIX `ngroups_max` static system tunable parameter specifies the maximum number of multiple groups to which a user may simultaneously belong. You should increase the number of groups by running the following command and then rebooting:

```
irix# systune ngroups_max value
```

Reporting IRIX Problems

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

```
irix# 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 5 Administration Guide for SGI InfiniteStorage*.

If a panic has occurred on an IRIX node, also retain the system core files in `/var/adm/crash`, including the following:

`analysis.`*number*
`unix.`*number*
`vmcore.`*number*`.comp`

Linux Platforms

CXFS supports a client-only node running the Red Hat Enterprise Linux (RHEL) or SUSE Linux Enterprise Server (SLES) operating system.

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 5 Administration Guide for SGI InfiniteStorage*.

This chapter contains the following sections:

For information about system tunable parameters, see *CXFS 5 Administration Guide for SGI InfiniteStorage*.

CXFS on Linux

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

Requirements for Linux

In addition to the items listed in "Requirements" on page 9, using a Linux node to support CXFS requires the following:

- One of the following
  - 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, serial lines and/or 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/grioqos
/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 "Linux Installation Procedure" on page 94.

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 5 Administration Guide for SGI InfiniteStorage*.

CXFS Mount Scripts on Linux

Linux supports the CXFS mount scripts. See "CXFS Mount Scripts" on page 7 and the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

For RHEL nodes, 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. For more information, see "Using cxfs-reprobe on IRIX Nodes" on page 63.

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 client 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.

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 293.

Using the `dmi` Mount Option on a SLES 10 or SLES 11 Node

By default, DMAPI 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 `dmi` mount option, you must set `DMAPI_PROBE="yes"` in the `/etc/sysconfig/sysctl` file on the node. Changes to the file will processed on the next reboot. After setting that system configuration file, you can immediately enable DMAPI by executing the following:

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

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

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 16.

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 15.

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

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.

Modifications Required for 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 `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:

Linux Installation Procedure

The CXFS software will be initially installed and configured by SGI personnel. This section provides an overview of those procedures. You can use the information in this section to verify the installation.

Table 5-1 and Table 5-2 provide examples of the differences in package extensions among the various processor classes supported by CXFS.

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

Table 5-1 RHEL Processor and Package Extension Examples

Class	Example Processors	User Package Architecture Extension	Kernel Package Architecture Extension
x86_64	AMD Opteron	`.x86_64.rpm`	`.x86_64.rpm`
	Intel Xeon EM64T	`.x86_64.rpm`	`.x86_64.rpm`
ia64	Intel Itanium 2	`.ia64.rpm`	`.ia64.rpm`

Table 5-2 SLES Processor and Package Extension Examples

Class	Example Processors	User and Kernel Package Architecture Extension
x86_64	AMD Opteron	`.x86_64.rpm`
	EM64T	`.x86_64.rpm`
ia64	Intel Itanium 2	`.ia64.rpm`

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

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/CLIENT_ARCHITECTURE/cxfs-client.tar.gz
```

   For example, for an Altix ia64 client, the location of the tarball on the server will be:

```
/usr/cluster/client-dist/5.6.0.3/linux/sles10sp2/ia64/cxfs-client.tar.gz
```

In this case, you could do the following:

```
cxfs_server# cd /usr/cluster/client-dist/5.6.0.3/linux/sles10sp2/ia64
cxfs_server# 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 `smp` (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 99 and the `cxfs_client`(1M) 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-open` from the SGI Foundation Software release
- `pcp-sgi` from the SGI ProPack release

These packages and are included with SGI Foundation Software. You can obtain the open source PCP package from ftp://oss.sgi.com/projects/pcp/download

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 5 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:                                      [  OK  ]
Mounting devfs filesystems:                                [  OK  ]
Starting cxfs client:                                      [  OK  ]
```

To stop `cxfs_client` manually, enter the following:

```
linux# service cxfs_client stop
Stopping cxfs client:                                      [  OK  ]
```

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

```
linux# service cxfs_client restart
Stopping cxfs client:                                      [  OK  ]
```

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 (707) - 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 100

- "Recognizing Storage Changes for Linux" on page 100

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 "CXFS Mount Scripts" on page 7.

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 either 0 (to disable the probe) or 1 (to force the probe) on the appropriate bus.

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 withing 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_BUSSES=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 needs to rescan 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 either 0 (to disable the probe) or 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 withing 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_BUSSES=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 "Linux Installation Procedure" on page 94 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.

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.

A Linux node can mount a GRIO-managed filesystem and supports node-level reservations. A Linux 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 11 and the *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
```

Following is an example of the `/etc/failover2.conf` file on a Linux SGI Foundation Software system:

```
/dev/xscsi/pci0004:00:01.1/node200900a0b813b982/port1/lun4/disc, affinity=1
/dev/xscsi/pci0004:00:01.1/node200900a0b813b982/port2/lun4/disc, affinity=2
/dev/xscsi/pci0004:00:01.0/node200900a0b813b982/port1/lun4/disc, affinity=1
/dev/xscsi/pci0004:00:01.0/node200900a0b813b982/port2/lun4/disc, affinity=2
/dev/xscsi/pci0004:00:01.1/node200800a0b813b982/port1/lun4/disc, affinity=4
/dev/xscsi/pci0004:00:01.1/node200800a0b813b982/port2/lun4/disc, affinity=3 preferred
/dev/xscsi/pci0004:00:01.0/node200800a0b813b982/port1/lun4/disc, affinity=4
/dev/xscsi/pci0004:00:01.0/node200800a0b813b982/port2/lun4/disc, affinity=3
```

For more information, see:

- The comments in the `/etc/failover2.conf.example` file
- "XVM Failover and CXFS" on page 11
- *CXFS 5 Administration Guide for SGI InfiniteStorage*
- *XVM Volume Manager Administrator's Guide*

Troubleshooting for Linux

This section discusses the following:

For general troubleshooting information, see Chapter 10, "General Troubleshooting" on page 273 and Appendix D, "Error Messages" on page 301.

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 98 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 106.
- 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 `/usr/cluster/bin/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 265.

– 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 10 or SLES 11 Node" on page 86.

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 84.

`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
```

Reporting Linux Problems

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

```
linux# 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.

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 5 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 system. This chapter contains the following sections:

CXFS on Mac OS X

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

• "Access Control Lists and Mac OS X" on page 118

Requirements for Mac OS X

In addition to the items listed in "Requirements" on page 9, using a Mac OS X node to support CXFS requires the following:

- Mac OS X Tiger (10.4.8 or later)
- Mac OS X Leopard (10.5.2 or later)
- One of the following single- or multi-processor Apple Computer hardware platforms:

  Mac Pro
  Power Mac G4
  Power Mac G5
  Xserve
  Xserve G4
  Xserve G5

- 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/grioqos
/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 11 and "GRIO on Mac OS X" on page 136.

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 5 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 for both Leopard and Tiger:

- Mac OS X is unable to safely memory-map a file on a filesystem whose block size is greater than 4 KB. This is a due to a bug in the Darwin kernel that may be fixed by Apple in a future OS update.
- 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 `SGIAVT` mode for XVM failover version 2. For more information, see the information about XVM failover in *CXFS 5 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.

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 293.

Limitations and Considerations on Mac OS X Leopard Only

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

To view drive icons and see CXFS filesystems in the **Finder** under Leopard, enable the following:

Finder
> Preferences
> General
> Connected Servers

Note: Due to a Finder limitation, CXFS drives will only show on the desktop and not in **Finder** under **Shared**.

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 `cxfsmac1`, then you would see the following when requesting the hostname:

```
macosx# /bin/hostname
cxfsmac1.local
```

The full hostname including `.local` is the hostname that the CXFS software will use to determine its identity in the cluster, not `cxfsmac1`.

Therefore, you must configure the node as `cxfsmac1.local` or specify the fully qualified hostname in `/etc/hostconfig`. For example:

```
HOSTNAME=cxfsmac1.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.

Specific procedures differ by platform:

- "Making UID, GID, or Group Changes for Tiger" on page 117
- "Making UID, GID, or Group Changes for Leopard" on page 117

Making UID, GID, or Group Changes for Tiger

To change the user's UID, GID, or other groups on Tiger requires changes to the NetInfo domain, whether local or distributed. Do the following:

- Run the NetInfo Manager tool:

  Applications
  > Utilities
  > NetInfo Manager

- Select the domain (if not the local domain):

  Domain
  > Open....

- Select the user in question:

  users
  > *username*

- Modify the `uid`, `gid`, or group fields as required.

Note: Changing a user's primary UID and/or GID will also require modifying all files owned by the user to the new UID and GID. Ideally, users should be created with the correct values.

Making UID, GID, or Group Changes for Leopard

Leopard's **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 enforcement of POSIX ACLs and the editing of POSIX ACLs from the Mac OS X node.

This section discusses the following:

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,chown
 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,chown
 5: group:everyone allow read,readattr,readextattr,readsecurity
 6: group:everyone deny write,execute,delete,append,writeattr,writeextattr,writesecurity,chown
```

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 6-1, 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 6-1 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:

```
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:

```
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:

```
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:

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

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,chown
  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,chown
 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,chown
 5: group:everyone allow read,readattr,readextattr,readsecurity
 6: group:everyone deny write,execute,delete,append,writeattr,writeextattr,
   writesecurity,chown
```

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 (line breaks shown here for readability):

```
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,chown
 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,chown
 5: group:ptg allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 6: group:ptg deny execute
 7: group:everyone allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 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,chown
 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,chown
 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,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,chown
 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,chown
 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,chown
 4: group:everyone allow list,search,readattr,readextattr,readsecurity
 5: group:everyone deny add_file,delete,add_subdirectory,delete_child,writeattr,
    writeextattr,writesecurity,chown
 6: user:userA allow list,add_file,search,delete,add_subdirectory,delete_child,
    readattr,writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown,
    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,chown,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,search,readattr,readextattr,readsecurity,file_inherit,
     directory_inherit,only_inherit
 12: group:ptg deny add_file,delete,add_subdirectory,delete_child,writeattr,
     writeextattr,writesecurity,chown,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,chown,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

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 112.

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

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 16.

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 115) and IP address on its public network interface.
2. *(Tiger Only)* Decide if the Mac OS X node will be part of a NetInfo domain or a standalone machine. If part of an `/etc/hosts` domain, configure the node into the domain before proceeding further.
3. 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`.
4. Install a second network interface card if necessary as per the manufacturer's instructions.

5. 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.

6. 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

The CXFS software will be initially installed and configured by SGI personnel. This section provides an overview of those procedures. You can use the information in this section to verify the installation.

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 general release notes in the `/docs` directory on the ISSP DVD and late-breaking caveats on Supportfolio.
2. Verify that the node is running the 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 value of `8.8.0` or later for Tiger or `9.2.0` or later for 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` 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. **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 5 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

Before updating CXFS software, ensure that no applications on the node are accessing files on a CXFS filesystem. You can then run the new CXFS software package, which will update all CXFS software. A reboot is required after updating the CXFS software.

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:

```
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.

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 11 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 127.

For more information, see "XVM Failover and CXFS" on page 11, the comments in the `/etc/failover2.conf` file, *CXFS 5 Administration Guide for SGI InfiniteStorage*, and the *XVM Volume Manager Administrator's Guide*.

Troubleshooting for Mac OS X

This section discusses the following:

For general troubleshooting information, see Chapter 10, "General Troubleshooting" on page 273 and Appendix D, "Error Messages" on page 301

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 134 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 114.

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 113.

Reporting Mac OS X Problems

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

```
macosx# 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 5 Administration Guide for SGI InfiniteStorage*.

Solaris Platform

CXFS supports a client-only node running the Solaris operating system. This chapter contains the following sections:

CXFS on Solaris

This section contains the following information about CXFS on Solaris:

Requirements for Solaris

In addition to the items listed in "Requirements" on page 9, using a Solaris node to support CXFS requires the following:

- Solaris operating system:
  - Solaris 10 May 08 (patch 120011-14)

Caution: All other releases of Solaris 10 are not supported and may cause system instabilities when used with CXFS.

- The following supported Fibre Channel HBAs (you can use only one vendor for HBA, either LSI Logic or QLogic; you cannot mix HBA vendors):
  - LSI Logic models using the 01030600 firmware or newer:

    LSI7102XP
    LSI7202XP
    LSI7402XP
    LSI7104XP
    LSI7204XP
    LSI7404XP

  - QLogic models sold by Sun Microsystems and running with the driver supplied by Sun. (If you have a Qlogic HBA, your system will only access disks with GPT labels. For more information about GPT labels and CXFS, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.)

    SG-XPCI1FC-QL2 (single-port 2 Gb)
    SG-XPCI2FC-QF2-Z (dual-port 2 Gb)

Note: CXFS does not automatically detect WWPNs for LSI HBAs or QLogic HBAs. See "I/O Fencing for Solaris" on page 159.

- Any system based on UltraSPARC III, IIIi, or IV with a spare 66-MHz (or faster) PCI slot for a Fibre Channel HBA and a spare 100-Mb/s (or faster) ethernet port for the CXFS private network. CXFS supports a Solaris node only on the SPARC platform. It is not supported on other hardware platforms.

For additional latest information, see the CXFS Solaris release notes.

CXFS Commands on Solaris

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

```
/usr/cxfs_cluster/bin/cxfs_client
/usr/cxfs_cluster/bin/cxfs_info
/usr/cxfs_cluster/bin/cxfsdump
/usr/cxfs_cluster/bin/frametest
/usr/sbin/grioadmin
/usr/sbin/griomon
/usr/sbin/grioqos
/usr/cxfs_cluster/bin/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 `pkgadd` output lists all software added; see "Solaris Installation Procedure" on page 156.

For more information, see the man pages. For additional information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 11 and "GRIO on Solaris" on page 163.

Log Files on Solaris

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

For information about the log files created on server-capable administration nodes, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

CXFS Mount Scripts on Solaris

Solaris supports the CXFS mount scripts. See "CXFS Mount Scripts" on page 7 and the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Limitations and Considerations on Solaris

Note the following:

- Although it is possible to mount a UFS or NFS filesystem on top of a Solaris CXFS filesystem, this is not recommended.
- After a crash, attempts to reclaim locks and commit asynchronous writes to a CXFS filesystem from an NFS client may result in a stale file handle.
- For optimal performance, you should set the value of the Solaris system tunable parameter `maxphys` in the `/etc/system` file. See "`maxphys` System Tunable for Solaris" on page 146.
- All disk devices attached to LSI Logic HBAs must be for use only by CXFS disks; do not attach non-disk devices to any Fibre Channel HBA that is configured for CXFS use. This restriction is required because all disk devices on these HBAs (configured for CXFS) make use of the whole disk volume, which must be conveyed to Solaris via modification in the HBA driver to the value returned by the `READ_CAPACITY` SCSI command.
- CXFS does not automatically detect WWPNs for LSI HBAs. See "I/O Fencing for Solaris" on page 159 for instructions to set up a fencing configuration.
- The `xvm` command displays duplicate entries of physvols. The number of duplicate entries correspond to the devices for each LUN.

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 293.

Access Control Lists and Solaris

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.

If you restore a CXFS file that had an ACL containing only owner-ACL entries (that is, owner/group/other/mask) from a Solaris node, upon restoration one of the following will happen:

- **When using `tar(1)`, `cpio(1)`, and Legato Networker:** The ACL will be lost because these tools behave "intelligently" by not calling `acl` to set an ACL if the file has only owner/group/other/mask entries. These tools will only set the file mode. However, this does not present a change in functionality because an access

permissions check on the mode and the ACL containing only owner entries will give the same result.

- **When using other backup/restore utilities:** A mask will be added to the ACL if the application calls `acl` for every file.

  A backup/restore utility that calls `acl` to set an ACL for every file will result in a file being restored with four ACL entries (that is, owner/group/other/mask), even though it may have originally had only three (that is, owner/group/other). This is due to a requirement in `getfacl` that it receive four ACL entries for the `GETACL` command to `acl`. (If fewer than four entries are returned, `getfacl` will report an error).

Note: Normally, Solaris filesystem ACLs can have up to 1024 entries for a file and a directory can have 1024 entries as well as an additional 1024 entries for the default ACL. However, CXFS filesystems on Solaris nodes in a multiOS cluster must maintain compatibility with the metadata server. The CXFS filesystems on a Solaris node are limited to a maximum of 25 ACL entries for a file and a maximum total of 50 for a directory (that is, the directory ACL plus the default ACL).

When using the `ls` command to display access permissions for a file with an ACL, the mode reported for a CXFS file follows Linux semantics instead of Solaris/UFS semantics.

On Solaris, a UFS file mode reports the group permission as the intersection of the `GROUP` and `MASK` entries in the ACL. If the `GROUP` entry is `r-x` and the `MASK` entry is `rw-`, the group permission will be reported as `r--`.

The CXFS model calls for reporting the ACL `MASK` for the group permission in the mode. Therefore, using the example above, the group permission will be reported as `rw-`. Although it appears that the group has write permission, it does not and an attempt to write to the file will be rejected. You can obtain the real (that is, effective) group permission by using the Solaris `getfacl` command.

`maxphys` System Tunable for Solaris

For optimal performance, you should set the value of the Solaris system tunable parameter `maxphys` in the `/etc/system` file. Do the following:

1. Make a backup copy of the `/etc/system` file.

   Note: Exercise extreme caution in changing `/etc/system` and always make a backup copy.

2. Change the value of maxphys to `0x800000` (hexadecimal) by adding the following to `/etc/system`:

   ```
   set maxphys=0x800000
   ```

3. Reboot the Solaris node. This causes the change to take effect.

4. Verify that the new value for `maxphys` is in effect by running the following command:

   ```
   solaris# echo "maxphys/X" | adb -k
            physmem 1f03f
            maxphys:
            maxphys:        800000
   ```

HBA Installation for Solaris

The QLogic driver is provided with Solaris 10.

This section discusses the following:

- "Installing the LSI Logic HBA" on page 147
- "Verifying the HBA Installation" on page 148
- "Setting Persistent Name Binding" on page 150

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

Installing the LSI Logic HBA

To install the LSI Logic HBA, perform the following steps. Additional details are provided in the *Fibre Channel to PCI-X Host Adapters User's Guide.*

1. Install the LSI Logic HBA into the Solaris system. See the chapter "Installing the Host Adapter" from the *Fibre Channel to PCI-X Host Adapters User's Guide.*
2. Bring the system back up.
3. Install the LSI Logic HBA driver software (`ITImpt`, version 5.07.00 or later) according to the instructions in the driver's `readme` file.

   Do the following:

   a. Retrieve the driver package from the following LSI Logic website:

      http://www.lsi.com/cm/DownloadSearch.do?locale=EN

   b. Install the driver package:

```
solaris# unzip itmpt-5.07.00.zip
solaris# uncompress itmpt_install.tar.Z
solaris# tar -xvf itmpt_install.tar
solaris# cd install
solaris# pkgadd -d .
```

   c. Install the `lsi` utilities package:

```
solaris# uncompress lsiutils_v60.tar.Z
solaris# tar -xvf lsiutils_v60.tar
solaris# cd install
solaris# pkgadd -d .
```

4. For each target/LUN pair to be used by the LSI Logic HBA, use the `lsiprobe` utility to add entries to `/kernel/drv/ssd.conf`.

   For example, to add entries for targets 0 through 5 (inclusive), with each of those targets scanning LUNs 0, 2, 4, 5, and 6:

```
solaris# lsiprobe -a target 0-5 lun 0,2,4-6
```

Note: If you modify `/kernel/drv/ssd.conf`, you must reboot the system (as in step 5) in order for changes to take effect.

---

5. Reboot the Solaris node:

   ```
   solaris# init 6
   ```

6. After the system has rebooted, verify that the driver attached correctly to the HBA by following the steps "Verifying the HBA Installation" on page 148. Do not proceed until the verification succeeds.

Verifying the HBA Installation

After the system reboots, you should verify that the devices were correctly configured by running the Solaris `format` command. You should see a list of each device you selected.

For example:

```
solaris# format

Searching for disks...done

c2t200400A0B80C268Cd1: configured with capacity of 67.75GB
c2t200400A0B80C268Cd3: configured with capacity of 136.64GB

AVAILABLE DISK SELECTIONS:
       0. c0t0d0           /pci@1c,600000/scsi@2/sd@0,0
       1. c0t1d0           /pci@1c,600000/scsi@2/sd@1,0
       2. c2t200400A0B80C268Cd1           /pci@1d,700000/SUNW,qlc@1/fp@0,0/ssd@w200400a0b80c268c,1
       3. c2t200400A0B80C268Cd3           /pci@1d,700000/SUNW,qlc@1/fp@0,0/ssd@w200400a0b80c268c,3

Specify disk (enter its number):
```

In this example, disks 2 and 3 are being addressed by the QLogic driver, as indicated by the presence of `SUNW,qlc@1` in the pathname.

You can also use the `luxadm` command to view the status of the HBA:

```
solaris# luxadm -e port
/devices/pci@1d,700000/SUNW,qlc@1/fp@0,0:devctl                  CONNECTED
/devices/pci@1d,700000/SUNW,qlc@1,1/fp@0,0:devctl                NOT CONNECTED

solaris# luxadm probe
No Network Array enclosures found in /dev/es

Found Fibre Channel device(s):
  Node WWN:200400a0b80c268b  Device Type:Disk device
    Logical Path:/dev/rdsk/c2t200400A0B80C268Cd1s2
  Node WWN:200400a0b80c268b  Device Type:Disk device
    Logical Path:/dev/rdsk/c2t200400A0B80C268Cd3s2
```

The system log and console display may display warning messages similar to the following:

```
WARNING: /pci@1d,700000/SUNW,qlc@1/fp@0,0/ssd@w200400a0b80c268c,3 (ssd0):
        Corrupt label; wrong magic number

WARNING: /pci@1d,700000/SUNW,qlc@1/fp@0,0/ssd@w200400a0b80c268c,1 (ssd1):
        Corrupt label; wrong magic number
```

For QLogic HBA, these messages means that the disk has a bad label or a DVH label, which is not supported. (QLogic HBAs support only GPT labels.)

Similar messages for an LSI Logic HBA will appear on boot for LUNs that have DVH labels. When the XVM module is loaded or started, it installs hooks into the HBA driver and automatically translates the DVH labels into SUN labels (you should not try to relabel the disks with the `format` command); after XVM translates the labels, you will not see these error messages.

Note: You can also use the `lsiutil` command to determine the number of LSI HBAs installed, the model numbers, and firmware versions.

If you are having trouble with the verification steps, see "New Storage is Not Recognized on Solaris" on page 166.

Setting Persistent Name Binding

Adding a new RAID can change the fabric name binding. To set up persistent binding, edit the `/kernel/drv/itmpt.conf` file and add entries of the following format, one for each target, where *portWWN* is the port world wide name of the RAID controller:

`target-`*X*`-wwn="`*portWWN*`"`

For example:

```
target-0-wwn="200800a0b8184c8e"
target-1-wwn="200900a0b8184c8d"
target-2-wwn="200400a0b80c268c"
target-3-wwn="200500a0b80c268c"
```

In this example, `target-0` will be bound to the device with the port WWN `200800a0b8184c8e` and the resulting devices will be:

```
/pci@1d,700000/IntraServer,fc@1/ssd@0,*
```

Preinstallation Steps for Solaris

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

- "Adding a Private Network for Solaris Nodes" on page 150
- "Verifying the Private and Public Networks for Solaris" on page 155

Adding a Private Network for Solaris Nodes

The following procedure provides an overview of the steps required to add a private network to the Solaris system. A private network is **required** for use with CXFS. See "Use a Private Network" on page 16.

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

1. If your system is already operational and on the network, skip to step 2.

If your Solaris system has **never** been set up, bring the system to single-user mode. For example, go to the PROM prompt and boot the Solaris node into single-user mode:

```
> boot -s
```

As a last resort, you can reach the PROM prompt by pressing the `L1-A` (or `Stop-A`) key sequence.

2. Edit the `/etc/inet/ipnodes` file so that it contains entries for each node in the cluster and its private interfaces.

   The `/etc/inet/ipnodes` 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/inet/ipnodes` 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/inet/ipnodes` 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/inet/ipnodes` 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/inet/ipnodes` on the other nodes in the cluster.

   For more information, see the `hosts`, `named`, and `nis` 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 15.

3. Edit the `/etc/nsswitch.conf` file so that local files are accessed before either NIS or DNS. That is, the `ipnodes` line in `/etc/nsswitch.conf` must list `files` first.

   For example:

   ```
   ipnodes:      files nis dns
   ```

   (The order of `nis` and `dns` is not significant to CXFS, but `files` must be first.)

4. Determine the name of the private interface by using the `ifconfig` command as follows:

   ```
   solaris# ifconfig -a
   ```

   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.

   For example, on an Ultra Enterprise 250, the integrated Ethernet is `hme0`; this is the public network. The following `ifconfig` output shows that only the public interface exists:

```
solaris# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 128.162.2.91 netmask ffffff00 broadcast 128.162.2.255
        ether 8:0:20:d2:29:c5
```

   If the second network does not appear, do the following:

   a. If you do not have the PCI card installed, install it. Refer to your PCI documentation for instructions.

      If your card is already installed, skip to step b.

b. Use the output from the dmesg command to determine the interface name for the private network; look for the network interface that immediately follows the public network; you may wish to search for Found. For example:

```
solaris# dmesg

Feb  6 09:38:36 ue250 last message repeated 42 times
Feb  6 11:38:40 ue250 pseudo: [ID 129642 kern.info] pseudo-device: devinfo0
Feb  6 11:38:40 ue250 genunix: [ID 936769 kern.info] devinfo0 is /pseudo/devinfo@0
Feb  6 11:38:41 ue250 hme: [ID 517527 kern.info] SUNW,hme0 : PCI IO 2.0 (Rev Id = c1) Found
Feb  6 11:38:41 ue250 genunix: [ID 936769 kern.info] hme0 is /pci@1f,4000/network@1,1
Feb  6 11:38:41 ue250 hme: [ID 517527 kern.info] SUNW,hme1 : PCI IO 2.0 (Rev Id = c1) Found
Feb  6 11:38:41 ue250 hme: [ID 517527 kern.info] SUNW,hme1 : Local Ethernet address = 8:0:20:cc:43:48
Feb  6 11:38:41 ue250 pcipsy: [ID 370704 kern.info] PCI-device: SUNW,hme@1,1, hme1
Feb  6 11:38:41 ue250 genunix: [ID 936769 kern.info] hme1 is /pci@1f,2000/SUNW,hme@1,1
```

The second network is hme1; this is the private network, and is displayed after hme0 in the dmesg output. In this example, hme1 is the value needed in step c and in step 5 below.

c. Initialize the private network's interface by using the ifconfig command as follows, where *interface* is the value determined in step b:

ifconfig *interface* plumb

For example:

```
solaris# ifconfig hme1 plumb
```

After performing the plumb, the hme1 interface will appear in the ifconfig output, although it will not contain the appropriate information (the correct information will be discovered after the system is rebooted later in step 8). For example, at this stage you would see the following:

```
solaris# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 128.162.2.91 netmask ffffff00 broadcast 128.162.2.255
        ether 8:0:20:d2:29:c5
hme1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 0.0.0.0 netmask ff000000 broadcast 255.0.0.0
        ether 8:0:20:d2:29:c5
```

5. Create a file named /etc/hostname.*interface*, where *interface* is the value determined in step 4. This file must contain the name of the **private** network. For example:

```
solaris# cat /etc/hostname.hme1
cxfssun3-priv
```

Note: In this scenario, /etc/hostname.hme0 must contain the same value as the /etc/nodename file. For example:

```
solaris# cat /etc/hostname.hme0
cxfssun3
solaris# cat /etc/nodename
cxfssun3
```

---

6. Edit the /etc/netmasks file to include the appropriate entries.

7. *(Optional)* Edit the /.rhosts file if you want to use remote access or if you want to use the connectivity diagnostics provided with CXFS. Ensure that the mode of the .rhosts file is set to 600 (read and write access for the owner only).

   Make sure that the /.rhosts file on each Solaris node allows all of the nodes in the cluster to have access to each other. The connectivity tests execute a ping command from the local node to all nodes and from all nodes to the local node. To execute ping on a remote node, CXFS uses rsh as user root.

   For example, suppose you have a cluster with three nodes: admin0, solaris1, and solaris2. The /.rhosts files could be as follows (the prompt denotes the node name):

```
admin0# cat /.rhosts
solaris1 root
solaris1-priv root
solaris2 root
solaris2-priv root

solaris1# cat /.rhosts
admin0 root
admin0-priv root
solaris2 root
solaris2-priv root
```

```
solaris2# cat /.rhosts
admin0 root
admin0-priv root
solaris1 root
solaris1-priv root
```

8. Reboot the Solaris system:

   ```
   solaris# init 6
   ```

   At this point, `ifconfig` will show the correct information for the private network.

   For example:

```
ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 128.162.2.91 netmask ffffff00 broadcast 128.162.2.255
        ether 8:0:20:d2:29:c5
hme1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 10.1.1.36 netmask ffffff00 broadcast 10.1.1.255
        ether 8:0:20:d2:29:c5
```

Verifying the Private and Public Networks for Solaris

For each private network on each Solaris node in the pool, verify access with the Solaris `ping` command. Enter the following, where *nodeIPaddress* is the IP address of the node:

```
solaris# /usr/sbin/ping -s -c 3 nodeIPaddress
```

For example:

```
solaris# /usr/sbin/ping -s -c 3 128.162.2.91
PING 128.162.2.91: 56 data bytes
64 bytes from cxfssun3.americas.sgi.com (128.162.2.91): icmp_seq=0. time=0. ms
64 bytes from cxfssun3.americas.sgi.com (128.162.2.91): icmp_seq=1. time=0. ms
64 bytes from cxfssun3.americas.sgi.com (128.162.2.91): icmp_seq=2. time=0. ms
64 bytes from cxfssun3.americas.sgi.com (128.162.2.91): icmp_seq=3. time=0. ms
```

Also execute a `ping` on the public networks. If `ping` fails, follow these steps:

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

```
solaris# /usr/sbin/ifconfig eri0
eri0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 128.162.2.127 netmask ffffff00 broadcast 128.162.2.255
        ether 0:3:ba:d:ad:77
```

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

2. Verify that the cables are correctly seated.

Repeat this procedure on each node.

Client Software Installation for Solaris

The CXFS software will be initially installed and configured by SGI personnel. This section provides an overview of those procedures:

- "Solaris Installation Procedure" on page 156
- "Verifying the Solaris Installation" on page 159

Solaris Installation Procedure

Installing the CXFS client software for Solaris requires approximately 20 MB of space.

To install the required software on a Solaris node, SGI personnel will do the following:

1. Read the *SGI InfiniteStorage Software Platform* release notes, CXFS general release notes, and CXFS Solaris release notes in the `/docs` directory on the ISSP DVD and any late-breaking caveats on Supportfolio
2. Verify that the node has been upgraded to Solaris 10 (also known as *SunOS 5.10*) according to the Solaris installation guide. Use the following command to display the currently installed system:

   ```
   solaris# uname -r
   ```

   This command should return a value of 5.10.

3. 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 package on the server will be as follows:

`/usr/cluster/client-dist/`*CXFS_VERSION*`/solaris/10/noarch/SGIcxfs-sol10-v`*CXFS_VERSION*`.pkg`

4. Install the package:

   ```
   solaris# pkgadd -d SGIcxfs*.pkg
   ```

   You must select the package to be installed and then confirm that you want allow the installation of the packages with superuser permission. For example:

```
solaris# pkgadd -d SGIcxfs*.pkg
The following packages are available:
  1  SGIcxfs     SGI CXFS client software
                 (sparc) 5.0.0.3

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1

Processing package instance <SGIcxfs> from
</tmp/SGIcxfs-sol10-v5.0.0.3.pkg>

SGI CXFS client software(sparc) 5.0.0.3
/*
 *  Copyright (c) 2008 Silicon Graphics, Inc.  ALL RIGHTS RESERVED

...
 */
Using </> as the package base directory.
## Processing package information.
## Processing system information.
   3 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <SGIcxfs> [y,n,?] y
```

```
Installing SGI CXFS client software as <SGIcxfs>

## Installing part 1 of 1.
/etc/init.d/cxfs_cluster
/etc/rc0.d/K28cxfs_cluster <symbolic link>
/etc/rc1.d/K28cxfs_cluster <symbolic link>
/etc/rc2.d/S77cxfs_cluster <symbolic link>
/etc/rc3.d/S77cxfs_cluster <symbolic link>
/etc/rcS.d/K28cxfs_cluster <symbolic link>
/usr/cxfs_cluster/bin/cxfs_client
/usr/cxfs_cluster/bin/cxfs_info
/usr/cxfs_cluster/bin/cxfsdump
/usr/cxfs_cluster/bin/frametest
/usr/cxfs_cluster/bin/xvm
/usr/kernel/drv/cell.conf
/usr/kernel/drv/sparcv9/cell
/usr/kernel/drv/sparcv9/xvm
/usr/kernel/drv/xvm.conf
/usr/kernel/fs/sparcv9/cxfs
/usr/kernel/misc/sparcv9/hba_hook
/usr/lib/sparcv9/libgrio.so
/usr/sbin/clmount
/usr/sbin/grioadmin
/usr/sbin/griomon
/usr/sbin/grioqos
/usr/sbin/idbg
/usr/share/man/man1/xvm.1
/usr/share/man/man1m/cxfs_client.1m
/usr/share/man/man1m/cxfs_info.1m
/usr/share/man/man1m/frametest.1m
/var/cluster/cxfs_client-scripts/cxfs-post-mount
/var/cluster/cxfs_client-scripts/cxfs-post-umount
/var/cluster/cxfs_client-scripts/cxfs-pre-mount
/var/cluster/cxfs_client-scripts/cxfs-pre-umount
/var/cluster/cxfs_client-scripts/cxfs-reprobe
[ verifying class <none> ]
## Executing postinstall script.
Starting CXFS services...
cxfs_client daemon started
```

Verifying the Solaris Installation

To verify that the CXFS software has been installed properly, use the `pkginfo` command as follows:

```
pkginfo -l SGIcxfs
```

For example, the following output indicates that the CXFS package installed properly:

```
solaris# pkginfo -l SGIcxfs
   PKGINST:  SGIcxfs
      NAME:  SGI CXFS client software
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  5.0.0.3
   BASEDIR:  /
    VENDOR:  Silicon Graphics Inc.
    PSTAMP:  cxfssun120080213222059
  INSTDATE:  Feb 14 2008 07:15
    STATUS:  completely installed
     FILES:       38 installed pathnames
                   5 directories
                  22 executables
               20179 blocks used (approx)
```

I/O Fencing for Solaris

I/O fencing is required on Solaris 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 Solaris 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 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.

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

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 (comment lines begin with #):

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

To configure fencing, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Start/Stop `cxfs_client` for Solaris

The `/etc/init.d/cxfs_cluster` 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:

```
solaris# /etc/init.d/cxfs_cluster start
```

To stop `cxfs_client` manually, enter the following:

```
solaris# /etc/init.d/cxfs_cluster stop
```

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

```
solaris# /etc/init.d/cxfs_cluster restart
```

Maintenance for Solaris

This section contains the following:

Updating the CXFS Software for Solaris

Note: Before upgrading CXFS software, ensure that no applications on the node are accessing files on a CXFS filesystem.

To upgrade CXFS on a Solaris system, do the following:

1. Remove the current package:

```
solaris# pkgrm SGIcxfs

The following package is currently installed:
   SGIcxfs          SGI CXFS client software
                    (sparc) releaselevel

Do you want to remove this package? [y,n,?,q] y

# Removing installed package instance <SGIcxfs>

This package contains scripts which will be executed with super-user
permission during the process of removing this package.

Do you want to continue with the removal of this package [y,n,?,q] y
# Verifying package dependencies
...
```

2. Reboot the Solaris system:

   ```
   solaris# reboot
   ```

3. Obtain the CXFS update software from Supportfolio according to the directions in *CXFS 5 Administration Guide for SGI InfiniteStorage*

4. Follow the installation instructions to install the new package. See "Client Software Installation for Solaris" on page 156.

Modifying the CXFS Software for Solaris

You can modify the behavior of the CXFS client daemon (`cxfs_client`) by placing options in the `/usr/cxfs_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:

```
solaris# ps -ef | grep cxfs
```

Recognizing Storage Changes for Solaris

On Solaris nodes, the `cxfs-enumerate-wwns` script enumerates the world wide names (WWNs) on the host that are known to CXFS. See "CXFS Mount Scripts" on page 7.

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
```

If you make changes to your storage configuration, you must rerun the HBA utilities to reprobe the storage. See "HBA Installation for Solaris" on page 146.

GRIO on Solaris

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the Solaris 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 node.

A Solaris node can mount a GRIO-managed filesystem and supports node-level reservations. A Solaris 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 11 and the *Guaranteed-Rate I/O Version 2 for Linux Guide*.

XVM Failover V2 on Solaris

Following is an example of the `/etc/failover2.conf` file on Solaris using a QLogic HBA:

```
/pci@1d,700000/SUNW,qlc@1/fp@0,0/ssd@w200400a0b80c268c,1 affinity=1 preferred
/pci@1d,700000/SUNW,qlc@1,1/fp@0,0/ssd@w200400a0b80c268c,1 affinity=1
/pci@1d,700000/SUNW,qlc@1/fp@0,0/ssd@w200500a0b80c268c,1 affinity=2
/pci@1d,700000/SUNW,qlc@1,1/fp@0,0/ssd@w200500a0b80c268c,1 affinity=2
```

In this case:

- `SUNW,qlc@1` is the first port on the PCI card
- `SUNW,qlc@1,1` is the second port on the PCI card
- `200400a0b80c268c` is controller A on the TP9*XXX*
- `200500a0b80c268c` is controller B on the TP9*XXX*

Following is an example using an LSI HBA:

```
<XVM physvol phys/cc_is4000-lun0>
     pci@1f,2000/IntraServer,fc@1,1/ssd@0,0 <dev 130> affinity=1
     pci@1f,2000/IntraServer,fc@1,1/ssd@2,0 <dev 146> affinity=0 preferred

<XVM physvol phys/cc_is4000-lun1>
```

```
     pci@1f,2000/IntraServer,fc@1,1/ssd@0,1 <dev 738> affinity=1 preferred
     pci@1f,2000/IntraServer,fc@1,1/ssd@2,1 <dev 930> affinity=0

<XVM physvol phys/cc_is4000-lun2>
     pci@1f,2000/IntraServer,fc@1,1/ssd@0,2 <dev 746> affinity=1
     pci@1f,2000/IntraServer,fc@1,1/ssd@2,2 <dev 938> affinity=0 preferred

<XVM physvol phys/cc_is4000-lun3>
     pci@1f,2000/IntraServer,fc@1,1/ssd@0,3 <dev 754> affinity=1 preferred
     pci@1f,2000/IntraServer,fc@1,1/ssd@2,3 <dev 946> affinity=0
```

For more information, see "XVM Failover and CXFS" on page 11, the comments in the `/etc/failover2.conf` file, *CXFS 5 Administration Guide for SGI InfiniteStorage*, and the *XVM Volume Manager Administrator's Guide*.

Troubleshooting for Solaris

This section contains the following:

For general troubleshooting information, see Chapter 10, "General Troubleshooting" on page 273 and Appendix D, "Error Messages" on page 301.

The `cxfs_client` Daemon is Not Started on Solaris

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

```
solaris# ps -ef | 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 98 and watch the `cxfs_client` log file for errors.

Filesystems Do Not Mount on Solaris

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 the LUNs. This is usually caused by misconfiguration of the HBA or the SAN fabric:
  - Can the HBA see all of the LUNs for the filesystems it is mounting?
  - Can the operating system kernel see all of the LUN devices?

  See "New Storage is Not Recognized on Solaris" on page 166.
- The `cxfs_client` daemon may not be running. See "The `cxfs_client` Daemon is Not Started on Solaris" on page 164.
- 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 `/usr/cxfs_cluster/bin/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 265.
  - 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.

New Storage is Not Recognized on Solaris

If you have a problem with an HBA, verify that you enabled fabric mode. See "Recognizing Storage Changes for Solaris" on page 162.

Large Log Files on Solaris

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/cxfs_cluster/bin/cxfs_client.options file.

For information about the log files created on server-capable administration nodes, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Changing the CXFS Heartbeat Value on Solaris

To view the CXFS heartbeat value on Solaris, use the following:

```
# echo mtcp_hb_period/D | adb -k
physmem 3df86
mtcp_hb_period:
mtcp_hb_period: 600
```

Using the -k option to the adb(1) debugger causes it to attach to a live kernel. Echoing the command allows you to put it on a single line.

For example, to reset the value to 15 seconds, enter the following (the value is in Hz):

```
# echo mtcp_hb_period/W0t1500 | adb -kw
physmem 3df86
mtcp_hb_period: 0x258           =       0x5dc
```

Reporting Solaris Problems

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

```
solaris# 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.

Also collect 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 5 Administration Guide for SGI InfiniteStorage*.
- If there is a system panic, retain the system core file in `/var/crash/`*hostname* on a Solaris node.
- Output from the `crash` utility.
- `mdb`(1M) modular debugger output:
  - For panics or generated dumps, use the following commands and save the output:

    ```
    $c (or $C)
    $r
    $<msgbuf
    ```

  - For dumps from hangs:

    ```
    $<threadlist
    $c (or $C)
    $r
    $<msgbuf
    ```

- A list of the Solaris patches that have been installed. Use the `showrev` command. The `showrev` command without options prints a summary and the `-p` option lists the revision information about patches.
- A list of the loaded Solaris kernel modules and versions. Use the `modinfo` command.
- Output from the LSI `/usr/sbin/lsiutil` command, which displays the number of LSI HBAs installed, the model numbers, and firmware versions.

If any of the above Solaris tools are not currently installed on your Solaris system, you should install them.

Windows Platforms

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

This chapter contains the following sections:

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
> Settings
> Control Panel

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

Start
> Control Panel

CXFS on Windows

This section contains the following information about CXFS on Windows:

Requirements for Windows

In addition to the items listed in "Requirements" on page 9, CXFS requires at least the following:

- A supported Windows operating system. 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 we therefore expect them to be fully functional, although we have 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
  - 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 2003: 1.26.01
  - Windows XP: 1.26.01
  - Windows Vista: 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 does 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: Business, Enterprise and Ultimate Editions v9.1.7.15 STOR Miniport for both 32– and 6 4–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.

- If two QLogic HBAs are installed for Windows Server 2003, you should also install the QLDirect Filter (8.01.12) in order to facilitate HBA failover and load balancing. If two different model HBAs are installed, you must install drivers for both models.

  Note: If the primary HBA path is at fault during the Windows boot up (for example, if the Fibre Channel cable is disconnected), no failover to the secondary HBA path will occur. This is a limitation of the QLogic driver.

- The following ATTO software from the http://www.attotech.com website:
  - Windows XP, Windows 2003, Windows Vista x86 (32 and 64 bit) version 2.62
  - 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\cxfscp.exe
%ProgramFiles%\CXFS\cxfsdump.exe
%ProgramFiles%\CXFS\grioadmin.exe
%ProgramFiles%\CXFS\griomon.exe
%ProgramFiles%\CXFS\grioqos.exe
%ProgramFiles%\CXFS\xvm.exe
```

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

A single CXFS Client service and a single CXFS filesystem driver are installed as part of the Windows installation. The service and the CXFS filesystem driver 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 265.

For information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 11 and "GRIO on Windows" on page 234.

Log Files and Cluster Status for Windows

This section discusses the following:

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
> Settings
> Control Panel
> Administrative Tools
> Event Viewer

For information about the log files created on server-capable administration nodes, see the *CXFS 5 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 8-5 on page 217) and may be adjusted by following the steps described in "Modifying the CXFS Software for Windows" on page 230.

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 also 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

Figure 8-1 shows an example of the **CXFS Info** window.

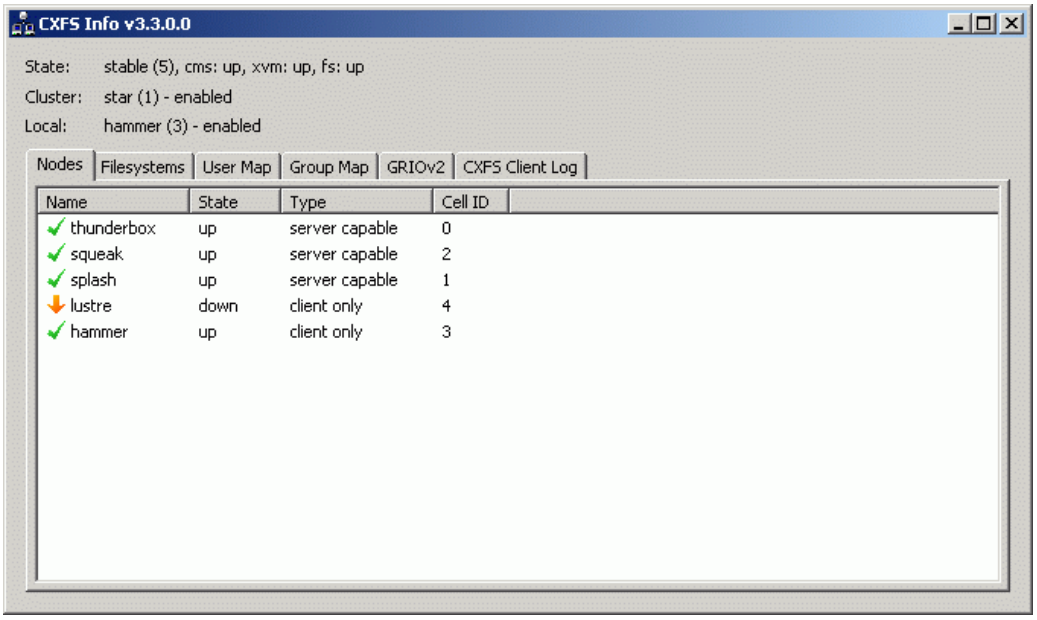

Figure 8-1 CXFS Info Window — Nodes Tab Display

The **CXFS Info** window also provides the following tabs to access further information:

- **Nodes** displays each node in the cluster, its state, and its cell ID number. For more information, see "Verifying the Cluster Status" on page 265.

- **Filesystems** displays each CXFS filesystem, its state, size, and other statistics. Figure 8-2 shows an example.

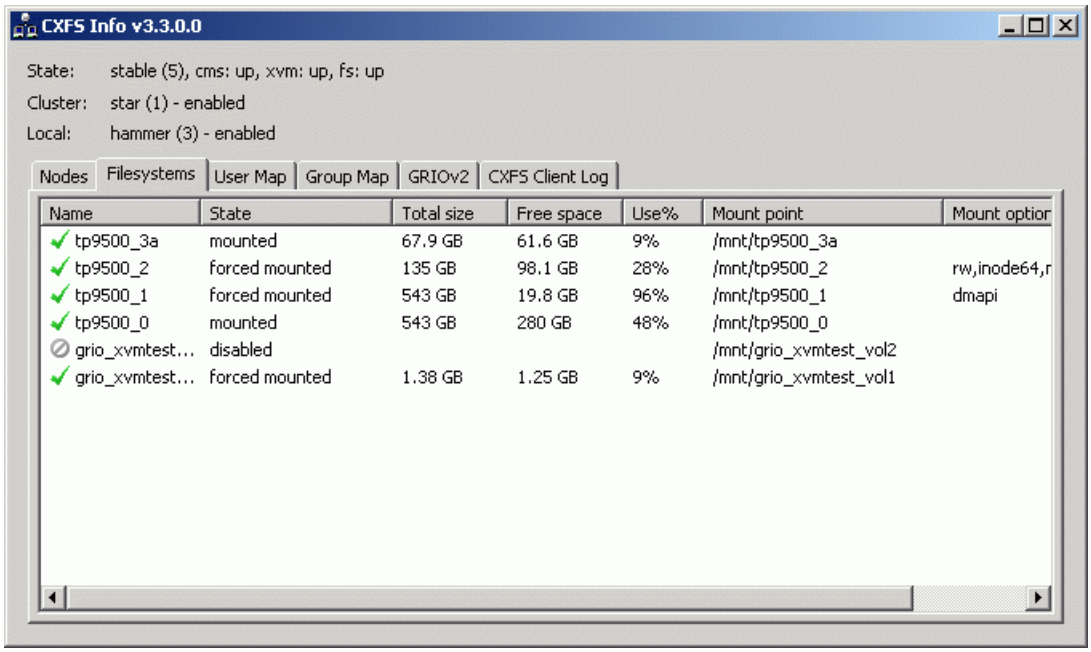

Figure 8-2 CXFS Info Window — Filesystems Tab

- **User Map** displays the usernames that are mapped to UNIX user identifiers.
- **Group Map** displays the groups that are mapped to UNIX group identifiers.
- **GRIOv2 Status** displays each guaranteed-rate I/O (GRIO) stream, its reservation size, and other statistics. See "GRIO on Windows" on page 234.
- **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 8-3 shows an example.

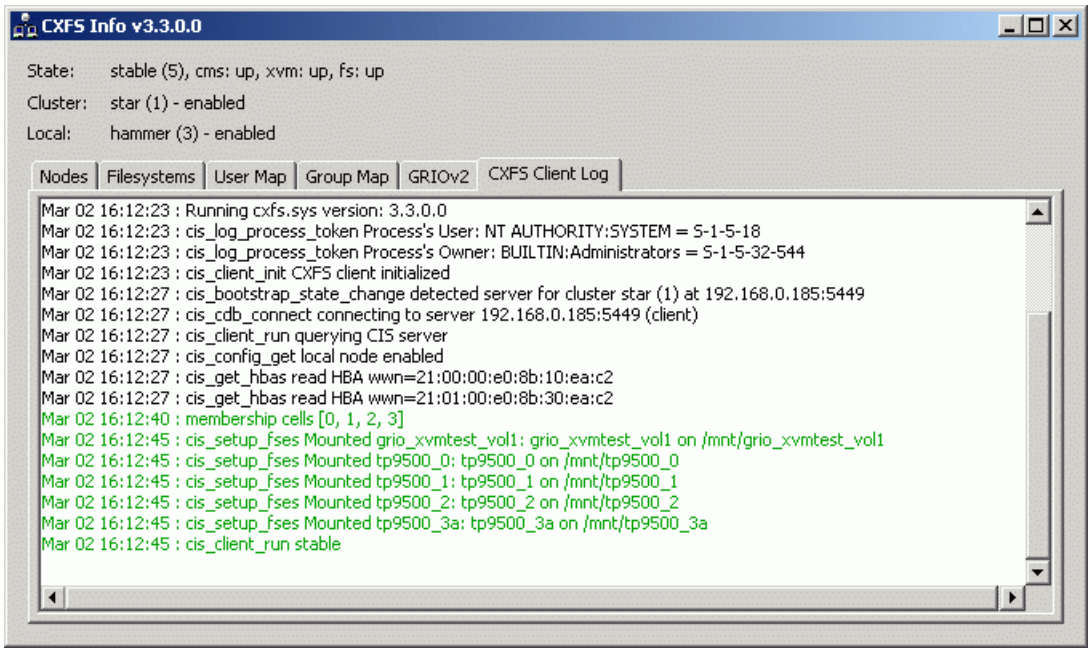

Figure 8-3 CXFS Info Window — CXFS Client Log Tab

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 8-13 on page 234.

Functional Limitations and Considerations for Windows

There are a number of limitations in the CXFS software that are unique to the Windows platform:

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 293.

Warning: DiskManager for Windows Vista and Windows 2008 Destroys Data

After CXFS is installed on the Windows Vista or Windows 2008 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 2008 Marks Newly Discovered Disks Offline

In a Windows 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`.

Use of TPSSM

If installing TPSSM on a Windows node, you must choose **Custom install** and uncheck the **TPSSM RDAC** option. This will prevent the RDAC pseudo/virtual LUNs from being installed onto the system (installing these LUNs would have a detrimental affect on XVM failover V2 for the Windows node).

Although the Windows platform allows access to properties on the disks and LUNs of the connected RAIDS, SGI recommends that you use only the TPSSM tools to perform all RAID configuration.

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 250.

- 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

# 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:

```
windows# /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

# /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 mail 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

SGI recommends that you enable the forced unmount feature on CXFS filesystems. See "Enable Forced Unmount When Appropriate" on page 19 and "Forced Unmount of CXFS Filesystems" on page 263.

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.

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

Windows XP and Windows 2003 Server (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 2003 Server 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 restart 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

By default, User Account Control is enabled for Windows Vista, 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 214.

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 2003 Server, do the following:

1. Select the following:

   Start
   > Settings
   > Control Panel
   > Power Settings

2. Select the **Power Options** item.
3. In the **Plugged in** scheme, select **Never** for **Turn off hard disks**

On Windows Vista and Windows 2008, do the following:

1. Select the following:

   Start
   > Settings
   > Control Panel
   > Power Settings

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** value setting to **Never**.
7. Click **OK** to save the changes.

Windows Time Service Default Synchronization

The Windows Time Service is capable of synchronizing with NTP servers, but the default configuration only 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 14:

- 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 187)

  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 1.
- 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 201.

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

The CXFS software supports several user identification mechanisms, which are described in "User Identification Mapping Methods for Windows" on page 192. Windows user and group names that match entries in the configured user list will be mapped to those user IDs (UIDs) and group IDs (GIDs).

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 one 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. If you select this method, you must install the `/etc/passwd` and `/etc/group` files immediately after installing the CXFS software, as described in "Performing User Configuration for Windows" on page 223.

- **ldap_activedir:** Windows Active Directory server with Services for UNIX (SFU) installed, which uses lightweight directory access protocol (LDAP).

  The **ldap_activedir** 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 247.

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

  – Windows 2003 Server 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.

For an example of the window, see Figure 8-5 on page 217.

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 230.

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.

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, otherwise 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:

- `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`.
- `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.

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 8-1. (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 8-1.

Table 8-1 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 mask 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 mask during CXFS installation or later by modifying the installation. For more information, see "Client Software Installation for Windows" on page 214 and "Inheritance and Default ACLs for Windows" on page 199.

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 mask 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:

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 to modify the registry. Do the following:

1. Back up the registry before making any changes.
2. Click **Start**, select **Run**, and open the `adit.exe` program.
3. Select **HKEY_LOCAL_MACHINE** and follow the tree structure down to the parameter you wish to change.

4. After making the change, reboot the system so that the change takes affect.

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 199.

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

CXFS for Windows prior to CXFS 3.2 broke down large direct I/O requests into requests no larger than 4 MB, which would result in additional network traffic to the metadata server and potentially multiple extents on disk when it could allocate a single extent. This limit has been increased to 16 MB and can be configured by modifying a new registry key in CXFS 3.2 and later.

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. If this parameter is not defined, it defaults to `0x1000000`, which is 16 MB. The upper bound for Windows is just under 64 MB.

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
   > ql2*xxx*
   > Parameters
   > Device

2. Double-click on **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. Shutdown and reboot the system.

Command Tag Queueing (CTQ) Used by the QLogic HBA

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 QLogic HBA will not recognize the CTQ setting placed on the disk by Linux nodes.

To configure the CTQ for the QLogic HBA, do the following:

1. Start the **SANsurfer Manager** program and connect.
2. Click the first adapter, for example **Adapter QLA2342**.
3. Click the **Settings** tab.
4. Click the **Select Settings section** drop down list and select **Advanced Adapter Settings**.
5. Enter a value in the range `1` through `256` in the **Execution Throttle** up-down edit control (the default is `16`). The value describes how many commands will be queued by the HBA.
6. Repeat step 5 for each HBA port used for CXFS.
7. Click **Save**, enter the password (`config` by default), and click **OK**.
8. Close the **SANsurfer Manager** program.
9. Reboot the system.

If you do not have SANsurfer Manager installed, you can also set the execution throttle in the QLogic BIOS during boot-up. To do this, press `ctrl-q` when you see the QLogic BIOS message. See the QLogic 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.

Memory-Mapped Files Flush Time for Windows

The `MmapFlushTimeSeconds` tunable allows the CXFS memory manager to periodically relinquish references to files that are currently memory-mapped but are not in use. This enables other nodes in the cluster to access the files.

The `MmapFlushTimeSeconds` registry value specifies the length of time in seconds that a CXFS flushing thread periodically awakens to flush the memory-mapped files that are not in use. The larger the value the longer the node will hold onto the tokens

for that file. The default is `30` seconds. Setting the value to `0` disables the flushing of memory-mapped files. (A negative value is invalid and will cause the setting to return to the default `30` seconds.)

Caution: Change the value for this parameter with caution. Increasing the `MmapFlushTimeSeconds` time can cause other nodes to increase their access wait time if memory-mapping coherency is enabled. Decreasing the value might cause unnecessary flushing and invalidation operations, which will hurt the system performance.

In **regedit**, navigate and edit the following value:

HKEY_LOCAL_MACHINE
> SYSTEM
> CurrentControlSet
> Services
> CXFS
> Parameters
> MmapFlushTimeSeconds

HBA Installation for Windows

The QLogic Fibre Channel host bus adapter (HBA) should be installed according to the QLogic hardware and driver installation instructions.

Information regarding large logical unit (LUN) support under Windows can be found in the QLogic documentation and also in 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

This section discusses the following:

Confirming the QLogic HBA Installation for Windows

To confirm that the QLogic HBA and driver are correctly installed, select the following to display all of the logical units (LUNs) visible to the HBA and listed within the Device Manager :

Start
> Settings
> 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 QLogic HBA within the Device Manager may differ. After it is located, any LUNS attached will be listed beneath it.

Configuring Multiple HBAs for Load Balancing on Windows

The QLogic HBA can be configured to mask particular targets so that I/O to one target will always use one HBA port, while I/O to another target will use another HBA port. This procedure assumes that the CXFS driver is already installed and working properly with one HBA.

Note: QLogic only supports load balancing of two or more HBAs when all the HBAs have Fibre Channel connections to the LUNs on startup. If the connection to one of the HBAs is not present upon boot, this feature may not function correctly.

To configure two HBAs for static load balancing, do the following:

1. Disable fencing for this node.
2. Determine the worldwide port name (WWPN) of the current adapter:
   a. Install SANsurfer QLogic Agent and Manager
   b. Run SANsurfer to determine the WWPN
   c. Record the WWPN on paper
3. Shut down Windows.
4. Install the second HBA and start Windows.
5. If the second HBA is a different model from the original one, install its mini port driver (for example, `ql2300.sys`).
6. Start the QLogic SANsurfer Manager and verify that two HBAs are detected. Verify that both of them mirror the same devices and logical units (LUNs). Notice that both HBAs have the same worldwide node name (WWNN) but different WWPNs. The original HBA can be recognized by its WWPN recorded in step 2.
7. If you are using SanSurfer, set the persistent binding to bind the target to the target ID. For more information, see "Mapping XVM Volumes to Storage Targets on Windows" on page 240[3] Otherwise, verify the driver parameters for the QLogic HBAs. In **regedit**, navigate to the following value:

   HKEY_LOCAL_MACHINE
   > SYSTEM
   > CurrentControlSet
   > Services
   > ql2*xxx*
   > Parameters
   > Device

   There should be a value named `DriverParameters`. This must contain at least the following semicolon-separated parameters:

   ```
   Buschange=0;FixupInquiry=1
   ```

   It will typically include `UseSameNN=1` as well. If the `Buschange` and `FixupInquiry` values are not there or are incorrect, edit the parameter list to correct them. Do not delete any other parameters.

8. Configure the HBA port (click **Configure**).

   Note: Ignore the following message, which appears when HBA/LAN configuration is done for the first time (line breaks added here for readability):

   ```
   An invalid device and LUN configuration has been detected. Auto
   configure run automatically. click OK to continue.
   ```

   The `HBA0` devices are automatically set to be visible for Windows applications (notice the open eye) and `HBA1` devices are set to be invisible (notice the closed eye).

9. Select the first device in the table, right click, and then select **Configure LUN(s)**.

   In the new window, select the following:

   Tools
   > Load Balance
   > All LUNs

   This will statically distribute the LAN's traffic load that is associated with this device between the two HBAs.

   Repeat step 9 for each of the other HBA devices.

10. Click **Apply** to save the new configuration.

11. Update the switch port information. Reenable fencing.

12. Reboot Windows.

For more information about using the CXFS GUI or `cxfs_admin` to perform these tasks, see *CXFS 5 Administration Guide for SGI InfiniteStorage*.

Configuring HBA Failover for Windows 2003

The QLogic HBA on Windows 2003 also supports a mechanism to configure the automatic failover to another HBA port if the configured port is no longer able to see the target.

Note: QLogic only supports failover of two or more HBAs when all the HBAs have Fibre Channel connections to the LUNs on startup. If the connection to one of the HBAs is not present upon boot, this feature may not function correctly.

To configure two HBAs for failover, do the following:

1. Install the QLdirect driver v8.01.12 by following all the default settings for the installation and verify that the CXFS node still operates normally.
2. Perform the procedure in "Configuring Multiple HBAs for Load Balancing on Windows" on page 209. With QLdirect installed, the targets can be masked but will also failover to another port if a connection is lost.

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

A private network is required for use with CXFS. See "Use a Private Network" on page 16.

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 XP SP2 Firewall for Windows

The Windows XP firewall will prevent a CXFS Windows node from achieving membership unless several ports are opened using the following applet:

Start
> Settings
> 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

The CXFS software will be initially installed and configured by SGI personnel. This section provides an overview of those procedures. You can use the information in this section to verify the installation.

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 general 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` or as an account with administrative privileges.

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 Only)* Disable User Account Control (requires administrator privileges). By default, User Account Control is enabled for Windows Vista, but it is not appropriate for use with CXFS; you should disable it before you install CXFS on a Windows Vista node. Do the following:

   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. Press the **Restart Now** button in the dialog box that says you must restart your computer to apply these changes. This will reboot your system.

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 on the **setup.exe** installation program to execute it.

7. Acknowledge the software license agreement when prompted and read the Windows release notes, which may contain corrections to this guide.

8. Install the CXFS software, as shown in Figure 8-4. If the software is to be installed in a nondefault directory, click **Browse** to select another directory. Click **Next** when finished.

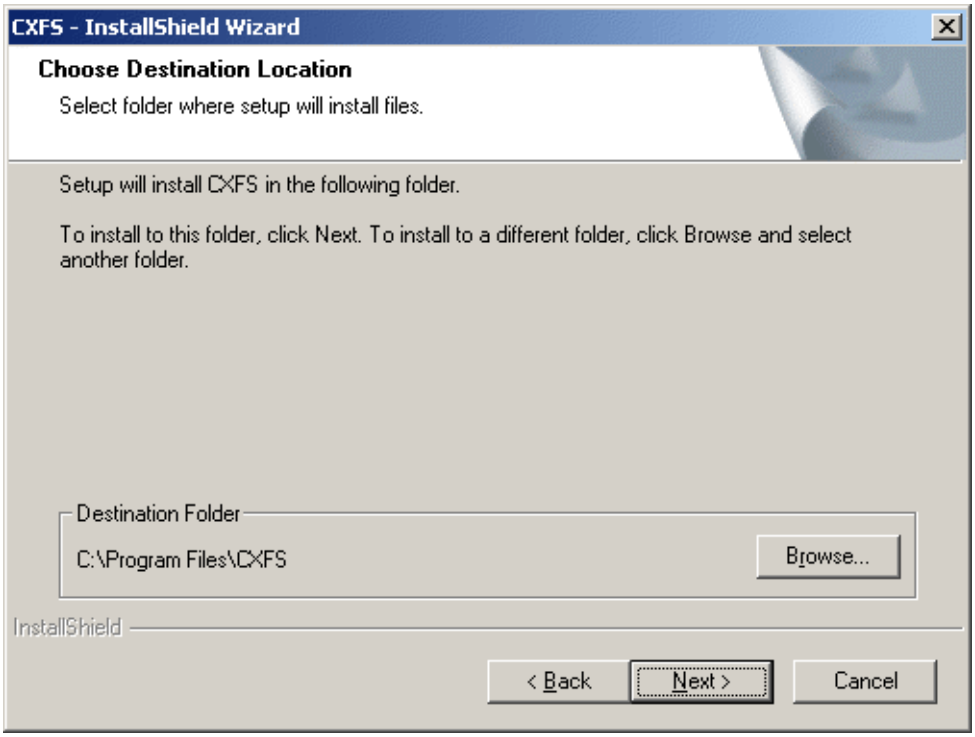

Figure 8-4 Choose Destination Location

9. Enter details for the following fields as shown in Figure 8-5 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 199.
   - **User ID mapping lookup sequence:** choose the appropriate primary and (optionally) secondary method. See "User Identification Mapping Methods for Windows" on page 192.
   - **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, from the command line type `cxfs_client -h`.

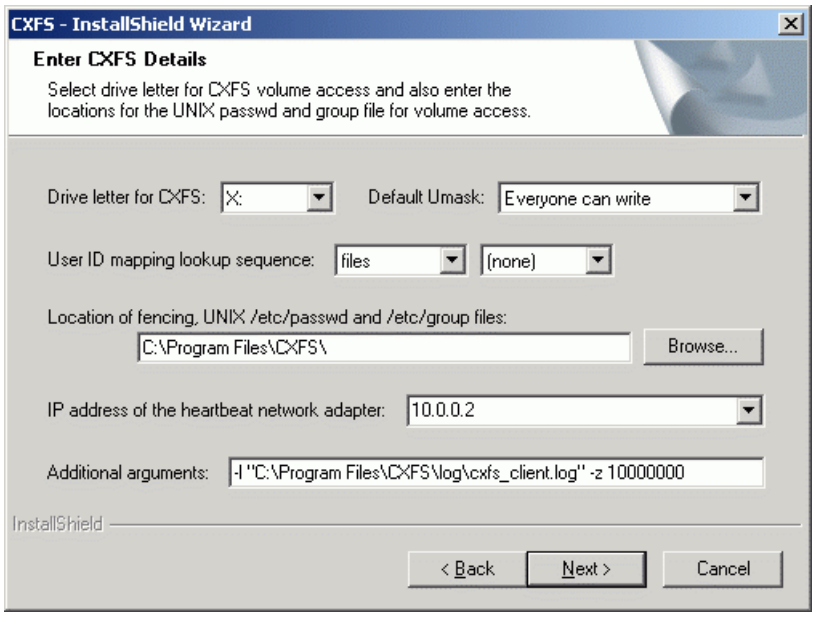

Figure 8-5 Enter CXFS Details

10. If you select **ldap_activedir** as the user ID mapping method, the dialog in Figure 8-6 is displayed after you click **Next**.

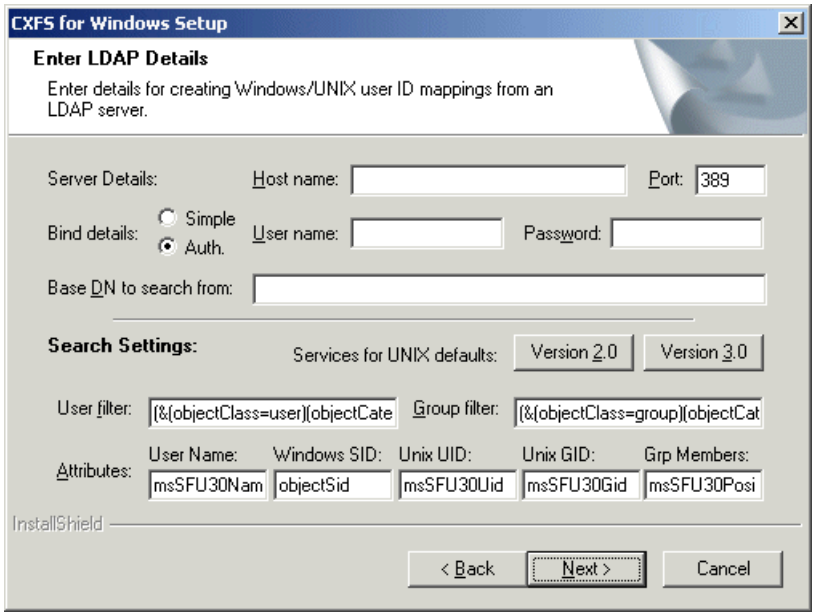

Figure 8-6 Active Directory Details

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 8-7 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**.

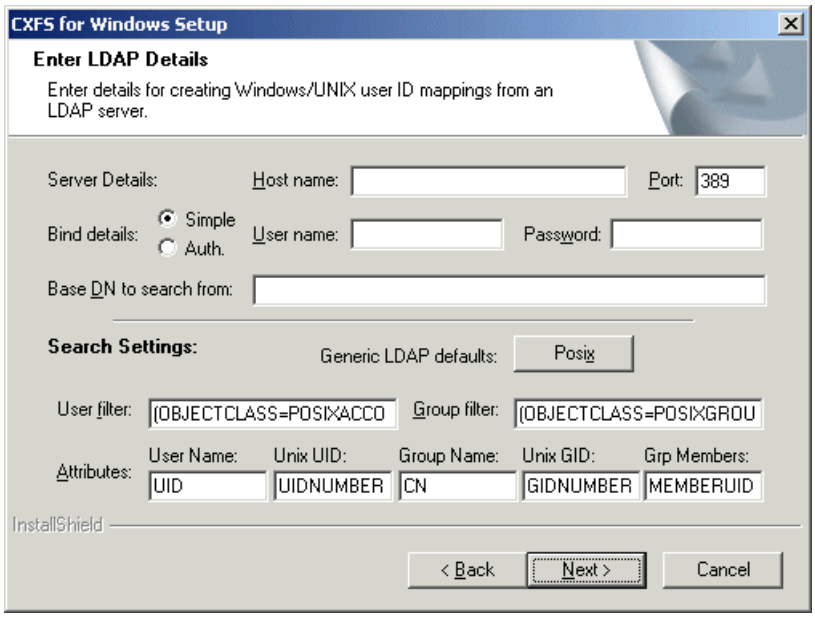

Figure 8-7 Generic LDAP Details

12. Review the settings, as shown in Figure 8-8. If they appear as you intended, click **Next**. If you need to make corrections, click **Back**.

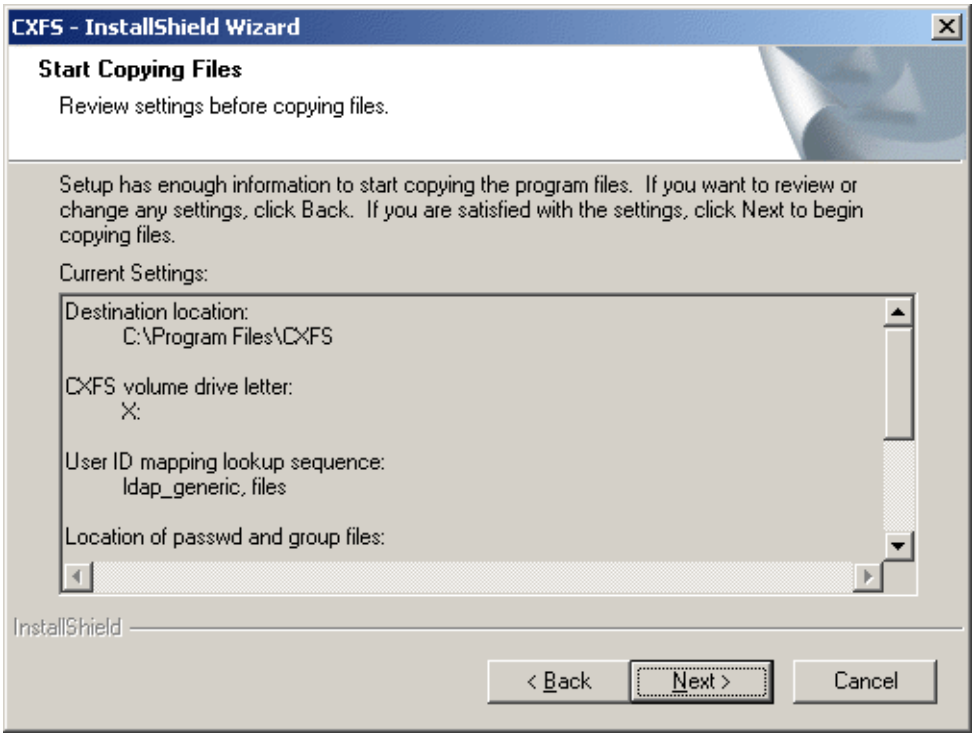

Figure 8-8 Review the Settings

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 8-9. 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.

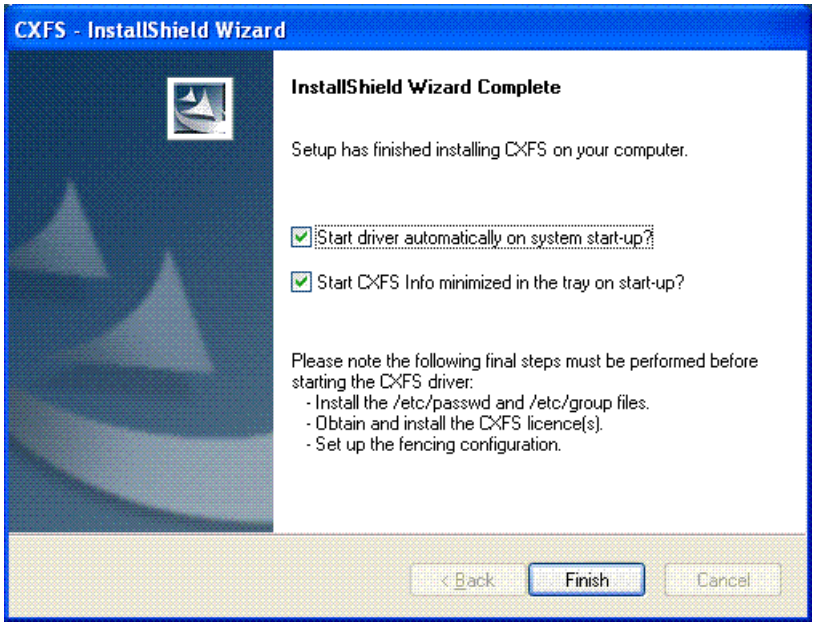

Figure 8-9 Start CXFS Driver

14. Choose to restart your computer later if you need to install `/etc/passwd` and `/etc/group` files or set up fencing; otherwise, choose to restart your computer now. The default is to restart later, as shown in Figure 8-10. (CXFS will not run until a restart has occurred.)

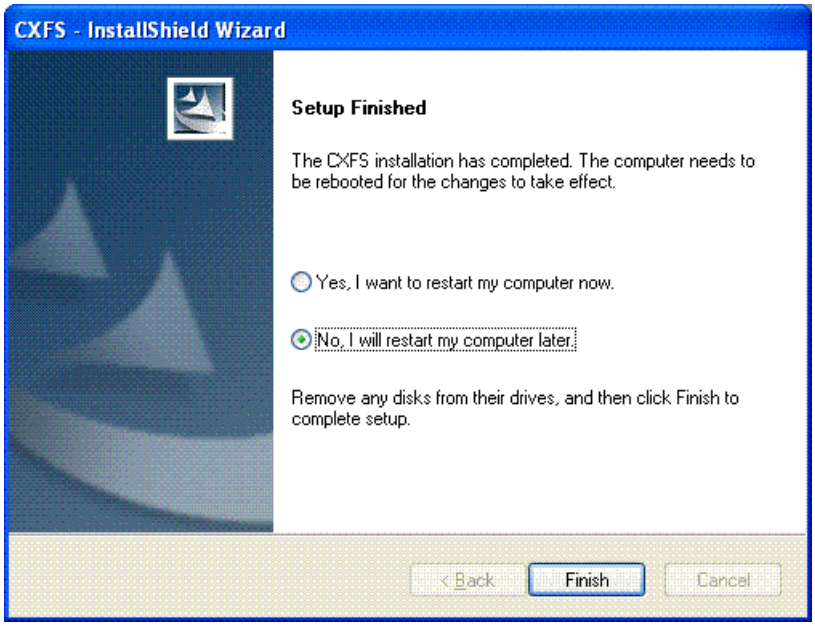

Figure 8-10 Restart the System

Postinstallation Steps for Windows

This section discusses the configuration steps that you should perform after installing CXFS software but before restarting 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 223
- "Performing User Configuration for Windows" on page 223

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 restarting it, you must install the `/etc/passwd` and `/etc/group` files from a server-capable administration node 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 223.

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 191.

If only the `Administrator` user is mapped, see "CXFS Client Service Cannot Map Users other than `Administrator` for Windows" on page 247.

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 can 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 225
- "Determining WWPN for a Brocade Switch" on page 227

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    Loc   Local                   Rem  Remote                  Unique
  Number  Type  PortWWN                 Type NodeWWN                 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 Node WWN` 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 5 Administration Guide for SGI InfiniteStorage*.

Determining 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 5 Administration Guide for SGI InfiniteStorage*.

Note: You could also use SANsurfer to determine the WWPN.

Start/Stop the CXFS Client Service for Windows

The CXFS Client service is automatically started when a Windows node is restarted. 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
> Settings
> 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

To change the location of the software and other configuration settings that were requested in "Client Software Installation for Windows" on page 214, perform the following steps:

1. Select the following:

   Start
   > Settings
   > Control Panel
   > Add/Remove Programs
   > CXFS
   > Add/Remove
   > Modify

   Figure 8-11 shows the screen that lets you modify the software.

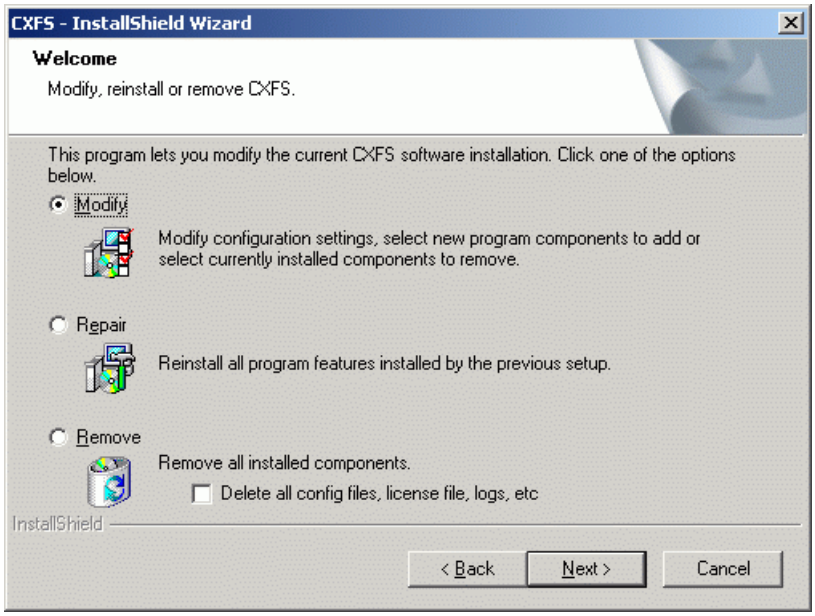

Figure 8-11 Modify CXFS for Windows

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. Restart the Windows node, which causes the changes to take effect.

Updating the CXFS Software for Windows

To upgrade the CXFS for Windows software, perform the following steps:

1. Obtain the CXFS update software from Supportfolio according to the directions in the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

2. 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 will be as follows:

`/usr/cluster/client-dist/`*CXFS_VERSION*`/windows/all/noarch/setup.exe`

3. Double-click on 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 8-12 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 214.

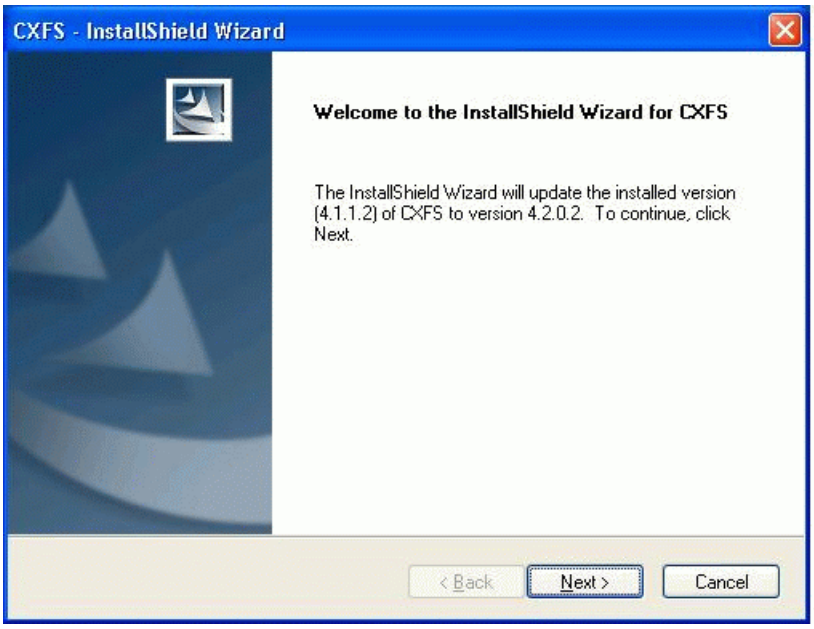

Figure 8-12 Upgrading the Windows Software

5. Restart the Windows node. The upgraded software will not activate until the Windows node is restarted.

Removing the CXFS Software for Windows

To remove the CXFS for Windows software, first ensure that no applications on this node are accessing files on a CXFS filesystem. Then, select the following sequence to remove all installed files and registry entries:

Start
> Settings
> Control Panel
> Add/Remove Programs
> CXFS
> Add/Remove
> Remove

Figure 8-11 on page 231 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**.

You should then restart the Windows node. This will cause the changes to take effect.

Downgrading the CXFS Software for Windows

To downgrade the CXFS software, follow the instructions to remove the software in "Removing the CXFS Software for Windows" on page 233 and then install the older version of the software as directed in "Client Software Installation for Windows" on page 214.

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.

Recognizing Storage Changes for Windows

If you make changes to your storage configuration, you must rerun the HBA utilities to reprobe the storage. See "HBA Installation for Windows" on page 208.

If new storage devices are added to the cluster, you must reboot the Windows node in order to discover those devices.

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 8-13 shows an example of the **CXFS Info** display for GRIO.

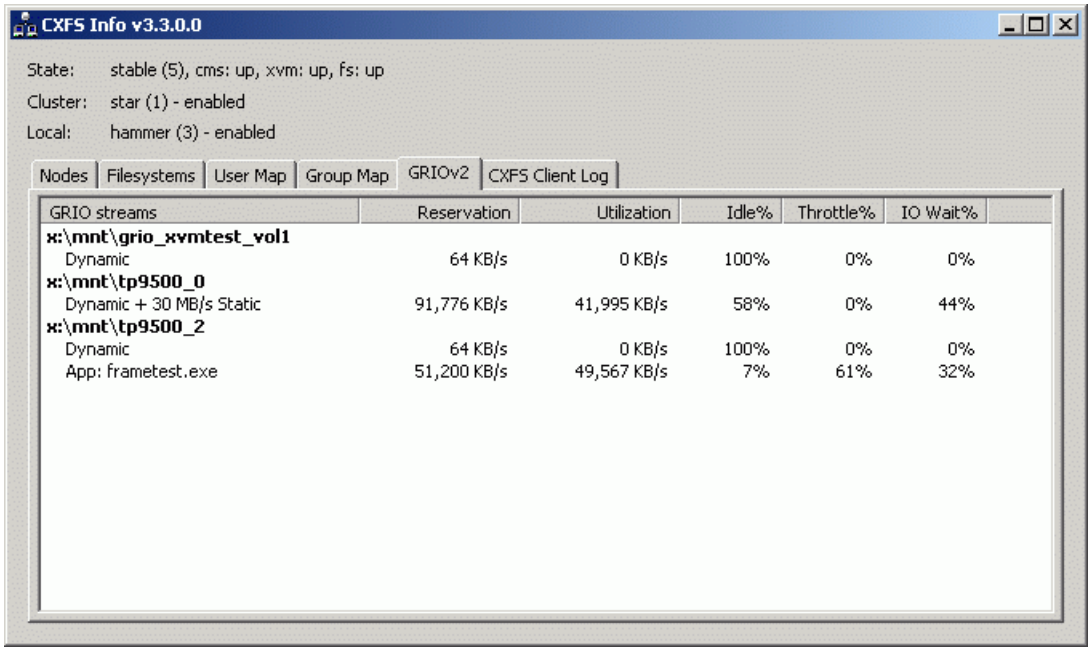

Figure 8-13 CXFS Info Display for GRIO for Windows

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 11 and the *Guaranteed-Rate I/O Version 2 for Linux Guide*.

XVM Failover V2 on Windows

Note: You must not install RDAC pseudo/virtual LUNs onto the Windows client. See "Use of TPSSM" on page 183.

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 240.

   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. Verify that the `failover2.conf` file has `affinity=0` set for the target ID corresponding to controller A and `affinity=1` set for the target ID corresponding to controller B. This is the default setting, but you must make sure that the settings are consistent across the cluster.

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. So there are 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=1
/dev/xscsi/pci08.03.1/node200500a0b82925e2/port1/lun0/disc affinity=1
/dev/xscsi/pci08.03.1/node200400a0b82925e2/port2/lun0/disc affinity=0
/dev/xscsi/pci08.03.1/node200400a0b82925e2/port1/lun0/disc affinity=0 preferred <current path>
```

In this configuration, controller A (`node200400a0b82925e2`) has an affinity of 0, controller B has an affinity of 1. 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 examples shown in the following sections (the format varies by the Windows OS version):

- "Windows XP SP2 and Windows 2003 Server R2 SP1 `failover2` Example " on page 237
- "Windows 2003 Server R2 SP2 and Windows Vista `failover2` Example" on page 239

For more information, see "XVM Failover and CXFS" on page 11, the comments in the `failover2.conf` file, *CXFS 5 Administration Guide for SGI InfiniteStorage*, and the *XVM Volume Manager Administrator's Guide*.

Windows XP SP2 and Windows 2003 Server R2 SP1 `failover2` Example

Windows XP SP 2 and Windows 2003 Server 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=0 preferred
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=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000 <dev 261> affinity=1
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 preferred
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&010 <dev 196> affinity=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000 <dev 176> affinity=1
```

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 0; controller B has an affinity of 1.

Windows 2003 Server R2 SP2 and Windows Vista `failover2` Example

Windows 2003 Server R2 SP2 and Windows Vista `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&00300 <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=0 preferred
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=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000000 <dev 261> affinity=1
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 preferred
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000100 <dev 196> affinity=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000000 <dev 176> affinity=1
```

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 0; controller B has an affinity of 1.

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 8-14.
- 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.

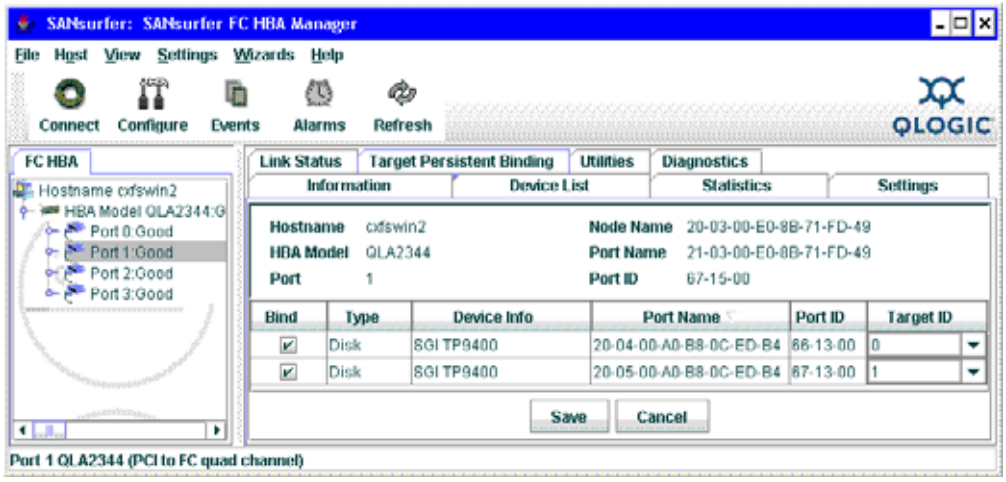

Figure 8-14 QLogic SANsurfer (Copyright QLogic® Corporation, all rights reserved)

Troubleshooting for Windows

This section discusses the following:

Also see:

- The Windows `cxfsdump` documentation located at `%ProgramFiles%\CXFS\cxfsdump.html`
- Chapter 10, "General Troubleshooting" on page 273

Verification that the CXFS Software is Running Correctly for Windows

To verify that the CXFS software is running correctly on a Windows node, do the following:

- Verify that the CXFS driver has started by selecting the following:

  Start
  > Settings
  > Control Panel
  > Administrative Tools
  > Computer Management
  > System Tools
  > Device Manager

  To show non-plug-and-play devices, select the following:

  View
  > Show hidden devices

  To show the CXFS driver, select the following:

  Non-Plug and Play Devices
  > CXFS
  > Properties

- Verify that the CXFS Client service has started by selecting the following:

  Start
  > Settings
  > 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 the following reasons:

- 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. For information specific to Windows, see "HBA Problems" on page 247.
  - 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
    > Settings
    > 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.
  - 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 on 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 265.
  - 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

Also, check the **CXFS Client Log** in **CXFS Info** for mount errors. They will be highlighted in red.

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 Sysinternals Filemon 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 about Filemon, see:

  http://www.sysinternals.com

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 182). 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 SysInternals Filemon 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 Filemon, see:

http://www.sysinternals.com

Note: There are some problems that will not be visible in a Sysinternals Filemon 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.

HBA Problems

If you have a problem with an HBA, check the following:

- Has plug-and-play been disabled?

  Plug-and-play functionality, which would normally discover new devices, is disabled by the QLogic HBA software so that it can perform path failover without Windows attempting to manage the change in available devices. Disabling the plug-and-play feature also enables CXFS to map CXFS volumes to the same devices if a Fibre Channel path was lost and then reestablished. If HBA path failover or CXFS rediscovering XVM volumes and filesystems does not appear to work, verify that plug-and-play is disabled.

- Are there QLogic management tool event and alarm log messages? Select the following:

  Start
  > Programs
  > QLogic Management Suite
  > SANsurfer

Also see "Recognizing Storage Changes for Windows" on page 234 and "Inability to Mount Filesystems on Windows" on page 243.

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
   > Settings
   > 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 the Client-Only Nodes" on page 262.

- 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 265.
- Check that the CXFS Client service has started. See "Start/Stop the CXFS Client Service for Windows" on page 228 and "Verification that the CXFS Software is Running Correctly for Windows" on page 243.
- 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 10MB.)

To verify that log rotation is enabled, check the **Addition** arguments by modifying the installation (see "Modifying the CXFS Software for Windows" on page 230) 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 228 for information on how to stop and start the CXFS Client service.

Windows Failure on Restart

If the CXFS Windows node fails to start and terminates in a blue screen, restart 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. Restart the computer.

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 then 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.

Windows Vista Node Loses Membership Due to Hibernation

If the Windows Vista node hibernates, it will lose membership in the CXFS cluster. Hibernation is turned on by default for Windows Vista and must be modified.

Do the following:

1. Select the following:

   Start
   > Settings
   > Control Panel
   > Power Options

2. Select the **High Performance** radio button.
3. Click **OK**.

Alternatively, you can use the following command:

```
C:\> powercfg -S SCHEME_MIN
```

Windows Vista Node Appears to be in Membership But Is Not

If the Windows Vista node appears to be in membership when the `cxfs_info` command is run from the Windows Vista 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 for Windows Vista).

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 214.

Windows Vista Node Unable to `cd` to a Mounted Filesystem

If you are unable to use the `cd` command on a Windows Vista 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 Windows Vista). For example, using a `cygwin` shell:

```
vista$ 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 214.

Slow Installation of Windows Vista or Windows 2008

If the installation of the Windows Vista or Windows 2008 operating system seems to take a long time or does not complete, it may be caused by the HBAs or SAN fabric.

You can resolve this problem by using the following steps:

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

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 254

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 5 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.

To do this, click the right mouse button on the **My Computer** icon and select the following:

Properties
> Advanced
> Startup and Recovery
> Write debugging information to

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.

These changes will take affect only after the node is restarted.

Saving Application Crash Dumps for Windows Vista and Windows 2008

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 on 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 254), 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.

In **regedit**, navigate to the following value:

HKEY_LOCAL_MACHINE
> SYSTEM
> CurrentControlSet
> Services
> i8042prt
> Parameters

Add a new entry by selecting the following:

Edit
> Add Value

Enter the following information:

- **Value Name:** `CrashOnCtrlScroll`
- **Data Type:** `REG_DWORD`
- **Value:** `1`

These changes will take affect only after the node is restarted.

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

Cluster Configuration Tasks to Include 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:

For detailed configuration instructions, see the *CXFS 5 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:

- AIX: use the value displayed by `/usr/bin/hostname`
- IRIX: use the value displayed by `/usr/bsd/hostname`
- Linux: use the value displayed by `/bin/hostname`
- Mac OS X: use the value displayed by `/bin/hostname`
- Solaris: use the value displayed by `/bin/hostname`
- Windows: select the following:

  Start
  > Settings
  > Network and Dial-up Connections
  > Advanced
  > Network Identification

When you specify that a node is running an operating system other Linux, the node will automatically be defined as a client-only node and you cannot change it. (These nodes cannot be potential metadata servers and are not counted when calculating the CXFS kernel membership quorum.) For client-only nodes, you must specify a unique node ID.

For example, the following shows the entries used to define a Solaris node named `solaris1` in the `mycluster` cluster:

```
# /usr/cluster/bin/cxfs_admin -i mycluster
cxfs_admin:mycluster> create node name=solaris1 os=solaris private_net=192.168.0.178
Event at [ Jan 21 15:58:02 ]
Node "solaris1" has been created, waiting for it to join the cluster...
Waiting for node solaris1, current status: Inactive
Waiting for node solaris1, current status: Establishing membership
Waiting for node solaris1, 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? solaris1
   os? solaris
   private_net? 192.168.0.178
Event at [ Jan 21 15:59:10 ]
Node "solaris1" has been created, waiting for it to join the cluster...
Waiting for node solaris1, current status: Inactive
Waiting for node solaris1, current status: Establishing membership
Waiting for node solaris1, current status: Probing XVM volumes
Operation completed successfully
```

When you specify that a node is running an operating system other Linux, the node will automatically be defined as a client-only node and you cannot change it. (These nodes cannot be potential metadata servers and are not counted when calculating the CXFS kernel membership quorum.) For client-only nodes, you must specify a unique node ID if you use the GUI; `cxfs_admin` provides a default node ID.

The following shows a `cxfs_admin` example in `basic` mode:

```
cxfs_admin:mycluster> create node
Specify the attributes for create node:
 name? cxfsopus5
 os? Linux
 private_net? 10.11.20.5
 type? client_only
Event at [ Jan 21 15:60:10 ]
Node "cxfsopus5" has been created, waiting for it to join the cluster...
```

For details about these commands, see the *CXFS 5 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`.

After you define all of the client-only nodes, you must add them to the cluster.

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 the Client-Only Nodes" on page 262.

Defining the Switch for I/O Fencing

You are required to use I/O fencing on client-only nodes in order to protect data integrity. 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:

```
admin# 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.

```
admin# 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:

```
admin# 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

admin# 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 on the Client-Only Nodes (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 on the Client-Only Nodes

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=solaris2
  ```

  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 19 and "Forced Unmount of CXFS Filesystems" on page 263.

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 5 Administration Guide for SGI InfiniteStorage*.

Restarting the Windows Node

After completing the steps in "Postinstallation Steps for Windows" on page 222 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 (Nodes Other Than Windows)

To verify general connectivity in a multicast environment (other than on a Windows node), you can execute a Linux `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`, `cmond`, `cad`, and `crsd`) or the `cxfs_client` daemon is running.

For example, to see the response for two packets sent from Solaris IP address `128.162.240.27` to the multicast address for CXFS heartbeat and ignore loopback, enter the following:

```
solaris# ping -i 128.162.240.27 -s -L 224.0.0.250 2
```

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 [ Jan 26 12:10:23 ]
Cluster         : clusterOne
Tiebreaker      :
Client Licenses : enterprise   allocated 0 of 256
                  workstation  allocated 2 of 50
------------------  --------  --------  ----------------------------------------------
Node                Cell ID   Age       Status
------------------  --------  --------  ----------------------------------------------
bert *              1         5         Stable
cxfsxe5 *           0         26        Stable
cxfs3               4         0         Disabled
penguin17           2         1         Stable
pg-27               3         12        Stable

------------------  ------------------  ----------------------------------------------
Filesystem          Mount Point         Status
------------------  ------------------  ----------------------------------------------
zj01s0              /mnt/zj01s0         Mounted (cxfsxe5) [4 of 5 nodes]
zj01s1              /mnt/zj01s1         Unmounted
zj0ds2              /mnt/zj0ds2         Mounted (cxfsxe5) [2 of 3 nodes]

------------------  ----------  ------------------------------------------------------
Switch              Port Count  Known Fenced Ports
------------------  ----------  ------------------------------------------------------
brocade26cp0        192         24, 25, 223
```

The following example for a different cluster shows `clconf_info` output:

```
admin# /usr/cluster/bin/clconf_info
Event at [2004-05-04 19:00:33]

Membership since Tue May  4 19:00:33 2004
_____________ ______ ________ ______ ______
Node          NodeID Status   Age    CellID

_____________ ______ ________ ______ ______
cxfs4              1 up           27      2
cxfs5              2 up           26      1
cxfs6              3 up           27      0
cxfswin4           5 up            1      5
cxfssun3           6 up            0      6
cxfsmac3.local.   17 up            0      7

_____________ ______ ________ ______ ______
2 CXFS FileSystems
/dev/cxvm/vol0 on /mnt/vol0  enabled  server=(cxfs4)  5
client(s)=(cxfs6,cxfs5,cxfswin4,cxfssun3,cxfsmac3.local.)  status=UP
/dev/cxvm/vol1 on /mnt/vol1  enabled  server=(cxfs5)  5
client(s)=(cxfs6,cxfs4,cxfswin4,cxfssun3,cxfsmac3.local.)  status=UP
```

On client-only nodes, the `cxfs_info` command serves a similar purpose. The command path is as follows:

- AIX and Solaris: `/usr/cxfs_cluster/bin/cxfs_info`
- IRIX and Linux: `/usr/cluster/bin/cxfs_info`
- Mac OS X: `/usr/cluster/bin/cxfs_info`
- Windows: `%ProgramFiles%\CXFS\cxfs_info.exe`

On AIX, Linux, Mac OS X, and Solaris nodes, 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 Solaris node:

```
solaris# /usr/cxfs_cluster/bin/cxfs_info
cxfs_client status [timestamp Jun 03 03:48:07 / generation 82342]

CXFS client:
     state: reconfigure (2), cms: up, xvm: up, fs: up
Cluster:
     performance (123) - enabled
Local:
     cxfssun3 (9) - enabled
Nodes:
     cxfs4  enabled  up    2
     cxfs5  enabled  up    1
     cxfs6  enabled  up    0
     cxfswin4  enabled  up    5
     cxfssun3  enabled  up    6
     cxfsmac3.local.  enabled  up    7
Filesystems:
     vol0       enabled  mounted          vol0                 /mnt/vol0
     vol1       enabled  mounted          vol1                 /mnt/vol1
```

The `CXFS client` line shows the state of the client in the cluster, which can be one of the following states:

`bootstrap`	Initial state after starting `cxfs_client`, while listening for bootstrap packets from the cluster.
`connect`	Connecting to the CXFS metadata server.
`query`	The client is downloading the cluster database from the metadata server.
`reconfigure`	The cluster database has changed, so the client is reconfiguring itself to match the cluster database.
`stable`	The client has been configured according to what is in the cluster database.
`stuck`	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.
`terminate`	The client is shutting down.

The `cms` field has the following states:

`unknown`	Initial state before connecting to the metadata server.
`down`	The client is not in membership.
`fetal`	The client is joining membership.
`up`	The client is in membership.
`quiesce`	The client is dropping out of membership.

The `xvm` field has the following states:

`unknown`	Initial state before connecting to the metadata server.
`down`	After membership, but before any XVM information has been gathered.
`fetal`	Gathering XVM information.
`up`	XVM volumes have been retrieved.

The `fs` field has the following states:

`unknown`	Initial state before connecting to the metadata server.
`down`	One or more filesystems are not in the desired state.
`up`	All filesystems are in the desired state.
`retry`	One or more filesystems cannot be mounted/unmounted, and will retry. See the "Filesystem" section of `cxfs_info` 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`(1M) command. For example:

```
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 258 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 AIX" on page 42
- "I/O Fencing for IRIX Nodes" on page 64
- "I/O Fencing for Linux" on page 97
- "I/O Fencing for Mac OS X" on page 132
- "I/O Fencing for Solaris" on page 159
- "I/O Fencing for Windows" on page 224

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:

```
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/stripe1         enabled
     device = /dev/cxvm/stripe1
     force = false
     options = []
     servers = cxfs5 (0), cxfs4 (1)
     clients = cxfs4, cxfs5, cxfs6, cxfsmac4, cxfssun1
     xvm:
         vol/stripe1                        0 online,open
             subvol/stripe1/data      2292668416 online,open
                 stripe/stripe1           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`.

```
admin# xvm show -e -t vol
vol/stripe1                       0 online,open
     subvol/stripe1/data      2292668416 online,open
         stripe/stripe1           2292668416 online,open (unit size: 1024)
             slice/d9400_0s0          1146334816 online,open (d9400_0:/dev/rdsk/200500a0b80cedb3/lun0vol/c2p1)
             slice/d9400_1s0          1146334816 online,open (d9400_1:/dev/rdsk/200400a0b80cedb3/lun1vol/c3p1)
```

On all platforms other than Windows, you can then run the xvm command on the client to identify the matching disk devices on the client. For example:

```
solaris# /usr/cxfs_cluster/bin/xvm show -e -t vol
vol/stripe1                         0 online,open
     subvol/stripe1/data      2292668416 online,open
         stripe/stripe1           2292668416 online,open (unit size: 1024)
             slice/d9400_0s0          1146334816 online,open (d9400_0:pci@9,600000/JNI,FCR@2,1/sd@2,0)
             slice/d9400_1s0          1146334816 online,open (d9400_1:pci@9,600000/JNI,FCR@2/sd@2,1)
```

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
```

For example:

```
solaris# /usr/cxfs_cluster/bin/xvm show -e -t subvol/stripe1
0 online,open,no physical connection
     subvol/stripe1/data      2292668416 online,open
         stripe/stripe1           2292668416 online,open (unit size: 1024)
             slice/d9400_0s0          1146334816 online,open (d9400_0:no direct attachment on this cell)
             slice/d9400_1s0          1146334816 online,open (d9400_1: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:

```
solaris# /usr/cxfs_cluster/bin/xvm show -v phys
```

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

General Troubleshooting

This chapter contains the following:

Also see the following platform-specific sections:

For more advanced cluster troubleshooting, see the *CXFS 5 Administration Guide for SGI InfiniteStorage.*

Identifying Problems

This section provides tips about identifying problems:

- "Are There Error Messages?" on page 275
- "What Is the Network Status?" on page 276
- "What is the Status of XVM Mirror Licenses?" on page 276

Is the Client-Only 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:

```
/usr/cluster/bin/cxfs-config -all
```

For more information, see "Verifying the Cluster Status" on page 265.

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 Client-Only Node in Membership?

To determine if the node is in the cluster membership, use the tools described in "Verifying the Cluster Status" on page 265.

If the client is not in membership, see the following:

- "Verifying the Cluster Configuration" on page 264
- "Verifying Connectivity in a Multicast Environment (Nodes Other Than Windows)" on page 264
- "Unable to Achieve Membership" on page 278

Is the Client-Only 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 265.

If the client has not mounted all filesystems, see the following:

- "Verifying the Cluster Configuration" on page 264
- "Verifying Access to XVM Volumes" on page 270

- "Determining If a Client-Only Node Is Fenced" on page 281
- Appendix C, "Mount Options Support" on page 297

Can the Client-Only 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 281 and "Verifying Access to XVM Volumes" on page 270.

If accessing the filesystem hangs the viewing process, see "Filesystem Appears to Be Hung" on page 279.

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 287 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 the following:

- "Log Files on AIX" on page 29
- "Log Files on IRIX" on page 56
- "Log Files on Linux" on page 84
- "Log Files on Mac OS X" on page 113
- "Log Files on Solaris" on page 143
- "Log Files and Cluster Status for Windows" on page 176

There are various logs also located on the CXFS server-capable administration nodes. For more information, see the *CXFS 5 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 Linux, Mac OS X, and Windows platforms support the `-s` option, which shows per-protocol statistics. The Linux and Windows systems also 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

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 one of the following sections as appropriate for your platform:
   - "Start/Stop cxfs_client for AIX" on page 43
   - "Start/Stop cxfs_client for IRIX" on page 65
   - "Start/Stop cxfs_client for Linux" on page 98

- "Start/Stop cxfs_client for Mac OS X" on page 134
- "Start/Stop cxfs_client for Solaris" on page 160
- "Start/Stop the CXFS Client Service for Windows" on page 228

2. Look for other warnings and error messages in the cxfs_client log file. See Appendix A, "Operating System Path Differences" on page 287 for the location of the log file on different platforms.
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:

```
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 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 hung in the filesystem, do the following:

1. Check that the client is still in membership and 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*). For example, on IRIX running `icrash` as `root`, the following message was received from cell 4 more than four minutes ago:

```
# icrash
    >>>> mesglist
    Cell:1
    THREAD ADDR         MSG ID TYPE CELL MESSAGE
    Time(Secs)
    ================== ======= ==== ==== ================================ ==========
    0xa80000004bc86400    10fc  Rcv    4                 I_dsxvn_allocate       4:20
```

3. If there is a stuck message, gather information for SGI support:

   - Find the stack trace for the stuck thread. For example:

```
>>>> kthread 0xa80000004bc86400

              KTHREAD TYPE                ID             WCHAN NAME
        ============================================================================
        a80000004bc86400    1         100000534  c000000002748008 mtcp_notify
        ============================================================================
        1 kthread struct found

        >>>> defkthread 0xa80000004bc86400

        Default kthread is 0xa80000004bc86400

        >>>> trace

        ============================================================================
        STACK TRACE FOR XTHREAD 0xa80000004bc86400 (mtcp_notify):

         1 istswtch[../os/swtch.c: 1526, 0xc00000000021764c]
         2 swtch[../os/swtch.c: 1026, 0xc000000000216de8]
         3 thread_block[../os/ksync/mutex.c: 178, 0xc00000000017dc8c]
         4 sv_queue[../os/ksync/mutex.c: 1595, 0xc00000000017f36c]
         5 sv_timedwait[../os/ksync/mutex.c: 2205, 0xc0000000001800a0]
         6 sv_wait[../os/ksync/mutex.c: 1392, 0xc00000000017f038]
         7 xlog_state_sync[../fs/xfs/xfs_log.c: 2986, 0xc0000000002a535c]
```

```
 8 xfs_log_force[../fs/xfs/xfs_log.c: 361, 0xc0000000002a25dc]
 9 cxfs_dsxvn_wait_inode_safe[../fs/cxfs/server/cxfs_dsxvn.c: 2011,
0xc00000000046a594]
10 dsvn_getobjects[../fs/cxfs/server/dsvn.c: 3266, 0xc0000000004676fc]
11 I_dsxvn_allocate[../fs/cxfs/server/cxfs_dsxvn.c: 1406, 0xc0000000004699c8]
12 dsxvn_msg_dispatcher[../IP27bootarea/I_dsxvn_stubs.c: 119,
0xc000000000456768]
13 mesg_demux[../cell/mesg/mesg.c: 1130, 0xc000000000408e88]
14 mtcp_notify[../cell/mesg/mesg_tcp.c: 1100, 0xc0000000004353d8]
15 tsv_thread[../cell/tsv.c: 303, 0xc000000000437738]
16 xthread_prologue[../os/swtch.c: 1638, 0xc00000000021782c]
17 xtresume[../os/swtch.c: 1686, 0xc0000000002178f8]
================================================================================
```

- 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 268.
- 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 282.
- 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 the following:

- "I/O Fencing for AIX" on page 42
- "I/O Fencing for IRIX Nodes" on page 64
- "I/O Fencing for Linux" on page 97
- "I/O Fencing for Mac OS X" on page 132
- "I/O Fencing for Solaris" on page 159
- "I/O Fencing for Windows" on page 224

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 5 Administration Guide for SGI InfiniteStorage*.

Verifying the XVM Mirror Licenses on Client-Only Nodes

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 AIX Problems" on page 51
- "Reporting IRIX Problems" on page 78
- "Reporting Linux Problems" on page 109
- "Reporting Mac OS X Problems" on page 138
- "Reporting Solaris Problems" on page 167
- "Reporting Windows Problems" on page 253

Appendix A

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 AIX Paths

Component	Path
CXFS client daemon	`/usr/cxfs_cluster/bin/cxfs_client`
Command that normally invokes the client daemon	`/usr/cxfs_cluster/bin/cxfs_cluster`
Log file	`/var/tmp/cxfs_client`
Options file	`/usr/cxfs_cluster/bin/cxfs_client.options`
CXFS status	`/usr/cxfs_cluster/bin/cxfs_info`
Hostname/address information	`/etc/hosts`
GRIO administration	`/usr/cxfs_cluster/bin/grioadmin`
GRIO monitoring	`/usr/cxfs_cluster/bin/griomon`
GRIO quality of service	`/usr/cxfs_cluster/bin/grioqos`
XVM query	`/usr/cxfs_cluster/bin/xvm`

Table A-2 IRIX Paths

Command / File	IRIX
System configuration	/sbin/chkconfig
Command that normally invokes the client daemon	/etc/init.d/cxfs_client
Log file	/var/adm/cxfs_client
CXFS status	/usr/cluster/bin/cxfs_info
Options file	/etc/config/cxfs_client.options
Hostname	/usr/bsd/hostname
GRIO administration	/usr/sbin/grioadmin
GRIO monitoring	/usr/sbin/griomon
GRIO quality of service	/usr/sbin/grioqos
XVM query	/sbin/xvm
Cluster daemon configuration files	/etc/config/
System log	/var/adm/SYSLOG

Table A-3 Linux Paths

Component	Path
CXFS client service:	/usr/cluster/bin/cxfs_client
Command that normally invokes the client daemon:	/etc/init.d/cxfs_client
Log file:	/var/log/cxfs_client
Options file:	/etc/cluster/config/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/grioqos
XVM query	/sbin/xvm

Table A-4 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/grioqos
XVM query	/usr/cluster/bin/xvm

Table A-5 Solaris Paths

Component	Path
CXFS client daemon:	/usr/cxfs_cluster/bin/cxfs_client
Command that normally invokes the client daemon:	/etc/init.d/cxfs_client
Log file:	/var/log/cxfs_client
Options file:	/usr/cxfs_cluster/bin/cxfs_client.options
CXFS status:	/usr/cxfs_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/grioqos
XVM query	/usr/cxfs_cluster/bin/xvm

Table A-6 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 228
Log file:	`%ProgramFiles%\CXFS\log\cxfs_client.log`
Options file:	See "Modifying the CXFS Software for Windows" on page 230
CXFS status:	`%ProgramFiles%\CXFS\cxfs_info.exe`
Hostname and address information:	`%SystemRoot%\system32\drivers\etc\hosts`
GRIO v2 administration:	`%ProgramFiles%\CXFS\grioadmin.exe`
GRIO monitoring	`%ProgramFiles%\CXFS\griomon.exe`
GRIO v2 quality of service:	`%ProgramFiles%\CXFS\grioqos.exe`
XVM query:	(unsupported)

Appendix B

Filesystem and Logical Unit Specifications

Table B-1 on page 294 summarizes filesystem and logical unit specifications differences among the supported client-only platforms.

Table B-1 Filesystem and Logical Unit Specifications

Item	AIX	IRIX	Linux x86_64	Linux ia64	Mac OS X	Solaris	Windows
Maximum filesystem size	2^{64} bytes [1]	2^{64} bytes	2^{64} bytes	2^{64} bytes	2^{64} bytes	2^{64} bytes	2^{64} bytes
Maximum file size/offset	16 TB [2]	2^{63}–1 bytes	2^{63}–1 bytes	2^{63}–1 bytes	2^{63}–1 bytes	2^{63}–1 bytes	2^{63}–1 bytes
Filesystem block size (in bytes)[3]	4096 (XFS default)	512, 1024, 2048, 4096, 8192, 16384, 32768, or 65536	512, 1024, 2048, or 4096	512, 1024, 2048, 4096, 8192, or 16384	4096, 8192, 16384, 32768, or 65536	2048, 4096, 8192, 16384, 32768, or 65536 [4]	512, 1024, 2048, 4096, 8192, 16384, 32768, or 65536
XVM device block size (in bytes)	512	512	512	512	512	512	512
Physical LUN limit for DVH-labeled disks	2 TB	2 TB	2 TB	2 TB	2 TB	1 TB	2 TB

1 About 18 million terabytes

2 Assumes the default `ulimit` is changed, see "Limitations and Considerations for AIX" on page 30.

3 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.

4 8192 is recommended

Item	AIX	IRIX	Linux x86_64	Linux ia64	Mac OS X	Solaris	Windows
Physical LUN limit for GPT-labeled disks [5]	2^{63} device blocks	IRIX 6.5.28 and 6.5.29: 2 TB IRIX 6.5.30: 2^{63} device blocks	2^{63} device blocks	2^{63} device blocks	2^{63} device blocks	2^{63} device blocks	2^{63} device blocks
Maximum concatenated slices	65536 [6]	65536	65536	65536	65536	65536	65536

5 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.

6 65536 concatenated slices is 130 PetaBytes

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)

A blank space within the table means that the option has not been verified.

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	AIX	IRIX	Linux 64	Mac OS X	Solaris	Windows
attr2	N	n	n	n	n	n	n
biosize	Y	S	S	S	S	S	S
client_timeout	Y	D	D	D	D	D	D
dmapi	Y	S	S	S	S	S	S
dmi	Y	S	S	S	S	S	S
filestreams[1]	Y	S	S	S	S	S	S
gqnoenforce	Y	S	S	S	S	S	S
gquota	Y	S	S	S	S	S	S
grpid	Y	S	S	S	S	S	S
grpquota	Y	S	S	S	S	S	S
inode64	Y	S	S	S	S	S	S
largeio	Y	S	S	S	S	S	S
logbsize	Y	S	S	S	S	S	S
logbufs	Y	S	S	S	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	AIX	IRIX	Linux 64	Mac OS X	Solaris	Windows
logdev	N	S	S	S	S	S	S
mrquota	Y	n	n	S	n	n	n
noalign	Y	n	n	S	n	n	S
noatime	Y	S	S	S	S	S	S
noattr2	N	n	n	n	n	n	n
noauto	N	n	n	n	n	n	n
nobarrier	N	S	S	S	S	S	S
nodev	N	S	S	S	S	S	S
nolargeio	N	n	n	S	n	S	n
noquota	Y	S	S	S	S	S	S
nosuid	Y	S	S	S	S	S	S
osyncisdsync	Y	n	n	S	n	n	n
pqnoenforce	Y	S	S	S	S	S	S
pquota	Y	S	S	S	S	S	S
prjquota	Y	S	S	S	S	S	S
qnoenforce	Y	S	S	S	S	S	S
quota	Y	S	S	S	S	S	S
ro	Y	S	S	S	S	S	S

Option	Checked by Client	AIX	IRIX	Linux 64	Mac OS X	Solaris	Windows
rtdev	N	n	n	n	n	n	n
rw	N	S	S	S	S	S	S
server_list	Y	D	D	D	D	D	D
server_timeout	Y	D	D	D	D	D	D
sunit	Y	S	S	S	S	S	S
swalloc	Y	S	S	S	S	S	S
swidth	Y	S	S	S	S	S	S
uqnoenforce	Y	S	S	S	S	S	S
uquota	Y	S	S	S	S	S	S
usrquota	Y	S	S	S	S	S	S
wsync	Y	S	S	S	S	S	S

Error Messages

The following are commonly seen error messages:

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 5 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 aixdisk0s0`

A filesystem mount has failed on an AIX node and will be retried

`cxfs_client:op_failed ERROR: Mount failed for concat0`

A filesystem mount has failed on an Linux 32–bit, Mac OS X, Solaris, or Windows node 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 ffffffffffffff00 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 on Solaris" on page 143 and "Log Files and Cluster Status for Windows" on page 176. Also see the log files on the server-capable administration node; for more information, see the *CXFS 5 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 retreving 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 230.

`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 223.

`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.

`cmgr` Examples

Caution: This appendix is included for convenience, but has not been updated to support the current release. With the exception of a few administrative `cmgr`commands, the preferred CXFS configuration tools are `cxfs_admin` and the CXFS graphical user interface (GUI). For more information about these commands, see *CXFS 5 Administration Guide for SGI InfiniteStorage*.

This appendix contains the following information about the `cmgr` command:

Example of Defining a Node Using `cmgr`

The following example shows the entries used to define a Solaris node named `solaris1` using the `cmgr` command in prompting mode:

```
# /usr/cluster/bin/cmgr -p
Welcome to SGI Cluster Manager Command-Line Interface

cmgr> define node solaris1
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ?
Is this a FailSafe node <true|false> ? false
Is this a CXFS node <true|false> ? true
```

```
Operating System <IRIX|Linux32|Linux64|AIX|MacOSX|Solaris|Windows> ? solaris
Node ID ? 7
Do you wish to define failure hierarchy[y/n]:y
Hierarchy option 0 <System|Fence|Shutdown>[optional] ? fence
Hierarchy option 1 <System|Fence|Shutdown>[optional] ? shutdown
Hierarchy option 2 <System|Fence|Shutdown>[optional] ?
Number of Network Interfaces ? (1)
NIC 1 - IP Address ? 163.154.18.172
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false> ? true
NIC 1 - (use network for control messages) <true|false> ? true
NIC 1 - Priority <1,2,...> ? 1
```

Adding the Client-Only Nodes to the Cluster Using cmgr

If you are using cmgr, you must add the defined nodes to the cluster. This happens by default if you are using cxfs_admin.

After you define all of the client-only nodes, you must add them to the cluster.

For example, if you have already defined a cluster named cxfscluster using cmgr and want to add the Solaris nodes solaris1 and solaris2, you could use the following cmgr command:

```
cmgr> modify cluster cxfscluster

cxfscluster ? add node solaris1
cxfscluster ? add node solaris2
cxfscluster ? done
```

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 the Client-Only Nodes" on page 262.

Defining the Switch for I/O Fencing Using cmgr

You are required to use I/O fencing on client-only nodes in order to protect data integrity. 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 CXFS server-capable administration node:

```
irix# 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.

```
irix# 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 CXFS server-capable administration node:

```
irix# 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

irix# cmgr -c admin fence update (No command necessary for cxfs_admin)

irix# 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 on the Client-Only Nodes Using cmgr

After adding the client-only nodes to the cluster with cmgr, 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.

For example:

```
cmgr> start cx_services on node solaris1 for cluster cxfscluster
cmgr> start cx_services on node solaris2 for cluster cxfscluster
```

Mounting Filesystems on New Client-Only Nodes Using `cmgr`

With `cmgr` command, 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. For example, to mount the `fs1` filesystem on all nodes in the cluster except `solaris2`, you could use the following commands:

```
cmgr> admin cxfs_unmount cxfs_filesystem fs1 in cluster cxfscluster
cmgr> modify cxfs_filesystem fs1 in cluster cxfscluster

cxfs_filesystem fs1 ? set dflt_local_status to enabled
cxfs_filesystem fs1 ? add disabled_node solaris2
cxfs_filesystem fs1 ? done
```

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 19 and "Forced Unmount of CXFS Filesystems" on page 263.

Forced Unmount of CXFS Filesystems Using `cmgr`

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 `cmgr`, define or modify the filesystem to unmount with force and then unmount the filesystem. For example:

```
define cxfs_filesystem logical_filesystem_name [in cluster clustername]
   set force to true

modify cxfs_filesystem logical_filesystem_name [in cluster clustername]
   set force to true

admin cxfs_unmount cxfs_filesystem filesystemname [on node nodename] [in cluster clustername]
```

For example, the following set of commands modifies the fs1 filesystem to allow forced unmount, then unmounts the filesystem on all nodes in the cxfscluster cluster:

```
cmgr> modify cxfs_filesystem fs1 in cluster cxfscluster
Enter commands, when finished enter either "done" or "cancel"cmgr>

cxfs_filesystem fs1 ? set force to true
cxfs_filesystem fs1 ? done
Successfully defined cxfs_filesystem fs1

cmgr> admin cxfs_unmount cxfs_filesystem fs1 in cluster cxfscluster
```

For details, see cmgr reference appendix in the *CXFS 5 Administration Guide for SGI InfiniteStorage.*

Summary of New Features from Previous Releases

This appendix contains a summary of the new features for each version of this guide.

CXFS MultiOS 2.0

Original publication (007-4507-001) supporting Solaris client-only nodes in a multiOS cluster with IRIX metadata servers.

CXFS MultiOS 2.1

The 007-4507–002 update contains the following:

- Support for Windows NT nodes in a CXFS multiOS cluster. Platform-specific information is grouped into separate chapters.
- Support for up to four JNI HBAs in each CXFS Solaris node.

Note: JNI supports a maximum of four JNI HBAs in operating environments with qualified Solaris platforms.

CXFS MultiOS 2.1.1

The 007-4507–003 update contains the following:

- References to using the latest software from the JNI website (http://www.jni.com/Drivers).
- Information about ensuring that appropriate software is installed on the IRIX nodes that are potential metadata servers.
- Clarifications to the use of I/O fencing and serial reset.
- Corrections to the procedure in the “Solaris Installation Overview” section and other editorial corrections.

CXFS MultiOS 2.2

The 007-4507–004 update contains the following:

- Support for Microsoft Windows 2000 nodes in a CXFS MultiOS cluster. This guide uses *Windows* to refer to both Microsoft Windows NT and Microsoft Windows 2000 systems.
- Support for SGI TP9100s. For additional details, see the release notes.
- A new section about configuring two HBAs for failover operation.
- Support for the JNI 5.1.1 and later driver on Solaris clients, which simplifies the installation steps.
- DMAPI support for all platforms.
- Removal of the Solaris limitation requiring more kernel threads.

CXFS MultiOS 2.3

The 007-4507–005 update contains the following:

- Updated Brocade Fibre Channel switch firmware levels.
- Filename corrections the chapters about FLEXlm licensing for Windows and modifying CXFS software on a Solaris system.

CXFS MultiOS 2.4

The 007-4507–006 update contains the following:

- Support for Sun Microsystems Solaris 9 and specific Sun Fire systems.
- Support for the JNI EZ Fibre release 2.2.1 or later.
- A cluster of as many as 32 nodes, of which as many as 16 can be CXFS administration nodes; the rest will be client-only nodes.
- Information about the **Node Function** field, which replaces node weight. For Solaris and Windows nodes, **Client-Only** is automatically selected for you. Similar fields are provided for the `cmgr` command. For more information, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

- Clarification that if the primary HBA path is at fault during the Windows boot up (for example, if the Fibre Channel cable is disconnected), no failover to the secondary HBA path will occur. This is a limitation of the QLogic driver.
- Reference to the availability of cluster information on Windows nodes.
- Information about enabling Brocade Fibre Channel switch ports.
- Additional information about functional limitations specific to Windows, and performance considerations, and access controls.

CXFS MultiOS 2.5

The 007-4507–007 update contains the following:

- Support for the IBM AIX platform, Linux on supported 32-bit platforms, SGI ProPack for Linux on Altix servers.
- Support for a cluster of up to 48 nodes, 16 of which can be CXFS administration nodes; the rest must be client-only nodes.
- For Windows nodes, user identification with lightweight directory access protocol (LDAP).
- Support of forced unmount of filesystems on Windows nodes.
- Information about protecting data integrity if JNI Fibre Channel cables are disconnected or fail.
- Support for the SGI TP9500 RAID.
- Support for the QLogic 2342 host bus adapter.
- Information about new `cxfs-reprobe` scripts on AIX, IRIX, Linux, and Solaris nodes. These scripts are run by either `clconfd` or `cxfs_client` when they need to reprobe the Fibre Channel controllers. The administrator may modify these scripts if needed.
- Information about setting the `ntcp_nodelay` system tunable parameter in order to provide adequate performance on file deletes.
- Automatic detection of HBAs is provided for Linux, Solaris, and Windows nodes.

CXFS MultiOS 3.0

The 007-4507–008 update contains the following:

- Support for the Microsoft Windows XP client.

Note: The CXFS multiOS 3.0 release is the last release that will support the Microsoft Windows NT 4.0 platform. The 3.1 release will not include software for Windows NT 4.0.

- Clarifications to the terminology and installation information for Linux 32-bit clients.
- Information about Linux 64-bit clients running SGI ProPack for Linux on SGI Altix 3000 systems has been removed and will appear in the *CXFS 5 Administration Guide for SGI InfiniteStorage* that support CXFS 3.0 for SGI ProPack 2.3 for Linux.

CXFS MultiOS 3.1

The 007-4507–009 update contains the following:

- Support for the Apple Computer, Inc. Mac OS X operating system on client-only nodes.
- Support for a cluster of up to 64 nodes.
- Information about the SGI TP9300, SGI TP9300S, and SGI TP9500S.
- Information about setting the LUN discovery method for Solaris systems using the SGI TP9100 1-Gbit controller
- Additional AIX troubleshooting information.

CXFS MultiOS 3.2

The 007–4507–010 update contains the following:

- Support for Mac OS X 10.3.5 and Apple host bust adapters (HBAs).

Note: Mac OS X 10.2.*x* and the Astera HBA are not supported with the CXFS 3.2 release.

- Support for Red Hat Enterprise Linux 3. If you are running a Red Hat Enterprise Linux 3 kernel and you want to use quotas on a CXFS filesystem, you must install the quota package.
- Support for the Sun Fire V210 server as a multiOS client platform.
- A summary of the maximum filesystem size, file size, and block size for each platform.
- Information about the environment variables you must define in the `/etc/cluster/config/cxfs_client.options` file in order for the `/etc/cluster/config/cxfs-reprobe` script to appropriately probe all of the targets on the SCSI bus for the Linux platform on third-party hardware.
- Availability of the new `xvm_maxdmasz` attribute to the AIX `chdev` command, used to change the maximum XVM direct memory access (DMA) size to improve direct I/O performance.
- Information about ensuring proper hostname configuration for a Windows node.
- XVM volume names are limited to 31 characters and subvolumes are limited to 26 characters.
- Information about mount options.
- Updates to the procedure for installing the AMCC JNI HBA.
- Clarification that the AMCC JNI HBA that is provided by Sun Microsystems **does not function with CXFS** and cannot be configured to do so. You must purchase the JNI HBA directly from AMCC.

CXFS MultiOS 3.3

The 007–4507–011 update contains the following:

- Support for Microsoft Windows Server 2003.
- Support for AMD AMD64, Intel EM64T, and Intel Itanium 2 third-party Linux systems as client-only nodes.

- Information about guaranteed-rate I/O (GRIO) version 2 (v2).
- Information about XVM failover v2.
- Platform-specific information about FLEXlm licenses and troubleshooting has been separated out into the various platform-specific chapters.
- Information about the recognizing changes to the storage systems.
- System tunables information for Solaris and Windows.
- Information about the SANshare license and XVM failover v2 on AIX.
- Information about configuring HBA failover on Windows.
- New sections about verifying the cluster configuration, connectivity, and status.
- Removed references to `xvmprobe`. The functionality of `xvmprobe` has been replaced by the `xvm` command.

CXFS MultiOS 3.4

The 007–4507–012 update contains the following:

- Support for SUSE Linux Enterprise Server 9 (SLES9)
- Best practices for client-only nodes
- Mapping XVM volumes to storage targets on AIX and Linux
- Remote core dump on Mac OS X
- Installing the LSI Logic HBA

CXFS 4.0

The 007–4507–013 update contains the following:

- Support for the following:
  - Red Hat Enterprise Linux 4.

Note: On Red Hat Enterprise Linux 4 (RHEL4) x86 nodes, you must fully disable SELinux and redirect `core` dump files in order to avoid a stack overflow panic.

  - Mac OS X 10.4, including full ACL support.
  - Solaris 10.

  The following are not included in CXFS 4.0:

  - AIX 5.2
  - Red Hat Enterprise Linux 3
  - Mac OS X 10.3.9
  - Solaris 8

- Support for the `cxfs_admin` command
- Information about choosing the correct version of XVM failover for your cluster.
- If Norton Ghost is installed on a Windows node, CXFS cannot mount filesystems on the mount point driver letter.
- Information about using fast copying for large CXFS files
- A platform-independent overview of client-only installation process
- Server-side CXFS client license keys are now supported on server-capable nodes, allowing a client without a node-locked client-side license key to request a license key from the server. Server-side license keys are optional on IRIX metadata servers, but are required on SGI ProPack metadata servers. The licensing software is based on the FLEXlm product from Macrovision Corporation. See *CXFS 5 Administration Guide for SGI InfiniteStorage*.
- Information about configuring firewalls for CXFS use and membership being prevented by inappropriate firewall configuration
- Information about the maximum CXFS I/O request size for AIX
- Support for Apple PCI Express HBA.
- Support for QLogic HBA for the Solaris platform.

- Support for the CXFS `autopsy` and `fabric_dump` scripts on Mac OS X.

CXFS 4.1

The 007–4507–014 update contains the following:

- Support for SUSE Linux Enterprise Server 10 (SLES 10) client-only nodes

  Note: DMAPI is disabled by default on SLES 10 systems. If you want to mount filesystems on a SLES 10 client-only node with the `dmi` mount option, you must enable DMAPI.

- Support for SGI License Key (LK) software on SGI ProPack server-capable nodes.

  Server-side licensing is required on the following client-only nodes (to determine the Linux architecture type, use the `uname -i` command):

  – SGI ProPack 5

  – Red Hat Enterprise Linux (RHEL) 4 on x86_64

  – SLES 9 on x86_64

  – SLES 10 on x86_64 or ia64

  (For specific release levels, see the release notes.)

  Other nodes can use either server-side or client-side licensing. However, if one node within a cluster requires server-side licensing, all nodes must use server-side licensing. If no nodes in the cluster require server-side licensing, the nodes can continue to use existing client-side licensing.

  Note: Server-side licensing is preferred, and no new client-side licenses will be issued. Customers with support contracts can exchange their existing client-side licenses for new server-side licenses. A future release will not support client-side licensing. For more information, contact SGI customer support.

  For licensing details, see the release notes and the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

- Support for changes in the Mac OS X device paths used by the `xvm` and and `failover2.conf` files.
- A new chapter to support SGI Altix XE as a client-only node.
- Updates to the supported mount options tables.

CXFS 4.2

The 007–4507–015 update contains the following:

- Support for the following new platforms:
  - Mac OS X on the Intel platform
  - Windows 2003 x86_64 platform
- As of CXFS 4.2, all server-capable nodes running 4.2 and client-only nodes running 4.2 require server-side licensing. If **all** existing client-only nodes are running a prior supported release, they may continue to use client-side license as part of the rolling upgrade policy until they are upgraded to 4.2. All client-only nodes in the cluster must use the same licensing type — if any client-only node in the cluster is upgraded to 4.2 or if a new 4.2 client-only node is added, then all nodes must use server-side licensing. Customers with support contracts can exchange their existing client-side licenses for new server-side licenses. For more information, contact SGI customer support.
- Support for 4Gb PICx and PCIe HBA support on Windows nodes
- Support for GPT labels on the Mac OS X and Windows platforms
- Memory-mapped files flush time for Windows
- Mapping XVM volumes to storage targets on Windows
- XVM failover V2 on Windows
- Documentation for the support of XVM failover version 2 on Windows nodes (first supported in the CXFS 4.1.1 release).
- Clarifications about support for the following:
  - Real-time subvolumes
  - External logs

- Information about the `cmgr` command has been moved to an appendix. The preferred CXFS configuration tools are `cxfs_admin` and the CXFS graphical user interface (GUI). As of the CXFS 5.0 release, the `cmgr` command will not be supported or documented.
- Removal of support for the following:
  - AIX 5.2
  - SLES 9 SP3
  - SGI ProPack 4 SP 3
  - Solaris 9
  - Windows 2000 and Windows XP SP 1

CXFS 5.0

The 007–4507–016 version includes the following changes:

- Support for the following new platforms:
  - Mac OS X Leopard (10.5).
  - SGI ProPack 5 SP 4 (client-only) and SGI ProPack 5 SP 5 (server and client-only).
  - Windows:

    Windows Server 2003 SP2
    Windows Server SP2 x64
    Windows Vista
    Windows Vista x64
- The IRIX platform as a client-only node.
- Removed support for Linux i386 architecture.
- The new section “Mapping Physical Device Names to XVM Physvols.”

CXFS 5.2

The 007–4507–017 version includes the following changes:

- CXFS server-capable nodes must run SGI Foundation Software 1.

  SGI Foundation Software 1 is a new product from SGI consisting of technical support tools, utilities, and driver software that enable SGI's Linux systems to run reliably and consistently. SGI ProPack 6 is the next generation of SGI's suite of performance-optimization libraries and tools that accelerate applications on SGI's Linux systems. SGI ProPack 6 may be optionally installed on any CXFS node running SGI Foundation Software 1. For more information on the content of these products, upgrades, ordering, service contracts, and licensing, see Supportfolio.

- Support for *edge serving*, in which CXFS client nodes can act as servers for NFS, Samba, CIFS, or any third-party network filesystem exporting files from a CXFS filesystem. However, there are no performance guarantees when using edge serving; for best performance, SGI still recommends that you use the active metadata server. If you require a high-performance solution, contact SGI Professional Services.

- Clarifications to the list of supported mount options for the Windows platform.

- Clarification that the physical LUN limit with GPT-labeled disks is 2 TB for IRIX 6.5.28 and IRIX 6.5.29 nodes.

CXFS 5.4

The 007–4507–018 version includes the following changes:

- Clarifications about the need to reboot a Linux node after enabling GRIO.

- Information about the fact that the `cxfs_client` software automatically detects the world wide port names (WWPNs) of any supported host bus adapters (HBAs) for Solaris nodes that are connected to a switch that is configured in the cluster database. (Introduced in CXFS 5.3.) See

- "Warning: DiskManager for Windows Vista and Windows 2008 Destroys Data".

- "Saving Application Crash Dumps for Windows Vista and Windows 2008".

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*.

ARP

Address resolution protocol.

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

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

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.

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*, *fencereset*, *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.

fencereset

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 withdrawl 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 Infinite Storage 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-foo` log only on the node `foo` and to write only minimal level-1 messages to the `crsd` log on all other nodes.

log file

A file containing notifications for a particular *log group*. A log file is part of the *log configuration* for a log group.

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 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 operating systems in addition to Linux with SGI Foundation Software, such as Solaris.

multiport serial adapter cable

A device that provides four DB9 serial ports from a 36-pin connector.

NFS edge serving

Exporting files from a CXFS filesystem with NFS from CXFS client nodes.

node

A *node* is an operating system (OS) image, usually an individual computer. (This use of the term *node* does not have the same meaning as a node in an SGI Origin 3000 or SGI 2000 system and 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 therefore) 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 is not the same thing as the *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 propagates 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 a serial line connected to the system controller. The reset may be a powercycle, serial reset, or NMI (nonmaskable interrupt).

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 unresponsive server-capable administration node and the lack of reset and/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

B

C

D

E

I

M

N

O

P

Q

R

S

T

U

V

W

X