sgi®

CXFS™ MultiOS Client-Only Guide for
SGI® InfiniteStorage

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 guide includes the following changes:

- 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. See "Limitations and Considerations for Linux" on page 56.

- 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 Administration Guide for SGI InfiniteStorage*.

- Support for changes in the Mac OS X device paths used by the `xvm` and and `failover2.conf` files. See "XVM Failover V2 on Mac OS X" on page 106 and "Mapping XVM Volumes to Storage Targets on Mac OS X" on page 107.

- Chapter 6, "SGI ProPack Client-Only Platform" on page 113 to support SGI Altix XE as a client-only node.

- Updates to the tables in Appendix C, "Mount Options Support" on page 281.

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 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

Contents

Figures

Tables

About This Guide

This publication documents the CXFS MultiOS Clients 4.1 release. This release supports Apple Computer Mac OS X, IBM AIX, Linux on supported third-party platforms, Sun Microsystems Solaris, Microsoft Windows 2000, Microsoft Windows 2003, and Microsoft Windows XP nodes. For more details, see the platform-specific release notes.

Prerequisites

This guide assumes the following:

- The IRIX or SGI ProPack for Linux CXFS cluster is 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 Administration Guide for SGI InfiniteStorage* and the platform's operating system and installation documentation.

Related Publications

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 Administration Guide for SGI InfiniteStorage* (TPL link)

- AMCC (formerly JNI) host bus adapter (HBA) card and driver documentation:

    - *Installation Guide, FCE-6460 and FCE2-6460 PCI-to-Fibre Channel Host Bus Adapters (Solaris, Windows NT/2000, Novell, AIX, HP-UX, MAC-OS, Linux) JNI FibreStar*

    - *Quick Installation Guide, Solaris, AIX and Windows JNI EZ Fibre*

LSI Logic HBA documentation: *Fibre Channel to PCI-X Host Adapters User's Guide*

Also see the AMCC website at:

http://www.amcc.com

- 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

- Linux third-party platform documentation:

  – Red Hat:

    - *Red Hat Enterprise Linux Installation Guide for x86, Itanium, AMD64, and Intel Extended Memory 64 Technology (Intel EM64T)*

    - *Red Hat Enterprise Linux System Administration Guide*

    - *Red Hat SELinux Guide*

      http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/selinux-guide/

    See:

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

  – SLES 9:

    - *SUSE Linux Enterprise Server 9 Administration and Installation*

    - *SUSE Linux Enterprise Server 9 Start-Up Guide*

    - *SUSE Linux Enterprise Server 9 Architecture-Specific Information for x86, AMD64, Intel EM64T, and Itanium Processor Family*

  – SLES 10:

    - *SLES 10 Start-Up Guide*

    - *SLES 10 Installation and Administration Guide*

See:

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*

  – *Solaris 9 Installation Guide*

  – *Solaris 9 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

- *Flexible License Manager End User Manual* from Macrovision Corporation (for client-side licensing)

The following man pages are provided on CXFS client-only nodes:

Client-Only Man Page	IRIX Subsystem	SGI ProPack Subsystem
cxfs_client(1M)	cxfs_client.man.man	cxfs_client
cxfs_info(1M)	cxfs_client.man.man	cxfs_client
cxfs-config(1M)	cxfs_util.man.man	cxfs_util
cxfscp(1)	cxfs_util.man.man	cxfs_util
cxfsdump(1M)	cxfs_util.man.man	cxfs_util
cxfslicense(1M)	cxfs_util.man.man	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, release notes, man pages, and other information.

- If it is installed on your IRIX SGI system, you can use InfoSearch, an online tool that provides a more limited set of online books, release notes, and man pages. On an IRIX system, enter infosearch at a command line or select **Help > InfoSearch** from the Toolchest.

- You can view the release notes in the /cdrom/docs directory.

- On all but Windows systems, you can view man pages by typing man *title* at a command line.

Conventions

This guide uses the following terminology abbreviations:

- *Solaris* to Solaris 9 and Solaris 10

- *Windows* to refer to Microsoft Windows 2000, Microsoft Windows 2003, and Microsoft Windows XP

- *Linux* used alone refers to the Linux operating system running on third-party hardware

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
  1140 East Arques Avenue
  Sunnyvale, CA 94085–4602

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 :

- AIX

- Mac OS X

- Red Hat Enterprise Linux

- SGI ProPack 5 running SUSE Linux Enterprise Server 10 (SLES 10) on SGI Altix XE systems

- SUSE Linux Enterprise Server 9 (SLES 9)

- SUSE Linux Enterprise Server 10 (SLES 10)

- Sun Microsystems Solaris

- Microsoft Windows 2000, Microsoft Windows Server 2003, and Microsoft Windows XP

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*. It will contain potential CXFS metadata servers and client-only nodes.

Nodes running SGI ProPack for Linux on SGI Altix machines or running IRIX can be either CXFS metadata server-capable nodes or client-only nodes. (*Metadata* is information that describes a file, such as the file's name, size, location, and permissions.) For more information about these nodes, see the *CXFS 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.

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

This chapter discusses the following:

- "When to Use CXFS" on page 2
- "CXFS on Client-Only Nodes" on page 3
- "License Keys" on page 9
- "Guaranteed-Rate I/O (GRIO) and CXFS" on page 10
- "XVM Failover and CXFS" on page 11
- "Monitoring CXFS" on page 12

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

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 Installation and Configuration Overview" on page 3
- "CXFS Processes" on page 4
- "Cluster Administration" on page 4
- "User Administration for CXFS" on page 5
- "User and Group Quotas " on page 5
- "CXFS Mount Scripts" on page 6
- "Requirements" on page 7

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 Administration Guide for SGI InfiniteStorage* and the release notes.

4. Install and verify the switch. See the *CXFS Administration Guide for SGI InfiniteStorage* and the release notes.

5. Obtain the CXFS server-side license key or use your existing client-side license key. For more information about licensing, see "License Keys" on page 9 and *CXFS 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 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 211.

8. Install the CXFS software.

9. Perform any required post-installation configuration steps.

10. Configure the cluster to define the new client-only node in the pool, add it to the cluster, start CXFS services, and mount filesystems. See Chapter 9, "Cluster Configuration" on page 247.

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 and Chapter 10, "General Troubleshooting" on page 263.

CXFS 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 administration node.

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 administration nodes contain the cluster database.

An administration node is required to perform administrative tasks, using the cxfs_admin command, the cmgr command, or the CXFS graphical user interface

(GUI). For more information about using these tools, see the *CXFS Administration Guide for SGI InfiniteStorage*.

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

Client-only nodes cannot view or edit user and group quotas because CXFS administration must be performed from a CXFS administration node. However, user and group quotas are enforced correctly by the metadata server.

To view or edit your quota information, you must log in to a CXFS administration node and make any necessary changes. If you want to provide a viewing command on the client-only node, such as `repquota`, 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 prior to and after a CXFS filesystem is mounted or unmounted on the following platforms:

- AIX

- IRIX

- Linux

- Solaris

- SGI ProPack

Note: NFS and Samba exports of CXFS filesystems are only supported from IRIX and SGI ProPack for Linux nodes.

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 CXFS mount scripts are used by CXFS to ensure that LUN path failover works after fencing. These scripts can be customized 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 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 with Red Hat Linux" on page 72.

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:

```
root@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
```

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 9.

- 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. Potential metadata servers should use serial reset lines. See "Protect Data Integrity" on page 17.

- There must be at least one server-capable node to act as the metadata server and from which to perform cluster administration tasks. You should install CXFS software on the server-capable 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 administration nodes; the rest must be client-only nodes. See "Make Most Nodes Client-Only Nodes" on page 16.

  A cluster in which both CXFS and FailSafe 2.1 or later are run (known as *coexecution*) is supported with a maximum of 64 nodes, as many as 8 of which can run FailSafe. FailSafe runs on IRIX nodes only.

- No nodes within the cluster running Trusted IRIX. A multiOS cluster cannot contain Trusted IRIX nodes.

- If you are using IRIX server-capable nodes, there are additional installation requirements. For example, if you want to use quotas and access control lists (ACLs) on any cluster node with IRIX metadata servers, the `eoe.sw.quotas`, `nfs.sw.acl_nfs`, and `eoe.sw.acl` subsystems must be installed on the administration nodes listed as potential metadata servers. Likewise, if using guaranteed-rate I/O (GRIO) version 2 in the cluster, ensure that `eoe.sw.grio2` and `cxfs.sw.grio2_cell` are installed on all IRIX nodes in the cluster.

  For more information, see the following:

  – *IRIX Admin: Disks and Filesystems*

  – *IRIX Admin: Backup, Security and Accounting*

  – *Guaranteed-Rate I/O Version 2 Guide*

  – Your site's IRIX system administrator

- Set the `mtcp_nodelay` system tunable parameter to `1` on potential metadata servers in order to provide adequate performance on file deletes.

Also see "Requirements for Solaris" on page 136, "Requirements for Windows" on page 178, and Chapter 2, "Best Practices for Client-Only Nodes" on page 13.

License Keys

CXFS requires the following license keys:

- CXFS license keys using either server-side licensing or existing client-side licensing on supported platforms.

    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

    – RHEL 4 on x86_64

    – SLES 9 on x86_64

    – SLES 10 on x86_64 or ia64

    Other client-only 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.

    The licensing used for SGI ProPack server-capable nodes is based the SGI License Key (LK) software. For IRIX server-capable nodes, the licensing is based on the FLEXlm product from Macrovision Corporation.

    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

    See the general release notes and the *CXFS Administration Guide for SGI InfiniteStorage* for more information.

- XVM cluster mirroring requires a license key on server-capable 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) requires a license key on the server-capable nodes.

- Fibre Channel switch license key. See the release notes.

- AIX using XVM failover version 2 also requires a SANshare license; see "SANshare and XVM Failover V2 for AIX" on page 32.

Guaranteed-Rate I/O (GRIO) and CXFS

CXFS supports guaranteed-rate I/O (GRIO) version 2 clients on all platforms, and GRIO servers on IRIX server-capable nodes or SGI ProPack server-capable nodes. However, GRIO is disabled by default on Linux. See "GRIO on Linux" on page 74.

Once installed in a cluster, 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 273, 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 43

- "GRIO on Linux" on page 74

- "GRIO on Mac OS X" on page 106

- "GRIO on SGI ProPack Client-Only Nodes" on page 127

- "GRIO on Solaris" on page 168

- "GRIO on Windows" on page 233

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

XVM Failover and CXFS

There are two versions of XVM failover. You must choose the appropriate version for your cluster:

- CXFS supports XVM failover v1 on the following platforms:

  IRIX
  SGI ProPack 3 or earlier

- CXFS supports XVM failover v2 on the following platforms:

  IRIX
  SGI ProPack
  AIX
  Linux third-party
  Mac OS X
  Solaris

Note: CXFS supports both v1 and v2 on IRIX and SGI ProPack 3 or earlier. However, v1 is not supported on the other CXFS platforms.

XVM failover v2 requires that the RAID be configured in AVT mode. AIX also requires a SANshare license; see "SANshare and XVM Failover V2 for AIX" on page 32.

To configure failover v2, you must create and edit the `/etc/failover2.conf` file. The easiest method to generate a default `failover2.conf` file is to run the following command:

```
xvm show -v phys | grep affinity > /etc/failover2.conf
```

The entries in the file only apply to already labeled devices. Values within < > angle brackets are comments; you can delete them or ignore them.

For more information, see the comments in the `/etc/failover2.conf` file on a CXFS administration node, *CXFS Administration Guide for SGI InfiniteStorage*, and the *XVM Volume Manager Administrator's Guide*.

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

- "XVM Failover V2 on AIX" on page 44

- "XVM Failover V2 on Linux" on page 74

- "XVM Failover V2 on Mac OS X" on page 106

- "XVM Failover V2 on SGI ProPack Client-Only Nodes" on page 128

- "XVM Failover V2 on Solaris" on page 168

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 administration node. For more information, see "Verifying the Cluster Status" on page 256.

Best Practices for Client-Only Nodes

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

- "Configuration Best Practices" on page 13
- "Administration Best Practices" on page 19

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

Configuration Best Practices

This section discusses the following:

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

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.

- When you are using NFS, Samba, or CIFS to export a CXFS filesystem from a CXFS client. Performance will be much better when the export is performed from the active CXFS metadata server than when it is performed from a CXFS client node. If a client is used, SGI will only support an NFS or Samba export from an IRIX or SGI ProPack for Linux client.

- 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

  - Create filesystems with sparse files disabled (`unwritten=0`)

- 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 for Windows" on page 185.

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:

- The first node you define must be an 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 administration node (which creates the cluster database), you must recreate the cluster database.

Use the `cxfs-config -check -ping` command line on an administration node to confirm network connectivity. For more information, see *CXFS 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:

- 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. Use client administration nodes only if a Failsafe co-execution node cannot be a potential

metadata server (Failsafe requires that a node be either a server-capable administration node or a client administration node).

The advantage to using client-only nodes is that they do not keep a copy of the cluster database; they contact an 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 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 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 administration nodes in the cluster.

See the following:

- "I/O Fencing for AIX" on page 40
- "I/O Fencing for Linux" on page 67
- "I/O Fencing for Mac OS X" on page 102
- "I/O Fencing for SGI ProPack Client-Only Nodes" on page 122
- "I/O Fencing for Solaris" on page 163
- "I/O Fencing for Windows" on page 225

Use a Client-Only Tiebreaker

SGI recommends that you always define a client-only node as the CXFS tiebreaker. (Server-capable 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 nodes when one of the server-capable nodes is removed from the cluster for maintenance (via a stop of CXFS services).

The following rules apply:

- If exactly two server-capable nodes are configured and there are no client-only nodes, **neither** server-capable 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 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 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 node as tiebreaker is recommended only when there are four or more server-capable 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

Enable the *forced unmount* feature for CXFS filesystems, which is turned off by default. 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.

Many sites have found that enabling this feature improves the stability of their CXFS cluster, particularly in situations where the filesystem must be unmounted. For more information, see "Forced Unmount of CXFS Filesystems" on page 254 and the *CXFS Administration Guide for SGI InfiniteStorage*.

Configure Firewalls for CXFS Use

Do one of the following:

- Configure firewalls to allow CXFS traffic. See *CXFS 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" on page 20

- "Understand the Platform-Specific Limitations and Considerations" on page 20

- "Shut Down Client-Only Nodes Properly" on page 21
- "Do Not Run Backups on a Client Node" on page 21
- "Use `cron` Jobs Properly" on page 21
- "Repair Filesystems with Care" on page 22
- "Disable CXFS Before Maintenance" on page 23
- "Running Power Management Software" on page 23
- "Use Fast Copying for Large CXFS Files" on page 23
- "Use the Appropriate Version of `lcrash` for SGI ProPack" on page 23

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:

- "Upgrading the CXFS Software for AIX" on page 42
- "Upgrading the CXFS Software for Mac OS X" on page 105
- "Upgrading the CXFS Software for Solaris" on page 167
- "Upgrading the CXFS Software for Windows" on page 230

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 28
- "Limitations and Considerations for Linux" on page 56

- "Limitations and Considerations on Mac OS X" on page 86

- "Limitations and Considerations for SGI ProPack Client-Only Nodes" on page 116

- "Limitations and Considerations on Solaris" on page 138

- "Functional Limitations for Windows" on page 185 and "Performance Considerations for Windows" on page 188

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 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`, `ls`, or Linux `slocate`. 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 '{}' ';'"
```

Edit the nodes' crontab file to only execute this find command on one metadata server of the cluster.

On Linux systems, there is often a cron job to execute updatedb, which can be problematic. You should remove this cron job or modify it to exclude CXFS directories. On Linux third-party systems, you can add xfs to the PRUNEFS configuration variable to exclude all CXFS filesystems. (This is not appropriate for SGI ProPack for Linux systems, which may use local XFS filesystems.)

Repair Filesystems with Care

Do not use any filesystem defragmenter software. You can use the IRIX fsr command or the 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, SGI ProPack, 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.

Use the Appropriate Version of `lcrash` for SGI ProPack

Do not use the version of lcrash that is shipped with SLES 9. For SGI ProPack 4, use the version of lcrashx that is available from Supportfolio. When using lcrashx, you must include the -x option to load the CXFS kerntypes. :

lcrashx -x /boot/sgi-cxfs-kerntypes-*kernelversion-architecturetype*

AIX Platform

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

- "CXFS on AIX" on page 25

- "SANshare and XVM Failover V2 for AIX" on page 32

- "HBA Installation for AIX" on page 33

- "Preinstallation Steps for AIX" on page 33

- "Client Software Installation for AIX" on page 37

- "I/O Fencing for AIX" on page 40

- "Start/Stop `cxfs_client` Daemon for AIX" on page 41

- "Maintenance for AIX" on page 42

- "GRIO on AIX" on page 43

- "XVM Failover V2 on AIX" on page 44

- "Mapping XVM Volumes to Storage Targets on AIX" on page 44

- "Troubleshooting for AIX" on page 44

- "Reporting AIX Problems" on page 50

CXFS on AIX

This section contains the following information about CXFS on AIX:

- "Requirements for AIX" on page 26

- "CXFS Commands on AIX" on page 27

- "Log Files on AIX" on page 27

- "CXFS Mount Scripts on AIX" on page 28

- "Limitations and Considerations for AIX" on page 28

- "Maximum CXFS I/O Request Size and AIX" on page 30

- "Access Control Lists and AIX " on page 32

Requirements for AIX

In addition to the items listed in "Requirements" on page 7, 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_info
/usr/cxfs_cluster/bin/cxfslicense
/usr/cxfs_cluster/bin/grioadmin
/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.

You can use the `cxfslicense` command to verify CXFS or XVM FLEXlm license keys for AIX nodes using client-side licensing.

The `lslpp` output lists all of the software added; see "Installing CXFS Software on AIX" on page 37.

For information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 10 and "GRIO on AIX" on page 43.

For more information on these commands, see the man pages.

Log Files on AIX

The `cxfs_client` command creates a `/usr/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 administration nodes, see the *CXFS Administration Guide for SGI InfiniteStorage*.

CXFS Mount Scripts on AIX

AIX supports the CXFS mount scripts. See "CXFS Mount Scripts" on page 6 and the *CXFS Administration Guide for SGI InfiniteStorage*.

Limitations and Considerations for AIX

Note the following:

- IRIX nodes do not permit nested mount points on CXFS filesystems; that is, you cannot mount an IRIX XFS or CXFS filesystem on top of an existing CXFS filesystem. 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 IRIX 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 IRIX.

- 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.

- When a CXFS mount is performed on a mirror volume created by XVM, the AIX system goes into panic status. The mirror volume cannot be mounted on the AIX CXFS system.

- 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 "License Keys" on page 9.

- 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 209715 (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=209715
```

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

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 279.

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, 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 IRIX semantics instead of AIX JFS semantics.

The IRIX 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 IRIX ACL type for a file and a directory.

SANshare and XVM Failover V2 for AIX

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 IRIX or Linux metadata servers: IRIX will disable CTQ and 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*. 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 and a separate Host Group for all the other hosts in the CXFS cluster. Set the Host Type for the AIX host group to `LINUX`. (The host type of `AIX` has AVT status disabled.) Set the Host Type of the second Host Group to `SGIAVT`.

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 33
- "Verifying the Private and Public Network for AIX" on page 36

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
```

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: irix0, aix1, and aix2. The /.rhosts files could be as follows (where the prompt denotes the node name):

```
irix0# cat /.rhosts
aix1 root
```

```
                              aix2 root

                              aix1# cat /.rhosts
                              irix0 root
                              aix2 root

                              aix2# cat /.rhosts
                              irix0 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" on page 37

- "Verifying the AIX Installation " on page 39

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 release notes to learn about any late-breaking changes in the installation procedure.

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. Insert the *CXFS MultiOS Client 4.1* CD.

4. Mount the CD-ROM:

   ```
   aix# mount -v cdrfs -o ro /dev/cd0 /mnt/cdrom
   ```

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

```
aix# installp -a -d /mnt/cdrom/aix/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 4.1.0.5                      # 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 4.1.0.5


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

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


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

Installation Summary
--------------------
Name                      Level          Part       Event       Result
-------------------------------------------------------------------------------
SGIcxfs-aix5L             4.1.0.5        USR        APPLY       SUCCESS
SGIcxfs-aix5L             4.1.0.5        ROOT       APPLY       SUCCESS
```

6. *(For client-side licensing only)* Use the `cxfslicense -d` command to verify that the existing CXFS client-side license key is installed in `/etc/flexlm/license.dat`. For example, the following shows a properly installed client-side license key:

```
# /usr/cxfs_cluster/bin/cxfslicense -d
No license required for XVM user feature.
No license required for XVM mirror feature.
CXFS_AIX license granted.
```

Note: If you are using server-side licensing, you should skip this step. If you execute this step on a node that uses server-side licensing at this point in the install procedure (before rebooting), you will get an error that can be ignored.

If you are using client-side licensing and there are errors, verify that you have installed the existing CXFS client-side license key. For more information, see "Verifying Existing Client-Side License Keys for AIX" on page 45.

7. To start AIX CXFS services without rebooting, see "Start/Stop `cxfs_client` Daemon for AIX" on page 41. To start CXFS services automatically, reboot.

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               4.1.0.5  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 Administration Guide for SGI InfiniteStorage*.

Start/Stop `cxfs_client` Daemon 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 after initial installation completes, enter the following:

aix# **/usr/cxfs_cluster/bin/cxfs_cluster init**

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:

- "Upgrading the CXFS Software for AIX" on page 42
- "Modifying the CXFS Software for AIX" on page 43
- "Recognizing Storage Changes for AIX" on page 43

Upgrading 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          4.1.0.5   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. Install the new version. See "Client Software Installation for AIX" on page 37.

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. 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.

An AIX client can mount a GRIO-managed filesystem and supports application- and node-level reservations. An AIX client 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 10 and the *Guaranteed-Rate I/O Version 2 Guide*.

XVM Failover V2 on AIX

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

```
/dev/hdisk199 affinity=0 preferred
/dev/hdisk135 affinity=0
/dev/hdisk231 affinity=1
/dev/hdisk167 affinity=1
```

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

Mapping XVM Volumes to Storage Targets on AIX

To map XVM volumes to storage targets on AIX, do the following:

1. Get visible controller port WWNs.

2. Display the desired fields by using the `lscfg` command. For example (line breaks shown for readability):

```
# lscfg |sed -n -e 's/000000000000//' -e 's/.*\(hdisk[0-9]*\)
.*-\(P.*-C.*-T.*\)-W\(.*\)-L\([0-9ABCDEF]*\) .*/\/dev\/\1 # HBA=\2 WWN=\3 LUN=\4/p'|sort -k5
/dev/hdisk137 # HBA=F8-P1-C1-T1 WWN=W21000080E5119F42 LUN=0
/dev/hdisk98 # HBA=F8-P1-C4-T1 WWN=W21000080E5119F42 LUN=0
/dev/hdisk127 # HBA=F8-P1-C4-T1 WWN=W21000080E5119F42 LUN=1
/dev/hdisk138 # HBA=F8-P1-C1-T1 WWN=W21000080E5119F42 LUN=1
```

Troubleshooting for AIX

This section discusses the following:

- "Verifying Existing Client-Side License Keys for AIX" on page 45

- "Reinstalling An Existing Client-Side License Key for AIX" on page 46

- "Unable to Mount Filesystems on AIX" on page 46

- "The `cxfs_client` Daemon is Not Started on AIX" on page 47

- "Filesystems Do Not Mount on AIX" on page 48

- "Panic Occurs when Executing `cxfs_cluster` on AIX " on page 49

- "A Memory Error Occurs with `cp -p` on AIX" on page 49

- "An ACL Problem Occurs with `cp -p` on AIX" on page 49

- "Large Log Files on AIX" on page 49

- "Error Reading Kernel XVM Cluster Mirror Status" on page 50

Also see Chapter 10, "General Troubleshooting" on page 263 and Appendix D, "Error Messages" on page 289.

Verifying Existing Client-Side License Keys for AIX

Note: Server-side licensing is preferred. AIX 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. See "License Keys" on page 9.

To verify that the existing client-side license key has been installed properly when using client-side licensing, use the `cxfslicense -d` command. For example:

- Client-side license key installed properly on an AIX node:

  ```
  aix# /usr/cxfs_cluster/bin/cxfslicense -d
  No license required for XVM user feature.
  No license required for XVM mirror feature.
  CXFS_AIX license granted.
  ```

- No client-side license key on an AIX node:

```
aix# /usr/cxfs_cluster/bin/cxfslicense -d
No license required for XVM user feature.
No license required for XVM mirror feature.
Cannot find valid version 1.0 license for CXFS_AIX
 -> : Cannot find license file
 The license files (or server network addresses) attempted are
 listed below.  Use LM_LICENSE_FILE to use a different license file,
 or contact your software provider for a license file.
Feature:        CXFS_AIX
Filename:       /etc/flexlm/license.dat
License path:   /etc/flexlm/license.dat
```

```
FLEXlm error:  -1,359.  System Error: 2 "No such file or directory"
For further information, refer to the FLEXlm End User Manual,
available at "www.macrovision.com".

No client-side license, will request a server-side license.
```

> The above output indicates that no client-side license key is installed and the CXFS software will request a server-side license key. If you are using client-side licensing, verify that you have installed the existing CXFS client-side license key.

Reinstalling An Existing Client-Side License Key for AIX

An existing AIX client-side license key is installed in the following location:

`/etc/flexlm/license.dat`

If you need to reinstall the existing client-side license, do the following:

1. Create the license key directory if necessary. For example:

   aix# **mkdir -p /etc/flexlm**

2. Copy the key to the `license.dat` file.

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 client 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 47.

- 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 /usr/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 administration node with the correct hostname? See "Configuring Hostnames on Mac OS X" on page 86.

    - Has the node been added to the cluster and enabled? See "Verifying the Cluster Status" on page 256.

  - – 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 /usr/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.

- If the license key has expired, which is indicated when the following message is output to the /usr/tmp/cxfs_client file:

  ```
  CXFS not properly licensed for this host
  ```

  In this case, execute the following command to determine why the license key check fails:

  ```
  aix# /usr/cluster/bin/cxfslicense -d
  ```

  If the license key is valid, check the cxfs_client log file for other errors.

  Restart cxfs_client as described in "Start/Stop cxfs_client Daemon for AIX" on page 41 and watch the cxfs_client log file for errors.

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 /usr/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 27.

Error Reading Kernel XVM Cluster Mirror Status

The following message will be output by the cxfslicense -d command if you execute it before rebooting the system:

```
error reading kernel XVM cluster mirror status. Check if XVM module is
started.
```

After you reboot the system and therefore load the XVM module, this message will no longer appear when you run cxfslicense -d. You can ignore this message if you have not rebooted.

Reporting AIX Problems

When reporting a problem about a CXFS AIX node to SGI, you should retain the following information:

- Information about the AIX node system dump and system configuration:

  aix# **snap -a -o /dev/rmt0**

- Console log:

  aix# **alog -o -t console**

- Current syslog file

- The /usr/tmp/cxfs_client CXFS log file

- 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 | grep bos.64bit
  ```

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

If any of these AIX tools are not currently installed on your AIX node, you should install them.

Linux Third-Party Platforms

CXFS supports a client-only node running the Linux operating system on supported third–party platforms.

Note: The term *Linux* in this guide always refers to Linux client-only nodes on third-party platforms. For information about SGI ProPack for Linux and CXFS, see Chapter 6, "SGI ProPack Client-Only Platform" on page 113, and *CXFS Administration Guide for SGI InfiniteStorage*.

On Linux systems, the use of XVM is supported only with CXFS; XVM does not support local Linux disk volumes.

This chapter contains the following sections:

- "CXFS on Linux" on page 54
- "HBA Installation for Linux" on page 58
- "Preinstallation Steps for Linux" on page 60
- "Client Software Installation for Linux" on page 64
- "I/O Fencing for Linux" on page 67
- "Start/Stop cxfs_client for Linux" on page 69
- "Maintenance for Linux" on page 71
- "Using cxfs-reprobe with Red Hat Linux" on page 72
- "GRIO on Linux" on page 74
- "XVM Failover V2 on Linux" on page 74
- "Mapping XVM Volumes to Storage Targets on Linux" on page 75
- "Troubleshooting for Linux" on page 75
- "Reporting Linux Problems" on page 80

CXFS on Linux

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

- "Requirements for Linux"

- "CXFS Commands on Linux" on page 55

- "Log Files on Linux" on page 56

- "CXFS Mount Scripts on Linux" on page 56

- "Limitations and Considerations for Linux" on page 56

- "Access Control Lists and Linux" on page 58

Requirements for Linux

In addition to the items listed in "Requirements" on page 7, using a Linux node to support CXFS requires the following:

- One of the following operating systems:

  - Red Hat Enterprise Linux 4 (RHEL4) update 4 for WS, ES, and AS (see the release notes for the supported update levels)

  - SUSE Linux Enterprise Server 9 (SLES 9) (see the release notes for the supported service pack levels)

  - SUSE Linux Enterprise Server 10 (SLES 10) (see the release notes for the supported service pack levels)

- A choice of at least one Fibre Channel host bus adapter (HBA):

  - QLogic QLA 2200, QLA2310, QLA2342, or QLA2344

  - LSI Logic LS17202XP-LC dual channel PCI-X HBA

- A CPU of the following class:

  - i386 architecture (i386 as reported by the uname -i command), such as:

    - Advanced Micro Devices AMD Athlon

    - AMD Duron

    - Intel Pentium 4

  – 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.

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/cluster/bin/cxfslicense
/usr/sbin/grioadmin
/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.

You can use the `cxfslicense` command to verify CXFS or XVM FLEXlm client-side license keys for Linux nodes using client-side licensing.

The `rpm` output lists all software added; see "Linux Installation Overview" on page 65.

For more information, see the man pages.

Log Files on Linux

The `cxfs_client` command creates a `/var/log/cxfs_client` log file. This file is rotated by default.

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 CXFS administration nodes, see the *CXFS Administration Guide for SGI InfiniteStorage*.

CXFS Mount Scripts on Linux

Linux supports the CXFS mount scripts. See "CXFS Mount Scripts" on page 6 and the *CXFS Administration Guide for SGI InfiniteStorage*.

For Red Hat 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` with Red Hat Linux" on page 72.

Limitations and Considerations for Linux

Note the following:

- By default, DMAPI is turned off on SLES 10 systems. If you want to mount filesystems on a SLES 10 client-only node with the `dmi` mount option, you must ensure that the `DMAPI_PROBE` system tunable parameter on the node is set to `yes` in the `/etc/sysconfig/sysctl` file. 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
  ```

- IRIX nodes do not permit nested mount points on CXFS filesystems; that is, you cannot mount an IRIX XFS or CXFS filesystem on top of an existing CXFS filesystem. Although it is possible to mount other filesystems on top of a Linux CXFS filesystem, this is not recommended.

- On Linux systems, the `mkfs.xfs` command does not discover log or realtime subvolumes. You must specify the log or realtime subvolumes on the command line. For more information, see the `mkfs.xfs`(8) man page.

- Due to Linux kernel limitations, CXFS filesystems cannot be mounted with the `inode64` mount option. For more information, see Appendix C, "Mount Options Support" on page 281.

- CXFS filesystems with XFS version 1 directory format cannot be mounted on Linux nodes.

- By default, the Linux kernel will only scan LUN 0 of a SCSI device. This can be altered by adding `max_scsi_luns=`*N* to the kernel boot arguments, where *N* is the number of LUNs that should be scanned. If not all devices on the fabric are found, this may resolve the issue.

- 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.

- RHEL4 i386 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 i386 RHEL4 client nodes (you cannot simply set it to `permissive` mode). See the *Red Hat SELinux Guide* for instructions:

  http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/selinux-guide/

  Note: This caveat does not apply to RHEL4 nodes with x86_64 or ia64 architectures.

  – You must redirect `core` dump files on RHEL4 i386 nodes to an absolute path. (By default, `core` dump files are in the current working directory of the

process, which might be on a CXFS filesystem and could cause a panic.) Add the following line to the /etc/sysctl.conf file:

```
kernel.core_pattern = /core
```

For more information, see the sysctl man page.

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 279.

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, 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 name is mptscsih. The LSI lsiutil command displays the number of LSI HBAs installed, the model numbers, and firmware versions.

- QLogic card: the drivers are supplied with the Linux kernel.

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

- **Automatic method:** For RHEL, add a new line such as the following to the /etc/modprobe.conf file:

```
alias scsi_hostadapter1 qla2200
```

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

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

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

- **Manual method:** recreate your initrd to include the appropriate HBA driver module. For more information, see the operating system documentation for the mkinitrd command.

You should then verify the appropriate initrd information:

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

  initrd /initrd-*version*.img

- If using the LILO loader, do the following:

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

     /boot/initrd-*version*.img

  2. Rerun LILO.

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

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

Preinstallation Steps for Linux

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

- "Adding a Private Network for Linux" on page 60
- "Modifications Required for CXFS Connectivity Diagnostics for Linux" on page 62
- "Verifying the Private and Public Networks for Linux" on page 63

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:

```
[root@linux root]# 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 Connectivity Diagnostics for Linux

In order to test node connectivity by using the GUI or the cmgr command, 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.

Following is one possible method that 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. Enter the following, where *nodeIPaddress* is the IP address of the node:

`ping` *nodeIPaddress*

For example:

```
[root@linux root]# 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, repeat the following procedure on each node:

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

```
[root@linux root]# 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.

Client Software Installation for Linux

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 4-1 and Table 4-2 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 4-1 RHEL Processor and Package Extension Examples

Class	Example Processors	User Package Architecture Extension	Kernel Package Architecture Extension
i386	AMD Athlon	`.i386.rpm`	`.athlon.rpm`
	AMD Duron	`.i386.rpm`	`.athlon.rpm`
	Intel Pentium 4	`.i386.rpm`	`.i686.rpm`
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 4-2 SLES Processor and Package Extension Examples

Class	Example Processors	User and Kernel Package Architecture Extension
i386	AMD Athlon	`.i586.rpm` (`.i386.rpm` for `sysadm` packages)
x86_64	AMD Opteron	`.x86_64.rpm`
	EM64T	`.x86_64.rpm`
ia64	Intel Itanium 2	`.ia64.rpm`

Linux Installation Overview

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

Installing the CXFS client CD 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, SGI personnel will do the following:

1. Read the release notes to learn about any late-breaking changes in the installation procedure.

2. Verify that the node is running a supported Linux distribution, according to the CXFS for Linux release notes. See the Red Hat `/etc/redhat-release` or SLES `/etc/SuSE-release` files.

   > **Note:** When installing the Linux OS, disconnect the system from the fabric or ensure that the drive you are installing on is not a SAN-attached drive.

3. (Red Hat systems only) Insert and mount the *CXFS MultiOS Client 4.1 CXFS for the RHEL4 Client* CD. Change to the directory containing the appropriate kernel RPM for your system, according to the information about upgrading the kernel in the operating system documentation. Then install the XFS kernel module, as follows:

```
[root@linux cdrom]# rpm -Uvh sgi-xfs-kmp-kernelvariant-kernelrelease-version.architecture.rpm
Preparing...                ########################################### [100%]
   1:sgi-xfs-kmp-kernelvariant-ker########################################### [100%]
```

4. Insert and mount the *CXFS MultiOS Client 4.1* CD.

5. Install the CXFS kernel modules:

```
[root@linux cdrom]# rpm -Uvh sgi-cxfs-kmp-kernelvariant-kernelrelease-version.architecture.rpm
Preparing...                ########################################### [100%]
   1:sgi-cxfs-kmp-kernelvariant-ker########################################### [100%]
```

Where:

- *kernelvariant* and *kernelrelease* are the kernel variant and release level as output by the `uname -r` command

- *version* is the version number

- *architecture* is the architecture type extension as listed in Table 4-1 on page 64 or Table 4-2 on page 64

Note: For SLES 9, the *kernelrelease* must match the stock kernel release provided by SUSE. For SLES 10 nodes, one version of CXFS may support one or more *kernelrelease* values. See the CXFS Linux release notes for the supported versions.

6. Install the user-space packages:

```
[root@linux cdrom]# rpm -Uvh cxfs_client* cxfs_util* cxfs-xvm-cmds* cxfs-doc*
Preparing...              ######################################### [100%]
   1:cxfs-xvm-cmds        ######################################### [ 25%]
   2:cxfs_util            ######################################### [ 50%]
   3:cxfs_client          ######################################### [ 75%]
cxfs_client                0:off  1:off  2:off  3:on   4:off  5:on   6:off
   4:cxfs-doc             ######################################### [100%]
```

Note: The order of RPMs listed on the command line is not necessarily the same order in which they will be displayed in the rpm command output.

7. If you are using GRIO, install the grio2-cmds packages:

```
[root@linux cdrom]# rpm -Uvh grio2-cmds*

Preparing...              #############################################[100%]
   1:grio2-cmds           #############################################[100%]
```

8. Edit the /etc/cluster/config/cxfs_client.options file as necessary. See the "Maintenance for Linux" on page 71 and the cxfs_client(1M) man page.

9. *(For client-side licensing only)* Use the cxfslicense -d command to verify that the client-side license key is installed in /etc/flexlm/license.dat. For example, the following shows properly installed client-side license keys on an i386 system:

```
[root@linux root]# /usr/cluster/bin/cxfslicense -d
Found 1 XVM_STD_LINUX32 license(s) of version 1.0 for XVM.
Didn't find XVM_PLEX_LINUX32 license of version 1.0 for XVM.
Found XVM_PLEX_CLUSTER license of version 4.0 for XVM.
```

```
CXFS_LINUX32 license granted.
```

Note: If you are using server-side licensing, you should skip this step. If you execute this step on a node that uses server-side licensing at this point in the install procedure (before rebooting), you will get an error that can be ignored.

If you are using client-side licensing and there are errors, verify that you have obtained and installed the CXFS client-side license key. For more information, see "Verifying the Linux Installation" on page 67.

10. Reboot the system with the newly installed kernel:

```
[root@linux root]# reboot
```

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, there will be messages about loading the HBA/SNIA library logged to the /var/log/cxfs_client 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 /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.

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. Determine the HBA WWPN: 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 Administration Guide for SGI InfiniteStorage*.

Start/Stop `cxfs_client` for Linux

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 up cxfs_client manually, enter the following:

```
[root@linux root]# /etc/init.d/cxfs_client start
Loading cxfs modules:                                    [  OK  ]
Mounting devfs filesystems:                              [  OK  ]
Starting cxfs client:                                    [  OK  ]
```

To stop cxfs_client manually, enter the following:

```
[root@linux root]# /etc/init.d/cxfs_client stop
Stopping cxfs client:                                    [  OK  ]
```

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

```
[root@linux root]# /etc/init.d/cxfs_client restart
Stopping cxfs client:                                    [  OK  ]
```

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

```
[root@ceara root]# /etc/init.d/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:

```
[root@linux root]# /etc/init.d/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

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:

```
[root@linux root]# 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
```

Recognizing Storage Changes for Linux

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 Red Hat 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 with Red Hat Linux" on page 72.

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 6.

Using `cxfs-reprobe` with Red Hat Linux

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 Red Hat Linux 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 /etc/init.d/cxfs_client script.

GRIO on Linux

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the Linux platform. However, GRIO is disabled by default on Linux. To enable GRIO, change the following line in `/etc/cluster/config/cxfs_client.options` from:

```
export GRIO2=off
```

to:

```
export GRIO2=on
```

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 Linux client can mount a GRIO-managed filesystem and supports application- and node-level reservations. A Linux client 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 10 and the *Guaranteed-Rate I/O Version 2 Guide*.

XVM Failover V2 on Linux

Following is an example of the `/etc/failover2.conf` file on a 32–bit system:

```
/dev/scsi/host1/bus0/target6/lun0/disc affinity=0 preferred
/dev/scsi/host1/bus0/target0/lun0/disc affinity=0
/dev/scsi/host1/bus0/target1/lun0/disc affinity=1
/dev/scsi/host1/bus0/target7/lun0/disc affinity=1
```

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

Mapping XVM Volumes to Storage Targets on Linux

To map XVM volumes to storage targets on Linux, do the following:

1. Get visible controller port WWNs.

2. Display the desired fields from the `/proc/scsi/qla*/[0-9]` files:

```
cat /proc/scsi/qla*/[0-9]* | grep target | cut -f2 -d"=" | cut -f1 -d";"
```

Troubleshooting for Linux

This section discusses the following:

- "Verifying Existing Client-Side License Keys for Linux" on page 76

- "Reinstalling Existing Client-Side License Keys for Linux" on page 77

- "Device Filesystem Enabled for Linux" on page 77

- "The `cxfs_client` Daemon is Not Started on Linux" on page 78

- "Filesystems Do Not Mount on Linux" on page 78

- "Large Log Files on Linux" on page 79

- "`xfs off` Output from `chkconfig`" on page 79

- "Error Reading Kernel XVM Cluster Mirror Status" on page 80

For general troubleshooting information, see Chapter 10, "General Troubleshooting" on page 263 and Appendix D, "Error Messages" on page 289.

Verifying Existing Client-Side License Keys for Linux

Note: Server-side licensing is preferred. Server-side licensing is required on the following Linux client-only nodes (to determine the architecture type, use the uname -i command):

- RHEL 4 on x86_64
- SLES 9 on x86_64
- SLES 10 on x86_64 or ia64

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. See "License Keys" on page 9.

To verify that the client-side license keys have been installed properly when using client-side licensing, use the cxfslicense -d command. For example:

- Client-side license key installed properly on a Linux node:

```
[root@linux root]# /usr/cluster/bin/cxfslicense -d
Found 1 XVM_STD_LINUX32 license(s) of version 1.0 for XVM.
Didn't find XVM_PLEX_LINUX32 license of version 1.0 for XVM.
Found XVM_PLEX_CLUSTER license of version 4.0 for XVM.
CXFS_LINUX32 license granted.
```

- No client-side license keys on a Linux i386 or ia64 node:

```
[root@linux root]# /usr/cluster/bin/cxfslicense -d
Found 1 XVM_STD_LINUX32 license(s) of version 1.0 for XVM.
Didn't find XVM_PLEX_LINUX32 license of version 1.0 for XVM.
Found XVM_PLEX_CLUSTER license of version 4.0 for XVM.
Cannot find valid version 1.0 license for CXFS_LINUX32
 -> : Cannot find license file
 The license files (or server network addresses) attempted are
 listed below.  Use LM_LICENSE_FILE to use a different license file,
 or contact your software provider for a license file.
Feature:       CXFS_LINUX32
Filename:      /var/flexlm/license.dat
License path:  /etc/flexlm/license.dat:/var/flexlm/license.dat
FLEXlm error:  -1,359.  System Error: 2 "No such file or directory"
For further information, refer to the FLEXlm End User Manual,
```

```
available at "www.macrovision.com".

No client-side license, will request a server-side license.
```

The above output indicates that no client-side license key is installed and the
CXFS software will request a server-side license key. If you are using client-side
licensing, verify that you have installed the existing client-side CXFS license key.

- No client-side license keys on a Linux x86_64 node :

  ```
  [root@linux root]# /usr/cluster/bin/cxfslicense -d

  Found 1 XVM_STD_IPF license(s) of version 1.0 for XVM.
  Didn't find XVM_PLEX_IPF license of version 1.0 for XVM.
  Found 1 XVM_PLEX_CLUSTER license(s) of version 4.0 for XVM.
  No client-side license, will request a server-side license.
  ```

  The above output indicates that no client-side license key is installed and the
  CXFS software will request a server-side license key. If you are using client-side
  licensing, verify that you have installed the existing client-side CXFS license key.

Reinstalling Existing Client-Side License Keys for Linux

Existing client-side license keys for CXFS and XVM base (and potentially XVM
mirroring) are installed in the following location:

/etc/flexlm/license.dat

Do the following:

1. Create the license keys directory if necessary. For example:

   ```
   [root@linux cdrom]# mkdir -p /etc/flexlm
   ```

2. Copy the key to the license.dat file.

Device Filesystem Enabled for Linux

The kernels provided for the Linux client 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**

The cxfs_client daemon might not start if the license key has expired, which is indicated when the following message is output to the /var/log/cxfs_client file:

CXFS not properly licensed for this host

In this case, execute the following command to determine why the license keys check fails:

linux# **/usr/cluster/bin/cxfslicense -d**

If the license keys is valid, check the cxfs_client log file for other errors.

Restart cxfs_client as described in "Start/Stop cxfs_client for Linux" on page 69 and watch the cxfs_client log file for errors.

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 client 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 78.

- 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 administration node with the correct hostname?

    - Has the node been added to the cluster and enabled? See "Verifying the Cluster Status" on page 256.

  - 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.

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 56.

`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
```

Error Reading Kernel XVM Cluster Mirror Status

The following message will be output by the `cxfslicense -d` command if you execute it before rebooting the system:

```
error reading kernel XVM cluster mirror status. Check if XVM module is
started.
```

After you reboot the system and therefore load the XVM module, this message will no longer appear when you run `cxfslicense -d`. You can ignore this message if you have not rebooted.

Reporting Linux Problems

When reporting a problem about a Linux node to SGI, you should retain the following information:

* The kernel you are running:

  ```
  [root@linux root]# uname -a
  ```

* The CXFS packages you are running:

```
[root@linux root]# rpm -q cxfs_client cxfs_utils cxfs-xvm-cmds \
 sgi-cxfs-kmp-kernelvariant-kernelrelease-version
```

* The numbers and types of the processors on your machine:

  ```
  [root@linux root]# cat /proc/cpuinfo
  ```

* The hardware installed on your machine:

  ```
  [root@linux root]# lspci
  ```

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

  ```
  [root@linux root]# lsiutil
  ```

* Modules that are loaded on your machine:

  ```
  [root@linux root]# lsmod
  ```

* The `/var/log/cxfs_client` log file

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

- Output about the cluster obtained from the `cxfsdump` utility run on a CXFS administration node. The `cxfsdump` command transfers all of the information back to the node where the command was issued. When run in local mode on a Linux node, it stores information in `/var/cluster/cxfsdump-data/`*nodename*`.tar.gz`.

- After a system kernel panic, the debugger information from the `kdb` built-in kernel debugger.

⚠️ **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:

```
[root@linux root]# 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.

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" on page 83
- "HBA Installation for Mac OS X" on page 96
- "Preinstallation Steps for Mac OS X" on page 98
- "Client Software Installation for Mac OS X" on page 100
- "I/O Fencing for Mac OS X" on page 102
- "Start/Stop cxfs_client for Mac OS X" on page 104
- "Maintenance for Mac OS X" on page 105
- "GRIO on Mac OS X" on page 106
- "XVM Failover V2 on Mac OS X" on page 106
- "Mapping XVM Volumes to Storage Targets on Mac OS X" on page 107
- "Troubleshooting for Mac OS X" on page 108
- "Reporting Mac OS X Problems" on page 111

CXFS on Mac OS X

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

- "Requirements for Mac OS X" on page 84
- "CXFS Commands on Mac OS X" on page 84
- "Log Files on Mac OS X" on page 85
- "Limitations and Considerations on Mac OS X" on page 86
- "Configuring Hostnames on Mac OS X" on page 86
- "Mapping User and Group Identifiers for Mac OS X" on page 87

- "Access Control Lists and Mac OS X" on page 88

Requirements for Mac OS X

In addition to the items listed in "Requirements" on page 7, using a Mac OS X node to support CXFS requires the following:

- Mac OS X operating system 10.4.4 or later Tiger

- One of the following single- or dual-processor Apple Computer hardware platforms:

  Power Mac G4
  Xserve G4
  Power Mac G5
  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/cxfslicense
/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/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.

You can use the cxfslicense command to verify CXFS or XVM FLEXlm client-side
license keys for Mac OS X nodes using client-side licensing.

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 information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and
CXFS" on page 10 and "GRIO on Mac OS X" on page 106.

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 CXFS administration nodes, see the
CXFS Administration Guide for SGI InfiniteStorage.

Limitations and Considerations on Mac OS X

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

- Mac OS X does not support the `inode64` mount option. For more information, see Appendix C, "Mount Options Support" on page 281.

- 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.

- Mac OS X is unable to memory-map a file larger than 2 GB.

- 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.

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 279.

Configuring Hostnames on Mac OS X

A Mac OS X node may use a combination of methods for obtaining the node's hostname, depending on if it is in a NetInfo domain or is standalone.

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 in the Mac OS X node's `/etc/hosts` file.

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 CXFS 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 CXFS administration nodes, so any 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 CXFS administration node, and the `groups` command on the administration node. For example:

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

irix% id
uid=1113(fred) gid=999(users)
irix% groups
users 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.

To change the user's UID, GID, or other groups 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.

Alternatively, you can change the UID and GID on the CXFS administration nodes and CXFS filesystems.

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 metadata server.

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

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 (used by IRIX and SGI ProPack for Linux) 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 5-1, 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 5-1 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 on IRIX is user, group, and other, as follows:

```
irix# ls -ldD newfile
-rw-r-xr--+   1 userA    ptg               4 Jan 18 09:49 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:

```
irix# chmod 167 newfile
irix# ls -ldD newfile
---xrw-rwx+   1 userA    ptg               4 Jan 18 09:49 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:

```
irix# ls -ldD newfile
-rw-rwxr--+   1 userA    ptg               4 Jan 18 09:49 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 IRIX ACL:

```
irix# ls -ldD newfile
-rw-r-xr--+   1 userA    ptg               4 Jan 18 09:49 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:

- IRIX:

  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:

- IRIX:

```
# ls -lD newfile
-rw--wxr--+   1 userA    ptg               4 Jan 18 09:49 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 ACLs on Mac OS X

To add, remove, or edit an 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
 7: group:everyone allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
```

```
 8: 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 IRIX 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 IRIX:

```
irix# ls -dD test
test [u::rwx,g::r--,o::---/u::rw-,g::rw-,o::r--,u:501:r--,m::rwx]
```

And like this on Mac OS X:

```
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" on page 97

- "Installing the Fibre Channel Utility for Mac OS X" on page 97
- "Configuring Two or More Apple HBA Ports" on page 98
- "Using `point-to-point` Fabric Setting for Apple HBAs" on page 98

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 84.

Configuring Two or More Apple HBA Ports

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"

- "Verifying the Private and Public Networks for Mac OS X" on page 100

- "Disabling Power Save Mode for Mac OS X" on page 100

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 86) and IP address on its public network interface.

2. Decide if the Mac OS X node will be part of a NetInfo domain or a standalone machine. If part of a NetInfo 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 NetInfo database and/or 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
   > Show**

   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 NetInfo database and/or 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 Save Mode for Mac OS X

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**

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.

Note: CXFS software can also be installed from the command line. For more information about command line installation using the `installer` command, see the Mac OS X documentation.

Installing the CXFS client CD 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 release notes to learn about any late-breaking changes in the installation procedure.

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 `8.4.0` or higher.

3. Insert the *CXFS MultiOS Client 4.1* CD.

4. Using the **Finder**, open `macosx/cxfs.dmg` from the CD. This will launch the installation application, which will do the following:

   • Display the release notes

   • Display the license agreement and request acceptance

   • Force you to select the boot disk if multiple local disk partitions are installed

   Before starting the actual file installation, you may use the following menu selection to view the installation process in more detail:

   File
   > Show Log

   This information is also appended to the `/var/log/cxfs_inst.log` file.

5. *(For client-side licensing only)* Use the `cxfslicense -d` command to verify that the client-side license key is installed in `/etc/flexlm/license.dat` when

using client-side licensing. For example, the following shows a properly installed client-side license key:

```
macosx# /usr/cluster/bin/cxfslicense -d
No license required for XVM user feature.
No license required for XVM mirror feature.
CXFS_MACOSX license granted.
```

Note: If you are using server-side licensing, you should skip this step. If you execute this step on a node that uses server-side licensing at this point in the install procedure (before rebooting), you will get an error that can be ignored.

If you are using client-side licensing and there are errors, verify that you have installed the existing CXFS client-side license key. For more information, see "Verifying Existing Client-Side License Keys for Mac OS X" on page 108.

6. Restart the 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 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:

- "Upgrading the CXFS Software for Mac OS X" on page 105

- "Modifying the CXFS Software for Mac OS X" on page 105

- "Removing the CXFS Software for Mac OS X" on page 106

- "Recognizing Storage Changes for Mac OS X" on page 106

Upgrading the CXFS Software for Mac OS X

Before upgrading 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 automatically upgrade all 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:

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

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. 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.

For more information, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 10 and the *Guaranteed-Rate I/O Version 2 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
```

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

Mapping XVM Volumes to Storage Targets on Mac OS X

To map XVM volumes to storage targets on Mac OS X, do the following:

1. Get visible controller node WWNs

2. Display the desired fields:

```
ls -1 /dev/xvm-* | cut -d'-' -f2 | sort -u
```

For example:

```
macosx# ls -1 /dev/xvm-* | cut -d'-' -f2 | sort -u
200400a0b80cd5fe
200500a0b80cd5fe
```

You can also map XVM volumes to devices and to targets on RAID controllers using the output from the xvm command and the device entries in the filesystem. You can use the xvm command to display the device names:

```
macosx# /usr/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:/dev/rxvm-200400a0b80cd5fe-000)
            slice/d9400_1s0   1146334816 online,open
(d9400_1:/dev/rxvm-200500a0b80cd5fe-001)
```

These devices include the controller WWN and LUN number in their name.

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 84.

For more information, see "Verifying Access to XVM Volumes" on page 261 and the *XVM Volume Manager Administrator's Guide*.

Troubleshooting for Mac OS X

This section discusses the following:

- "Verifying Existing Client-Side License Keys for Mac OS X" on page 108
- "Reinstalling An Existing Client-Side License Key for Mac OS X" on page 109
- "The `cxfs_client` Daemon is Not Started on Mac OS X" on page 109
- "XVM Volume Name is Too Long on Mac OS X" on page 110
- "Large Log Files on Mac OS X" on page 110
- "Error Reading Kernel XVM Cluster Mirror Status" on page 110

For general troubleshooting information, see Chapter 10, "General Troubleshooting" on page 263 and Appendix D, "Error Messages" on page 289

Verifying Existing Client-Side License Keys for Mac OS X

Note: Server-side licensing is preferred. Mac OS X 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. See "License Keys" on page 9.

To verify that the client-side license key has been installed properly on a Mac OS X when using client-side licensing, use the `cxfslicense -d` command. For example:

- Client-side license key installed properly on a Mac OS X node:

  ```
  macosx# /usr/cluster/bin/cxfslicense -d
  No license required for XVM user feature.
  No license required for XVM mirror feature.
  CXFS_MACOSX license granted.
  ```

- No client-side license key on a Mac OS X node:

```
macosx# /usr/cluster/bin/cxfslicense -d
No license required for XVM user feature.
No license required for XVM mirror feature.
Error on CXFS_MACOSX license check out.
```

```
checkout failed: Cannot find license file
 The license files (or server network addresses) attempted are
 listed below.  Use LM_LICENSE_FILE to use a different license file,
 or contact your software provider for a license file.
Feature:        CXFS_MACOSX
Filename:       /etc/flexlm/license.dat
License path:   /etc/flexlm/license.dat
FLEXlm error:   -1,359.  System Error: 2 "No such file or directory"
For further information, refer to the FLEXlm End User Manual,
available at "www.macrovision.com".

No client-side license, will request a server-side license.
```

The above output indicates that no client-side license is installed and the CXFS software will request a server-side license key. If you are using client-side licensing, verify that you have obtained and installed the CXFS license keys.

Reinstalling An Existing Client-Side License Key for Mac OS X

An existing Mac OS X client-side license key is installed in the following location:

/etc/flexlm/license.dat

Do the following:

1. Create the license key directory if necessary. For example:

   macosx# **mkdir -p /etc/flexlm**

2. Copy the key to the license.dat file.

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**

The cxfs_client daemon might not start if the license key has expired, which is indicated when the following message is output to the /var/log/cxfs_client file:

CXFS not properly licensed for this host

In this case, execute the following command to determine why the license key check fails:

```
macosx# /usr/cluster/bin/cxfslicense -d
```

If the license key is valid, check the cxfs_client log file for other errors.

Restart cxfs_client as described in "Start/Stop cxfs_client for Mac OS X" on page 104 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 86.

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 85.

Error Reading Kernel XVM Cluster Mirror Status

The following message will be output by the cxfslicense -d command if you execute it before rebooting the system:

```
error reading kernel XVM cluster mirror status. Check if XVM module is
started.
```

After you reboot the system and therefore load the XVM module, this message will no longer appear when you run cxfslicense -d. You can ignore this message if you have not rebooted.

Reporting Mac OS X Problems

When reporting a problem about a CXFS Mac OS X node to SGI, you should run `/usr/cluster/bin/cxfs_dump` and send the `tar.gz` file that is created in the `/var/cluster/cxfsdump-data/`*date_time* directory to SGI.

SGI ProPack Client-Only Platform

SGI ProPack for Linux is an overlay product that adds or enhances features in the supported Linux base distributions. This chapter discusses the following:

- "CXFS on SGI ProPack Client-Only Nodes" on page 113
- "Client Software Installation for SGI ProPack Client-Only Nodes" on page 118
- "I/O Fencing for SGI ProPack Client-Only Nodes" on page 122
- "Start/Stop cxfs_client for SGI ProPack Client-Only Nodes" on page 124
- "Maintenance for SGI ProPack Client-Only Nodes" on page 125
- "XVM Failover V2 on SGI ProPack Client-Only Nodes" on page 128
- "Mapping XVM Volumes to Storage Targets on SGI ProPack" on page 130
- "Troubleshooting for SGI ProPack Client-Only Nodes" on page 131
- "Reporting SGI ProPack Client-Only Nodes Problems" on page 133

For information about SGI ProPack server-capable administration nodes, see *CXFS Administration Guide for SGI InfiniteStorage*.

CXFS on SGI ProPack Client-Only Nodes

This section discusses the following:

- "Requirements for SGI ProPack Client-Only Nodes" on page 114
- "CXFS Commands on SGI ProPack Client-Only Nodes" on page 115
- "Log Files on SGI ProPack Client-Only Nodes" on page 115
- "CXFS Mount Scripts on SGI ProPack Client-Only Nodes" on page 116
- "Limitations and Considerations for SGI ProPack Client-Only Nodes" on page 116

Requirements for SGI ProPack Client-Only Nodes

In addition to the items listed in "Requirements" on page 7, using an SGI ProPack client-only node to support CXFS requires the following:

- SGI ProPack 4 SP3 sn2 or rtgfx kernel on Altix systems.

  Note: On SGI Altix systems, CXFS supports either a *server-capable administration node* containing the cluster administration daemons (fs2d, crsd, cad, and cmond), the CXFS control daemon (clconfd), and the cluster database or a *client-only node* containing the cxfs_client daemon. The software you install on a node determines the node type.

  Nodes that you intend to run as metadata servers must be installed as administration nodes; all other nodes should be client-only nodes.

- SGI ProPack 5 default kernel on Altix systems

- SGI ProPack 5 smp kernel on Altix XE systems

CXFS requires the hardware and software specified in the release notes:

- A supported SAN hardware configuration.

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

- Use a network switch. (A network hub is not supported.) The switch should be at least 100baseT.

- A private 100baseT or Gigabit Ethernet TCP/IP network connected to each node.

  Note: When using Gigabit Ethernet, do not use jumbo frames. For more information, see the tgconfig man page.

- 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.

- At least one host bus adapter (HBA) as specified in the release notes.

- RAID hardware as specified in the release notes.

- The XVM volume manager, which is provided with the CXFS release.

- 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.

- SGI ProPack 5 client-only nodes require server-side licensing. For more information, see *CXFS Administration Guide for SGI InfiniteStorage*.

CXFS Commands on SGI ProPack Client-Only Nodes

The following commands are shipped as part of the CXFS SGI ProPack 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/cluster/bin/cxfslicense
/usr/sbin/grioadmin
/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. For more information, see the man pages.

Log Files on SGI ProPack Client-Only Nodes

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 SGI ProPack platform uses the `logrotate` system utility to rotate the `cxfs_client` logs:

- 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

CXFS Mount Scripts on SGI ProPack Client-Only Nodes

SGI ProPack supports the CXFS mount scripts. See "CXFS Mount Scripts" on page 6 and the *CXFS Administration Guide for SGI InfiniteStorage*.

Limitations and Considerations for SGI ProPack Client-Only Nodes

The following sections highlight limitations and considerations for SGI ProPack nodes.

Limitations and Considerations for Any SGI ProPack Node

The following limitations and considerations apply to any SGI ProPack node (client-only or server-capable):

- By default, DMAPI is turned off on SGI ProPack 5 systems. When you install DMF on a server-capable node, it automatically enables DMAPI. However, if you want to mount filesystems on an SGI ProPack 5 client-only node with the `dmi` mount option, you must ensure that the `DMAPI_PROBE` system tunable parameter on the node is set to `yes` in the `/etc/sysconfig/sysctl` file. Changes to the file will be 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
  ```

  If you run a DMAPI application other than DMF, you must also change parameter on the SGI ProPack 5 server-capable nodes.

- On SGI ProPack systems, the `mkfs.xfs` command does not discover log or realtime subvolumes. You must specify the log or realtime subvolumes on the command line. For more information, see the `mkfs.xfs`(8) man page.

- GPT partition tables, often created by operating system installers or the `parted` partitioning tool, store labels in two locations. If you reuse a disk that previously had a GPT label, you must be careful; using tools such as `fdisk` to repartition the drive will not eliminate the backup GPT label. When you reboot, EFI scans the disks before the operating system is started. It assumes any backup labels it finds are valid and restores them. This can corrupt or destroy filesystems. You can use the `parted` tool to detect this situation and fix it.

> **Note:** The `parted` tool has a `mkpartsect` command that accepts start and end
> values for partitions being created in sectors rather than MB. For more
> information, see the *XVM Volume Manager Administrator's Guide* and
> http://support.sgi.com/content_request/838562/index.html on Supportfolio.

- CXFS filesystems with XFS version 1 directory format cannot be mounted on SGI
  ProPack nodes.

- Whenever you install a new kernel patch, you must also install the corresponding
  CXFS package. This is required because the kernel patch causes the kernel version
  number to be increased. Failure to install the corresponding CXFS package will
  result in the inability to run CXFS. To obtain the required CXFS package, see your
  SGI support contact.

- After upgrading CXFS, you should reboot the system in order to make the new
  updates to take effect. A reboot is not required if you are performing a fresh
  installation.

- 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).

Limitations and Considerations for SGI ProPack Client-Only Nodes

Client-only SGI ProPack nodes have the following limitations and considerations:

- Client-only nodes cannot view or edit user and group quotas. However, user and
  group quotas are enforced correctly by the metadata server.

- To view or edit your quota information, you must log in to an administration
  node and make any necessary changes. If you would like to provide a viewing
  command such as `repquota`, you could construct shell script similar to the
  following on the SGI ProPack node:

```
#! /bin/sh
#

# Where repquota lives on administration node
```

```
repquota=/usr/etc/repquota

# The name of an administration node in the cluster
adminnode=cain

rsh $adminnode "$repquota $*"
exit
```

- On SGI Altix systems 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
```

Client Software Installation for SGI ProPack Client-Only Nodes

The CXFS client-only 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: Package version numbers shown here are examples; your installed system may differ.

This section covers the following:

- "SGI ProPack Client-Only Installation Overview" on page 118
- "Installing the Performance Co-Pilot Agent" on page 122
- "Verifying the SGI ProPack Client-Only Installation" on page 122

SGI ProPack Client-Only Installation Overview

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

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

To install the required software on an SGI ProPack client-only node, SGI personnel will do the following:

1. Read the release notes to learn about any late-breaking changes in the installation procedure.

2. Install the SGI ProPack release, according to the directions in the SGI ProPack documentation. Ensure that you select the `SGI Licensed` package group. You must install the `pcp-open` package from the SGI ProPack release.

   Note: When installing the Linux OS, disconnect the system from the fabric or ensure that the drive you are installing on is not a SAN-attached drive.

3. Install any required patches. See the SGI ProPack `releasenotes/README` file for more information.

   ⚠ **Caution:** You **must** update the operating system with all security fixes, bug fixes, and enhancements available from the operating system vendor.

4. Verify that the node is running the supported Linux distribution and SGI ProPack overlay, according to the CXFS for SGI ProPack release notes. See the `/etc/SuSE-release` and `/etc/sgi-release` files.

5. If you have previously installed XVM in standalone mode, remove any remaining `sgi-xvm-standalone` package. To find and remove the package:

```
[root@linux CXFS_CDROM]# rpm -e --allmatches `rpm -qa | grep xvm-standalone`
```

   If installing on an SGI ProPack 5 client, you may also need to remove weak-updates links from the sgi-xvm-standalone RPM. If you are running the 2.6.16.21-0.25 kernel, you would do the following:

```
[root@linux CXFS_CDROM]# rm -rf /lib/modules/2.6.16.21-0.25-default/weak-updates/os_lib
[root@linux CXFS_CDROM]# rm -rf /lib/modules/2.6.16.21-0.25-default/weak-updates/xvm
```

6. Insert and mount the *CXFS MultiOS Client 4.1* CD.

7. Install the CXFS kernel modules:

> **Note:** This procedure uses the `rpm -U` option to update RPMs, which works for an initial installation as well as updates. For an initial installation, you could also use `-i`.

```
[root@linux cdrom]# rpm -Uvh sgi-cxfs-kmp-kernelvariant-kernelrelease-version.architecture.rpm
Preparing...                 ######################################### [100%]
   1:sgi-cxfs-kmp-kernelvariant-######################################### [100%]
```

Where:

- *kernelvariant* and *kernelrelease* are the kernel variant and release level as output by the `uname -r` command

- *version* is the version number

- *architecture* is the processor architecture type

> **Note:** For SGI ProPack 4, the *kernelrelease* must match the stock kernel release provided by SUSE. For SGI ProPack 5 running SLES 10, one version of CXFS may support one or more *kernelrelease* values. See the CXFS SGI ProPack release notes for the supported versions.

8. Install the user-space packages:

```
[root@linux cdrom]# rpm -Uvh cxfs_client* cxfs_util* cxfs-xvm-cmds* cxfs-doc*
Preparing...                 ######################################### [100%]
   1:cxfs-xvm-cmds            ######################################### [ 25%]
boot.xvm              0:off  1:off  2:off  3:off  4:off  5:off  6:off
   2:cxfs_util              ######################################### [ 50%]
   3:cxfs_client            ######################################### [ 75%]
cxfs_client           0:off  1:off  2:off  3:on   4:off  5:on   6:off
   4:cxfs-doc               ######################################### [100%]
boot.xvm              0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

> **Note:** The order of RPMs listed on the command line is not necessarily the same order in which they will be displayed in the `rpm` command output.

9. If you are using GRIO, install the `grio2-cmds` package:

```
[root@linux cdrom]# rpm  -Uvh grio2-cmds*
Preparing...              ###########################################[100%]
   1:grio2-cmds           ###########################################[100%]
```

10. Edit the `/etc/cluster/config/cxfs_client.options` file as necessary. See "Modifying the CXFS Software for SGI ProPack" on page 125 and the `cxfs_client`(1M) man page.

11. Reboot the system with the newly installed kernel:

    ```
    [root@linux cdrom]# reboot
    ```

12. *(For client-side licensing only)* Use the `cxfslicense -d` command to verify that the existing CXFS client-side license key is installed in `/etc/flexlm/license.dat`. For example, the following shows a properly installed client-side license key:

    ```
    # /usr/cluster/bin/cxfslicense -d
    Found 1 XVM_STD_IPF license(s) of version 1.0 for XVM.
    Didn't find XVM_PLEX_IPF license of version 1.0 for XVM.
    Found 1 XVM_PLEX_CLUSTER license(s) of version 4.0 for XVM.

    Found version 1.0 license for CXFS_IPF.
    ```

    Note: If you are using server-side licensing, you should skip this step. If you execute this step on a node that uses server-side licensing at this point in the install procedure (before rebooting), you will get an error that can be ignored.

    If you are using client-side licensing and there are errors, verify that you have installed the existing CXFS client-side license key. For more information, see "Verifying Existing Client-Side License Keys for AIX" on page 45.

13. Modify `updatedb` behavior so that it avoids CXFS filesystems.

⚠ **Caution:** If XVM standalone was in use prior to CXFS installation, you must reboot the system before starting CXFS services to ensure that the new `xvm` modules are loaded.

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
pcp-sgi
```

These packages and are included on the first and second SGI ProPack CDs (respectively). You can obtain the open source PCP package from ftp://oss.sgi.com/projects/pcp/download

Verifying the SGI ProPack Client-Only Installation

To verify that the CXFS software has been installed properly, use the `rpm -q` command to query the packages.

To verify the SGI ProPack release, display the `/etc/sgi-release` file.

I/O Fencing for SGI ProPack Client-Only Nodes

On the SGI ProPack 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/log/cxfs_client` file.

If no WWPNs are detected, you can manually specify the WWPNs in the `/etc/fencing.conf` 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.

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. Determine the HBA WWPN: 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 Administration Guide for SGI InfiniteStorage*.

Start/Stop `cxfs_client` for SGI ProPack Client-Only Nodes

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:

```
# /etc/init.d/cxfs_client start
Loading cxfs modules:                                          [  OK  ]
Mounting devfs filesystems:                                    [  OK  ]
Starting cxfs client:                                          [  OK  ]
```

To stop CXFS processes manually, enter the following command:

```
# /etc/init.d/cxfs_client stop
Stopping cxfs client:                                    [  OK  ]
```

Maintenance for SGI ProPack Client-Only Nodes

This section discusses the following:

- "Modifying the CXFS Software for SGI ProPack" on page 125
- "Recognizing Storage Changes for SGI ProPack" on page 125

Modifying the CXFS Software for SGI ProPack

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

`/etc/cluster/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`, enter the following:

```
propack# 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
```

Recognizing Storage Changes for SGI ProPack

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 ProPack due to limitations in the Linux 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` 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, for SGI ProPack 4:

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

For SGI ProPack 5:

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

GRIO on SGI ProPack Client-Only Nodes

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the SGI ProPack client-only platform.

GRIO is disabled by default on SGI ProPack. To enable GRIO, change the following line in `/etc/cluster/config/cxfs_client.options` from:

`export GRIO2=off`

to:

`export GRIO2=on`

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.

An SGI ProPack client-only node can mount a GRIO-managed filesystem and supports application- and node-level reservations. An SGI ProPack client-only node client 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 10 and the *Guaranteed-Rate I/O Version 2 Guide*.

XVM Failover V2 on SGI ProPack Client-Only Nodes

The following example for SGI ProPack 4 groups the paths for `lun3` and the paths for `lun4`. The order of paths in the file is not significant. Paths to the same LUN are detected automatically. Without this file, all paths to each LUN would have affinity 0 and there would be no preferred path. Setting a `preferred` path ensures that multiple paths will be used for performance. If no path is designated as `preferred`, the path used to the LUN is arbitrary based on the order of device discovery. There is no interaction between the preferred path and the affinity values.

This file uses affinity to group the RAID controllers for a particular path. Each controller has been assigned an affinity value. It shows the following:

- There is one PCI card with two ports off of the HBA (`pci04.01.1` and `pci04.01.0`)

- There are two RAID controllers, `node200800a0b813b982` and `node200900a0b813b982`

- Each RAID controller has two ports that are identified by `port1` or `port2`

- Each LUN has eight paths (via two PCI cards, two RAID controllers, and two ports on the controllers)

- There are two affinity groups for each LUN, `affinity=0` and `affinity=1`

- There is a `preferred` path for each LUN

```
/dev/xscsi/pci04.01.1/node200900a0b813b982/port1/lun3/disc,  affinity=0
/dev/xscsi/pci04.01.1/node200900a0b813b982/port2/lun3/disc,  affinity=0
/dev/xscsi/pci04.01.0/node200900a0b813b982/port1/lun3/disc,  affinity=0
/dev/xscsi/pci04.01.0/node200900a0b813b982/port2/lun3/disc,  affinity=0   preferred
/dev/xscsi/pci04.01.1/node200800a0b813b982/port1/lun3/disc,  affinity=1
/dev/xscsi/pci04.01.0/node200800a0b813b982/port1/lun3/disc,  affinity=1
/dev/xscsi/pci04.01.1/node200800a0b813b982/port2/lun3/disc,  affinity=1
/dev/xscsi/pci04.01.0/node200800a0b813b982/port2/lun3/disc,  affinity=1

/dev/xscsi/pci04.01.1/node200900a0b813b982/port1/lun4/disc, affinity=0
/dev/xscsi/pci04.01.1/node200900a0b813b982/port2/lun4/disc, affinity=0
/dev/xscsi/pci04.01.0/node200900a0b813b982/port1/lun4/disc, affinity=0
/dev/xscsi/pci04.01.0/node200900a0b813b982/port2/lun4/disc, affinity=0
/dev/xscsi/pci04.01.1/node200800a0b813b982/port1/lun4/disc, affinity=1
/dev/xscsi/pci04.01.1/node200800a0b813b982/port2/lun4/disc, affinity=1 preferred
/dev/xscsi/pci04.01.0/node200800a0b813b982/port1/lun4/disc, affinity=1
/dev/xscsi/pci04.01.0/node200800a0b813b982/port2/lun4/disc, affinity=1
```

In SGI ProPack 5, the file would be:

```
/dev/xscsi/pci0004:00:01.1/node200900a0b813b982/port1/lun3/disc,  affinity=0
/dev/xscsi/pci0004:00:01.1/node200900a0b813b982/port2/lun3/disc,  affinity=0
/dev/xscsi/pci0004:00:01.0/node200900a0b813b982/port1/lun3/disc,  affinity=0
/dev/xscsi/pci0004:00:01.0/node200900a0b813b982/port2/lun3/disc,  affinity=0   preferred
/dev/xscsi/pci0004:00:01.1/node200800a0b813b982/port1/lun3/disc,  affinity=1
/dev/xscsi/pci0004:00:01.0/node200800a0b813b982/port1/lun3/disc,  affinity=1
/dev/xscsi/pci0004:00:01.1/node200800a0b813b982/port2/lun3/disc,  affinity=1
/dev/xscsi/pci0004:00:01.0/node200800a0b813b982/port2/lun3/disc,  affinity=1

/dev/xscsi/pci0004:00:01.1/node200900a0b813b982/port1/lun4/disc, affinity=0
/dev/xscsi/pci0004:00:01.1/node200900a0b813b982/port2/lun4/disc, affinity=0
/dev/xscsi/pci0004:00:01.0/node200900a0b813b982/port1/lun4/disc, affinity=0
/dev/xscsi/pci0004:00:01.0/node200900a0b813b982/port2/lun4/disc, affinity=0
/dev/xscsi/pci0004:00:01.1/node200800a0b813b982/port1/lun4/disc, affinity=1
/dev/xscsi/pci0004:00:01.1/node200800a0b813b982/port2/lun4/disc, affinity=1 preferred
/dev/xscsi/pci0004:00:01.0/node200800a0b813b982/port1/lun4/disc, affinity=1
/dev/xscsi/pci0004:00:01.0/node200800a0b813b982/port2/lun4/disc, affinity=1
```

Given the above, failover will exhaust all paths to `lun3` from RAID controller `node200900a0b813b982` (with `affinity=0` and the `preferred` path) before moving to RAID controller `node200800a0b813b982` paths (with `affinity=1`)

The following example for SGI ProPack 4 shows an additional grouping of PCI cards. The preferred path has an affinity of 2. If that path is not available, the failover mechanism will try the next path on the same PCI card (with `affinity=2`). If that is not successful, it will move to `affinity=3`, which is the other PCI port on the same RAID controller (`node200800a0b813b982`).

```
/dev/xscsi/pci04.01.1/node200900a0b813b982/port1/lun4/disc, affinity=0
/dev/xscsi/pci04.01.1/node200900a0b813b982/port2/lun4/disc, affinity=1
/dev/xscsi/pci04.01.0/node200900a0b813b982/port1/lun4/disc, affinity=0
/dev/xscsi/pci04.01.0/node200900a0b813b982/port2/lun4/disc, affinity=1
/dev/xscsi/pci04.01.1/node200800a0b813b982/port1/lun4/disc, affinity=3
/dev/xscsi/pci04.01.1/node200800a0b813b982/port2/lun4/disc, affinity=2 preferred
/dev/xscsi/pci04.01.0/node200800a0b813b982/port1/lun4/disc, affinity=3
/dev/xscsi/pci04.01.0/node200800a0b813b982/port2/lun4/disc, affinity=2
```

The example file for SGI ProPack 5 would be:

```
/dev/xscsi/pci0004:00:01.1/node200900a0b813b982/port1/lun4/disc, affinity=0
/dev/xscsi/pci0004:00:01.1/node200900a0b813b982/port2/lun4/disc, affinity=1
/dev/xscsi/pci0004:00:01.0/node200900a0b813b982/port1/lun4/disc, affinity=0
/dev/xscsi/pci0004:00:01.0/node200900a0b813b982/port2/lun4/disc, affinity=1
/dev/xscsi/pci0004:00:01.1/node200800a0b813b982/port1/lun4/disc, affinity=3
/dev/xscsi/pci0004:00:01.1/node200800a0b813b982/port2/lun4/disc, affinity=2 preferred
/dev/xscsi/pci0004:00:01.0/node200800a0b813b982/port1/lun4/disc, affinity=3
/dev/xscsi/pci0004:00:01.0/node200800a0b813b982/port2/lun4/disc, affinity=2
```

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

Mapping XVM Volumes to Storage Targets on SGI ProPack

You can use the `cxfs-enumerate-wwns` script to map XVM volumes to storage targets (assuming that CXFS software is installed.)

```
# /var/cluster/cxfs_client-scripts/cxfs-enumerate-wwns | grep -v "#"| sort -u
```

Troubleshooting for SGI ProPack Client-Only Nodes

This section discusses the following:

- "lcrash and SLES 9" on page 131

- "Verifying Existing Client-Side License Keys for SGI ProPack 4" on page 131

- "Reinstalling An Existing Client-Side License Key for SGI ProPack 4" on page 132

- "Error Reading Kernel XVM Cluster Mirror Status" on page 132

For general troubleshooting information, see Chapter 10, "General Troubleshooting" on page 263 and Appendix D, "Error Messages" on page 289. Also see *CXFS Administration Guide for SGI InfiniteStorage*.

lcrash and SLES 9

Do not use the version of lcrash that is shipped with SLES 9. Use the version of lcrash that is available from Supportfolio with the -x option to load the CXFS kerntypes:

```
# lcrash -x /boot/sgi-cxfs-kerntypes-kernelversion-architecturetype
```

Verifying Existing Client-Side License Keys for SGI ProPack 4

Note: Server-side licensing is preferred and is required for SGI ProPack 5. SGI ProPack 4 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. See "License Keys" on page 9.

To verify that the existing SGI ProPack 4 client-side license keys have been installed properly when using client-side licensing, use the cxfslicense -d command. For example:

- Client-side license key installed on an SGI ProPack 4 node:

```
propack4# /usr/cluster/bin/cxfslicense -d
Found 1 XVM_STD_IPF license(s) of version 1.0 for XVM.
Didn't find XVM_PLEX_IPF license of version 1.0 for XVM.
```

```
                              Found 1 XVM_PLEX_CLUSTER license(s) of version 4.0 for XVM.
                              Found version 1.0 license for CXFS_IPF.
```

- No client-side license key on an SGI ProPack 4 node:

```
propack4# /usr/cluster/bin/cxfslicense -d
Found 1 XVM_STD_IPF license(s) of version 1.0 for XVM.
Didn't find XVM_PLEX_IPF license of version 1.0 for XVM.
Found 1 XVM_PLEX_CLUSTER license(s) of version 4.0 for XVM.

No client-side license, will request a server-side license.
```

The above output indicates that no client-side license key is installed and the CXFS software will request a server-side license key.

If there are errors, verify that you have obtained and installed the CXFS license keys.

Reinstalling An Existing Client-Side License Key for SGI ProPack 4

An existing SGI ProPack 4 client-side license key is installed in the following location:

```
/etc/flexlm/license.dat
```

Do the following:

1. Create the license key directory if necessary. For example:

   ```
   solaris# mkdir -p /etc/flexlm
   ```

2. Copy the key to the license.dat file.

Error Reading Kernel XVM Cluster Mirror Status

The following message will be output by the cxfslicense -d command if you execute it before rebooting the system:

```
error reading kernel XVM cluster mirror status. Check if XVM module is
started.
```

After you reboot the system and therefore load the XVM module, this message will no longer appear when you run cxfslicense -d. You can ignore this message if you have not rebooted.

Reporting SGI ProPack Client-Only Nodes Problems

Retain the following information for SGI ProPack nodes:

- The kernel you are running:

  [root@linux root]# **uname -a**

- The CXFS packages you are running:

[root@linux root]# **rpm -q cxfs_client sgi-cxfs-kmp cxfs_utils cxfs-xvm-cmds**

- The number and types of processors in your machine:

  [root@linux root]# **cat /proc/cpuinfo**

- The hardware installed on your machine:

  [root@linux root]# **/sbin/lspci**

- Modules that are loaded on your machine:

  [root@linux root]# **/sbin/lsmod**

- The /var/log/cxfs_client log file

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

- Output about the cluster obtained from the cxfsdump utility run on an administration node.

- After a system kernel panic, the debugger information from the KDB built-in kernel debugger.

- 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.

- Output from the following commands:

– Information from the following files:

```
/var/log/messages
/var/log/cxfs_client      (for client-only nodes)
/etc/failover.conf        (for XVM failover version 1)
/etc/failover2.conf       (for XVM failover version 2)
/etc/hosts
/proc/discontig
```

– Output from the following commands:

```
/usr/cluster/bin/cdbutil gettree '#'
/usr/bin/hinv
/usr/bin/topology
/sbin/xvm show -v phys
/sbin/xvm show -top -v vol
/bin/netstat -ia
```

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:

- "Access Control Lists and Solaris" on page 139

- "`maxphys` System Tunable for Solaris" on page 140

Requirements for Solaris

In addition to the items listed in "Requirements" on page 7, using a Solaris node to support CXFS requires the following:

- Solaris operating system:

  – Solaris 10 January 06 (patch 118822-25)

  ⚠️ **Caution:** This release of CXFS Solaris only supports the January 06 release (patch 118822-25) of Solaris 10. All other releases of Solaris 10 are not supported and may cause system instabilities when used with CXFS.

  – Solaris 9 March 06 (patch 112233–06) or later.

- Supported Fibre Channel HBAs:

  – AMCC JNI FibreStar models:

  FCE-6460
  FCX-6562
  FCX2-6562

  Note: 1-Gbit HBAs and Sbus HBAs are not supported.

  – LSI Logic models:

  LSI7102XP
  LSI7202XP
  LSI7402XP
  LSI7104XP
  LSI7204XP
  LSI7404XP

– QLogic models sold by Sun Microsystems:

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

⚠ **Caution:** SGI supports only the AMCC JNI HBA that is purchased directly from AMCC. The AMCC JNI HBA that is provided by Sun Microsystems **does not function with CXFS** and cannot be configured to do so.

• 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.

Note: CXFS supports a Solaris client only on the SPARC platform. It is not supported on other hardware platforms.

For the 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/cxfslicense
/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.

You can use the `cxfslicense` command to verify CXFS or XVM FLEXlm client-side license keys for Solaris nodes using client-side licensing.

The `pkgadd` output lists all software added; see "Solaris Installation Overview" on page 161.

For more information, see the man pages.

For information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 10 and "GRIO on Solaris" on page 168.

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 CXFS administration nodes, see the *CXFS Administration Guide for SGI InfiniteStorage*.

CXFS Mount Scripts on Solaris

Solaris supports the CXFS mount scripts. See "CXFS Mount Scripts" on page 6 and the *CXFS Administration Guide for SGI InfiniteStorage*.

Limitations and Considerations on Solaris

Note the following:

- IRIX nodes do not permit nested mount points on CXFS filesystems; that is, you cannot mount an IRIX XFS or CXFS filesystem on top of an existing CXFS filesystem. 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 140.

- All disk devices attached to AMCC JNI or 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 163 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 279.

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, 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 IRIX 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 IRIX 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) in the `/etc/system` file.

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

Solaris supports the following host bus adapters (HBAs):

- AMCC JNI HBA

- LSI Logic HBA

- QLogic HBA

The QLogic driver is provided with Solaris 10.

The remainder of this section discusses the following:

- "Installing the AMCC JNI HBA"

- "Installing the LSI Logic HBA" on page 152

- "Verifying the HBA Installation" on page 154

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.

Note: CXFS is not compatible with the firmware installed on a Sun OEM JNI HBA. JNI HBAs should be sourced from AMCC or an AMCC reseller.

Installing the AMCC JNI HBA

To install the AMCC JNI HBA, perform the following steps. Details are provided in the *Installation Guide, FCE-6460 and FCE2-6460 PCI-to-Fibre Channel Host Bus Adapters (Solaris, Windows NT/2000, Novell, AIX, HP-UX, Mac OS, Linux) JNI FibreStar*, as noted.

1. Install the AMCC JNI host bus adapter (HBA) into the Solaris system. Perform the steps in the following chapter:

   • "Hardware Installation"

   Note: The AMCC JNI card **must** be installed in a 66MHz slot; if it is installed in another type of slot, CXFS will not work properly.

2. Bring the system back up using the steps listed in the following "Verifying" sections (the following represents the location of these sections in the manual):

   • "Unix Server DriverSuite"

     – "Solaris Driver"

       • "Verifying Hardware in OpenBoot PROM"

       • "Verifying Hardware in Solaris"

   You will be required to perform a Solaris `boot -r` after installing hardware.

   Caution: If you do not see the expected results, do not attempt to continue. Instead, diagnose why the card is not being seen by the hardware.

3. Install the AMCC JNI HBA driver software (`JNIC146x`) v5.3.*x.x* according to the instructions in the following "Installing" section:

   • "Unix Server DriverSuite"

     – "Solaris Driver"

       • "Installing the Software"

   Do the following:

   a. Retrieve the driver package from the following AMCC website:

      http://www.amcc.com/drivers

   b. Install the driver package with the following command:

      solaris# **pkgadd -d** *package*

      Choose to install both the HBA driver and the SNIA HBA library.

Verify that the driver attached correctly to the HBA and that the package installed correctly by following the verification steps at the end of the section. Do not proceed until the verification succeeds.

4. Set the HBA to fabric mode:

   a. In the `/kernel/drv/jnic146x.conf` file, change the following lines:

      ```
      # FcLoopEnabled=1;
      # FcFabricEnabled=0;
      ```

      Delete the # character at the beginning of each line to uncomment it and change the values so that loop is disabled (0) and fabric is enabled (1). When you are done, the lines will appear as follows:

      ```
      FcLoopEnabled=0;
      FcFabricEnabled=1;
      ```

   b. Reboot the Solaris node by entering the following command:

      ```
      solaris# init 6
      ```

Setting the LUN Discovery Method for SGI TP9100

If you are using an SGI TP9100 controller and the v5.3 AMCC JNI driver, you must set the LUN discovery method to 0 (meaning "SCSI Inquiry") in the `/kernel/drv/jnic146x.conf` file. Change the following line from:

```
# LunDiscoveryMethod = 1;
```

to:

```
LunDiscoveryMethod = 0;
```

Failure to do this will limit access to only LUN 0 on each target.

Installing the JNI AMCC EZ Fibre Configuration Utility

After you have verified the installation of the HBA and the driver's attachment to it, you are ready to install and run the EZ Fibre program. This graphical user interface (GUI) will modify the driver's configuration file, `/kernel/drv/jnic146x.conf`, so that it lists the worldwide node name (WWNN) and worldwide port name (WWPN) for the devices on your Fibre Channel.

For general installation information, see *Quick Installation Guide, Solaris, AIX and Windows JNI EZ Fibre*.

Do the following:

1. Download the latest EZ Fibre GUI from the following website:

   http://www.jni.com/drivers

2. Install the EZ Fibre GUI:

   a. Extract the saved file using the `tar` command. For example:

      solaris# **tar xvf EZF_*xxx*.tar**

   b. Change to the directory where the extracted GUI command is located:

      solaris# **cd EZF_*xx***

   c. Run the `install.sh` script:

```
solaris# ./install.sh
Checking for required and recommended patches...
checkpatches.sh: Note - The following OS vendor recommended patches are
not installed or have been superseded -- please consult the EZ-Fibre
read me:
 108434-01 108435-01
<press enter to continue>

InstallAnywhere is preparing to install...
Installer using temporary disk space in '/tmp' ($TMPDIR not set).
```

You should install the GUI into the default location. When you see
Congratulations!, click on **Done**.

3. Change to the following directory and read any README files you find there:

   solaris# **cd /opt/jni/ezfibre/standalone**

Protecting Data Integrity

It is possible to put data integrity at risk if Fibre Channel cables are disconnected or
fail. To protect against this issue, you must uncomment the `FailoverDelay`
parameter in the `/kernel/drv/jnic146x.conf` file and change it to 1. (A setting
of 0 disables XVM failover v2). It then should appear as follows:

```
FailoverDelay = 1;
```

This change ensures data integrity in the CXFS filesystem in the event that the Solaris
client machine is unable to see the filesystems.

Discovering and Configuring LUNs

Do the following:

1. Invoke the EZ Fibre GUI from within the same directory by entering the following command:

   ```
   solaris# ./ezf
   ```

   Two windows will appear. The first, titled **Refresh SAN information**, will say **Discovering LUNs for HBA#0**.

   After a short while, this window will be replaced by a larger window, as shown in the example in Figure 7-1. (The example screen snaps in this document may not exactly match the display you see.)

File View Help	
System View	
cxfssun4 (SunOS)	

System Information | Failover/Dyn Load Balancing

System Variable	System Value
System Hostname	cxfssun4
IP Address	128.162.89.129
Operating System	SunOS
Release	5.8
Version	Generic_108528-12
Machine	sun4u
System Architecture	sparcv9+vis
Platform	SUNW,Ultra-250
EZ Fibre Agent Version	2.2d build 011106-10 64-bit

Reboot System   Help

Ready

Figure 7-1 Example: Second Window: EZ Fibre Configuration Utility - Standalone

The left-hand pane of this window displays a listing of systems. Find the system you are configuring and click on the + sign next to it; this action expands the display so that it shows the installed JNI HBA on the system. Figure 7-2 highlights the + sign.
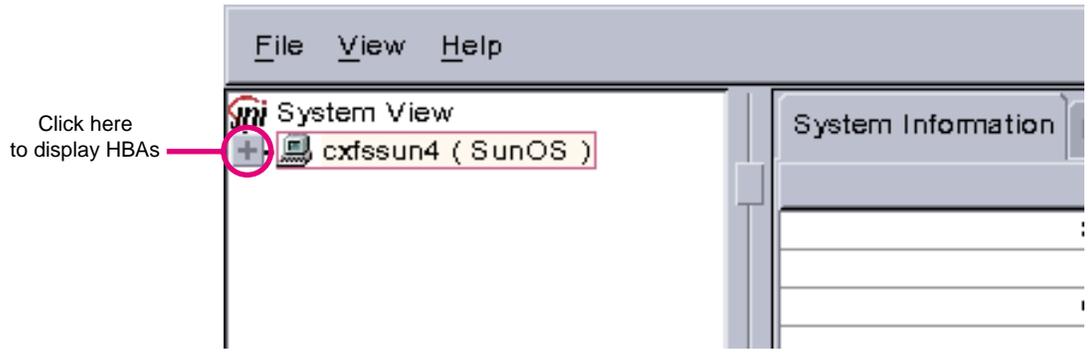
Figure 7-2 Location of icon (+) to Display the HBA

Figure 7-3 shows an example of the display after clicking on the + sign for cxfssun4, which shows the JNI HBA.
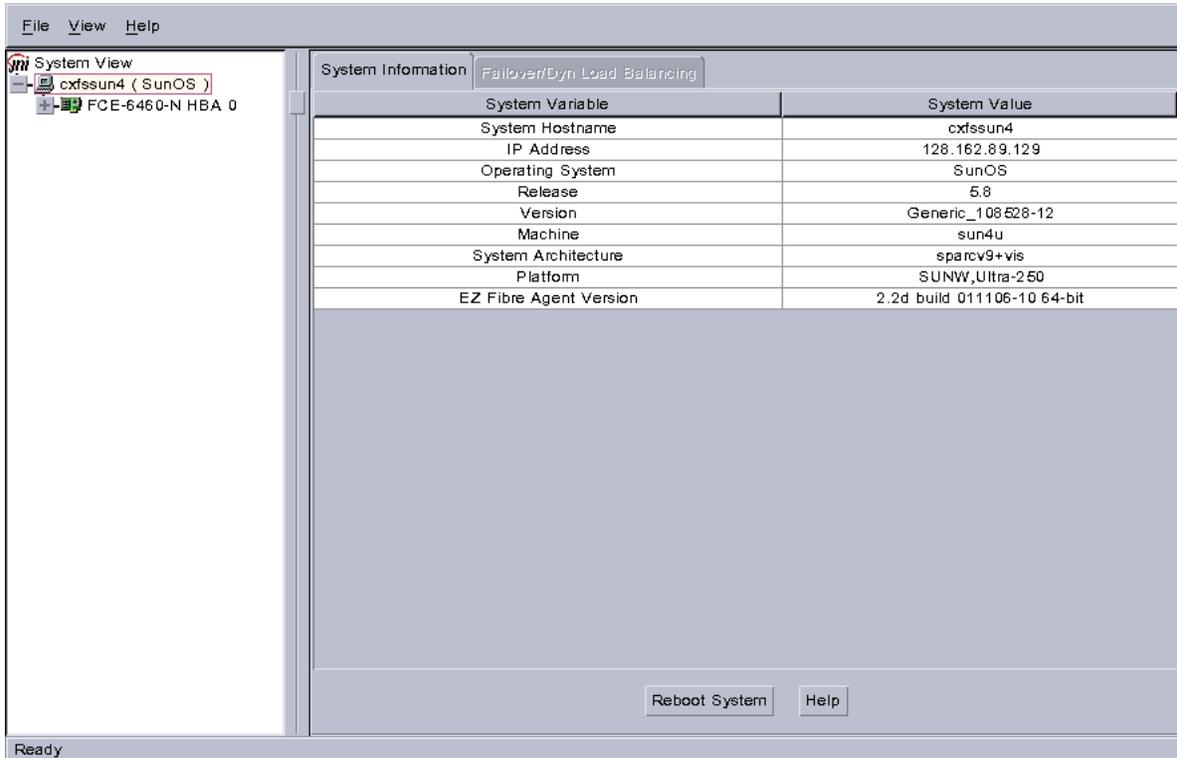
Figure 7-3 Example: After Clicking + to Display the HBA

2. Click on the icon to the right (not the + sign to the left). Figure 7-4 shows the icon.

Click here to display
Adapter Parameters
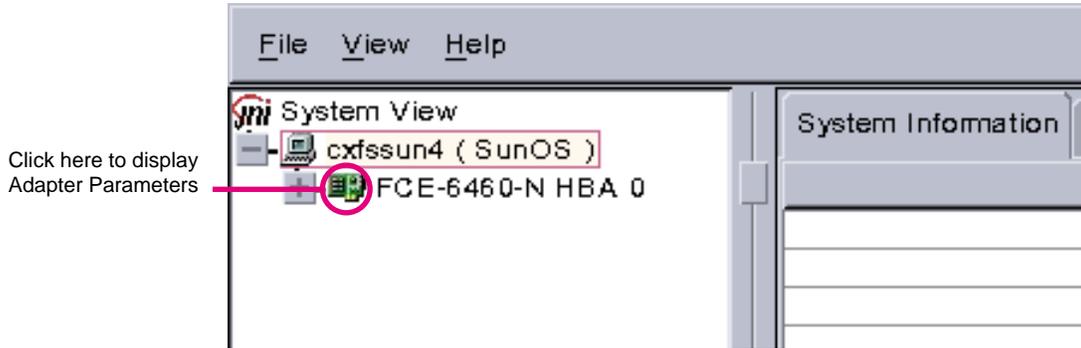
Figure 7-4 Location of the Icon to Display the Adapter Parameters

The right-hand pane will change to show **Adapter Parameters** for the selected HBA, as shown in Figure 7-5.
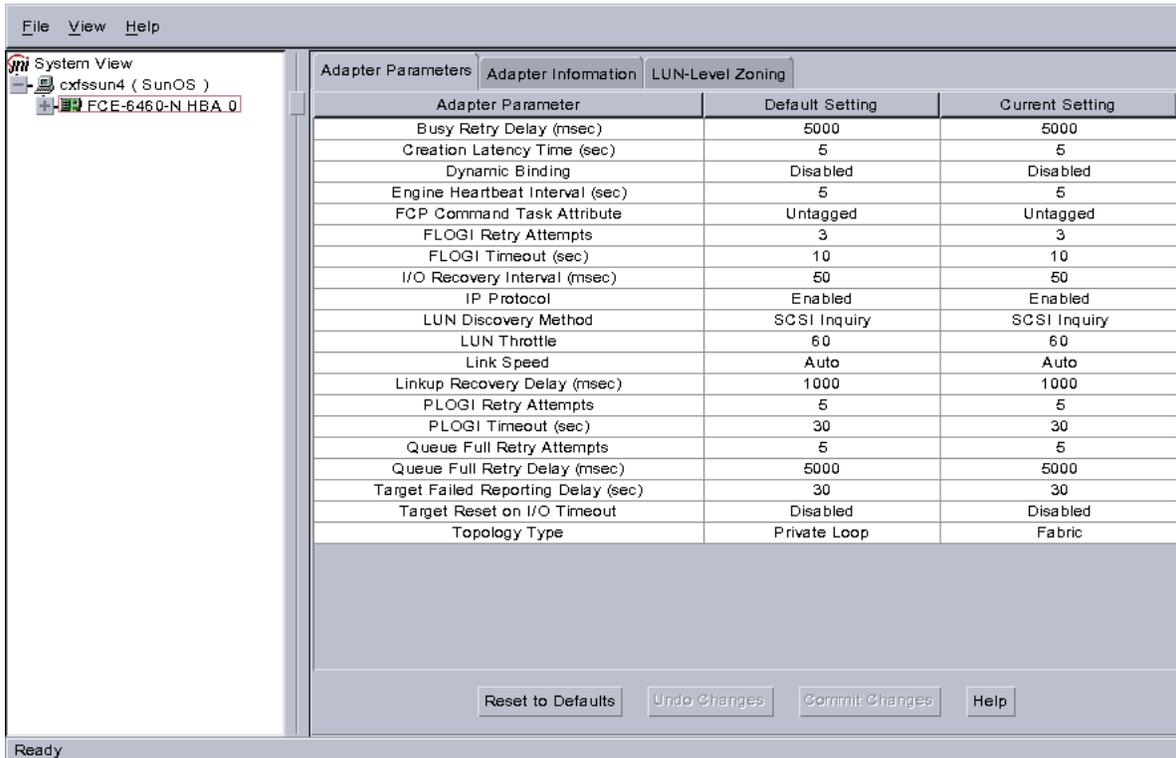
Figure 7-5 Example: After Clicking the HBA Icon to Show the Adapter Parameters

a. Click on the **Adapter Information** tab to see the information in Figure 7-6.

The last two lines show the WWNN and WWPN of the JNI HBA. These may be used in the the `/etc/fencing.conf` file, if automatic detection does not work; see "No HBA WWPNs are Detected" on page 270.
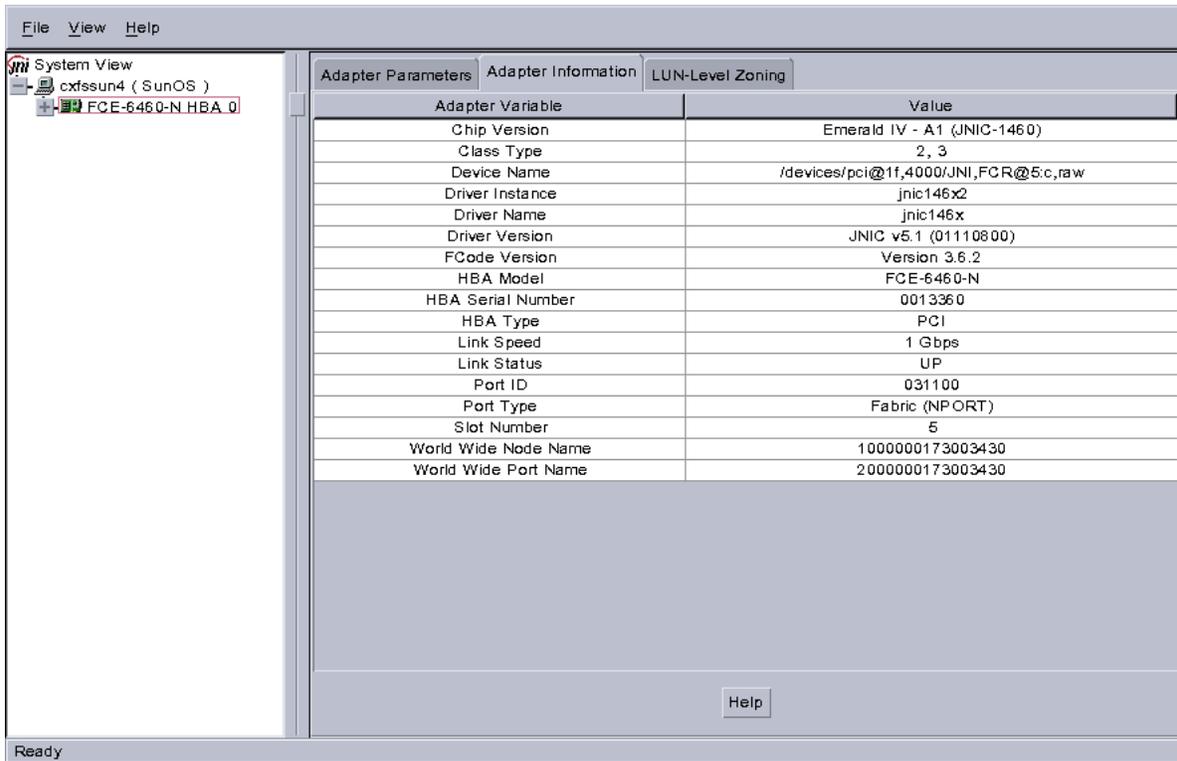
Figure 7-6 After Clicking the Adapter Information Tab

    b.   Click on the **LUN-Level Zoning** tab in the left-hand pane to display a list of all the known devices on the selected HBA, as shown in Figure 7-7.
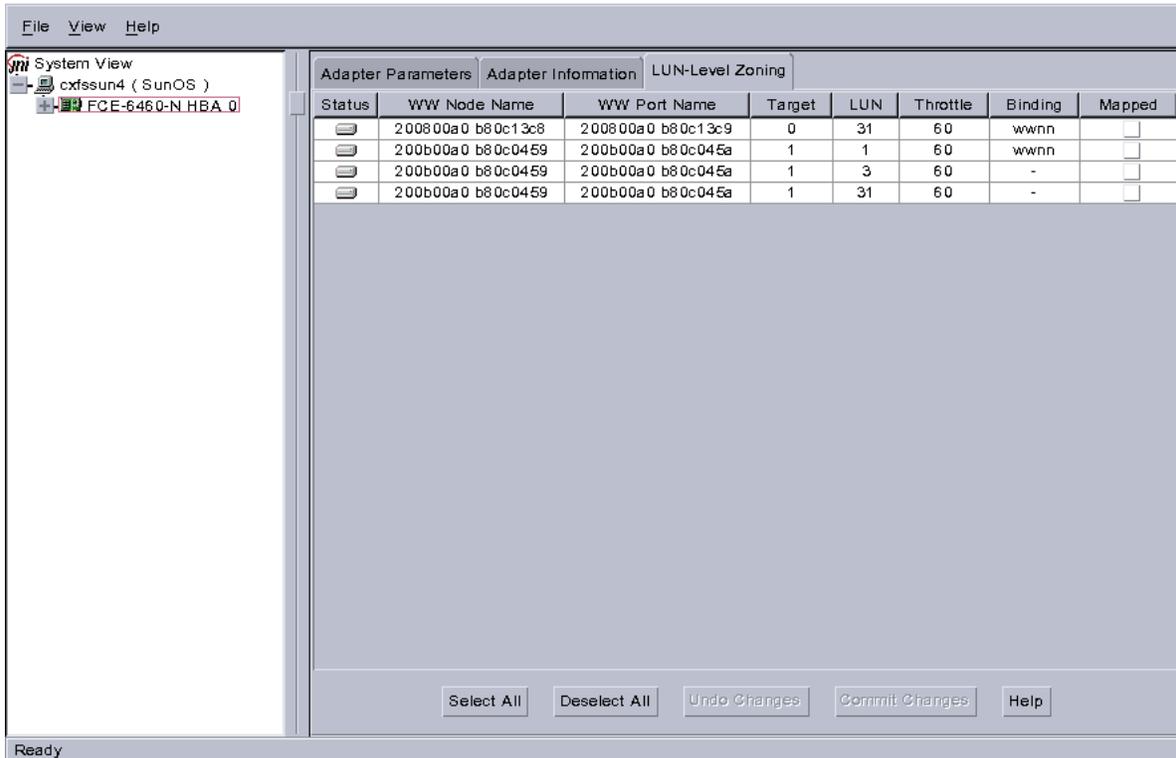
Figure 7-7 After Clicking on LUN-Level Zoning

3. Select the devices that should be accessed through the HBA.

   For each device you want to access, click on the corresponding box in the
   Mapped column to make a check mark appear, as shown in Figure 7-8. After you
   have selected all the desired devices for the HBA, click on **Commit Changes**. The
   LUNs you map will depend upon your own site's needs.

   ⚠ **Caution:** In this example, LUN 31 is used for administration by the TP9*x*00. This
   LUN must not be used for other purposes; do not map it or use it for XVM
   volumes.
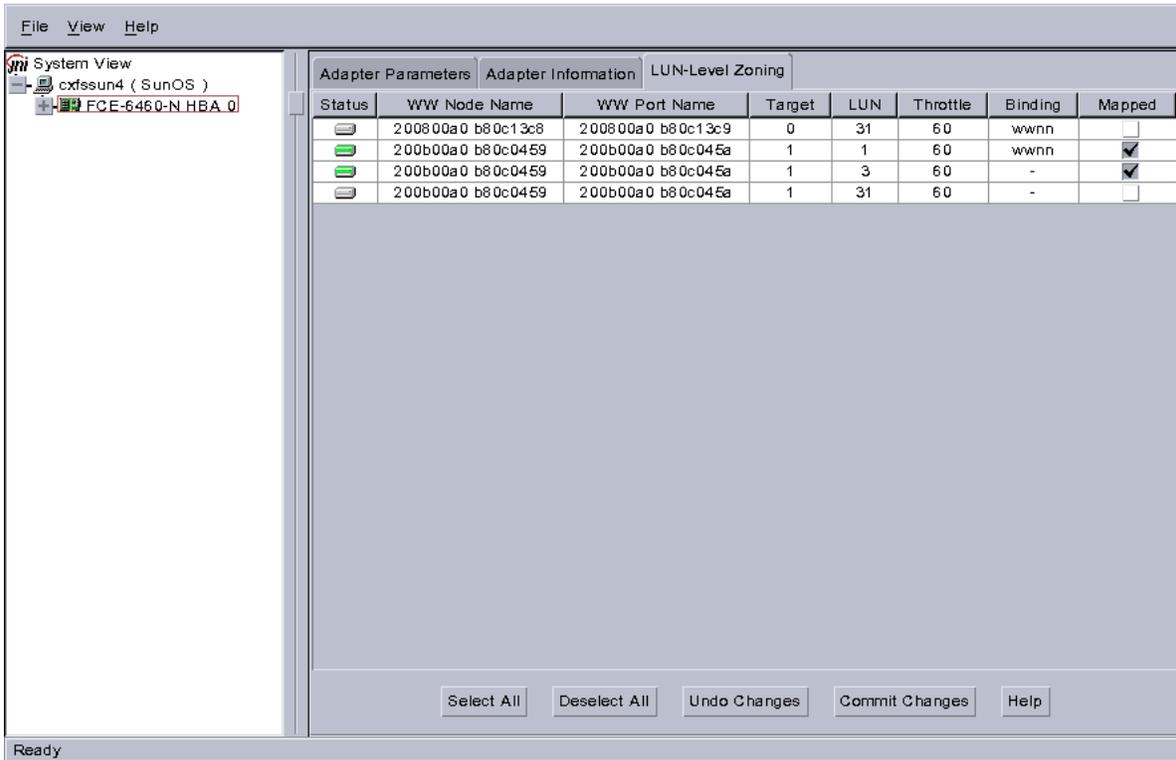
Figure 7-8 Example: After Mapping the LUNs and Committing the Changes

4. Reboot the system to make the changes take effect:

   ```
   solaris# init 6
   ```

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.lsilogic.com/downloads/selectDownload.do

   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 154. 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

c4t1d1: configured with capacity of 133.99GB
c4t1d3: configured with capacity of 133.99GB


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /pci@1f,4000/scsi@3/sd@0,0
       1. c4t1d1 <SGI-TP9400-0401 cyl 65533 alt 2 hd 64 sec 67>
          /pci@1f,4000/JNI,FCR@5/sd@1,1
       2. c4t1d3 <GI-TP9400-0401 cyl 65533 alt 2 hd 64 sec 67>
          /pci@1f,4000/JNI,FCR@5/sd@1,3
Specify disk (enter its number):
```

In this example, disks 1 and 2 are being addressed by the JNI driver, as indicated by the presence of JNI,FCR in the pathname.

The system log and console display may display warning messages because the disks have IRIX labels on them. For example:

```
Mar  5 14:17:33 cxfssun4 scsi: WARNING: /pci@1f,4000/JNI,FCR@5/sd@1,1 (sd154):
Mar  5 14:17:33 cxfssun4         corrupt label - wrong magic number
Mar  5 14:17:33 cxfssun4 scsi:     Vendor 'SGI', product 'TP9400', 284203008 512 byte blocks
Mar  5 14:17:33 cxfssun4 scsi: WARNING: /pci@1f,4000/JNI,FCR@5/sd@1,3 (sd155):
Mar  5 14:17:33 cxfssun4         corrupt label - wrong magic number
Mar  5 14:17:33 cxfssun4 scsi:     Vendor 'SGI', product 'TP9400', 284203008 512 byte blocks
```

This situation will be corrected automatically by CXFS after it is installed.

Note: You should not be alarmed by the preceding messages, nor should you try to relabel the disks with the format command. At this point, you are only trying to achieve connectivity to the devices, and the content is not important.

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 173.

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 155

- "Verifying the Private and Public Networks for Solaris" on page 160

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
```

The Solaris `/etc/nodename` file is analogous to the IRIX `/etc/sys_id` file.

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: `irix0`, `solaris1`, and `solaris2`. The `/.rhosts` files could be as follows (the prompt denotes the node name):

   ```
   irix0# cat /.rhosts
   solaris1 root
   solaris1-priv root
   solaris2 root
   solaris2-priv root

   solaris1# cat /.rhosts
   irix0 root
   irix0-priv root
   solaris2 root
   solaris2-priv root

   solaris2# cat /.rhosts
   irix0 root
   irix0-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. You can use the information in this section to verify the installation.

Solaris Installation Overview

Installing the CXFS client CD 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 release notes to learn about any late-breaking changes in the installation procedure.

2. Verify that the node has been upgraded to Solaris 9 (also known as *SunOS 5.9*) or 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 5.9 or 5.10.

3. Insert the *CXFS MultiOS Client 4.1* CD.

4. Read the already inserted CD as follows:

   - Solaris 10:

     ```
     solaris# pkgadd -d /cdrom/cdrom01/solaris/SGIcxfs-sol10.pkg
     ```

   - Solaris 9

     ```
     solaris# pkgadd -d /cdrom/cdrom01/solaris/SGIcxfs-sol9.pkg
     ```

For example, installing `SGIcxfs-sol10.pkg` under Solaris 10 will display at least the following output, although the exact version numbers may differ:

```
solaris# pkgadd -d /cdrom/cdrom01/solaris/SGIcxfs-sol10.pkg
The following packages are available:
  1 SGIcxfs      SGI CXFS client software
                 (sparc) release X.X

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

Processing package instance <SGIcxfs> from </cdrom/solaris/SGIcxfs-sol10.pkg>
. . .
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>
...
```

5. *(For client-side licensing only)* Use the `cxfslicense -d` command to verify that the existing client-side license key is installed in `/etc/flexlm/license.dat`.

Note: If you are using server-side licensing, you should skip this step. If you execute this step on a node that uses server-side licensing at this point in the install procedure (before rebooting), you will get an error that can be ignored.

For example, the following shows a properly installed client-side license key:

```
solaris# /usr/cxfs_cluster/bin/cxfslicense -d
No license required for XVM user feature.
No license required for XVM mirror feature.
CXFS_SOLARIS license granted.
```

If you are using client-side licensing and there are errors, verify that you have correctly installed the CXFS license keys; see "Reinstalling An Existing Client-Side License Key for Windows " on page 236. For more information, see "Verifying the Linux Installation" on page 67.

6. If CXFS previously failed to load because of a license key error, restart CXFS now.

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:

```
% pkginfo -l SGIcxfs
   PKGINST:  SGIcxfs
      NAME:  SGI CXFS MultiOS client software
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  release 2.4
   BASEDIR:  /
    VENDOR:  Silicon Graphics Inc.
```

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 AMCC JNI HBAs for Solaris nodes that are connected to a switch that is configured in the cluster database. These HBAs are available for fencing.

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

If no WWPNs are detected, there will be messages about loading the HBA/SNIA library logged to the `/var/log/cxfs_client` file.

For LSI and QLogic HBAs, or if no WWPNs are detected, you can manually specify the WWPNs in the fencing file.

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.

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. To determine the WWNs, use the following command:

```
fcinfo hba-port
```

For example:

```
cxfssun # fcinfo hba-port
HBA Port WWN: 2000000173007392
        OS Device Name: /devices/pci@4,2000/JNI,FCR@1:a,raw
        Manufacturer: JNI Corporation
        Model: FCE-6460-N
        Type: unknown
        State: link down
        Supported Speeds: not established
        Current Speed: not established
        Node WWN: 1000000173007392
HBA Port WWN: 20000001730081f9
        OS Device Name: /devices/pci@6,2000/JNI,FCR@1:b,raw
        Manufacturer: JNI Corporation
        Model: FCX-6562-N
        Type: N-port
        State: online
        Supported Speeds: 1Gb
        Current Speed: 1Gb
        Node WWN: 10000001730081f9
HBA Port WWN: 210000e08b19233e
        OS Device Name: /dev/cfg/c2
        Manufacturer: QLogic Corp.
        Model: 375-3102-xx
```

```
                 Type: N-port
                 State: online
                 Supported Speeds: 1Gb 2Gb
                 Current Speed: 1Gb
                 Node WWN: 200000e08b19233e
```

5. If the `fchba-info` command is not available, 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).

6. 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
```

7. To configure fencing, see the *CXFS 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:

- "Upgrading the CXFS Software for Solaris" on page 167
- "Modifying the CXFS Software for Solaris" on page 167
- "Recognizing Storage Changes for Solaris" on page 168

Upgrading 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. Follow the installation instructions to install the new package. See "Client Software Installation for Solaris" on page 161.

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

If you make changes to your storage configuration, you must rerun the HBA utilities to reprobe the storage. See "Discovering and Configuring LUNs" on page 145 and "HBA Installation for Solaris" on page 141.

GRIO on Solaris

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

For more information, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 10 and the *Guaranteed-Rate I/O Version 2 Guide*.

XVM Failover V2 on Solaris

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

```
/devices/pci@1f,2000/JNI,FCR@1/sd@4,2 affinity=0
/devices/pci@1f,2000/JNI,FCR@1/sd@5,2 affinity=0 preferred
/devices/pci@1f,2000/JNI,FCR@1/sd@8,2 affinity=1
/devices/pci@1f,2000/JNI,FCR@1/sd@6,2 affinity=1
```

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

Mapping XVM Volumes to Storage Targets on Solaris

You can map XVM volumes to devices and to targets on RAID controllers using the output from the xvm command, the EZ Fibre utility, and the device entries in the filesystem.

You can use the xvm command to display the device names. 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)
```

These devices are created in the /devices directory (note the *:c,raw):

```
solaris# ls /devices/pci@9,600000/JNI,FCR@2,1/sd@2,0*:c,raw
/devices/pci@9,600000/JNI,FCR@2,1/sd@2,0:c,raw
```

These devices are linked from the disk devices under /dev/rdsk:

```
solaris# ls -l /dev/rdsk/c*s2 /dev/dsk/c*s2 | grep
'pci@9,600000/JNI,FCR@2,1/sd@2,0'
lrwxrwxrwx   1 root     root            51 May 14  2004 /dev/rdsk/c2t2d0s2
-> ../../devices/pci@9,600000/JNI,FCR@2,1/sd@2,0:c,raw
```

Using the EZ Fibre utility, it is possible to find the LUN and therefore the target WWN by exploring the LUNs to find the matching /dev/rdsk entry. See "Installing the JNI AMCC EZ Fibre Configuration Utility" on page 143.

For more information, see "Verifying Access to XVM Volumes" on page 261 and the *XVM Volume Manager Administrator's Guide*.

Troubleshooting for Solaris

This section contains the following:

- "Verifying Existing Client-Side License Keys for Solaris" on page 170
- "Reinstalling An Existing Client-Side License Key for Solaris" on page 171
- "The `cxfs_client` Daemon is Not Started on Solaris" on page 171
- "Filesystems Do Not Mount on Solaris" on page 172
- "New Storage is Not Recognized on Solaris" on page 173
- "Large Log Files on Solaris" on page 173
- "Changing the CXFS Heartbeat Value on Solaris" on page 173
- "Error Reading Kernel XVM Cluster Mirror Status" on page 174

For general troubleshooting information, see Chapter 10, "General Troubleshooting" on page 263 and Appendix D, "Error Messages" on page 289.

Verifying Existing Client-Side License Keys for Solaris

> **Note:** Server-side licensing is preferred. Solaris 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. See "License Keys" on page 9.

To verify that the existing client-side license keys have been installed properly when using client-side licensing, use the cxfslicense -d command. For example:

- Client-side license key installed on a Solaris node:

```
solaris# /usr/cxfs_cluster/bin/cxfslicense -d
No license required for XVM user feature.
No license required for XVM mirror feature.
CXFS_SOLARIS license granted.
```

- No client-side license key on a Solaris node:

```
solaris# /usr/cxfs_cluster/bin/cxfslicense -d
No license required for XVM user feature.
No license required for XVM mirror feature
Error on CXFS_SOLARIS license check out.
checkout failed: Invalid license file syntax
Feature:       CXFS_SOLARIS
License path:  /etc/flexlm/license.dat
FLEXlm error:  -2,413
For further information, refer to the FLEXlm End User Manual,
available at "www.macrovision.com".

No client-side license, will request a server-side license.
```

  The above output indicates that no client-side license key is installed and the CXFS software will request a server-side license key.

  If you are using client-side licensing and there are errors, verify that you have obtained and installed the CXFS license keys.

Reinstalling An Existing Client-Side License Key for Solaris

An existing Solaris client-side license key is installed in the following location:

`/etc/flexlm/license.dat`

Do the following:

1. Create the license key directory if necessary. For example:

   solaris# **mkdir -p /etc/flexlm**

2. Copy the key to the `license.dat` file.

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**

The `cxfs_client` daemon might not start if the license key has expired, which is indicated when the following message is output to the `/var/log/cxfs_client` file:

```
CXFS not properly licensed for this host
```

In this case, execute the following command to determine why the license key check fails:

```
linux# /usr/cxfs_cluster/bin/cxfslicense -d
```

If the license key is valid, check the `cxfs_client` log file for other errors.

Restart `cxfs_client` as described in "Start/Stop `cxfs_client` for Linux" on page 69 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 client 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? Verify that the LUNs used by this filesystem have been mapped in the EZ Fibre configuration utility. See "Installing the JNI AMCC EZ Fibre Configuration Utility" on page 143.

    See "New Storage is Not Recognized on Solaris" on page 173.

- The `cxfs_client` daemon may not be running. See "The `cxfs_client` Daemon is Not Started on Solaris" on page 171.

- 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 administration node with the correct hostname?

- Has the node been added to the cluster and enabled? See "Verifying the Cluster Status" on page 256.

– 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 168.

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 CXFS administration nodes, see the *CXFS 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
```

Error Reading Kernel XVM Cluster Mirror Status

The following message will be output by the cxfslicense -d command if you execute it before rebooting the system:

```
error reading kernel XVM cluster mirror status. Check if XVM module is
started.
```

After you reboot the system and therefore load the XVM module, this message will no longer appear when you run cxfslicense -d. You can ignore this message if you have not rebooted.

Reporting Solaris Problems

When reporting a problem about a CXFS Solaris node to SGI, you should retain the following information:

- 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 installed CXFS packages. Use the `pkginfo` command as follows:

  # **pkginfo -l SGIcxfs**

- 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 about the cluster obtained from the `cxfsdump` utility run on a CXFS administration node. When run in local mode on a Solaris node, it stores information in `/var/cluster/cxfsdump-data/`*nodename*`.tar.gz`.

- 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 2000, Windows Server 2003, or Windows XP 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 Windows XP or Windows Server 2003 **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" on page 178
- "CXFS Commands on Windows" on page 180
- "Log Files and Cluster Status for Windows" on page 180
- "Functional Limitations for Windows" on page 185
- "Performance Considerations for Windows" on page 188
- "Access Controls for Windows" on page 190
- "System Tunables for Windows" on page 201

Requirements for Windows

In addition to the items listed in "Requirements" on page 7, CXFS requires at least the following:

- An Intel Pentium or compatible processor
- Minimum RAM requirements (more will improve performance):
    - Windows 2000: 128 MB
    - Windows XP, Windows Server 2003: 256 MB
- A minimum of 10 MB of free disk space
- Host bus adapter (HBA):
    - LSI Logic LSI7204XP-LC dual-port 4Gb PCI-X
    - A QLogic QLA2200, QLA2310, QLA2342, or QLA2344
- The following LSI Logic software from the http://www.lsilogic.com website:
    - Windows 2000 and Windows 2003: 1.21.03
    - Windows XP: 1.21.04
    - MyStorage 3.00.00 revision

- The following QLogic software from the http://www.qlogic.com website:
  - QLA2200:
    - Windows 2000, Windows Server 2003: v8.1.5.15
    - Windows XP: v8.1.5.12
  - QLA2310, QLA2342 and QLA2344:
    - Windows 2000, Windows XP, Windows Server 2003: v9.1.2.11
  - SANsurfer Pro for Windows 2.0.30 Build 78

  You should install the documentation associated with the software. See the SANsurfer Pro README for the default password. Follow the QLogic instructions to install the driver, the SANsurfer NT Agent, and the SANsurfer Manager software.

- If two QLogic HBAs are installed for Window 2000 or 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.

- Windows versions:
  - Windows 2000 Service Pack 4
  - Windows XP Service Pack 1 or Service Pack 2
  - Windows Server 2003 Service Pack 1
  - Windows Server 2003 R2

  Note: CXFS will not work on 64-bit versions of Windows.

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\cxfslicense.exe
%ProgramFiles%\CXFS\grioadmin.exe
%ProgramFiles%\CXFS\griomon.exe
%ProgramFiles%\CXFS\grioqos.exe
```

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\cxfslicense.exe` assists with license validation for client-side licensing. See "Verifying Existing Client-Side License Keys for Windows" on page 235.

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 256.

For information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 10 and "GRIO on Windows" on page 233.

Log Files and Cluster Status for Windows

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 CXFS administration nodes, see the *CXFS 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 8-6 on page 217) and may be adjusted by following the steps described in "Modifying the CXFS Software for Windows" on page 229.

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

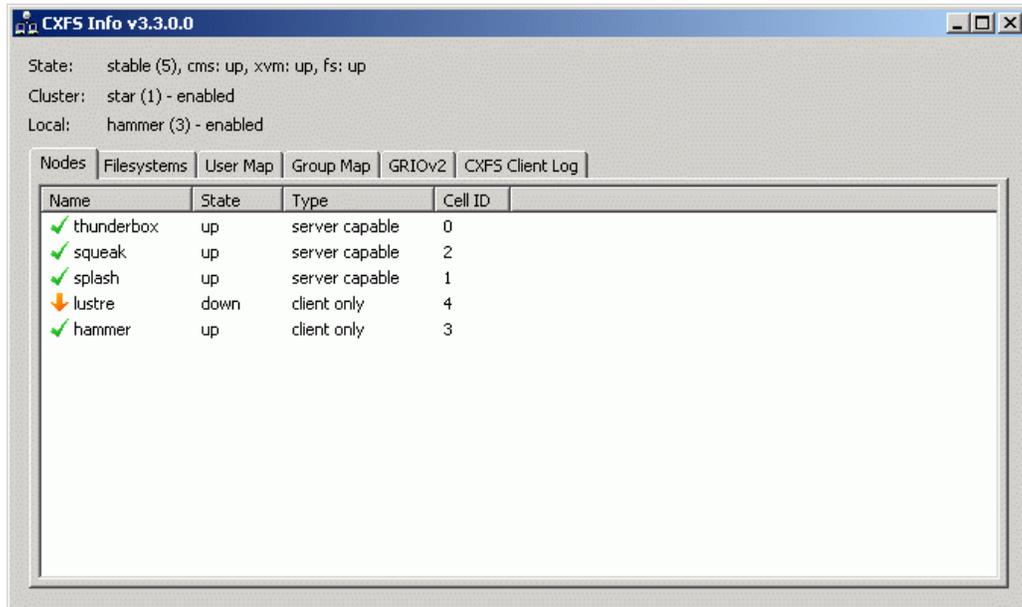Figure 8-1 shows an example of the **CXFS Info** window.

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 256.

- **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 233.

- **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
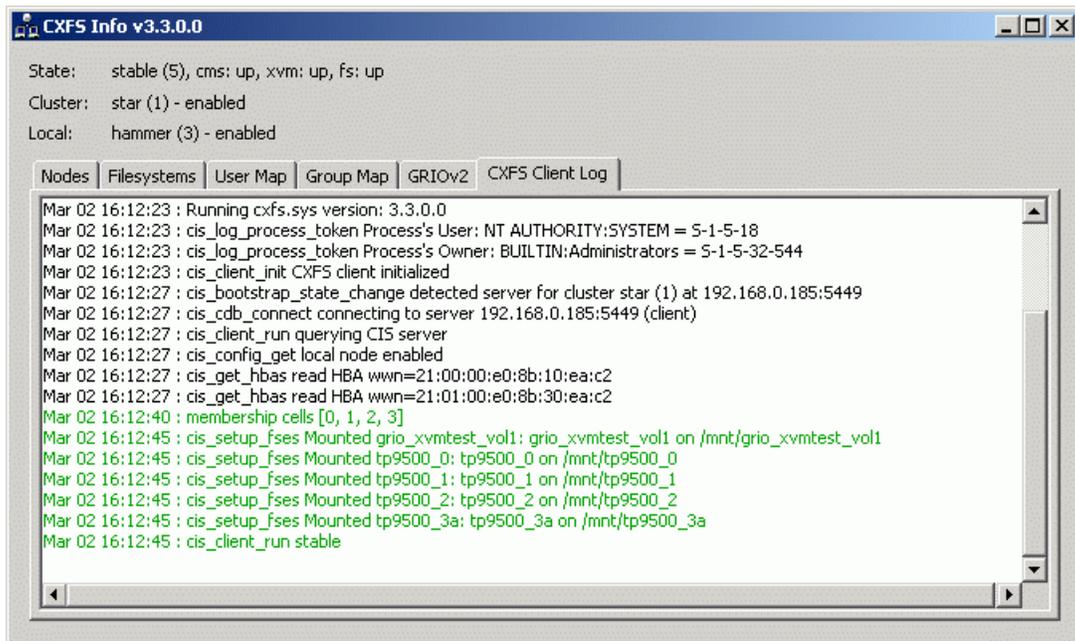
Figure 8-3 shows an example.

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 client in the cluster:

• Green indicates that the client 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 8-14 on page 234.

Functional Limitations for Windows

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

- "XVM Failover V2 Is Not Supported" on page 185

- "UNIX Perspective of CXFS for Windows" on page 185

- "Windows Perspective of CXFS for Windows" on page 186

- "Forced Unmount on Windows" on page 187

- "Define LUN 0 on All Storage Devices for Windows" on page 187

- "Memory-Mapping Large Files for Windows" on page 188

- "CXFS Mount Scripts for Windows" on page 188

- "Norton Ghost Prevents Mounting Filesystems" on page 188

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 279.

XVM Failover V2 Is Not Supported

Windows does not support XVM failover v2. However, it does support a mechanism to configure the automatic failover to another HBA port if the configured port is no longer able to see the target. See "Configuring HBA Failover for Windows 2000 and Windows 2003" on page 210.

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 CXFS 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 244.

- 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.

- You can move, rename, or delete a symbolic link; however, you cannot copy a symbolic link. Copying a valid 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.

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.

- 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" on page 19 and "Forced Unmount of CXFS Filesystems" on page 254.

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

Windows, and therefore CXFS, may 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. This problem has occurred on Windows XP CXFS clients. For more information about this issue, see:

http://support.microsoft.com/default.aspx?scid=kb;en-us;821666

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.

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

    SGI recommends that you use Samba on the CXFS 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 CXFS Windows client should perform well when their I/O access patterns are similar to those described in "When to Use CXFS" on page 2.

- 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" on page 190
- "User Identification Mapping Methods for Windows" on page 191
- "Enforcing Access to Files and Directories for Windows" on page 193
- "Viewing and Changing File Attributes with Windows Explorer" on page 194
- "Viewing and Changing File Permissions with Windows Explorer" on page 194
- "Viewing and Changing File Access Control Lists (ACLs) for Windows" on page 197
- "Effective Access for Windows" on page 198
- "Restrictions with file ACLs for Windows" on page 198
- "Inheritance and Default ACLs for Windows" on page 199

User Identification for Windows

The CXFS software supports several user identification mechanisms, which are described in "User Identification Mapping Methods for Windows" on page 191. 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 241.

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

– Windows 2000 Server or 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 8-6 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 229.

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 IRIX 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 IRIX metadata server:

```
irix# 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 IRIX ACL mask. See the `chacl`(1) man page for an explanation of IRIX 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 IRIX 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 215 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:

```
irix% ls -lda .
drwxr-xr-x   3 fred     video             41 Nov 21 18:01 ./

irix% umask
22

irix% touch file3
irix% ls -l file3
-rw-r--r--   1 fred     video              0 Nov 21 18:23 file3
```

For more information, see the umask man page.

Creating a file in Windows Explorer on a Windows node will have the same result.

An IRIX 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, an IRIX 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`:

```
irix% 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
irix% touch dir1/newfile
irix% mkdir dir1/newdir
irix% 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 an IRIX ACL cannot differentiate between the two.

System Tunables for Windows

This section discusses the following topics:

- "Registry Modification" on page 201

- "Default Umask for Windows" on page 202

- "Maximum DMA Size for Windows" on page 202

- "Memory-Mapping Coherency for Windows" on page 203

- "DNLC Size for Windows" on page 203

- "Mandatory Locks for Windows" on page 204

- "User Identification Map Updates for Windows" on page 205

- "I/O Size Issues Within the QLogic HBA" on page 206

- "Command Tag Queueing (CTQ) Used by the QLogic HBA" on page 206

Note: These system tunables are removed when the software is removed. They may need to be reset when downgrading the CXFS for Windows software.

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 `Regedit.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 and edit 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 client 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 and edit 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 client 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 and edit 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 client.

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

> **HKEY_LOCAL_MACHINE**
>> **> SYSTEM**
>>> **> CurrentControlSet**
>>>> **> Services**
>>>>> **> CXFS_Client**
>>>>>> **> Parameters**

In the `regedit` menu:

> **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.

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

> **HKEY_LOCAL_MACHINE**
> > **> SYSTEM**
> > > **> CurrentControlSet**
> > > > **> ql2xxx**
> > > > > **> Parameters**
> > > > > > **> Device**

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 IRIX 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 (qlogic 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.

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" on page 208

- "Configuring Multiple HBAs for Load Balancing on Windows" on page 208

- "Configuring HBA Failover for Windows 2000 and Windows 2003" on page 210

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. Verify the driver parameters for the QLogic HBAs. Run **regedit** and go to the following key:

   HKEY_LOCAL_MACHINE
      > SYSTEM
         > CurrentControlSet
            > Services
               > ql2xxx
                  > Parameters
                     > Device

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

   ```
   Buschange=0;FixupInquriry=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, cxfs_admin, or cmgr to perform these tasks, see *CXFS Administration Guide for SGI InfiniteStorage*.

Configuring HBA Failover for Windows 2000 and Windows 2003

The QLogic HBA on Windows 2000 and 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 client still operates normally.

2. Perform the procedure in "Configuring Multiple HBAs for Load Balancing on Windows" on page 208. 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" on page 211

- "Verifying the Private and Public Networks for Windows" on page 213

- "Configuring the Windows XP SP2 Firewall for Windows" on page 214

Adding a Private Network for Windows

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

Procedure to Add a Private Network for Windows

The following procedure provides an overview of the steps required to add a private network to the Windows node. You may skip some steps, depending upon the starting conditions at your site.

Do the following:

1. Install the second network adapter in the Windows node as per the network adapter vendor instructions. In some cases you must remove all network setups, restart, and then add network services to each network adapter from scratch.

2. Ensure that the node recognizes two network adapters in the system. Select the following:

   Start
   > **Settings**
   > **Network and Dial-up Connections**

3. Specify the private network settings (IP address, subnet mask, default gateway) on one of the network adapters. Select the following:

Start
    > Settings
        > Network and Dial-up Connections

Then right-mouse click the private network adapter and click **Properties**.
Uncheck all check boxes except **Internet Protocol (TCP/IP)**, thereby enabling only **Internet Protocol (TCP/IP)**, as shown in Figure 8-4.
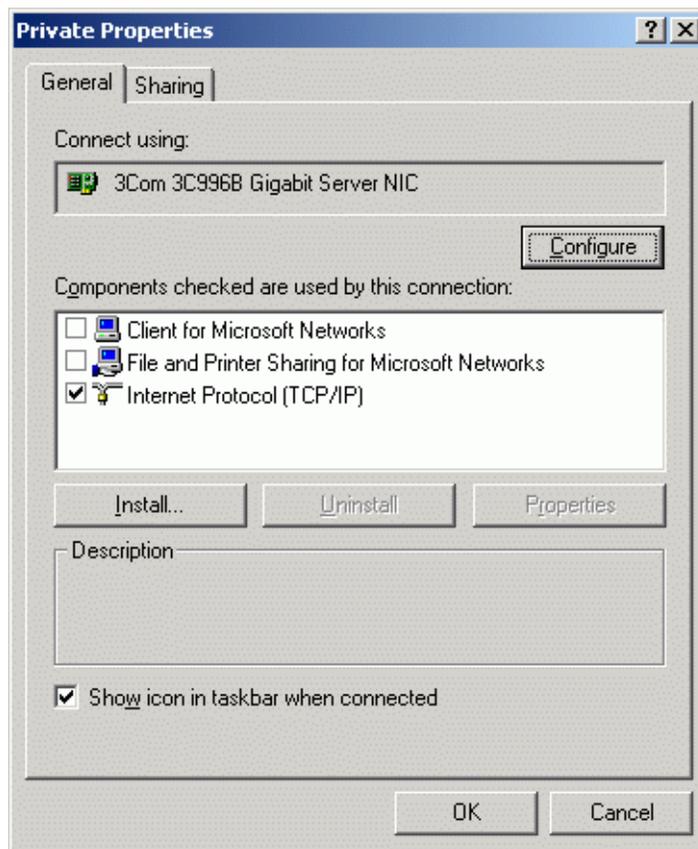
Figure 8-4 Private Properties: Selecting only TCP/IP

4. Select `Internet Protocol (TCP/IP)` and then click **Properties**. Specify the static IP address and DNS server. The private network IP address must be a fixed address and cannot be configured by DHCP.

The location of the host information is:

`%SystemRoot%\system32\drivers\etc\hosts`

Ensuring Proper Hostname Configuration for Windows

It is very important, especially in the presence of private network failover, to ensure that the hostnames and IP addresses for the various network interfaces are properly configured on both the Windows node and the CXFS metadata servers. (For more information about configuring private network failover, see the *CXFS Administration Guide for SGI InfiniteStorage.*)

For example, problems may occur if the cluster is configured using hostnames and the primary network interface on the Windows node is used for the CXFS private network. In this situation, the Windows node may associate the computer name to the primary network interface rather than the private network name configured in DNS.

To avoid such problems, SGI recommends that you specify the private network addresses for the Windows node using IP addresses rather than hostnames.

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 release notes for the Windows platform to learn about any late-breaking changes in the installation procedure.

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**

4. Insert the *CXFS MultiOS Client 4.1* CD into the Windows host. Normally, the setup program will automatically run, otherwise run the following program from the CD:

   CD_drive:`Windows/setup.exe`

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

6. Install the CXFS software, as shown in Figure 8-5. If the software is to be installed in a nondefault directory, click **Browse** to select another directory. Click **Next** when finished.
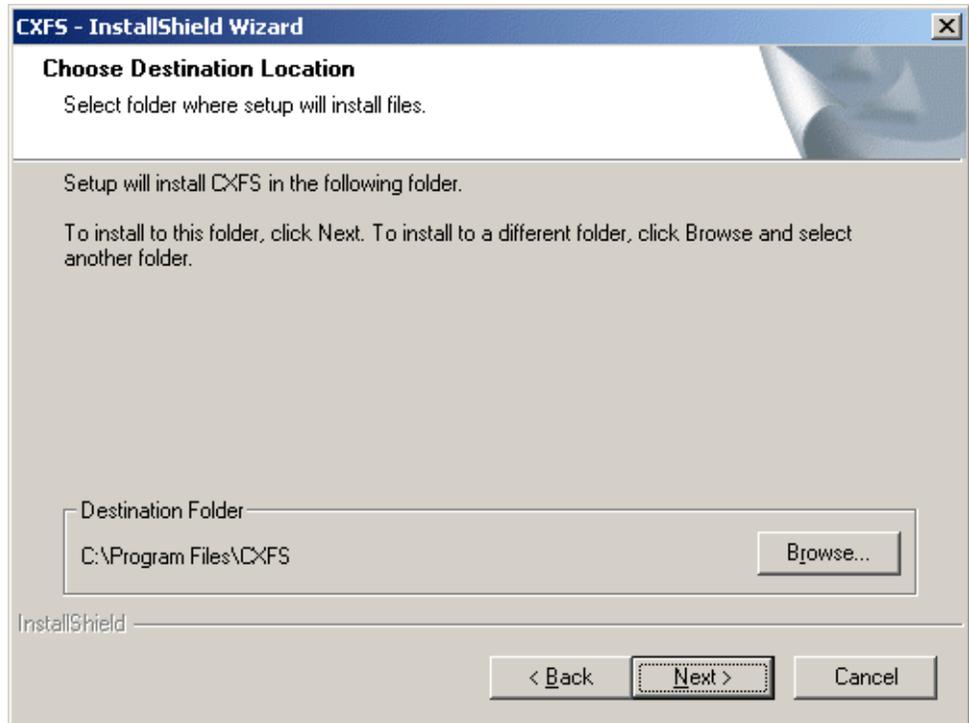
Figure 8-5 Choose Destination Location

7. Enter details for the following fields as shown in Figure 8-6 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 191.

- **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`.
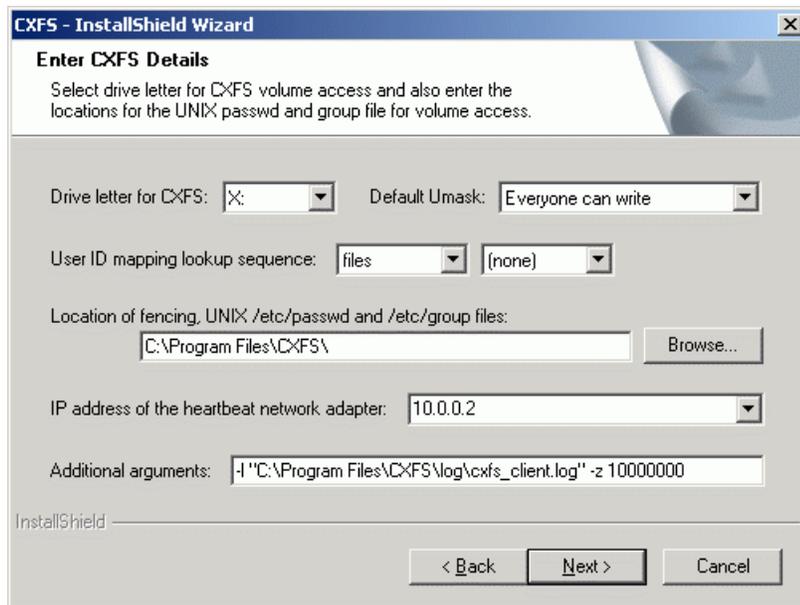
Figure 8-6 Enter CXFS Details

8. If you select **ldap_activedir** as the user ID mapping method, the dialog in Figure 8-7 is displayed after you click **Next**.
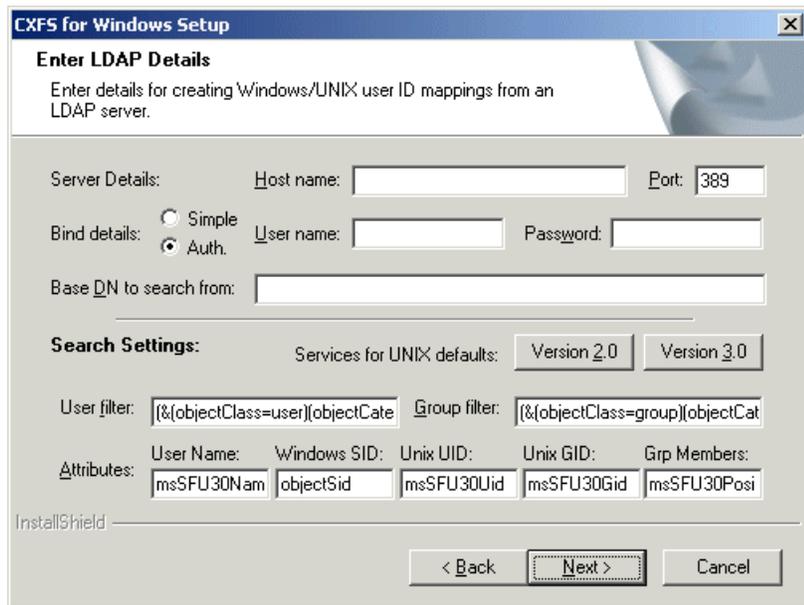
Figure 8-7 Active Directory Details

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.

9. If you select **ldap_generic** as the user ID mapping method, the dialog in Figure 8-8 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 8-8 Generic LDAP Details

10. Review the settings, as shown in Figure 8-9. If they appear as you intended, click **Next**. If you need to make corrections, click **Back**.
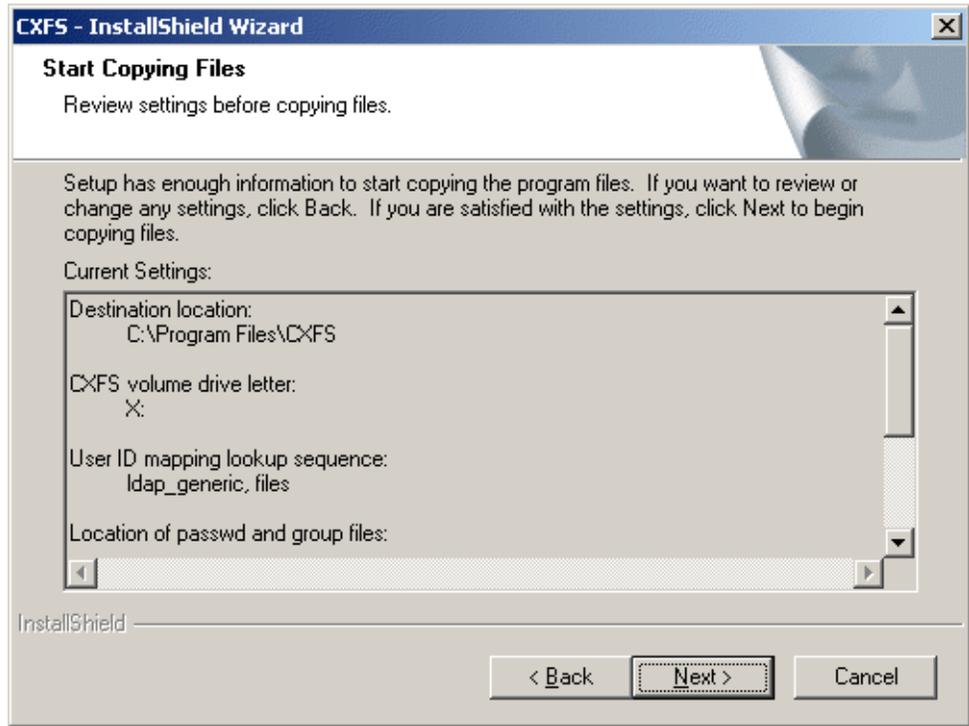
Figure 8-9 Review the Settings

After you click **Next**, the CXFS software will be installed.

11. You will be given the option to start the driver at system start-up, as shown in Figure 8-10. 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.
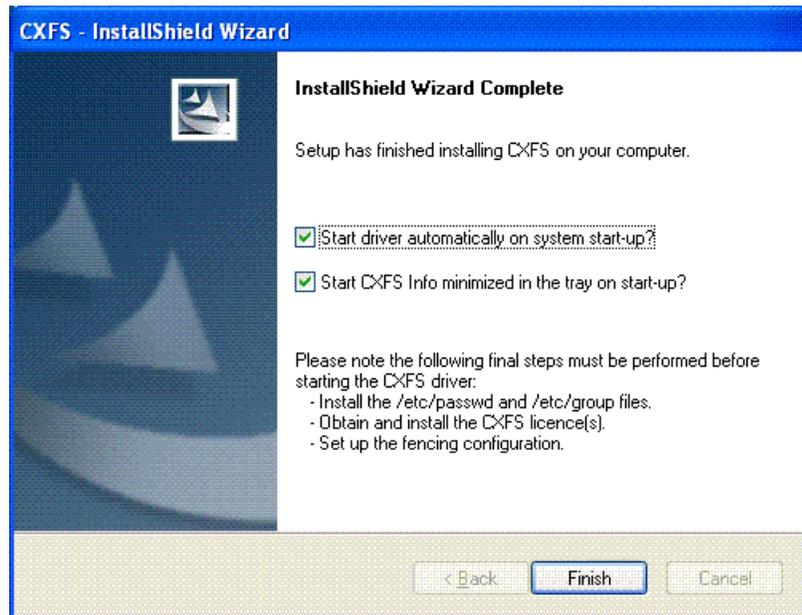
Figure 8-10 Start CXFS Driver

12. 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 8-11. (CXFS will not run until a restart has occurred.)
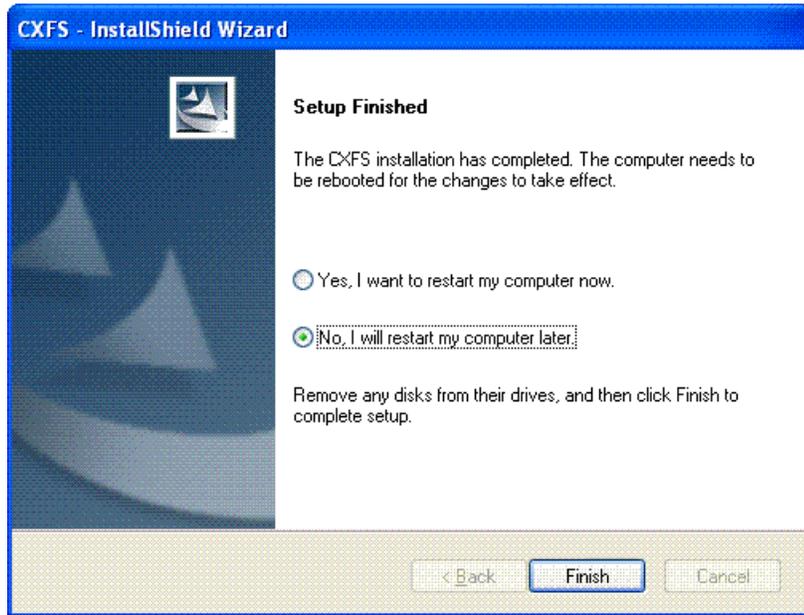
Figure 8-11 Restart the System

13. *(For client-side licensing only)* Use the `cxfslicense` command to verify that the existing client-side CXFS license key is installed in `%ProgramFiles%\CXFS\lib\license.dat`.

Note: If you are using server-side licensing, you should skip this step. If you execute this step on a node that uses server-side licensing at this point in the install procedure (before rebooting), you will get an error that can be ignored.

If there are errors, verify that the client-side license is installed correctly; see "Reinstalling An Existing Client-Side License Key for Solaris" on page 171.

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 CXFS 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 190.

If only the `Administrator` user is mapped, see "CXFS Client Service Cannot Map Users other than `Administrator` for Windows" on page 241.

I/O Fencing for Windows

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.

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 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 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.

It is recommended 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" on page 229
- "Upgrading the CXFS Software for Windows" on page 230
- "Removing the CXFS Software for Windows" on page 231
- "Downgrading the CXFS Software for Windows" on page 232
- "Recognizing Storage Changes for Windows" on page 232

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 215, perform the following steps:

1. Select the following:

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

Figure 8-12 shows the screen that lets you modify the software.

Figure 8-12 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.

Upgrading the CXFS Software for Windows

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

1. Insert the CD containing the upgraded software to run the setup program. If the
    setup program does not automatically start, run *CD_drive*:Windows/Setup.exe
    from the CD.

2. A welcome screen will appear that displays the version you are upgrading from
    and the version you are upgrading to. Figure 8-13 shows the screen that appears

when you are upgrading the software. All the configuration options are available to update as discussed in "Client Software Installation for Windows" on page 215.
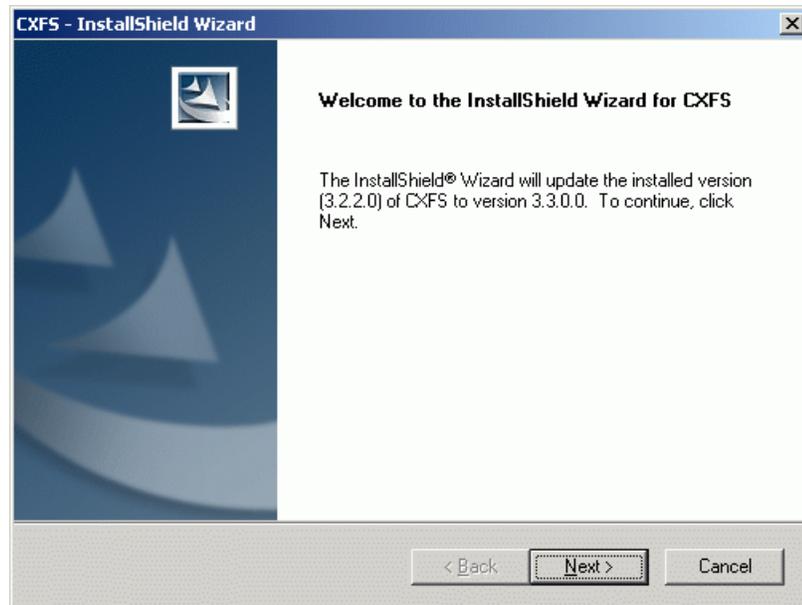
Figure 8-13 Upgrading the Windows Software

3. 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 8-12 on page 230 shows the screen that lets you remove the software.

Note: By default, the `passwd`, `group`, `license` 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 231 and then install the older version of the software as directed in "Client Software Installation for Windows" on page 215.

Note: The removal process may remove the configuration and client-side license key files. You should back up the configuration file and (if you are using client-side licensing) the license key file before removing the CXFS software so that you can easily restore them 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 207.

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

To verify the driver parameters for the QLogic HBAs, run `regedit` and go to the following key:

> **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;FixupInquriry=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 it. Do not delete any other parameters.

GRIO on Windows

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

Figure 8-14 shows the **CXFS Info** display for GRIO.

Figure 8-14 CXFS Info Display for GRIO for Windows

For more information, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 10 and the *Guaranteed-Rate I/O Version 2 Guide*.

Troubleshooting for Windows

This section contains the following common Windows problems:

- "Verifying Existing Client-Side License Keys for Windows" on page 235

- "Reinstalling An Existing Client-Side License Key for Windows " on page 236

- "Verifying that the CXFS Software is Running Correctly for Windows" on page 237

- "Unable to Mount Filesystems on Windows" on page 237

- "Access-Denied Error when Accessing Filesystem on Windows" on page 239

- "Application Works with NTFS but not CXFS for Windows" on page 239

- "Delayed-Write Error Dialog is Generated by the Windows Kernel" on page 240

- "CXFS Client Service Does Not Start on Windows" on page 241

- "HBA Problems" on page 241

- "CXFS Client Service Cannot Map Users other than `Administrator` for Windows" on page 241

- "Filesystems Are Not Displayed on Windows" on page 242

- "Large Log Files on Windows" on page 243

- "Windows Failure on Restart" on page 243

- "Memory Configuration for Windows" on page 244

- "Application Cannot Create File Under CXFS Drive Letter" on page 244

- "Error Reading Kernel XVM Cluster Mirror Status" on page 244

Also see:

- The Windows `cxfsdump` documentation located at `%ProgramFiles%\CXFS\cxfsdump.html`

- Chapter 10, "General Troubleshooting" on page 263

Verifying Existing Client-Side License Keys for Windows

Note: Server-side licensing is preferred. Windows 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. See "License Keys" on page 9.

Existing client-side license keys for CXFS are valid for all versions of Windows that support CXFS. Typically, the following feature names are supported for all versions of Windows:

- CXFS: `CXFS_NT`, `CXFS_W2K`, `CXFS_WXP`, and `CXFS_WNET`

- XVM Mirroring: `XVM_NT`, `XVM_W2K`, `XVM_WXP`, and `XVM_WNET`.

This allows a client running Windows 2000 to be upgraded to Windows XP or Windows Server 2003 without purchasing a new license key from SGI. (The hostid cannot be changed; the hostid in the license key must match the hostid on the Windows node.)

To verify that the existing client-side license keys have been installed properly when using client-side licensing, use the cxfslicense command in a DOS command shell.

Create a DOS command shell with the following sequence:

> **Start**
>   **> Programs**
>   **> Accessories**
>   **> Command Prompt**

To run cxfslicense, enter the following command:

```
C:\> cxfslicense
```

If a valid client-side license key has been correctly specified for a Windows node, the following will be displayed:

```
Checking all supported Windows licenses
Found a valid license for CXFS for Windows
No license required for XVM user feature.
No license required for XVM mirror feature.
```

If you are using client-side licensing and there are errors, verify that you have installed the CXFS license keys as described in "Reinstalling An Existing Client-Side License Key for Windows " on page 236. For more information about licenses, see *CXFS Administration Guide for SGI InfiniteStorage*.

Reinstalling An Existing Client-Side License Key for Windows

An existing Windows client-side license is installed in the following location:

```
%ProgramFiles%\CXFS\lib\license.dat
```

If you need to reinstall the existing client-side license, do the following:

1. Create the license key directory %ProgramFiles%\CXFS\lib and the file license.dat if necessary.

2. Add the key to the `license.dat` file.

Verifying 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**

Unable 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 client 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 241.

  – 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 administration node with the correct hostname? See "Configuring Hostnames on Mac OS X" on page 86.

    - Has the node been added to the cluster and enabled? See "Verifying the Cluster Status" on page 256.

  - 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 for Windows" on page 185). 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 client being fenced and the filesystem being unmounted. Check that the Windows client is still in membership and that there are no unmount messages in the cxfs_client.log file.

*   Loss of Fibre Channel connection to the Fibre Channel switch or RAID. Check the Fibre Channel connections and use the SanManager tool to verify that the HBA can still see all of the LUNs. Make sure the filesystems are still mounted.

*   The metadata server returned an I/O error. Check the system log on the metadata server for any I/O errors on the filesystem and take corrective action on the server if required.

CXFS Client Service Does Not Start on Windows

The following error may be seen when the CXFS Client service attempts to start:

`Error 10038: An operation was attempted on something that is not a socket.`

Check the **CXFS Client Log** in **CXFS Info** for information on why the CXFS client failed to start.

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 232 and "Unable to Mount Filesystems on Windows" on page 237.

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 253 .

- 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 256.

- Check that the CXFS Client service has started. See "Start/Stop the CXFS Client Service for Windows" on page 228 and "Verifying that the CXFS Software is Running Correctly for Windows" on page 237.

- Check the **CXFS Client Log** in **CXFS Info** for warnings and errors regarding license keys or 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 229) 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.

Memory Configuration for Windows

A Windows problem may affect Windows CXFS nodes performing large asynchronous I/O operations. If the Windows node crashes with a NO_MORE_SYSTEM_PTES message, the work-around described in the following link should be considered:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/winxppro/reskit/prmd_stp_fztl.asp

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
```

Error Reading Kernel XVM Cluster Mirror Status

The following message will be output by the cxfslicense -d command if you execute it before rebooting the system:

```
error reading kernel XVM cluster mirror status. Check if XVM module is
started.
```

After you reboot the system and therefore load the XVM module, this message will no longer appear when you run cxfslicense -d.

Reporting Windows Problems

To report problems about a Windows node, you should retain platform-specific information and save crash dumps.

Retain Windows Information

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

- License key information (if using existing client-side licensing)

- CXFS version information

- Network settings

- Event log

- *(optionally)* Windows crash dump, as described in "Save Crash Dumps for Windows" on page 245

You can obtain information about the cluster by running the cxfsdump utility on a CXFS administration node.

Save 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.

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 "Save Crash Dumps for Windows" on page 245), you can modify the registry so that a combination of key strokes will cause the Windows node to crash. This will only work on machines with a PS/2 keyboard.

To do this, run the registry editor as follows:

> **Start**
> > **> Run**
> > > **> regedit**

Then navigate to:

```
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

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 CXFS administration node), the cxfs_admin command on any host that has access permission to the cluster, or the cmgr command on a CXFS administration node. The GUI and cxfs_admin provide a guided configuration and setup help for defining a cluster.

This section discusses the following tasks in cluster configuration:

- "Defining the Client-Only Nodes" on page 248
- "Adding the Client-Only Nodes to the Cluster (cmgr and GUI)" on page 250
- "Defining the Switch for I/O Fencing" on page 250
- "Starting CXFS Services on the Client-Only Nodes (cmgr and GUI)" on page 252
- "Verifying LUN Masking" on page 252
- "Mounting Filesystems on the Client-Only Nodes" on page 253
- "Unmounting Filesystems" on page 254
- "Forced Unmount of CXFS Filesystems" on page 254
- "Restarting the Windows Node" on page 255
- "Verifying the Cluster Configuration" on page 255
- "Verifying Connectivity in a Multicast Environment" on page 255
- "Verifying the Cluster Status" on page 256
- "Verifying the I/O Fencing Configuration" on page 259
- "Verifying Access to XVM Volumes" on page 261

For detailed configuration instructions, see the *CXFS 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

- 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 than IRIX or 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
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
cxfs_admin:mycluster> create node
  Specify the attributes for create node:
   name? solaris1
   os? solaris
   private_net? 192.168.0.178
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 than IRIX or 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 or cmgr; cxfs_admin provides a default node ID.

For example, the following 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|HPUX|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
```

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
Node "cxfsopus5" has been created, waiting for it to join the cluster...
```

For details about these commands, see the *CXFS Administration Guide for SGI InfiniteStorage*.

Adding the Client-Only Nodes to the Cluster (`cmgr` and GUI)

If you are using the GUI or 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 253.

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 CXFS 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 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 (`cmgr` and GUI)

After adding the client-only nodes to the cluster with the GUI or `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
```

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 or the 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" on page 19 and "Forced Unmount of CXFS Filesystems" on page 254.

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 or the cmgr command, 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 the "CXFS Filesystems Tasks with the GUI" sections of the GUI or the `cmgr` reference chapters in the *CXFS Administration Guide for SGI InfiniteStorage*.

Restarting the Windows Node

> After completing the steps in "Postinstallation Steps for Windows" on page 223 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:

```
irix# /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

> To verify general connectivity in a multicast environment, you can execute a UNIX `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 CXFS administration node) and the `cxfs_info` command (on a client-only node).

For example, using `cxfs_admin`:

```
cxfs_admin:mycluster> status
  Cluster    : mycluster
  Tiebreaker : node3
  Licenses   : n/a
  ----------------- -------- --------------------------------------------------
  Node              Cell ID  Status
  ----------------- -------- --------------------------------------------------
  node1 *           4        Disabled
  node2 *           5        Disabled
  node3             2        Stable
  node4 *           1        Stable
  node5             3        Disabled
  node6 *           0        Stable


  ----------------- ----------------- ---------------------------------------
  Filesystem        Mount Point       Status
  ----------------- ----------------- ---------------------------------------
```

```
    tp9500_0               /mnt/cxfs_vol        Mounted

    -----------------  ----------  ------------------------------------------------
    Switch             Port Count  Known Fenced Ports
    -----------------  ----------  ------------------------------------------------
    myswitch0          16          10
```

For example, using `clconf_info`:

```
irix# /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
_____ _____ _____ _____ _____
cxfsirix4         1 up           27      2
cxfsirix5         2 up           26      1
cxfsirix6         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=(cxfsirix4)  5
client(s)=(cxfsirix6,cxfsirix5,cxfswin4,cxfssun3,cxfsmac3.local.)  status=UP
/dev/cxvm/vol1 on /mnt/vol1  enabled  server=(cxfsirix5)  5
client(s)=(cxfsirix6,cxfsirix4,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, IRIX, Linux, and Solaris: /usr/cxfs_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:
     cxfsirix4  enabled  up    2
     cxfsirix5  enabled  up    1
     cxfsirix6  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 CXFS administration node and use the `cxfs-config`(1M) command. For example:

```
irix# /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 248 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 cxfsirix2
    switch 2: 16 port brocade admin@asg-fcsw8 <no ports masked>
        port 5: 210300e08b71fd49 cxfswin2
        port 14: 210000e08b12d914 cxfsirix2
```

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 40
- "I/O Fencing for Linux" on page 67
- "I/O Fencing for Mac OS X" on page 102
- "I/O Fencing for SGI ProPack Client-Only Nodes" on page 122
- "I/O Fencing for Solaris" on page 163
- "I/O Fencing for Windows" on page 225

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 CXFS administration node and run:

irix# **/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 = cxfsirix5 (0), cxfsirix4 (1)
    clients = cxfsirix4, cxfsirix5, cxfsirix6, 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
```

It is then possible to 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.

```
irix# 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, it is then possible to run the xvm command on the client to identify the matching disk devices on the client:

```
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)
```

The process to map device names to identify the target WWN is platform-specific. See:

- "Mapping XVM Volumes to Storage Targets on AIX" on page 44

- "Mapping XVM Volumes to Storage Targets on Linux" on page 75

- "Mapping XVM Volumes to Storage Targets on Mac OS X" on page 107

- "Mapping XVM Volumes to Storage Targets on SGI ProPack" on page 130

- "Mapping XVM Volumes to Storage Targets on Solaris" on page 169

Note: There is no xvm command on the Windows platform and therefore no method to map XVM volumes directly to disk devices under Windows.

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 volvol/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 CXFS 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:

- "Identifying Problems" on page 263
- "Typical Problems and Solutions" on page 266
- "Reporting Problems to SGI" on page 271

Also see the following platform-specific sections:

- "Troubleshooting for AIX" on page 44
- "Troubleshooting for Linux" on page 75
- "Troubleshooting for Mac OS X" on page 108
- "Troubleshooting for SGI ProPack Client-Only Nodes" on page 131
- "Troubleshooting for Solaris" on page 170
- "Troubleshooting for Windows" on page 234

For more advanced cluster troubleshooting, see the *CXFS Administration Guide for SGI InfiniteStorage*.

Identifying Problems

This section provides tips about identifying problems:

- "Is the Client-Only Node Configured Correctly? " on page 264
- "Is the Client-Only Node in Membership?" on page 264
- "Is the Client-Only Node Mounting All Filesystems?" on page 264
- "Can the Client-Only Node Access All Filesystems?" on page 265
- "Are There Error Messages?" on page 265
- "What Is the Network Status?" on page 265

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 administration node:

```
/usr/cluster/bin/cxfs-config -all
```

For more information, see "Verifying the Cluster Status" on page 256.

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 256.

If the client is not in membership, see the following:

- "Verifying the Cluster Configuration" on page 255

- "Verifying Connectivity in a Multicast Environment" on page 255

- "Unable to Achieve Membership" on page 266

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 256.

If the client has not mounted all filesystems, see the following:

- "Verifying the Cluster Configuration" on page 255

- Appendix C, "Mount Options Support" on page 281

- "Determining If a Client-Only Node Is Fenced" on page 269

- "Verifying Access to XVM Volumes" on page 261

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 269 and "Verifying Access to XVM Volumes" on page 261.

If accessing the filesystem hangs the viewing process, see "Filesystem Appears to Be Hung" on page 267.

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 273 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 27
- "Log Files on Linux" on page 56
- "Log Files on Mac OS X" on page 85
- "Troubleshooting for SGI ProPack Client-Only Nodes" on page 131
- "Log Files on Solaris" on page 138
- "Log Files and Cluster Status for Windows" on page 180

There are various logs also located on the CXFS administration nodes. For more information, see the *CXFS Administration Guide for SGI InfiniteStorage*.

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.

Typical Problems and Solutions

This section contains the following typical problems that apply to any platform:

- "Unable to Achieve Membership" on page 266
- "Filesystem Appears to Be Hung" on page 267
- "Determining If a Client-Only Node Is Fenced" on page 269
- "No HBA WWPNs are Detected" on page 270
- "Membership Is Prevented by Firewalls" on page 271

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 Daemon for AIX" on page 41
   - "Start/Stop cxfs_client for Linux" on page 69
   - "Start/Stop cxfs_client for Mac OS X" on page 104
   - "Mapping XVM Volumes to Storage Targets on SGI ProPack" on page 130
   - "Start/Stop cxfs_client for Solaris" on page 166
   - "Start/Stop the CXFS Client Service for Windows" on page 228

2. Verify that the license check has succeeded in the cxfs_client log file.

3. Look for other warnings and error messages in the cxfs_client log file. See Appendix A, "Operating System Path Differences" on page 273 for the location of the log file on different platforms.

4. Check cxfs-config output on the CXFS administration node to ensure that the client is correctly configured and is reachable via the configured CXFS private network. For example:

   irix# /**usr/cluster/bin/cxfs-config -all**

5. Check that the client is enabled into the cluster by running clconf_info on a CXFS administration node.

6. 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.

7. Check that the metadata server has the node correctly configured in its hostname lookup scheme (/etc/host file or DNS).

8. If you are still unable to resolve the problem, reboot the client node.

9. If rebooting the client node in step 8 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.

10. If restarting cluster administration daemons in step 9 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 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:

```
irix# /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`:

```
irix# /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 259.

- 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 270.

- 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 40
- "I/O Fencing for Linux" on page 67
- "I/O Fencing for Mac OS X" on page 102
- "I/O Fencing for SGI ProPack Client-Only Nodes" on page 122
- "I/O Fencing for Solaris" on page 163
- "I/O Fencing for Windows" on page 225

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.

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 50
- "Reporting Linux Problems" on page 80
- "Reporting Mac OS X Problems" on page 111
- "Reporting SGI ProPack Client-Only Nodes Problems" on page 133
- "Reporting Solaris Problems" on page 174
- "Reporting Windows Problems" on page 244

Operating System Path Differences

This appendix lists the location of CXFS-specific commands and files. For SGI ProPack paths, see the *CXFS Administration Guide for SGI InfiniteStorage* 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:	/usr/tmp/cxfs_client
Options file:	/usr/cxfs_cluster/bin/cxfs_client.options
CXFS status:	/usr/cxfs_cluster/bin/cxfs_info
CXFS client-side license file:	/etc/flexlm/license.dat
CXFS client-side license verification command:	/usr/cxfs_cluster/bin/cxfslicense
Hostname/address information	/etc/hosts
GRIO v2 administration	/usr/cxfs_cluster/bin/grioadmin
GRIO v2 quality of service	/usr/cxfs_cluster/bin/grioqos
XVM query	/usr/cxfs_cluster/bin/xvm

Table A-2 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`
CXFS client-side license file:	`/etc/flexlm/license.dat`
CXFS client-side license verification command:	`/usr/cluster/bin/cxfslicense`
Hostname/address information	`/etc/hosts`
GRIO v2 administration	`/usr/sbin/grioadmin`
GRIO v2 quality of service	`/usr/sbin/grioqos`
XVM query	`/sbin/xvm`

Table A-3 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:	`%ProgramFiles%\CXFS\cxfs_info.exe`
CXFS client-side license file:	`/etc/flexlm/license.dat`
CXFS client-side license verification command:	`/usr/cluster/bin/cxfslicense`
Hostname/address information	`/etc/hosts`
GRIO v2 administration	`/usr/sbin/grioadmin`
GRIO v2 quality of service	`/usr/sbin/grioqos`
XVM query	`/usr/cluster/bin/xvm`

Table A-4 Solaris Paths

Component	Path
CXFS client daemon:	`/usr/cxfs_cluster/bin/cxfs_client`
Command that normally invokes the client daemon:	`/etc/init.d/cxfs_cluster`
Log file:	`/var/log/cxfs_client`
Options file:	`/usr/cxfs_cluster/bin/cxfs_client.options`
CXFS status:	`/usr/cxfs_cluster/bin/cxfs_info`
CXFS client-side license file:	`/etc/flexlm/license.dat`
CXFS client-side license verification command:	`/usr/cxfs_cluster/bin/cxfslicense`
Hostname/address information	`/etc/hosts`
GRIO v2 administration	`/usr/sbin/grioadmin`
GRIO v2 quality of service	`/usr/sbin/grioqos`
XVM query	`/usr/cxfs_cluster/bin/xvm`

Table A-5 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 229
CXFS status:	`%ProgramFiles%\CXFS\cxfs_info.exe`
CXFS client-side license file:	`%ProgramFiles%\CXFS\lib\license.dat`
CXFS client-side license verification command:	`c:\Program Files\cxfs\cxfslicense`
Hostname and address information:	`%SystemRoot%\system32\drivers\etc\hosts`
GRIO v2 administration:	`%ProgramFiles%\CXFS\grioadmin.exe`
GRIO v2 quality of service:	`%ProgramFiles%\CXFS\grioqos.exe`
XVM query:	(unsupported)

Filesystem and Logical Unit Specifications

Table B-1 on page 280 summarizes filesystem and logical unit specifications differences among the supported client-only platforms.

Table B-1 Filesystem and Logical Unit Specifications

Item	AIX	Linux i386[1]	Linux x86_64	Linux ia64	Mac OS X	SGI ProPack	Solaris	Windows
Maximum filesystem size	2^{64} bytes [2]	16 TB	2^{64} bytes	2^{64} bytes	2^{64} bytes	2^{64} bytes	2^{64} bytes	2^{64} bytes
Maximum file size/offset	16 TB [3]	16 TB [4]	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)[5]	4096 (XFS default)	512, 1024, 2048, or 4096	512, 1024, 2048, or 4096	512, 1024, 2048, 4096, 8192, or 16384	4096, 8192, 16384, 32768, or 65536	512, 1024, 2048, 4096, 8192, or 16384	2048, 4096, 8192, 16384, 32768, or 65536 [6]	512, 1024, 2048, 4096, 8192, 16384, 32768, or 65536
XVM device block size (in bytes)	512	512	512	512	512	512	512	512
Physical LUN limit	2 TB	2 TB	2 TB	2 TB	2 TB	2 TB	1 TB or 2 TB [7]	2 TB
Maximum concatenated slices	65536 [8]	65536	65536	65536	65536	65536	65536	65536

---

1   The Linux architecture is reported by the `uname -i` command
2   About 18 million terabytes
3   Assumes the default `ulimit` is changed, see "Limitations and Considerations for AIX" on page 28.
4   Using large file support (`O_LARGEFILE`)
5   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.
6   8192 is recommended
7   DVH labels for Solaris 9 and Solaris 10 have a limit of 1 TB. GPT labels for Solaris 10 have a limit of 2 TB. (Solaris 9 does not support GPT labels.)
8   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. Both the IRIX and the SGI ProPack for Linux mount commands support 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 IRIX mount(1M) and Linux mount(8) man pages.

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) 32–bit Linux on i386 architecture and 64–bit Linux on x86_64 and ia64 architectures.

Table C-1 Mount Options Support for Client-Only Platforms in an IRIX Cluster

Option	Checked by Client	AIX	IRIX	Linux 32	Linux 64	Mac OS X	Solaris	Windows
attr2	Y		S					
biosize	Y	S	S	S	S	S	S	S
client_timeout	N	D	D	D	D	D	D	D
dmapi	N	n	n	n	n	n	n	n
dmi	N		S	S	S	S	S	S
filestreams	Y	S	S	S	S	S	S	S
gqnoenforce	N		S		S	S	S	S
gquota	N		S		S	S	S	S
grpid	N		S		n	S	S	S
inode64	Y	S	S	n	S	n	S	S
logbsize	Y		S					S
logbufs	Y		S					S
logdev	N							
noalign	Y							
noatime	Y		S		S	S	S	S
noattr2	N		S					
noauto	N	n	n	n	n	n	n	n

Option	Checked by Client	AIX	IRIX	Linux 32	Linux 64	Mac OS X	Solaris	Windows
nodev	N		S		S	S	n	S
noquota	N		S		S	S	S	S
nosuid	N		S		S	S	S	S
osyncisdsync	Y		S					S
pqnoenforce	N		S					S
pquota	N		S					S
qnoenforce	N		S		S	S	S	S
quota	N		S		S	S	S	S
ro	N	S	S	S	S	S	S	S
rtdev	N	n	S	n	n	n	n	n
rw	N		S		S	S	S	S
server_list	N	D	D	D	D	D	D	D
server_timeout	N	D	D	D	D	D	D	D
shared	Y							
sunit	N		S					S
swalloc	Y		S					S
swidth	N		S					S
uqnoenforce	N		S		S	S	S	S

Option	Checked by Client	AIX	IRIX	Linux 32	Linux 64	Mac OS X	Solaris	Windows
`uquota`	N		S		S	S	S	S
`wsync`	Y		S					S

Table C-2 Mount Options Support for Client-Only Platforms in an SGI ProPack Cluster

Option	Checked by Client	AIX	IRIX	Linux 32	Linux 64	Mac OS X	Solaris	Windows
attr2	Y							
biosize	Y	S	S	S	S	S	S	
client_timeout	N	D	D	D	D	D	D	D
dmapi	N	n	n	n	n	n	n	n
dmi	N		S	S	S	S	S	
filestreams	Y	S	S	S	S	S	S	S
gqnoenforce	N		S		S	S	S	
gquota	N		S		S	S	S	
grpid	N	n	n	n	n	n	n	n
inode64	Y		S	n	S	n	S	
logbsize	Y							
logbufs	Y							
logdev	N	S		S	S	S	S	S
noalign	Y							
noatime	Y		S		S	S	S	n
noattr2	N							
noauto	N	n	n	n	n	n	n	n

Option	Checked by Client	AIX	IRIX	Linux 32	Linux 64	Mac OS X	Solaris	Windows
nodev	N		S		S	S	n	
noquota	N		S		S	S	S	
nosuid	N		S		S	S	S	
osyncisdsync	Y							
pqnoenforce	N							
pquota	N							
qnoenforce	N		S		S	S	S	
quota	N		S		S	S	S	
ro	N	S	S	S	S	S	S	S
rtdev	N	n	n	n	n	n	n	n
rw	N		S		S	S	S	
server_list	N	D	D	D	D	D	D	D
server_timeout	N	D	D	D	D	D	D	D
shared	Y							
sunit	N							
swalloc	Y							
swidth	N							
uqnoenforce	N		S		S	S	S	

Option	Checked by Client	AIX	IRIX	Linux 32	Linux 64	Mac OS X	Solaris	Windows
uquota	N		S		S	S	S	
wsync	Y							

Error Messages

The following are commonly seen error messages:

- "Could Not Start CXFS Client Error Messages" on page 289
- "CMS Error Messages" on page 290
- "Mount Messages" on page 290
- "Network Connectivity Messages" on page 290
- "Device Busy Message" on page 291
- "Windows Messages" on page 291

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:

- For server-side licenses, see the *CXFS Administration Guide for SGI InfiniteStorage.*
- For existing client-side licenses, see:

  – "Reinstalling An Existing Client-Side License Key for AIX" on page 46

  – "Reinstalling Existing Client-Side License Keys for Linux" on page 77

  – "Reinstalling An Existing Client-Side License Key for Mac OS X" on page 109

  – "Reinstalling An Existing Client-Side License Key for SGI ProPack 4" on page 132

  – "Reinstalling An Existing Client-Side License Key for Solaris" on page 171

  – "Reinstalling An Existing Client-Side License Key for Windows " on page 236

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 138 and "Log Files and Cluster Status for Windows" on page

180. Also see the log files on the CXFS administration node; for more information, see the *CXFS 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 229.

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.

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 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 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 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.

Glossary

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.

administration node

A node in the pool that is installed with the cluster_admin.sw.base software product, allowing the node to perform cluster administration tasks and contain a copy of the cluster database. There are two types of administration nodes: *server-capable administration nodes* and *client administration nodes*.

administrative stop

See *forced CXFS shutdown*

cell ID

A number associated with a node that is used by the CXFS software and appears in messages.

CLI

Underlying command line interface commands used by the CXFS Manager graphical user interface (GUI) and the cmgr command.

client

See *CXFS client node*, *CXFS client-only node* and *administration node*.

client administration node

A node that is installed with the cluster_admin software product, allowing the node to perform cluster administration tasks and contain a copy of the cluster database, but is not capable of coordinating CXFS metadata. Only supported for IRIX nodes running in coexecution with FailSafe.

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*

client-side licensing

Licensing that requires a node-locked license on each node in the cluster. *Server-side licensing* is preferred over client-side licensing; existing client-side licensing is supported on a limited set of platforms. 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. For details, see the *CXFS Administration Guide for SGI InfiniteStorage*.

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: `fs2d`, `cad`, `cmond`, `crsd`.

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 domain

XVM concept in which a filesystem applies to the entire cluster, not just to the local node. See also *local domain*.

cluster database membership

The group of 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 administration nodes in the pool. (Also known as *user-space membership* and *fs2d database membership*.)

cluster ID

A unique number within your network in the range 1 through 128. 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 resets any node for which it detects 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.

cluster node

A node that is defined as part of the cluster. See also *node*.

coexecution

The ability to run CXFS and IRIS FailSafe together.

control messages

Messages that cluster software sends between the cluster nodes to request operations on or distribute information about cluster nodes. Control messages and heartbeat messages are sent through a node's network interfaces that have been attached to a control network.

control network

The network that connects nodes through their network interfaces (typically Ethernet) such that CXFS can send heartbeat messages and control messages through the network to the attached nodes. CXFS uses the highest priority network interface on the control network; it uses a network interface with lower priority when all higher-priority network interfaces on the control network fail.

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 an 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* and FailSafe membership. For more information about FailSafe, see *FailSafe Administrator's Guide for SGI InfiniteStorage*.

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 an 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 `cmgr` command.

CXFS services stop

To disable a node, which changes a flag in the cluster database, by using a CXFS graphical user interface (GUI) or the `cmgr` 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. The CXFS tiebreaker differs from the FailSafe tiebreaker; see *FailSafe Administrator's Guide for SGI InfiniteStorage*.

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 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*.

FailSafe membership

The group of nodes that are actively sharing resources in the cluster, which may be a subset of the nodes defined in a cluster. FailSafe membership differs from *CXFS kernel membership* and *cluster database membership*. For more information about FailSafe, see *FailSafe Administrator's Guide for SGI InfiniteStorage*.

failure action hierarchy

See *failpolicy methods*

failpolicy methods

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*, *fenceresetreset*, and *shutdown*. Also known as *failure action hierarchy*

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 nodes (unless there is a single server-capable 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 file systems 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*.

heartbeat messages

Messages that cluster software sends between the nodes that indicate a node is up and running. Heartbeat messages and *control messages* are sent through the node's network interfaces that have been attached to a control network.

heartbeat interval

The time between heartbeat messages. The node timeout value must be at least 10 times the heartbeat interval for proper CXFS operation. The higher the number of heartbeats (smaller heartbeat interval), the greater the potential for slowing down the network. See also *dynamic heartbeat monitoring* and *static heartbeat monitoring*.

I/O fencing

See *fence*.

kernel-space membership

See *CXFS kernel membership*.

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.

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 administration node that coordinates updating of meta data 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

A cluster that is running multiple operating systems, such as IRIX and Solaris.

multiport serial adapter cable

A device that provides four DB9 serial ports from a 36-pin connector.

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 *administration node*, *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 in the pool. If you do not specify a number, CXFS will calculate an ID for you. You must not change the node ID number after the node has been defined.

node membership

The list of nodes that are active (have CXFS kernel membership) in a cluster.

node timeout

If no heartbeat is received from a node in this period of time, the node is considered to be dead. The node timeout value must be at least 10 times the heartbeat interval for proper CXFS operation.

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.

pool

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

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

port password

The password for the system controller port, usually set once in firmware or by setting jumper wires. (This is not the same as the node's root password.)

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.

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 nodes in the cluster are required to **form** an initial membership

  – Half (**50%**) of the server-capable 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.

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).

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 nodes; it does not require node-locked license keys on CXFS client-only nodes. The license keys are node-locked to each server-capable node and specify the number and size of client-only nodes that

may join the cluster membership. Server-side licensing is preferred over client-side licensing; existing *client-side licensing* is supported on a limited set of platforms. 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. For details, see the *CXFS Administration Guide for SGI InfiniteStorage*.

shutdown

The fail action hierarchy selection 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*.

snooping

A security breach involving illicit viewing.

split-brain syndrome

A situation in which multiple clusters are formed due to a network partition and the lack of reset and/or CXFS tiebreaker capability.

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 constantly at 1-second intervals and declares a timeout after 5 consecutive missed seconds (default). See also *dynamic heartbeat monitoring*.

storage area network (SAN)

A dedicated, high-speed, scalable network of servers and storage devices designed to enhance the storage, retrieval, and management of data

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*.

user-space membership

See *cluster database membership*.

view area

The portion of the GUI window that displays components graphically. See also *details area*.

Index

100baseT, 114

A

ACL problem and AIX, 49
acledit, 28
aclget, 28
aclput, 28
ACLs
    AIX, 28, 32
    Linux, 58
    Mac OS X, 88
    Solaris, 139
    Windows, 190
Active Directory user ID mapping method, 217
adapter parameters, 148, 151
admin account, 17
admin cxfs_unmount, 253
administrative tasks, 5
AIX
    ACLs, 28
    client software installation, 37
    commands installed by CXFS , 27
    common problems, 47
    FLEXlm license key verification, 39, 121
    hardware, 26
    HBA installation, 33
    identifying problems, 263
    ifconfig, 35
    kernel extensions, 51
    limitations, 28
    log files, 27
    manual CXFS startup/shutdown, 42
    modify the CXFS software, 43
    NFS export scripts, 6
    operating system version, 26
    postinstallation steps, 40
    preinstallation steps, 33
    problem reporting, 50
    requirements, 26
    software
        maintenance, 42
        upgrades, 42
    space requirements, 37
alog, 50
AMCC JNI HBA, 136
appropriate use of CXFS, 14

B

backups, 21
bandwidth, 2, 14
best practices, 13
    administration tasks, 19
    appropriate use of CXFS, 14
    backups, 21
    client-only nodes, 16
    configuration tasks, 13
    cron jobs, 21
    fast copying, 23
    filesystem repair, 22
    firewall configuration, 19
    forced unmount, 19
    hostname resolution rules, 15
    maintenance of CXFS, 23
    mix of software releases, 17
    network configuration rules, 15
    network issues, 16
    node shutdown, 21
    platform-specific limitations, 20
    power mangement software, 23
    private network, 16