CXFS™ MultiOS Client-Only Guide for SGI® InfiniteStorage

007–4507–009

CONTRIBUTORS

Written by Lori Johnson

Illustrated by Chrystie Danzer

Production by Karen Jacobson

Engineering contributions to the book by Vlad Apostolov, Neil Bannister, Dale Brantly, David Chatterton, Mark Cruciani, Tad Dolphay, Dave Ellis, Eric Eppe, Andrew Gildfind, Dennis Kender, Aaron Mantel, Troy McCorkell, Ken McDonell, Bill Mckevitt, Terry Merth, Daniel Moore, Max Matveev, Fujio Nakajima, Barry Naujok, Mike Raskie, Eric Sandeen, Tim Sirianni, Wesley Smith, Michael Umansky, Madan Valluri, Geoffrey Wehrman

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

New Features in this Guide

Note: Be sure to read the `README` files for your platforms to learn about any late-breaking changes in the installation and configuration procedures.

This guide contains the following new features:

- Support for the Apple Computer, Inc. Mac OS X operating system on client-only nodes. See:
  - "Mac OS X Overview" on page 15
  - "Mac OS X Host Information" on page 32
  - Chapter 7, "Mac OS X Platform" on page 67
  - "Defining the Client-Only Nodes" on page 159
  - "Mac OS X Error Messages" on page 170
  - "Common Mac OS X Problems" on page 182
  - "Reporting Mac OS X Problems" on page 189
- Support for a cluster of up to 64 nodes. See "Requirements" on page 6.
- Information about the SGI TP9300, SGI TP9300S, and SGI TP9500S. See Chapter 2, "SGI RAID Firmware" on page 19.
- Information about setting the LUN discovery method for Solaris systems using the SGI TP9100 1-Gbit controller; see "Setting the LUN Discovery Method for SGI TP9100" on page 94.
- Additional AIX troubleshooting information. See "Common AIX Problems" on page 180.

Note: The Microsoft Windows NT 4.0 platform is no longer supported.

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 the 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 Infinite Storage*.
009	February 2004 Revised to support the 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.

Contents

Figures

About This Guide

This publication documents the CXFS MultiOS Clients 3.1 release. This release supports Apple Computer Mac OS X, IBM AIX, Linux 32-bit on supported platforms, Sun Microsystems Solaris, Microsoft Windows 2000, and Microsoft Windows XP nodes.

Prerequisites

This guide assumes the following:

- The IRIX or Linux 64-bit 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 Infinite Storage* 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 Infinite Storage* (TPL link)
- SGI TP9400 and SGI TP9500 documentation :
  - *SGI TPSSM Software Concepts Guide* (TPL link)
  - *SGI® Total Performance 9400 and SGI® Total Performance 9500 RAID User's Guide* (TPL link)
  - *SGI TPSSM Administration Guide* (TPL link)

The SGI TP9400 documentation is available on the release CD in the following files:

- `tp9400_sw_concepts_guide.pdf`
- `tp9400_owners_guide.pdf`
- `tp9400_admin_guide.pdf`

- SGI TP9100:
  - *TPM Installation Instructions and User's Guide for TP9100*
- 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*

  Also see the AMCC website at:

  http://www.amcc.com
- Astera Technologies HBA card and driver documentation:
  - *JumanJi User Guide*

  See the Astera Technologies website at:

  http://www.asteratech.com
- QLogic HBA card and driver documentation:
  - *SANblade 2x00 Series User's Guide*
  - *SANsurfer Applications User's Guide*

  See the QLogic website at:

  http://www.qlogic.com
- AIX documentation:
  - *AIX 5L V5.1 Installation Guide*
  - *AIX V5.1 Network Installation Mangement Guide and Reference*
  - *AIX 5L V5.1 Installation in a Partitioned Environment Guide*

See the IBM website at:

http://www.ibm.com

- Linux 32-bit platforms documentation:
  - *The Official Red Hat Linux x86 Installation Guide*
  - *The Official Red Hat Linux Reference Guide*
- 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 8 Installation Guide*
  - *Solaris 8 System Administration Collection*
  - *Solaris 8 Advanced Installation Guide*
  - *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.

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 as follows:
- On IRIX systems, use either `grelnotes` or `relnotes`
- On Linux 32-bit systems, see `linux-IA32/README_CXFS_LINUXIA32_3.1.0.txt` on the CXFS multiOS CD
- On Mac OS X systems, see `/usr/cluster/docs/ReadMe.html`
- On Solaris and AIX systems, see `/usr/cxfs_cluster/doc/relnotes`
- On Windows systems, see `C:\Program Files\CXFS\relnotes.rtf`. You can access this by selecting:

  Start
  > Programs
  > CXFS
  > Release Notes

- On IRIX and Linux systems, you can view man pages by typing `man` *title* at a command line.

Conventions

Note: This guide uses *Solaris* to Solaris 8 and Solaris 9 and *Windows* to refer to Microsoft Windows 2000 and Microsoft Windows XP nodes, when the information applies equally to all. Information that applies to only one of these types of nodes is identified.

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

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

  http://docs.sgi.com

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

- Send mail to the following address:

  Technical Publications
  SGI
  1500 Crittenden Lane, M/S 535
  Mountain View, California 94043–1351

SGI values your comments and will respond to them promptly.

Introduction

This guide provides an overview of the installation and configuration procedures for CXFS client-only nodes running SGI CXFS clustered filesystems. A *CXFS client-only node* runs a subset of CXFS processes and services.

This release supports client-only nodes running the following operating systems:

- IBM AIX 5L version 5.1 ML 4 (64-bit kernel mode) APAR number IY42428
- Apple Computer, Inc., Mac OS X operating system 10.2.8
- Red Hat Linux on supported 32-bit platforms (see "Requirements Specific to Linux 32-bit Platforms" on page 52):
  - Red Hat Linux 8.0
  - Red Hat Linux 9
- Sun Microsystems Solaris:
  - Solaris 8 plus appropriate patch (see the release notes)
  - Solaris 9 plus appropriate patch
- Microsoft Windows:
  - Windows 2000 Service Pack 3 or Service Pack 4
  - Windows XP Service Pack 1

Note: This guide uses *Solaris* to refer to both Solaris 8 and Solaris 9 and *Windows* to refer to Windows 2000 and Windows XP nodes when the information applies to all platforms. Information that applies to only one operating system is identified.

A cluster running multiple operating systems is known as a *multiOS cluster*.

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.

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. Contact your local SGI sales representative for details.

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

This chapter discusses the following:

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.

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. (*Metadata* is information that describes a file, such as the file's name, size, location, and permissions.) CXFS is faster than NFS because the data does not go through the network.

CXFS on Client-Only Nodes

This section contains the following:

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.

Licenses

You must have the following licenses:

- Brocade license. See "Required Brocade Fibre Channel Switch Firmware and License" on page 23.
- CXFS FLEXlm license installed on every node in the cluster; see Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29.

Note: XVM provides a mirroring feature. If you want to access a mirrored volume from a given node in the cluster, you must install a FLEXlm mirroring license on that node. Only those nodes that will access the mirrored volume must be licensed. For information about purchasing this license, see your SGI sales representative.

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.) 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 either the `cmgr` command or the CXFS graphical user interface (GUI). For more information about using these tools, see the *CXFS Administration Guide for SGI Infinite Storage*.

Note: The NFS export scripts are supported on AIX nodes, IRIX nodes , Linux 32-bit nodes, Linux 64-bit on SGI Altix 3000 nodes, and Solaris nodes; they are not supported on Mac OS X or Windows nodes. The scripts behave the same on these platforms, but the pathnames differ between client-only nodes and administration nodes:

On client-only nodes running AIX, IRIX, Linux 32-bit, Linux 64-bit, and Solaris:

```
/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
```

On administration nodes:

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

The following `cxfs-reprobe` scripts are run by either `clconfd` or `cxfs_client` when they need to reprobe the Fibre Channel controllers:

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

You may modify any of these scripts if needed.

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

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 the same 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 on different platforms.

Performance Considerations

CXFS may not give optimal performance under the following circumstances:

- 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.
- 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.

- When distributed applications write to shared files that are memory mapped.

Also see "Functional Limitations Specific to Windows" on page 121.

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
```

Requirements

Using a client-only node in a 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.

- At least one server-capable administration node to act as the metadata server and from which to perform cluster administration tasks. CXFS should be installed on the administration node before CXFS is installed on the client-only nodes.
- 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.
- A FLEXlm license key for CXFS and optionally XVM. The CXFS license is required for all nodes in the pool; a license is required for each node from which you want to access a mirrored XVM volume.
- A Brocade Fibre Channel 2400, 2800, 3200, 3800, 3900, or 12000 switch that is supported by SGI. The switch is required to protect data integrity.

  Nodes with system controllers use serial reset lines or I/O fencing to protect the integrity of the data stored in the cluster. (One of these methods is mandatory for the administration nodes in a cluster with only two server-capable nodes. Larger clusters should have an odd number of server-capable nodes.)

  The *I/O fencing* feature isolates a problem node so that it cannot access I/O devices and therefore cannot corrupt data in the shared CXFS filesystem. This feature can only be used with a Brocade Fibre Channel switch; therefore, the Brocade switch is a required piece of hardware in a multiOS cluster.

  I/O fencing differs from zoning:

  – *Fencing* is a generic cluster term that means to erect a barrier between a host and shared cluster resources.

  – *Zoning* is the ability to define logical subsets of the switch (zones), with the ability to include or exclude hosts and media from a given zone. A host can only access media that are included in its zone. Zoning is one possible implementation of fencing.

Zoning implementation is complex and does not have uniform availability across switches. Therefore, SGI chose to implement a simpler form of fencing, enabling/disabling a host's Brocade ports.

If there are problems with a node, the I/O fencing software sends a message via the `telnet` protocol to the appropriate Fibre Channel switch. The switch only allows one `telnet` session at a time; therefore, if you are using I/O fencing, you must keep the `telnet` port on the Fibre Channel switch free at all times.

Caution: Do not perform a `telnet` to the switch and leave the session connected.

- At least one administration node must be a server-capable administration node in order to be a CXFS metadata server; other nodes can 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. All AIX, Linux 32-bit, Mac OS X, Solaris, and Windows nodes are CXFS client-only nodes.

  A cluster in which both CXFS and IRIS 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. However, FailSafe cannot run on AIX, Linux 32-bit, Mac OS X, Solaris, or Windows nodes.

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

- Ensure that the appropriate software is installed on the potential metadata server nodes. For example, if you want to use quotas and access control lists (ACLs) on any cluster node, 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. For more information, see *IRIX Admin: Disks and Filesystems*, *IRIX Admin: Backup, Security and Accounting*, and 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 Specific to Solaris" on page 88 and "Requirements Specific to Windows" on page 118.

Recommendations

SGI recommends the following when running CXFS on a client-only node:

- Fix SAN connectivity issues before trying to use CXFS.
- Fix any network issues on the private network before trying to use CXFS.
- Use an Ethernet network switch rather than a hub for performance and control.
- Configure a production cluster with an odd number of server-capable administration nodes.
- For large clusters, SGI recommends that you define only the first three server-capable administration nodes and then continue on with the steps to define the cluster. After you have a successful small cluster, go back and add the remaining nodes.
- Launch any task initiated using `cron` on a CXFS filesystem from a single node in the cluster, preferably from the active metadata server.

  The `cron` daemon can cause severe stress on a CXFS filesystem if multiple nodes in a cluster start the same filesystem-intensive task simultaneously. An example of such a task is one that uses the `find` command to search files in a filesystem.
- Do not run any defragmentation software on CXFS filesystems. This includes the IRIX `fsr`(1M) command and any similar commands on other supported operating systems.
- Be very careful when running `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. SGI recommends that you contact SGI technical support before using `xfs_repair`. For more details, see the *CXFS Administration Guide for SGI Infinite Storage*.
- Only those nodes that you want to be potential metadata servers should be CXFS administration nodes (installed with the `cluster_admin` software product). CXFS client administration nodes should only be used when necessary for coexecution with IRIS FailSafe. All other nodes should be client-only nodes (installed with `cxfs_client`). Use an odd number of server-capable administration nodes.
- Shut down cluster services before maintenance.
- In this release, relocation is disabled by default and recovery is supported only when using standby nodes.

A *standby node* is a metadata 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. To use recovery, you must not run any applications on any of the potential metadata servers for a given filesystem; after the active metadata server has been chosen by the system, you can then run applications that use the filesystem on the active metadata server and client-only nodes.

Relocation and recovery are fully implemented, but the number of associated problems prevents full support of these features in the current release. Although data integrity is not compromised, cluster node panics or hangs are likely to occur. Relocation and recovery will be fully supported in a future release when these issues are resolved.

- Use the following good practices:
  - Unmount the filesystems from the metadata server, shut down the node, and remount the filesystem when possible.
  - Do the following before shutting down a node:
    - Unmount filesystems.
    - Shut down cluster services.
- Do not run power management software, which may interfere with the CXFS cluster.
- 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.

  The method used to implement this feature is platform-specific:

  - On IRIX nodes, this feature uses the `umount -k` option. The `-k` option attempts to kill processes that have open files or current directories in the appropriate filesystems and then unmount them. That is, it attempts to terminate any I/O going to the filesystem, so that it can unmount it promptly, rather than having to wait for the I/O to finish on its own, causing the unmount to possibly fail.

- On AIX nodes, a similar function is performed with the `fuser -k` command and `umount` command.
- On Linux 32-bit and Linux 64-bit nodes, a similar function is performed with the `fuser -m -k` command and the `umount` command
- On Mac OS X nodes, a similar function is performed with a modified version of the `lsof` command, followed by `unmount`
- On Solaris nodes, a similar function is performed with the `fuser -c -k` command and the `umount -f` command.
- On Windows nodes, all processes with open files on the CXFS filesystem are killed. For details, see "Forced Unmount on a Windows Node" on page 123.

This feature is available on an administration node with the following CXFS GUI menu:

Tasks
> Filesystems
> Unmount a Filesystem

In the CXFS GUI, click the **Force** toggle in the **Unmount Filesystem** task.

You can also specify this feature using the `cmgr`(1M) commands to define the filesystem. For more information, see "Forced Unmount of CXFS Filesystems" on page 168.

You must use `cmgr` from an administration node, and the GUI must be connected to an administration node.

For more information, see the *CXFS Administration Guide for SGI Infinite Storage*, and the `fuser` and `umount` man pages.

- Enable system dumps; see the operating system documentation for instructions.

Overview of the Installation and Configuration Steps

This section provides an overview of the installation, verification, and configuration steps for each platform type:

- "Hostname Resolution and Network Configuration Rules for All Platforms" on page 12

Hostname Resolution and Network Configuration Rules for All Platforms

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 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 cluster 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.

AIX Overview

Note: If you run into problems, see Chapter 11, "Troubleshooting" on page 169.

Following is the order of installation and configuration steps for a CXFS AIX node:

1. Read the release notes `README` file for the AIX platform to learn about any late-breaking changes in the installation procedure.
2. Install the AIX 5L version 5.1 operating system according to the directions in the AIX documentation (if not already done).
3. Install and verify the SGI RAID. See Chapter 2, "SGI RAID Firmware" on page 19.
4. Install and verify the Brocade Fibre Channel switch. See Chapter 3, "Brocade Fibre Channel Switch Verification" on page 23.
5. Obtain and install the CXFS license. If you want to access an XVM mirrored volume from a given node in the cluster, you must purchase the mirroring software option and obtain and install a FLEXlm license. Only those nodes that will access the mirrored volume must be licensed. For information about purchasing this license, see your sales representative. See Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29.
6. Install and verify the host bus adapter (HBA). See "Host Bus Adapter Installation and Configuration for AIX" on page 40.
7. Prepare the AIX node, including adding a private network. See "Preinstallation Steps for AIX" on page 40.
8. Install the CXFS software. See "Client Software Installation Steps for AIX" on page 44.
9. Create the I/O fencing file. See "Postinstallation Steps for AIX: Creating the I/O Fencing File" on page 46.
10. Configure the cluster to define the new AIX node in the pool, add it to the cluster, start CXFS services, and mount filesystems. See Chapter 10, "Cluster Configuration" on page 159.

Linux 32-bit Platforms Overview

Note: If you run into problems, see Chapter 11, "Troubleshooting" on page 169.

Following is the order of installation and configuration steps for a CXFS Linux 32-bit node on supported platforms:

1. Read the release notes `README` file for the Linux 32-bit platform to learn about any late-breaking changes in the installation procedure.
2. Install the Red Hat operating system according to the directions in the Red Hat documentation (if not already done).
3. Install and verify the SGI RAID. See Chapter 2, "SGI RAID Firmware" on page 19.
4. Install and verify the Brocade Fibre Channel switch. See Chapter 3, "Brocade Fibre Channel Switch Verification" on page 23.
5. Obtain and install the CXFS license. If you want to access an XVM mirrored volume from a given node in the cluster, you must purchase a mirroring software option and obtain and install a FLEXlm license. Only those nodes that will access the mirrored volume must be licensed. For information about purchasing this license, see your sales representative. See Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29.
6. Install and verify the host bus adapter (HBA). See "Host Bus Adapter Installation and Configuration for Linux 32-bit Platforms" on page 55.
7. Prepare the Linux 32-bit node, including adding a private network. See "Adding a Private Network for Linux 32-bit Nodes" on page 57.
8. Install the CXFS kernel and user space software packages. See "Client Software Installation Steps for Linux 32-bit Platforms" on page 61.
9. Configure the cluster to define the new Linux 32-bit node in the pool, add it to the cluster, start CXFS services, and mount filesystems. See Chapter 10, "Cluster Configuration" on page 159.

Mac OS X Overview

Note: If you run into problems, see Chapter 11, "Troubleshooting" on page 169.

Following is the order of installation and configuration steps for a CXFS Mac OS X node:

1. Read the release notes `ReadMe.html` file for the Mac OS X platform to learn about any late-breaking changes in the installation procedure.
2. Install the Mac OS X operating system according to the directions in the Mac OS X documentation (if not already done).
3. Install and verify the SGI RAID. See Chapter 2, "SGI RAID Firmware" on page 19.
4. Install and verify the Brocade Fibre Channel switch. See Chapter 3, "Brocade Fibre Channel Switch Verification" on page 23.
5. Obtain and install the CXFS license. If you want to access an XVM mirrored volume from a given node in the cluster, you must purchase a mirroring software option and obtain and install a FLEXlm license. Only those nodes that will access the mirrored volume must be licensed. For information about purchasing this license, see your sales representative. See Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29.
6. Install and verify the Astera Technologies host bus adapter (HBA). You will install the Rhino–3000 HDFC Driver package, which provides software for the Fibre Channel card and driver. See"Host Bus Adapter Installation and Configuration for Solaris" on page 92.
7. Prepare the Mac OS X node, including adding a private network. See "Preinstallation Steps for Mac OS X" on page 80..
8. Install the CXFS software. See "Client Software Installation Steps for Mac OS X" on page 83.
9. Configure the cluster to define the new Mac OS X node in the pool, add it to the cluster, start CXFS services, and mount filesystems. See Chapter 10, "Cluster Configuration" on page 159.

Solaris Overview

This information applies to both Solaris 8 and Solaris 9 unless otherwise noted.

Note: If you run into problems, see Chapter 11, "Troubleshooting" on page 169.

Following is the order of installation and configuration steps for a CXFS Solaris node:

1. Read the release notes `README` file for the Solaris platform to learn about any late-breaking changes in the installation procedure.
2. Install the Solaris 8 or Solaris 9 operating system according to the directions in the Solaris documentation (if not already done).
3. Install and verify the SGI RAID. See Chapter 2, "SGI RAID Firmware" on page 19.
4. Install and verify the Brocade Fibre Channel switch. See Chapter 3, "Brocade Fibre Channel Switch Verification" on page 23.
5. Obtain and install the CXFS license. If you want to access an XVM mirrored volume from a given node in the cluster, you must purchase a mirroring software option and obtain and install a FLEXlm license. Only those nodes that will access the mirrored volume must be licensed. For information about purchasing this license, see your sales representative. See Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29.
6. Install and verify the AMCC JNI host bus adapter (HBA). You will install the JNIC146x package, which provides software for the Fibre Channel card and driver. See "Host Bus Adapter Installation and Configuration for Solaris" on page 92.
7. Prepare the Solaris node, including adding a private network. See "Preinstallation Steps for Solaris" on page 106.
8. Install the CXFS software. See "Client Software Installation Steps for Solaris" on page 112.
9. Configure the cluster to define the new Solaris node in the pool, add it to the cluster, start CXFS services, and mount filesystems. See Chapter 10, "Cluster Configuration" on page 159.

Windows Overview

Note: If you run into problems, see Chapter 11, "Troubleshooting" on page 169.

Following is the order of installation and configuration steps for a CXFS Windows node:

1. Read the release notes for the Windows platform to learn about any late-breaking changes in the installation procedure.
2. Install the Windows operating system according to the directions in the Windows documentation (if not already done).
3. Install Windows software according to the directions in the Windows documentation (if not already done).
4. Install and verify the SGI RAID. See Chapter 2, "SGI RAID Firmware" on page 19.
5. Install and verify the Brocade Fibre Channel switch. See Chapter 3, "Brocade Fibre Channel Switch Verification" on page 23.
6. Obtain the CXFS license. If you want to access an XVM mirrored volume from a given node in the cluster, you must purchase a mirroring software option and obtain and install a FLEXlm license. Only those nodes that will access the mirrored volume must be licensed. For information about purchasing this license, see your sales representative. See Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29.
7. Install and verify the QLogic host bus adapter (HBA) and driver. See "Host Bus Adapter Installation for Windows" on page 136.
8. Prepare the Windows node, including adding a private network. See "Preinstallation Steps for Windows" on page 139.
9. Install the CXFS software. See "Client Software Installation Steps for Windows" on page 142.
10. Perform post-installation configuration steps:
    - "Configuring the FLEXlm License for Windows" on page 150
    - "Performing User Configuration" on page 151
    - "Creating a New Hardware Profile" on page 152

11. Configure the cluster to define the new Windows node in the pool, add it to the cluster, start CXFS services, and mount filesystems. See Chapter 10, "Cluster Configuration" on page 159.
12. Start CXFS services on the Windows node to see the mounted filesystems under the configured drive letter.

SGI RAID Firmware

The SGI RAID will be initially installed and configured by SGI personnel. For more information, see the documentation listed in the preface.

Supported SGI RAID Hardware

SGI supports the following RAID hardware:

- SGI TP9500S (serial ATA)
- SGI TP9500
- SGI TP9400
- SGI TP9300S (serial ATA)
- SGI TP9300
- SGI TP9100

By default, the RAID firmware supports a maximum number of logical units (LUNs). If additional LUNs are required, you must obtain a separate software-enabling key; this key will support a larger number of LUNs in separate partitions, which requires that the Fibre Channel ports be mapped to a partition. Contact your SGI sales representative for the SGI software partitioning key.

The default maximum number of LUNs supported depends upon the code installed, as show in Table 2-1.

Table 2-1 LUN Maximums

Firmware Level	Default LUN Maximum	LUN Maximum with a Partioning Key
04.01.*xx.xx*	32	128
04.02.*xx.xx*	32	128
05.30.*xx.xx*	32	1024
05.33.*xx.xx*	32	2048
05.40.*xx.xx*	256	2048

Required SGI RAID Firmware

This section describes the required RAID firmware.

Required SGI TP9500S RAID Firmware

The TP9500S 8.0 CD contains the required controller firmware and NVSRAM files. The 05.40.*xx.xx* code or later must be installed.

Required SGI TP9500 RAID Firmware

The TP9400/TP9500 6.0 CD contains the required controller firmware and NVSRAM files. The 05.30.*xx.xx* code or later must be installed.

Required SGI TP9400 RAID Firmware

The TP9400 4.0 CD contains the required controller firmware and NVSRAM files for the 4774 or 4884 units:

- If you have a 4774 unit, the 04.01.*xx.xx*, 04.02.*xx.xx*, or 05.30.*xx.xx* code must be installed
- If you have a 4884 unit, the 04.02.*xx.xx* code or later is installed by default.

Required SGI TP9300S RAID Firmware

The TP9300S 8.0 CD contains the required controller firmware and NVSRAM files. The 05.40.*xx.xx* code or later must be installed.

Required SGI TP9300 RAID Firmware

The TP9300 7.0 CD contains the required controller firmware and NVSRAM files. The 05.33.*xx.xx* code or later must be installed.

Required SGI TP9100 RAID Firmware

The TP9100 4.0 CD contains the required version 7.75 controller firmware for the 1-Gbit TP9100. The TP9100 5.0 CD contains the required version 8.40 firmware for the 2-Gbit TP9100. The TP9100 is limited to 64 host connections.

RAID Firmware Verification

To verify that the SGI RAID is properly installed and ready for use with CXFS, you can dump the RAID's profile and verify the controller software revisions.

Brocade Fibre Channel Switch Verification

In order to protect data integrity, AIX, Linux, Mac OS X, Solaris, and Windows nodes must use the *I/O fencing* feature, which 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 is also available for IRIX nodes and Linux 64–bit nodes on the SGI Altix platform. This feature can only be used with a Brocade Fibre Channel switch supported by SGI; therefore, the Brocade switch is a required piece of hardware in a multiOS cluster.

The Brocade Fibre Channel switches will be initially installed and configured by SGI personnel. You can use the information in this chapter to verify the installation.

Required Brocade Fibre Channel Switch Firmware and License

This release supports Brocade Silkworm Fibre Channel switches that are supported by SGI:

- 2400 (8-port)
- 2800 (16-port)
- 3200 (8-port, 2-Gbit)
- 3800 (16-port, 2-Gbit)
- 3900 (32-port, 2-Gbit)
- 12000 (32-, 64-, or dual 64-port, 2-Gbit)

All Brocade switches contained within the SAN fabric must have the appropriate Brocade license key installed. The following firmware is required:

- 2400 and 2800 switches: 2.6.0d or later
- 3200 and 3800 switches: 3.0.2c or later
- 3900 and 12000 switches: v4.0.2c or later

If the current firmware level of the switches must be upgraded, please contact your local SGI service representative or customer support center.

The Brocade switch must be configured so that its Ethernet interface is accessible(using `telnet`) from all CXFS administration nodes. The fencing network connected to the Brocade switch must be physically separate from the private heartbeat network.

Caution: The `telnet` port must be kept free in order for I/O fencing to succeed.

The 3900 and 12000 series switches permit multiple `telnet` sessions. However, CXFS I/O fencing requires a `telnet` lockout for global mutual exclusion when a fencing race occurs. Therefore, you must configure the 3900 and 12000 series switches to set the maximum allowed simultaneous `telnet` sessions for the `admin` user to one. Only the 3900 and 12000 switches require this configuration (other Brocade switch models are shipped with the required restrictions configured by default). See "Configuring the Brocade Silkworm 3900" on page 25 and "Configuring the Brocade Silkworm 12000" on page 26.

Verifying the Brocade License

To verify the Brocade license, log into the switch as user `admin` and use the `licenseshow` command, as shown in the following example:

```
brocade:admin> licenseshow
dcRyzyScSedSz0p:
    Web license
    Zoning license
    SES license
    Fabric license
SQQQSyddQ9TRRdUP:
    Release v2.2 license
```

Verifying the Brocade Switch Firmware Version

To verify the firmware version, log into the switch as user `admin` and use the `version` command, as shown in the following example:

```
workstation% telnet brocade1
Trying 169.238.221.224...
Connected to brocade1.acme.com
Escape character is '^]'.
```

```
Fabric OS (tm)  Release v2.6.0d

login: admin
Password:
brocade1:admin> version
Kernel:     5.4
Fabric OS:  v2.6.0d                  <== Firmware Revision
Made on:    Fri May 17 16:33:09 PDT 2002
Flash:      Fri May 17 16:34:55 PDT 2002
BootProm:   Thu Jun 17 15:20:39 PDT 1999
brocade1:admin>
```

Configuring the Brocade Silkworm 3900

To limit the maximum allowed simultaneous `telnet` sessions for the `admin` user to one on the Brocade Silkworm 3900, do the following:

1. Connect to the switch via the `telnet` command and login as `root`.
2. Issue the `sync` command to avoid filesystem corruption:

   ```
   # sync
   ```

3. Edit the `/etc/profile` file to change the `max_telnet_sessions` from `2` to `1` and place the information in a new file. For example:

```
# cd /etc
# sed -e 's/max_telnet_sessions=2/max_telnet_sessions=1/' profile >profile.new
```

4. Distribute the edited `profile` file to both partitions on both central processors. For example:

   ```
   # cp profile.new profile
   # cp profile.new /mnt/etc/profile
   ```

5. Issue the `sync` command again to avoid filesystem corruption:

   ```
   # sync
   ```

Configuring the Brocade Silkworm 12000

To limit the maximum allowed simultaneous `telnet` sessions for the `admin` user to `1` on the Brocade Silkworm 12000, do the following:

1. Connect to the switch via the `telnet` command and login as `root`.
2. Use the `haShow` command to make sure that both central processors are up. This is indicated by the message `Heartbeat Up` within the output of the `haShow` command. If it is not up, wait a few minutes and run `haShow` again to check for the status.
3. Issue the `sync` command on the filesystems to avoid filesystem corruption:

   ```
   # rsh 10.0.0.5 sync
   # rsh 10.0.0.6 sync
   ```

4. Edit the `/etc/profile` file to change the `max_telnet_sessions` from `2` to `1` and place the information in a new file. For example:

```
# cd /etc
# sed -e 's/max_telnet_sessions=2/max_telnet_sessions=1/' profile >profile.new
```

5. Distribute the new profile to both partitions and central processors. For example:

   ```
   # rcp /etc/profile.new 10.0.0.5:/etc/profile
   # rcp /etc/profile.new 10.0.0.5:/mnt/etc/profile
   # rcp /etc/profile.new 10.0.0.6:/etc/profile
   # rcp /etc/profile.new 10.0.0.6:/mnt/etc/profile
   ```

6. Issue the `sync` command again to avoid filesystem corruption:

   ```
   # rsh 10.0.0.5 sync
   # rsh 10.0.0.6 sync
   ```

Changing the Brocade FC Cable Connections

To change Brocade Fibre Channel cable connections used by nodes in the CXFS cluster, do the following:

1. Cleanly shut down CXFS services on the nodes affected by the cable change. Use either the CXFS GUI or the `cmgr` command.
2. Rearrange the cables as required.
3. Restart CXFS services.

4. Reconfigure I/O fencing if required. You must perform this step if I/O fencing is enabled on the cluster and if you added/removed any Brocade switches. You must use the CXFS GUI or the `cmgr`(1M) command to add/remove switches from the CXFS configuration as required.
5. If any CXFS client nodes are connected to a new (or different) Brocade switch, restart CXFS services on those nodes. This will ensure that the CXFS administration servers can correctly identify the Brocade ports used by all clients.

Consult the *CXFS Administration Guide for SGI Infinite Storage* for instructions to configure I/O fencing.

Obtaining CXFS and XVM FLEXlm Licenses

The software licensing used by CXFS is based on the FLEXlm product from Macrovision Corporation. For all supported platforms, a FLEXlm license is required to use CXFS. Perform the procedures in this chapter to satisfy this requirement.

XVM provides a mirroring feature. If you want to access a mirrored volume from a given node in the cluster, you must install a FLEXlm mirroring license on that node. Only those nodes that will access the mirrored volume must be licensed. For information about purchasing this license, see your SGI sales representative.

Obtaining the Host Information Required for the License

When you order CXFS, you will receive an entitlement identifier (ID). You must submit the system host ID, host name, and entitlement ID when requesting your permanent CXFS license. The method used to obtain this information is platform-specific.

AIX Host Information

To obtain the host ID and hostname of the system on which you will run CXFS, execute the following AIX commands:

```
/usr/bin/uname -m
/usr/bin/hostname
```

The `uname -m` command lists the host ID as a 12-digit number. For CXFS license purposes, you will drop the first two digits and the last two digits and supply the remaining eight digits.

For example:

```
aix# /usr/bin/uname -m
000276513100
aix# /usr/bin/hostname
cxfsaix3
```

When you are asked for the license manager host ID, you would provide the following information:

- Host ID: `02765131`

  That is, the output from the `uname -m` command minus the first two digits (`00`) and the last two digits (`00`)

- Hostname: `cxfsaix3`

You must have a separate license for each host on which CXFS is installed.

Linux 32-bit Host Information

To obtain the host ID and hostname of the system on which you will run CXFS, execute the following Linux commands:

```
/sbin/ifconfig eth0
/bin/hostname
```

The host ID is the value for the `HWaddr` field, minus the colons.

For example:

```
[root@linux32 root]# /bin/hostname
cxfslinux
[root@linux32 root]# /sbin/ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:E0:81:24:77:D1
          inet addr:128.162.240.135  Bcast:128.162.240.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17941247 errors:2 dropped:0 overruns:0 frame:2
          TX packets:16373834 errors:10 dropped:0 overruns:0 carrier:10
          collisions:0 txqueuelen:100
          RX bytes:1539518033 (1468.1 Mb)  TX bytes:1573522873 (1500.6 Mb)
          Interrupt:19 Base address:0xb880 Memory:fe3fe000-fe3fe038
```

In this case, the host ID is `00E0812477D1`

To obtain the physical CPU count, find the information in the `/proc/cpuinfo` file as follows:

```
[root@linux32 root]# cat /proc/cpuinfo
```

To determine the number of physical CPUs, count the number of `processor` lines and divide by the value for `siblings`. A value of `1` for `siblings` indicates that all of the processors are physical; a value of `2` indicates that the processors are hyperthreaded, and therefore the number of physical CPUs is half of the number of `processor` lines displayed.

For example (with truncated output indicated by `...`) :

```
[root@linux32 root]# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Xeon(TM) CPU 2.00GHz
stepping        : 7
cpu MHz         : 1999.795
cache size      : 512 KB
physical id     : 0
siblings        : 2
...

processor       : 1
vendor_id       : GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Xeon(TM) CPU 2.00GHz
stepping        : 7
cpu MHz         : 1999.795
cache size      : 512 KB
physical id     : 0
siblings        : 2
...

processor       : 2
vendor_id       : GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Xeon(TM) CPU 2.00GHz
stepping        : 7
cpu MHz         : 1999.795
cache size      : 512 KB
physical id     : 3
```

```
siblings        : 2
...

processor       : 3
vendor_id       : GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Xeon(TM) CPU 2.00GHz
stepping        : 7
cpu MHz         : 1999.795
cache size      : 512 KB
physical id     : 3
siblings        : 2
...
```

The above output shows that there are four `processor` lines, but they are hyperthreaded because the value of `siblings` is `2`. Therefore, the physical CPU count is 2 (4/2=2).

When you are asked for the license manager host ID, provide this information. You must have a separate license for each host on which CXFS is installed.

Mac OS X Host Information

To obtain the hostname of the system on which you run CXFS, execute the following Mac OS X command in a terminal to obtain the hostname and host ID from the onboard Ethernet interface:

```
/bin/hostname -s
/sbin/ifconfig en0 | grep ether
```

For example:

```
macosx# /bin/hostname -s
cxfsmac1
macosx# /sbin/ifconfig en0 | grep ether
ether 00:03:93:cc:3a:e4
```

The Ethernet address is `00:03:93:cc:3a:e4`, so the host ID is `000393cc3ae4`.

Alternatively, the Ethernet address is also displayed with the interface by using the following menu selection:

System Preferences
> Network

Note: Ensure that the address is for the onboard Ethernet interface and not any additional interface cards.

Solaris Host Information

To obtain the host ID and hostname of the system on which you will run CXFS, execute the following Solaris commands:

```
/usr/bin/hostid
/usr/bin/hostname
```

For example:

```
solaris# /usr/bin/hostid
830dad77
solaris# /usr/bin/hostname
cxfssun2
```

When you are asked for the license manager host ID, provide this information. You must have a separate license for each host on which CXFS is installed.

Windows Host Information

FLEXlm requires that you supply the Ethernet (MAC) address in order to generate the FLEXlm license. This address is known as the *Physical Address* in Windows. You can obtain this information in one of the following ways:

- View the network adapter properties in the **Windows Control Panel**
- Open a command prompt window and run the following command:

  ```
  C:\> ipconfig /all
  ```

If the machine has more than one network interface, you should use the Physical Address of the private network interface.

Note: If you are upgrading a Windows node (such as from Windows 2000 to Windows XP), you must obtain a new license.

Obtaining and Installing the Licenses

Along with your entitlement number, you will receive a URL to a key generation page. To obtain your permanent CXFS and XVM licenses, follow the instructions on the key generation page. After the required information is provided, a key will be generated and displayed on the webpage along with installation instructions.

See also:

- "FLEXlm License Verification for AIX" on page 39
- "FLEXlm License Verification for Linux 32-bit Platforms" on page 54
- "FLEXlm License Verification for Solaris" on page 92
- "Configuring the FLEXlm License for Windows" on page 150

For More Information

For more information about licensing, see the following webpage:

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

If you cannot use the web key generation page, you can contact the SGI order desk at 800 800 4SGI (800 800 4744).

For more information on FLEXlm, you may order the *Flexible License Manager End User Manual* from Macrovision Corporation.

AIX Platform

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

CXFS on AIX

This section contains the following information about CXFS on AIX:

Requirements Specific to AIX

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

- An AIX 5L version 5.1 MR 4 operating system (64-bit kernel mode) APAR number IY42428
- IBM FC6228 2-Gbit Fibre Channel host bus adapters (HBAs)

- One or more of the following IBM hardware platforms:
  - pSeries 610
  - pSeries 620
  - pSeries 630
  - pSeries 640
  - pSeries 650
  - pSeries 660
  - pSeries 670
  - pSeries 680
  - pSeries 690

Note: This release was tested with the following:

- pSeries 610 Model 6E1
- pSeries 630 Model 6E4

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.

CXFS Commands Installed 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/xvm`
- `/usr/cxfs_cluster/bin/xvmprobe`

These commands provide all of the services needed to include an AIX node in a CXFS cluster. The `lslpp` output lists all of the software added; see "AIX Installation Overview" on page 44.

For more information, see the `cxfs_client`, `cxfs_info`, `xvm`, and `xvmprobe` man pages.

Log Files on AIX

The `cxfs_client` command creates a `/var/tmp/cxfs_client` log file. (There is no `/var/cluster` log on AIX nodes.) To rotate this log file, use the `-z` option in the `/usr/cxfs_cluster/bin/cxfs_client.options` file; see the `cxfs_client.options` man page for details.

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

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

For information about the log files created on administration nodes, see the *CXFS Administration Guide for SGI Infinite Storage*.

Limitations on AIX

CXFS on AIX has the following limitations:

- The block size supported is 4 KB, which is the XFS default block size.
- Unlike IRIX, 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.

- The 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.

Maximum CXFS Filesystem Size and Offset Within a File on AIX

The maximum size of a CXFS filesystem on AIX is 2^{64} (about 18 million TB). The maximum offset within a file is 1 TB. An attempt to write beyond this limit will result in an `Invalid argument` or a `File too large` error.

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.

FLEXlm License Verification for AIX

Use the `cxfslicense` command with the `-d` option to verify that the FLEXlm licenses have been installed properly.

If the CXFS license is properly installed, you will see the following:

```
# /usr/cxfs_cluster/bin/cxfslicense -d
XVM_AIX license granted.
CXFS_AIX license granted.
```

If you do not have the CXFS license properly installed, you will see the following error on the console when trying to run CXFS:

```
Starting CXFS services> ....
CXFS not properly licensed for this host.  Run
   '/usr/cxfs_cluster/bin/cxfslicense -d'
for detailed failure information. After fixing the
license, please run '/usr/cxfs_cluster/bin/cxfs_cluster restart'.
```

An error such as the following example will appear in the `SYSLOG` file (line breaks added here for readability):

```
Jan 25 10:24:03 ncc1701:Jan 25 10:24:03 cxfs_client:
cis_main FATAL: cxfs_client failed the CXFS license check.
Use the cxfslicense command to diagnose the license problem
```

Host Bus Adapter Installation and Configuration for AIX

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

You are required to have a second network interface that must be used for the private metadata network. If you do not already have a second interface installed, you must install a network interface card. You may wish to install that card at this time.

Preinstallation Steps for AIX

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 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.

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 Nodes" on page 40
- "Verifying the Private and Public Network for AIX" on page 43

Adding a Private Network for AIX Nodes

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

Note: A private network is required for use with CXFS. Only the private network is used by CXFS for heartbeat/control messages.

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 5L V5.1 Installation Guide*.

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 "Hostname Resolution and Network Configuration Rules for All Platforms" on page 12 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 using `ifconfig`. 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 Steps for AIX

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

- "AIX Installation Overview"
- "Verifying the AIX Installation " on page 46

AIX Installation Overview

Installing the CXFS client CD 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 `README` file for the AIX platform to learn about any late-breaking changes in the installation procedure.
2. Verify that the node has been upgraded to AIX 5.1L according to the *AIX 5L V5.1 Installation Guide*. Use the following command to display the currently installed system:

   ```
   aix# uname -rv
   1 5
   ```

   This output indicates `1` is the revision and `5` is the version.
3. Insert the *CXFS MultiOS Client 3.1* CD-ROM.
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 3.1.0.0                       # 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 3.1.0.0


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

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

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

Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
SGIcxfs-aix5L               3.1.0.0         USR         APPLY       SUCCESS
SGIcxfs-aix5L               3.1.0.0         ROOT        APPLY       SUCCESS
```

6. Verify that the CXFS license key has been installed. See Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29. For example:

```
# /usr/cxfs_cluster/bin/cxfslicense -d
CXFS license granted.
```

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              3.1.0.0    C     F    CXFS CLIENT for AIX


State codes:
 A -- Applied.
 B -- Broken.
 C -- Committed.
 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
```

Postinstallation Steps for AIX: Creating the I/O Fencing File

I/O fencing is required to protect data integrity for AIX nodes. To use I/O fencing, you must create the AIX `/etc/fencing.conf` file, which lists the worldwide 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.

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.

The `/etc/fencing.conf` file enumerates the WWPN for all of the host bus adapters (HBA) that will be used to mount a CXFS filesystem. The `/etc/fencing.conf` file must contain a simple list of WWPNs, as a 64-bit hexadecimal number, one per line.

If you are not completely certain which number you should use, 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 Brocade Fibre Channel 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 Brocade Fibre Channel 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 Brocade Fibre Channel switch, your `/etc/fencing.conf` file should appear as follows:

   ```
   2000000173002c0b
   ```

5. After the AIX node is added to the cluster (see Chapter 10, "Cluster Configuration" on page 159), enable the fencing feature by using the CXFS GUI or `cmgr` command on a CXFS administration node; for more information, see the *CXFS Administration Guide for SGI Infinite Storage*.

Manual CXFS Start/Shutdown 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 process manually on your AIX node, enter the following:

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

To stop CXFS processes manually, enter the following:

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

Software Maintenance for AIX

This section contains information about upgrading and modifying the CXFS software on an AIX system.

Upgrading the CXFS Software on an AIX System

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         3.1.0.0    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 Steps for AIX" on page 44.

Modifying the CXFS Software on an AIX System

You can modify the CXFS client service (`/usr/cxfs_cluster/bin/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.

Linux 32-bit Platforms

CXFS supports a client-only node running the Linux operating system on supported 32-bit platforms.

Note: On Linux 32-bit systems, the use of XVM is supported only with CXFS; XVM does not support local Linux 32-bit disk volumes.

This chapter contains the following sections:

CXFS on Linux 32-bit Platforms

This section contains the following information about CXFS on Linux 32-bit systems:

- "Access Control Lists and Linux 32-bit Platforms" on page 54

Requirements Specific to Linux 32-bit Platforms

In addition to the items listed in "Requirements" on page 6, using a Linux 32-bit node to support CXFS requires the following:

- One of the following operating systems:
  - Red Hat Linux 8.0
  - Red Hat Linux 9

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

- A choice of at least one Fibre Channel host bus adapter (HBA):
  - QLogic 2200, QLogic 2310, or QLogic 2342 HBA
  - JNI FCX-6562-N 2 Gb 133 MHz PCI-X-to-Fibre Channel HBAs

    Note: 1-Gbit HBAs and Sbus HBAs are not supported with any Red Hat operating system.

  - LSI Logic LS17202XP-LC dual channel PCI-X HBA
- A CPU of the following type:
  - Intel Pentium III
  - Intel Pentium 4
  - Advanced Micro Devices AMD Athlon
  - Advanced Micro Devices AMD Duron

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)

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 32-bit CXFS filesystem, this is not recommended.

CXFS Commands Installed on Linux 32-bit Platforms

The following commands are shipped as part of the CXFS Linux 32-bit package:

- `/usr/cluster/bin/cxfs_client` (the CXFS client service)
- `/usr/cluster/bin/cxfs_info`
- `/usr/cluster/bin/cxfslicense`
- `/sbin/xvm`

These commands provide all of the services needed to include a Linux 32-bit node in a CXFS cluster. The `rpm` output lists all software added; see "Installation Overview " on page 61.

For more information, see the `cxfs_client` and `xvm` man pages.

Log Files on Linux 32-bit Platforms

The `cxfs_client` command creates a `/var/log/cxfs_client` log file. (There is no `/var/cluster` log on Linux 32-bit nodes.) This file is rotated by default.

The Linux 32-bit 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 Infinite Storage*.

Limitations and Considerations for Linux 32-bit Platforms

CXFS for Linux 32-bit has the following limitations and considerations:

- The maximum block size supported is 4 KB, determined by the kernel page size. (XFS uses a default block size of 4 KB unless overridden by an administrator to a different blocksize value, for example 1 KB or 2 KB.)
- Due to Linux kernel limitations, CXFS filesystems cannot be mounted with the `inode64` mount option.
- CXFS filesystems with XFS version 1 directory format cannot be mounted on Linux 32-bit nodes.

Maximum CXFS Filesystem Size and Offset Within a File on Linux 32-bit Platforms

The maximum size of a CXFS filesystem on a supported Linux 32-bit client is 2 TB; this is due to a Linux 32-bit kernel restriction. The maximum offset within a file is 16 TB, using Large File Support (`O_LARGEFILE`). An attempt to write beyond this limit will result in an `Invalid argument` or a `File too large` error.

Access Control Lists and Linux 32-bit Platforms

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.

Red Hat Linux file utilities do not provide ACL support, so commands such as `ls` and `cp` will not show or preserve ACLs. However, the commands in the `acl` package will allow manipulation of the ACLs of files on CXFS filesystems.

FLEXlm License Verification for Linux 32-bit Platforms

Use the `cxfslicense` command with the `-d` option to verify that the FLEXlm licenses have been installed properly.

If the CXFS license is properly installed, you will see the following:

```
[root@linux32 root]# /usr/cluster/bin/cxfslicense -d
CXFS license granted.
```

If you do not have the CXFS license properly installed, you will see the following error on the console and in the `cxfs client` logs when trying to run CXFS:

```
May 12 14:40:17 cxfs_client: cis_main FATAL: cxfs_client failed the CXFS
license check. Use the cxfslicense command to diagnose the license
problem.
May 12 14:40:17 cxfs_client: FATAL: aborting on fatal error
```

Host Bus Adapter Installation and Configuration for Linux 32-bit Platforms

This section provides an overview of the Fibre Channel host bus adapter (HBA) installation information for Linux 32-bit nodes.

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

The driver requirements are as follows:

- LSI Logic card: the drivers are supplied with the Red Hat Linux kernel. The module name is `mptscsih.o`.
- QLogic card: the drivers are supplied with the Red Hat Linux kernel. The module names are `qla2200.o` and `qla2300.o`.
- JNI card: the drivers are supplied on the JNI website:

  http://www.jni.com

You must build the AMCC JNI module from what is provided by AMCC JNI, following their instructions.

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: add the following line to the `/etc/modules.conf` file prior to installing the kernel RPM:

  ```
  alias scsi_hostadapter qla2200
  ```

  If a `scsi_hostadapter` line already exists, you can add a new line such as the following:

  ```
  alias scsi_hostadapter1 qla2200
  ```

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

- **Manual** method: enter the following command:

```
# mkinitrd -f -v --with=qla2200 /boot/initrd-2.4.20-19.8.sgi1.img 2.4.20-19.8.sgi1
```

  Where:

  `-f` overwrites any existing image of the same name

  `-v` displays verbose output

  `qla2200` is the name of the module to include

  `/boot/2.4.20-19.8.sgi1.img` is the created image

  `2.4.20-19.8.sgi1` is the output of the `uname -r` command for the corresponding kernel

You should then verify the appropriate `initrd` information:

- If using the LILO loader, do the following:

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

     ```
     initrd=/boot/2.4.20-19.8.sgi1.img
     ```

  2. Rerun LILO.

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

  ```
  initrd /2.4.20-19.8.sgi1.img
  ```

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 32-bit Platforms

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.

This section provides an overview of the steps that you will perform on your Linux 32-bit nodes prior to installing the CXFS software. It contains the following sections:

- "Adding a Private Network for Linux 32-bit Nodes" on page 57
- "Verifying the Private and Public Networks for Linux 32-bit Nodes" on page 60

Adding a Private Network for Linux 32-bit Nodes

The following procedure provides an overview of the steps required to add a private network to the Linux 32-bit system.

Note: A private network is required for use with CXFS. Only the private network is used by CXFS for heartbeat/control messages.

You may skip some steps, depending upon the starting conditions at your site. For details about any of these steps, see the Red Hat Linux 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 "Hostname Resolution and Network Configuration Rules for All Platforms" on page 12.

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 32-bit distribution manual. To verify that the private interface is operational, issue the following command:

```
[root@linux32 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 32-bit

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. 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 Red Hat documentation about PAM and the `hosts.equiv` man page.

Verifying the Private and Public Networks for Linux 32-bit Nodes

For each private network on each Linux 32-bit 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@linux32 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, follow these steps:

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

```
[root@linux32 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.

Repeat this procedure on each node.

Client Software Installation Steps for Linux 32-bit Platforms

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.

Installation Overview

Installing the CXFS client CD for Linux 32-bit requires approximately 50–200 MB of space, depending upon the packages installed at your site.

To install the required software on a Linux 32-bit node, SGI personnel will do the following:

1. Read the `README` file for the Linux 32-bit platform to learn about any late-breaking changes in the installation procedure.

2. Verify that the node is running a supported Linux 32-bit distribution, according to the CXFS for Linux 32-bit release notes. For example, use the following command to display the currently installed Red Hat system:

```
[root@linux32 root]# cat /etc/redhat-release
Red Hat Linux release 9 (Shrike)
```

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

3. Insert and mount the *CXFS MultiOS Client 3.1 Kernel distribution for Linux 32 bit Client* CD-ROM.

4. Install the appropriate kernel RPM for the system, according to the information about upgrading the kernel in the operating system documentation.

Caution: You should not use the `nodeps` or `force` flag during RPM installations.

For example, to install the kernel on an Intel Pentium III dual-processor machine, enter the following (the RPM version numbers may differ from the released product):

```
[root@linux32 cdrom]# rpm -ivh kernel-smp-2.4.20-13.7.sgi4.i686.rpm
Preparing...                ########################################### [100%]
   1:kernel-smp             ########################################### [100%]
```

5. Install the `xfs-modules` RPM. The `xfs` modules package architecture (`i686` or `athlon`) and type (`smp`, `non-smp`, or `bigmem`) must match the kernel package.

```
[root@linux32 cdrom]# rpm -ivh xfs-modules-smp-1.3pre1-2.4.20_13.7.sgi4_sgi3.i686.rpm
Preparing...                ########################################### [100%]
   1:xfs-modules            ########################################### [100%]
```

6. Configure the new kernel into the bootloader according to the instructions in the operating system documentation and make it the default kernel:

   - If you are using the LILO loader, verify that `/etc/lilo.conf` contains a stanza with the correct entries for the newly installed kernel and then run the

following command to install the boot loader and verify that all information is placed appropriately:

```
[root@linux32 cdrom]# lilo -v
```

- If you are using the GRUB loader, verify that `/boot/grub/grub.conf` contains a stanza with the correct entries for the newly installed kernel.

7. Install the `libacl` and `libattr`, and XFS filesystem packages (line break added here for readability):

```
[root@linux32 cdrom]# rpm -Uvh acl-2.2.7-0.i386.rpm attr-2.4.1-0.i386.rpm dmapi-2.0.6-0.i386.rpm \
xfsdump-2.2.8-0.i386.rpm xfsprogs-2.4.5-0.i386.rpm libacl-2.2.7-0.i386.rpm libattr-2.4.1-0.i386.rpm
Preparing...                ########################################### [100%]
   1:libacl                 ########################################### [ 14%]
   2:libattr                ########################################### [ 29%]
   3:acl                    ########################################### [ 43%]
   4:attr                   ########################################### [ 57%]
   5:dmapi                  ########################################### [ 71%]
   6:xfsdump                ########################################### [ 86%]
   7:xfsprogs               ########################################### [100%]
```

8. Insert and mount the *CXFS MultiOS Client 3.1* CD-ROM.

9. Install the CXFS kernel modules that correspond to the kernel installed in step 4. The `cxfs` modules package architecture (`i686` or `athlon`) and type (`smp`, `non-smp`, or `bigmem`) must match the kernel package. The following is an example of installing CXFS kernel modules:

```
[root@linux32 cdrom]# rpm -ivh cxfs-modules-smp-2.5-2.4.20_13.7.sgi4_sgi16.i686.rpm
Preparing...                ########################################### [100%]
   1:cxfs-modules           ########################################### [100%]
```

The following is an example of installing user-space packages (line break added for readability):

```
[root@linux32 cdrom]# rpm -Uvh cxfs_client-3.1-RH9_sgi6.i386.rpm \
cxfs_util-3.1-RH9_sgi6.i386.rpm \
xvm-cmds-3.1-RH9_sgi8.i386.rpm

Preparing...                ########################################### [100%]
   1:cxfs_util              ########################################### [ 33%]
   2:cxfs_client            ########################################### [ 66%]
   3:xvm-cmds               ########################################### [ 100%]
```

10. Edit the `/etc/cluster/config/cxfs_client.options` file as necessary. See the "Software Maintenance: Modifying the CXFS Software on a Linux 32-bit Platforms" on page 65 and the `cxfs_client`(1M) man page.

11. Create the `/etc/flexlm` directory:

    ```
    [root@linux32 cdrom]# mkdir -p /etc/flexlm
    ```

12. Copy your key to `/etc/flexlm/license.dat`. See Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29.

13. Verify that your key has been installed. For example:

    ```
    [root@linux32 root]# /usr/cluster/bin/cxfslicense -d
    CXFS license granted.
    ```

14. Reboot the system with the newly installed kernel:

    ```
    [root@linux32 root]# reboot
    ```

Verifying the Linux 32-bit Installation

Use the `uname -r` command to ensure the kernel installed in step 9 above is running. For example:

```
[root@linux32 root]# uname -r
2.4.18-27.7.x.sgismp
```

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

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

```
[root@linux32 root]# rpm -q cxfs-modules cxfs_util cxfs_client \
xvm-cmds

cxfs-modules-3.1-sgi
cxfs_util-3.1-sgi
cxfs_client-3.1-sgi
xvm-cmds-3.1-sgi
```

Manual CXFS Startup/Shutdown for Linux 32-bit Platforms

The `/etc/init.d/cxfs_client` 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 processes manually on your Linux 32-bit node, enter the following:

```
[root@linux32 root]# /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:

```
[root@linux32 root]# /etc/init.d/cxfs_client stop
Stopping cxfs client:                                      [  OK  ]
```

To see the current status of the CXFS processes, use the `status` argument. For example, the following output shows that the service is running:

```
[root@linux32 root]# /etc/init.d/cxfs_client status
cxfs_client (pid 3226) is running...
```

For example, if the service is stopped:

```
[root@linux32 root]# /etc/init.d/cxfs_client status
cxfs_client is stopped
```

Software Maintenance: Modifying the CXFS Software on a Linux 32-bit Platforms

You can modify the CXFS client service (`/usr/cluster/bin/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.

The first line in the `cxfs_client.options` file must contain the options you want `cxfs_client` to process; you cannot include a comment as the first line.

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

```
[root@linux32 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
```

Mac OS X Platform

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

CXFS on Mac OS X

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

Requirements Specific to Mac OS X

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

- Mac OS X operating system 10.2.8
- One or the following single- or dual-processor Apple Computer hardware platforms:
  - G4 Power Mac
  - G4 Xserve
  - G5 Power Mac
  - G5 Xserve
- One Astera Technologies Rhino host bus adapter (HBA) running Astera HDFC Driver v2.1.0:
  - HA3120F single-port 2-Gbit optical HBA
  - HA3200F dual-port 2-Gbit optical HBA

Note: The Apple Fibre Channel PCI card is not currently supported by CXFS.

CXFS Commands Installed on Mac OS X

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

- `/usr/cluster/bin/cxfs_client` (the CXFS client daemon)
- `/usr/cluster/bin/cxfs_info`
- `/usr/cluster/bin/cxfslicense`
- `/usr/cluster/bin/install-cxfs`
- `/usr/cluster/bin/uninstall-cxfs`
- `/usr/cluster/bin/XVMprobe`
- `/Library/StartupItems/cxfs/cxfs`

The `cxfs_client` and `XVMprobe` commands are needed to include a Mac OS X node in a CXFS cluster. The `cxfs_info` command reports the current status of this node in the CXFS cluster. For more information on these commands, see the `cxfs_client`, `cxfs_info` and `XVMprobe` man pages.

You can use the `cxfslicense` command to validate a CXFS or XVM FLEXlm license.

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.

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 supports filesystem block sizes of 4 KB only.
- Mac OS X does not support the `inode64` mount option.
- Mac OS X is unable to memory map a file larger than 2 GB.

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.

The hostname is normally specified using the following menu selection:

System Preferences
> Sharing
> Computer Name

The hostname entry in `/etc/hostconfig` is `HOSTNAME=-AUTOMATIC-`, which will normally result in the hostname specified for the machine with the following domain:

```
.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 Rendezvous, 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 other hosts in the cluster are specified in the Mac OS X node's `/etc/hosts` file.

Mapping User and Group Identifiers

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, in particular 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 UNIX mode bits (read, write, and execute) and optionally an access control list (ACL). For more information, see the `chmod` and `chacl` man pages on an IRIX or Linux 64–bit node.

Mac OS X supports UNIX mode bits but has no native support for ACLs, so they cannot be viewed or modified from a Mac OS X node. However, they are enforced by the Mac OS X node if applied to files by other nodes in the cluster.

When using `tar`, `cpio`, or other third-party backup tools from a Mac OS X node, the ACL will be lost because those tools are unaware of the ACL on the file.

FLEXlm License Verification for Mac OS X

Use the `cxfslicense` command with the `-d` option to verify that the FLEXlm licenses have been properly installed. If the CXFS license is properly installed, you will see the following:

```
macosx# /usr/cluster/bin/cxfslicense -d
CXFS license granted.
```

If you do not have the CXFS license properly installed, you will see the following error in the `/var/log/cxfs_client` file:

```
cxfs_client: cis_main FATAL: cxfs_client failed the CXFS license check.
Use the cxfslicense command to diagnose the license problem.
cxfs_client: FATAL: exiting on fatal error
```

Host Bus Adapter Installation and Configuration for Mac OS X

This section provides an overview of the Astera Technologies Fibre Channel host bus adapter (HBA) installation and verification for Mac OS X nodes:

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

Installing the Astera Technologies HBA

Do the following:

1. Install the Astera Technologies Rhino-3000 HBA into a spare PCI-X or PCI slot in the Mac OS X node, as per the manufacturer's instructions. Do not connect the Fibre Channel cable to the HBA at this time.
2. Open the `AsteraHDFCDrive.pkg` file that came with the HBA, or was downloaded from http://www.asteratech.com, and follow the instructions.
3. Reboot the node.

Installing and Running the JumanJi Configuration GUI

Do the following:

1. Open the `JumanJi.pkg` that came with the HBA, or was downloaded from http://www.asteratech.com, and follow the instructions. This will install JumanJi into `/Applications/Astera Technologies`.
2. Run JumanJi. You will see the HBA listed with crosses through it because there is no Fibre Channel connection to the RAID. Ensure that the driver version in the information Window is supported by CXFS.

   Figure 7-1 shows that the HBA has two ports that are not connected.

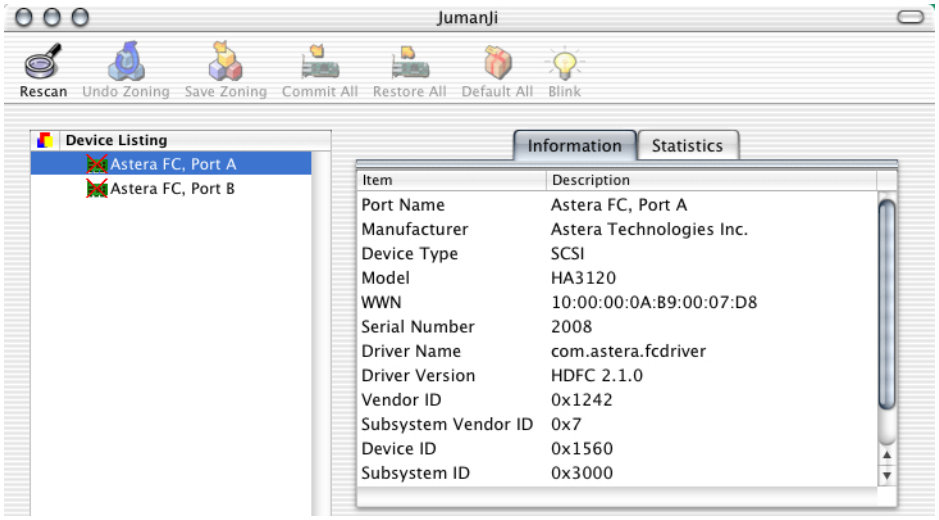

Figure 7-1 Two Ports, Not Connected

3. Connect the Fibre Channel cable to the HBA. If the HBA has more than one port, connect to only one port at this time.
4. Press **Rescan**, which will cause JumanJi to display the targets and logical units (LUNs) in the storage area network (SAN).

   Figure 7-2 shows an example where the SAN has two targets, each with five LUNs.

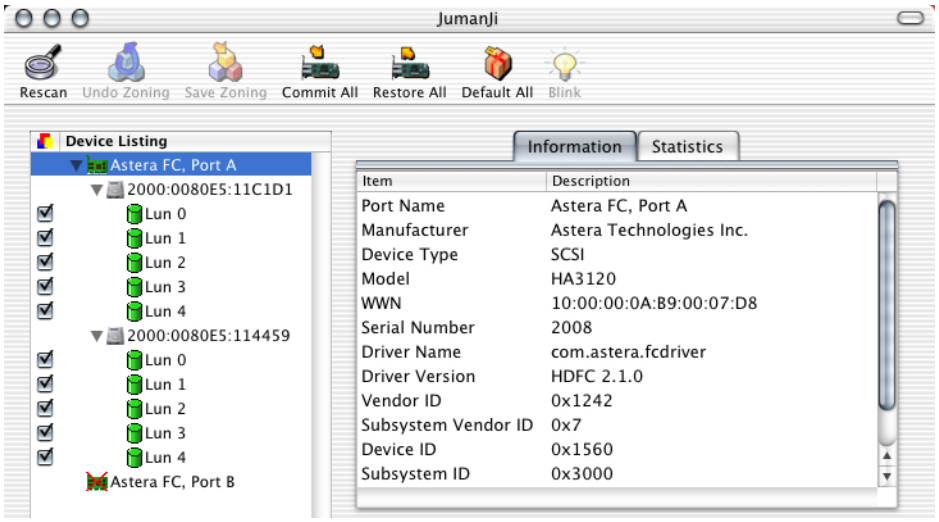

Figure 7-2 Two Targets

5. Select the **Statistics** pane to confirm that the HBA has detected the expected link speed, number of targets, and LUNs.

   Figure 7-3 shows an example where the SAN is 1 GB and there are 10 LUNs.

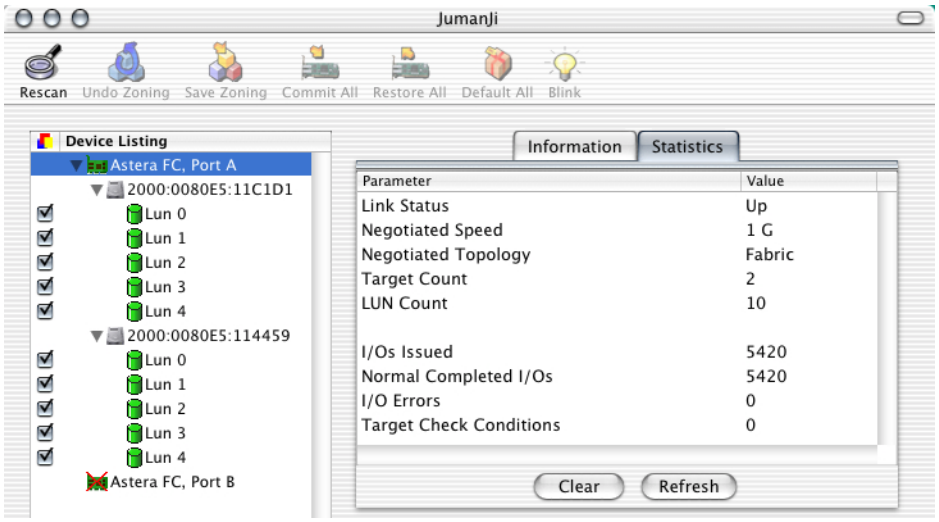

Figure 7-3 Statistics

6. The default settings for the HBA are the most appropriate for CXFS. To confirm the default settings, do the following:

   a. Select the padlock in the lower left-hand corner.

   b. Select **Defaults**.

   Figure 7-4 shows an example.

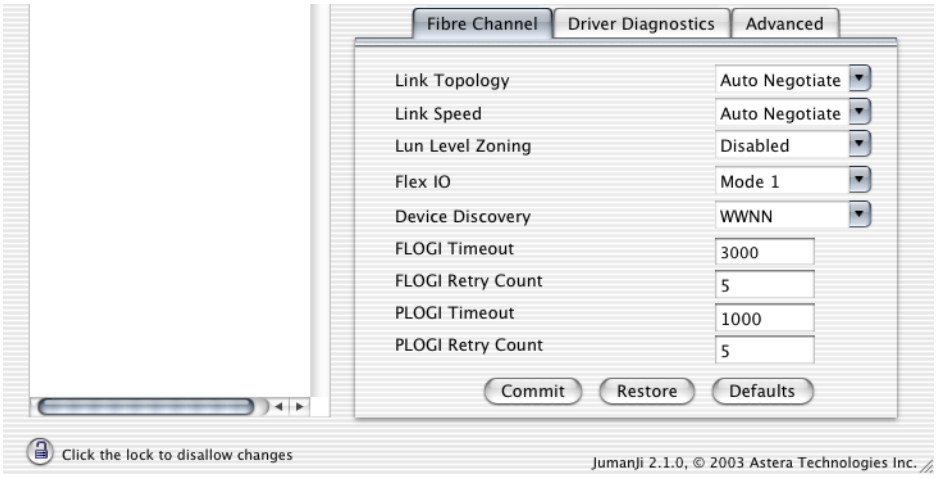

Figure 7-4 Defaults

If you change any settings, press **Commit** and reboot the node to make the settings take affect.

For more information, see the *JumanJi User Guide* from Astera Technologies.

Using the TP9300, TP9400, or TP9500 with Mac OS X

If you use a TP9300, TP9400, or TP9500 RAID with Mac OS X, there will be a pseudo LUN (LUN 31) that is used for the RAID configuration.

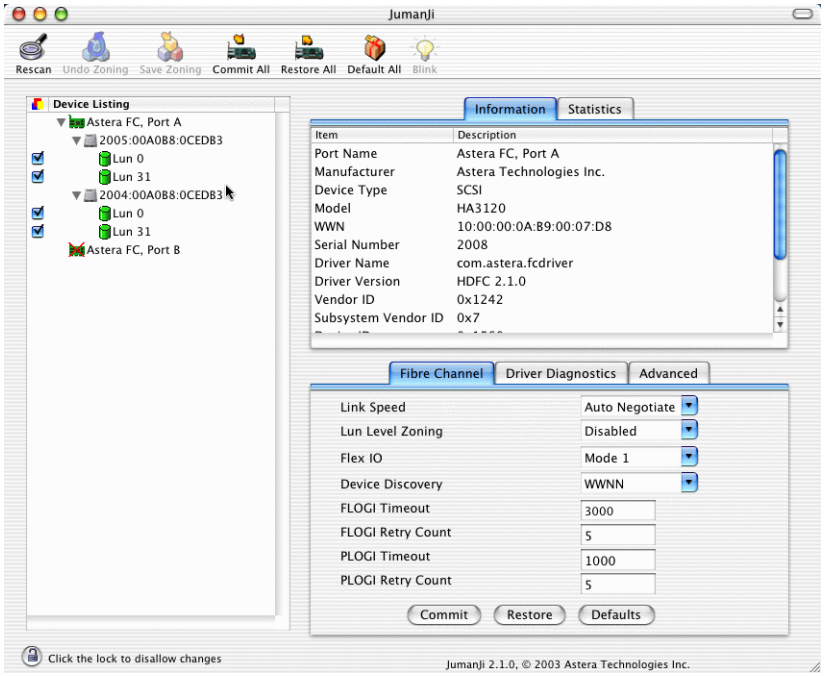

Figure 7-5 Lun Level Zoning on a TP9300, TP9400, or TP9500 RAID

However, when this LUN exists, the Mac OS X CXFS driver will not load properly. You must deselect path to this LUN from JumanJi.

Do the following:

1. Deselect LUN 31 from all active ports
2. Select:

   Tools
   > Save Zoning Changes

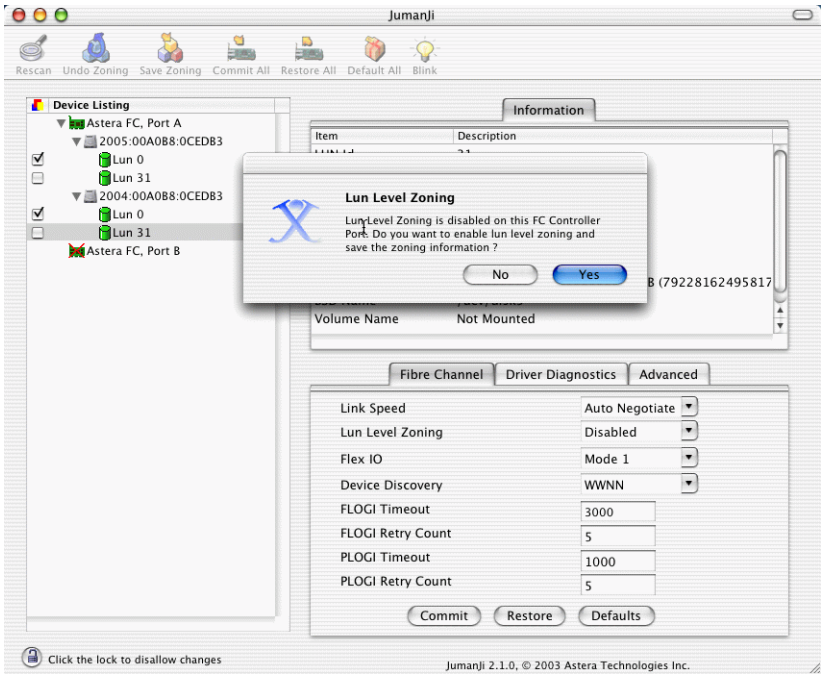

Figure 7-6 LUN 31 Disabled

3. Click **Yes** when prompted to enable **Lun Level Zoning**.

Configuring Two or More Astera HBA Ports

If the Mac OS X node has a dual-port HBA and/or multiple HBAs, then the JumanJi tool will list all the ports that are attached to the SAN. This may result in multiple

paths to the same LUN. By default, only the first mapping will be used, but it is possible to select and deselect LUNs so that different paths are taken.

Caution: Deselecting all paths to a particular LUN will cause CXFS to fail to mount any filesystems that span that LUN. This can be difficult to diagnose because the mapping from filesystem to XVM volumes to LUNs can be convoluted.

If an XVM volume has two stripes, each stripe on a different LUN, then it is possible to configure JumanJi so that I/O to one stripe takes a different path than I/O to the other stripe. In theory, this can improve I/O performance if the I/O and stripe width are of an appropriate size. This can be confirmed by monitoring the ports on the Fibre Channel switch and observing the amount of I/O via each port connected to the Mac OS X HBA ports.

Note: You should not connect to the Fibre Channel switch using the `telnet` command for any significant length of time if fencing is enabled, because this will prevent any fencing changes from occurring.

Any changes to the LUN mapping must be saved and will take affect on reboot.

Preinstallation Steps for Mac OS X

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.

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 83
- "Disabling Power Save Mode for Mac OS X" on page 83

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.

Note: A private network is required for use with CXFS. Only the private network is used by CXFS for heartbeat/control messages.

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 69) 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 as per the manufacturer's instructions.
5. Configure the second network interface card 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 interface may vary.

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

7. *(Optional)* Enable the `rshd` daemon and edit the `~root/.rhosts` file if you want to use the connectivity diagnostics provided with CXFS.

Caution: Mac OS X by default enables `ssh` connections and not `rsh` connections, because `rsh` connections are less secure. You should consider the security implications of enabling `rsh` access to the Mac OS X node before performing this action.

To enable `rshd` on Mac OS X, edit the `/etc/inetd.conf` file and uncomment the following lines:

```
ftp     stream  tcp     nowait  root    /usr/libexec/tcpd               ftpd -l
login   stream  tcp     nowait  root    /usr/libexec/tcpd               rlogind
shell   stream  tcp     nowait  root    /usr/libexec/tcpd               rshd
telnet  stream  tcp     nowait  root    /usr/libexec/tcpd               telnetd
```

Then restart the `inetd` process by entering the following:

```
macosx% sudo kill -HUP 'cat /var/run/inetd.pid'
```

Edit the `root` user's `.rhosts` file to allow `root` users on other CXFS nodes, notably CXFS administration nodes, to access this machine using `rsh` without a password. For example, if `cxfsmds1` is the metadata server, add the following to the `.rhosts` file:

```
macosx% sudo vi ~root/.rhosts cxfsmds1 root
macosx% sudo chmod 600 ~root/.rhosts
```

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`. Likewise, the CXFS diagnostic tool `cxfsdump` uses `rsh` from an administration node to collect information from the Mac OS X node.

Verifying the Private and Public Networks for Mac OS X

Verify each interface 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 power save mode:

System Preferences
> Energy Saver
> Put the computer to sleep when it is inactive for
> Never

Client Software Installation Steps for Mac OS X

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

Installing the CXFS client 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 `ReadMe` file for the Mac OS X platform to learn about any late-breaking changes in the installation procedure.
2. Verify that the node is running Mac OS X operating system 10.2.8 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 `6.8` because the Mac OS X kernel, Darwin, is version 6.*X* in Mac OS X 10.2.*X*.
3. Insert the *CXFS MultiOS Client 3.1* CD-ROM.
4. Using the **Finder**, open `macosx/cxfs.pkg` from the CD-ROM. This will launch the installation application, which will do the following:

   - Display the `ReadMe` file
   - 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. Verify that the CXFS license key has been installed. See Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29.

   For example:

   ```
   macosx# /usr/cluster/bin/cxfslicense -d
   CXFS license granted.
   ```
6. Restart the machine.

Manual CXFS Startup/Shutdown 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 processes required to run CXFS.

To start up CXFS processes manually on your Mac OS X node, enter the following:

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

To stop CXFS processes manually, enter the following:

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

To prevent the automatic startup of CXFS on boot, or to perform any other actions when starting or stopping CXFS, edit the `/Library/StartupItems/cxfs/cxfs` file.

Software Maintenance for Mac OS X

This section contains the following:

- "Upgrading the CXFS Software on a Mac OS X System"
- "Modifying the CXFS Software on a Mac OS X System" on page 85
- "Removing the CXFS Software from a Mac OS X System " on page 86

Upgrading the CXFS Software on a Mac OS X System

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 on a Mac OS X System

You can modify the CXFS client service (`/usr/cluster/bin/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.

The first line in the `cxfs_client.options` file must contain the options you want `cxfs_client` to process; you cannot include a comment as the first line.

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 from a Mac OS X System

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.

Solaris Platform

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

CXFS on Solaris

This section contains the following information about CXFS on Solaris:

Requirements Specific to Solaris

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

- Solaris operating system:
  - Solaris 8 and patch 108528–22 (July 7, 2003)
  - Solaris 9 and patch 112233–06 (March 03)
- One to four AMCC JNI FibreStar FCE-6460-N (PCI) 2-Gbit or JNI FCX-6562-N 2 Gb 133 MHz PCI-X-to-Fibre Channel host bus adapters (HBAs).

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

- One or more of the following Sun Microsystems hardware platform series:
  - Sun Blade 2000
  - Sun Fire 280R
  - Sun Fire V480
  - Sun Fire V880
  - Sun Fire 4800/4810 (PCI slots only, cPCI is not supported)
  - Sun Fire 6800 (PCI slots only, cPCI is not supported)
  - Sun Fire 12K
  - Sun Fire 15K
  - Ultra Enterprise 250
  - Ultra Enterprise 450
  - Ultra Enterprise 4000
  - Ultra Enterprise 3000
  - Ultra Enterprise 5000
  - Ultra Enterprise 6000
  - Ultra Enterprise 10000

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.

CXFS Commands Installed on Solaris

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

- `/usr/cxfs_cluster/bin/cxfs_client` (the CXFS client service)
- `/usr/cxfs_cluster/bin/cxfs_info`
- `/usr/cxfs_cluster/bin/cxfslicense`
- `/usr/cxfs_cluster/bin/xvm`
- `/usr/cxfs_cluster/bin/xvmprobe`

These commands provide all of the services needed to include a Solaris node in a CXFS cluster. The `pkgadd` output lists all software added; see "Solaris Installation Overview" on page 112.

For more information, see the `cxfs_client`, `xvm`, and `xvmprobe` man pages.

Log Files on Solaris

The `cxfs_client` command creates a `/var/log/cxfs_client` log file. (There is no `/var/cluster` log on Solaris nodes.) To rotate this log file, use the `-z` option in the `/usr/cxfs_cluster/bin/cxfs_client.options` file; see the `cxfs_client.options` man page for details.

For information about the log files created on CXFS administration nodes, see the *CXFS Administration Guide for SGI Infinite Storage*.

Limitations and Considerations on Solaris

CXFS for Solaris has the following limitations and considerations:

- 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
```

- Solaris supports block sizes in the range 2 KB through 64 KB.
- All disk devices attached to AMCC JNI controllers must be for use only by CXFS disks; do not attach non-disk devices to any AMCC JNI controllers that are configured for CXFS use. This restriction is required because all disk devices on AMCC JNI controllers configured for CXFS make use of the whole disk volume, which must be conveyed to Solaris via modification in the AMCC JNI driver to the value returned by the `READ_CAPACITY` SCSI command.

Maximum CXFS Filesystem Size and Offset Within a File on Solaris

The maximum size of a CXFS filesystem on a Solaris node is 2^{64} bytes (about 18 million terabytes). The maximum offset within a CXFS file is 2^{63}-1 bytes (about 9 million terabytes). An attempt to write beyond this limit will result in an `EFBIG (File too large)` error and the process will be sent a `SIGXFSZ (Filesize limit exceeded)` signal.

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.

FLEXlm License Verification for Solaris

Use the `cxfslicense` command with the `-d` option to verify that the FLEXlm licenses have been installed properly.

If the CXFS license is properly installed, you will see the following:

```
solaris# /usr/cxfs_cluster/bin/cxfslicense -d
CXFS license granted.
```

If you do not have the CXFS license properly installed, you will see the following error on the console when trying to run CXFS:

```
Cluster services:CXFS not properly licensed for this host.  Run
        '/usr/cxfs_cluster/bin/cxfslicense -d'
for detailed failure information.  After fixing the
license, please run '/etc/init.d/cxfs_cluster restart'.
```

An error such as the following example will appear in the `SYSLOG` file:

```
Mar  4 12:58:05 6X:typhoon-q32 crsd[533]: <<CI> N crs 0> Crsd restarted.
Mar  4 12:58:05 6X:typhoon-q32 clconfd[537]: <<CI> N clconf 0>
Mar  4 12:58:05 5B:typhoon-q32 CLCONFD failed the CXFS license check.Use the
Mar  4 12:58:05 5B:typhoon-q32    '/usr/cxfs_cluster/bin/cxfslicense -d'
Mar  4 12:58:05 5B:typhoon-q32 command to diagnose the license problem.
```

Host Bus Adapter Installation and Configuration for Solaris

This section provides an overview of the AMCC JNI Fibre Channel host bus adapter (HBA) installation and verification for Solaris nodes:

- "Installing the AMCC JNI HBA"
- "Protecting Data Integrity" on page 95
- "Installing and Running the EZ Fibre Configuration GUI" on page 95
- "Verifying the JNI HBA Installation" on page 104

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

Installing the AMCC JNI HBA

You can use one to four AMCC JNI HBAs for CXFS per Sun machine. (Other HBAs may be present that are not shared with the CXFS cluster.)

To install the AMCC JNI HBA, perform the following steps. Additional details are provided in various chapters/sections of 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 and Storage Networking Industry Association (SNIA) application programming interface package (`JNIsnia`), according to the instructions in the following "Installing" section:

   - "Unix Server DriverSuite"
     - "Solaris Driver"

- "Installing the Software"

You can retrieve the driver and SNIA package from the following website:

http://www.jni.com/drivers

a. Under **Locate Driver by Product**, click on **FCE-6460**.

b. Under the **Solaris** section, left click **JNIC146x.pkg** and save as the following pathname:

```
/var/tmp/JNIC146x.pkg
```

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 1-Gbit 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.

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 0. It then should appear as follows:

```
FailoverDelay = 0;
```

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

Installing and Running the EZ Fibre Configuration GUI

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. Install the GUI and change to the appropriate directory:
   a. Download the latest EZ Fibre GUI from the following website:
      i. Go to the following website:

         http://www.jni.com/drivers

      ii. Under **Locate Driver by Product**, click on **FCE-6460**
      iii. Under the **Solaris** section, left click **EZF_*xxx*.tar** or later and save as the following pathname (for example):

         `/var/tmp/EZF_`*`xxx`*`.tar`

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

```
solaris# tar xvf /var/tmp/EZF_xxx.tar
```

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

```
solaris# cd /var/tmp/EZF_xx
```

d. 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**.

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

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

3. 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 8-1. (The example screen snaps in this document may not exactly match the display you see.)

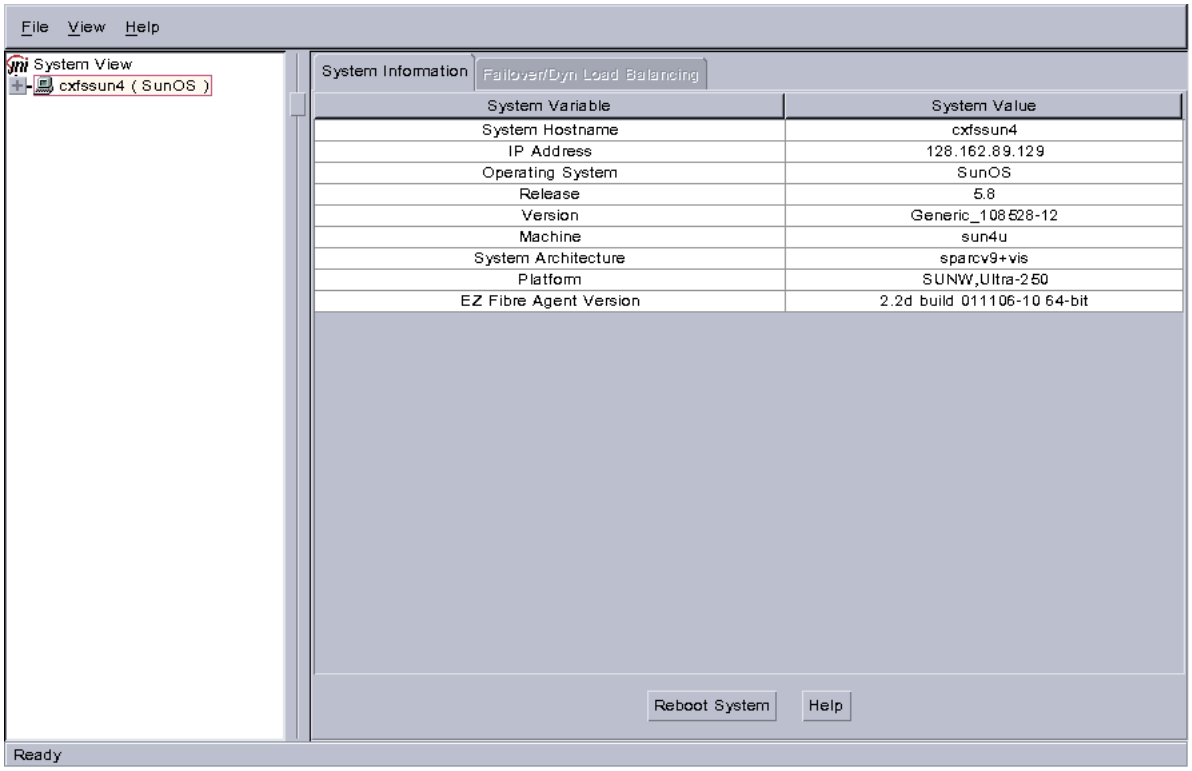

Figure 8-1 Example: Second Window: EZ Fibre Configuration Utility - Standalone

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 8-2 highlights the + sign.

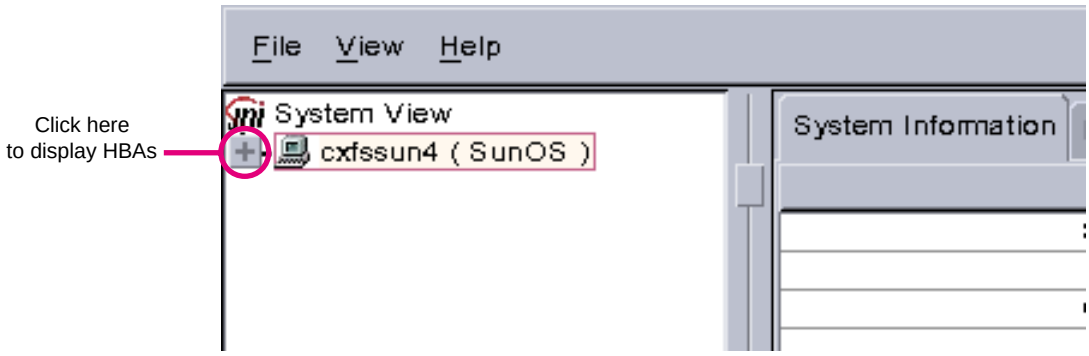

Figure 8-2 Location of icon (+) to Display the HBA

Figure 8-3 shows an example of the display after clicking on the + sign for `cxfssun4`, which shows the JNI HBA.

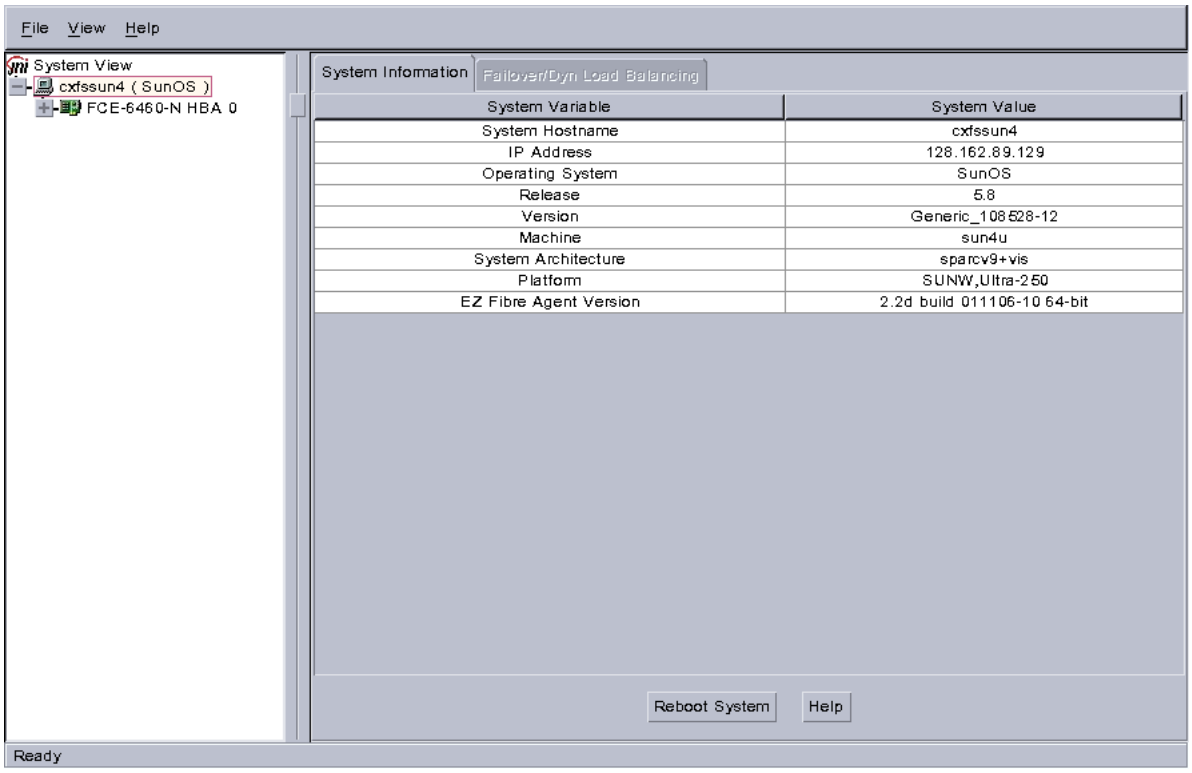

Figure 8-3 Example: After Clicking + to Display the HBA

4. Click on the icon to the right (not the + sign to the left). Figure 8-4 shows the icon.

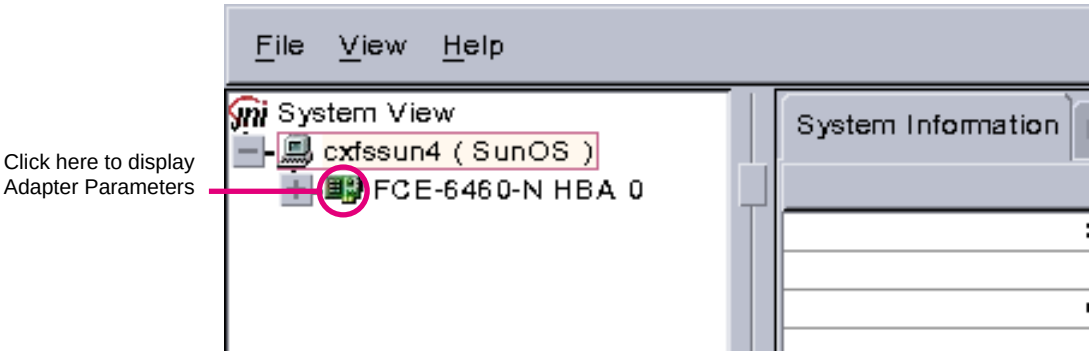

Figure 8-4 Location of the Icon to Display the Adapter Parameters

The right-hand pane will change to show **Adapter Parameters** for the selected HBA, as shown in Figure 8-5.

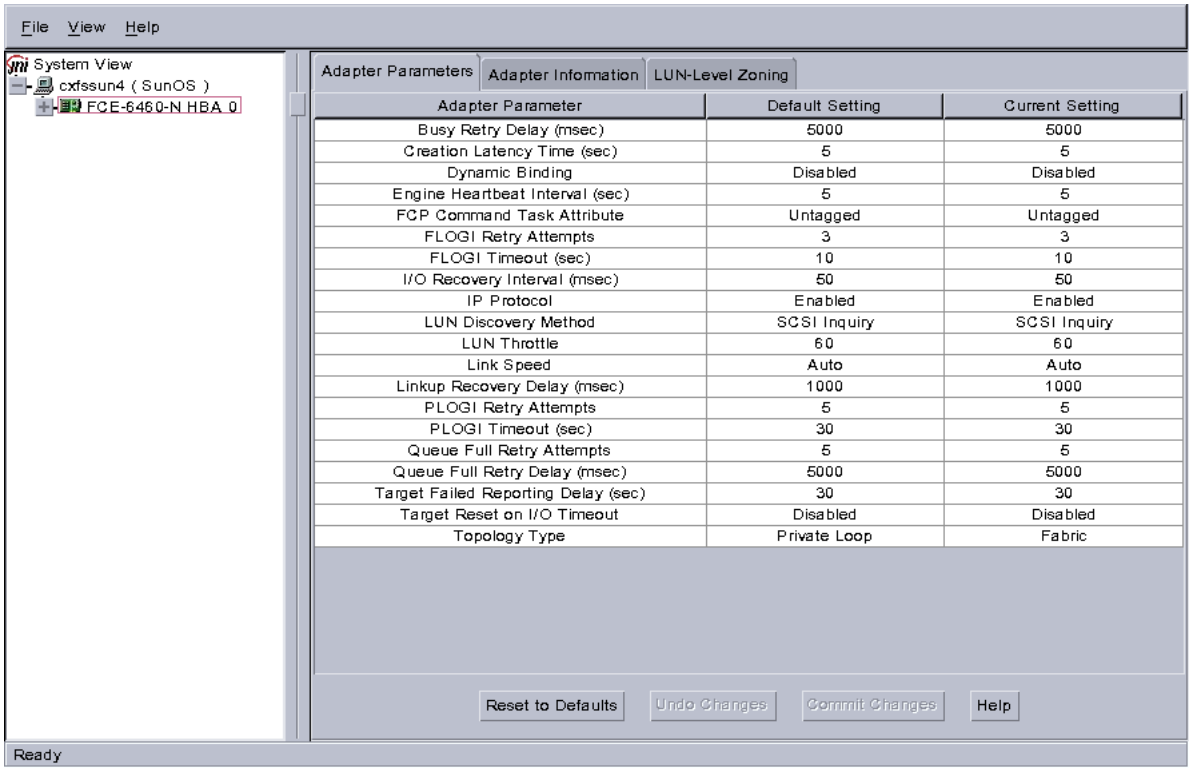

Figure 8-5 Example: After Clicking the HBA Icon to Show the Adapter Parameters

a. Click on the **Adapter Information** tab to see the information in Figure 8-6.

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 176.

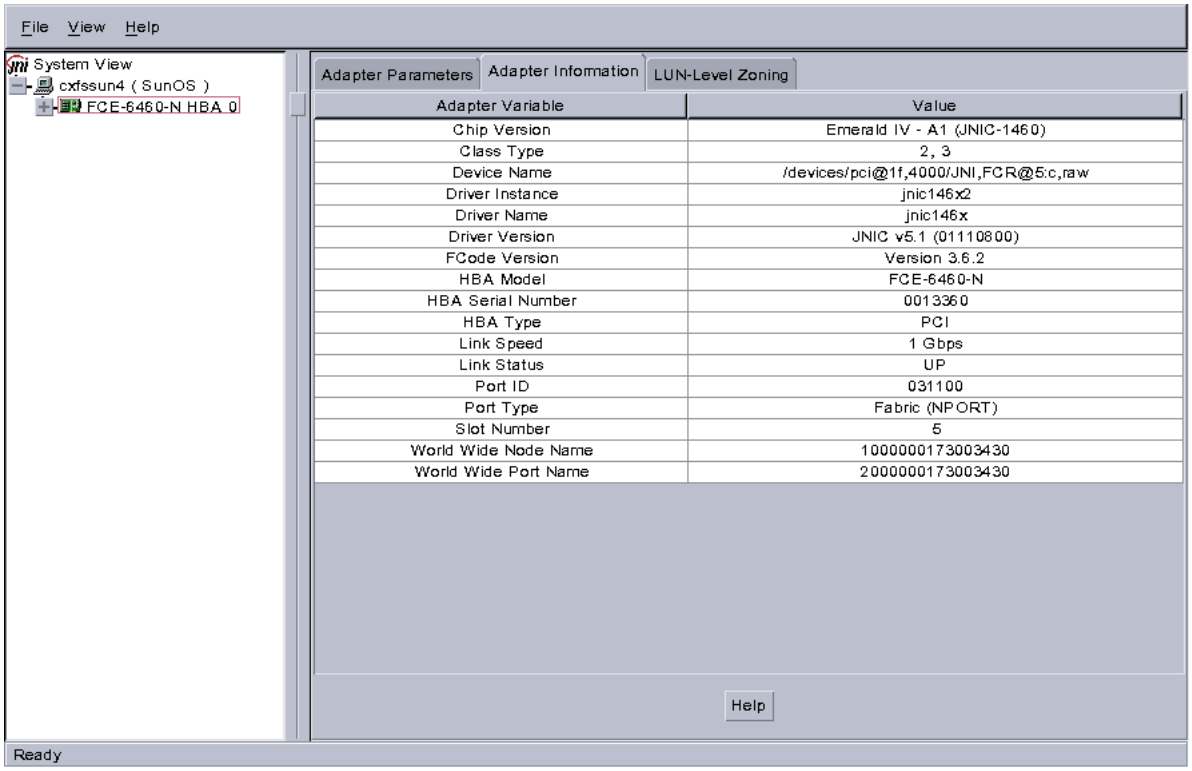

Figure 8-6 After Clicking the Adapter Information Tab

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 8-7.

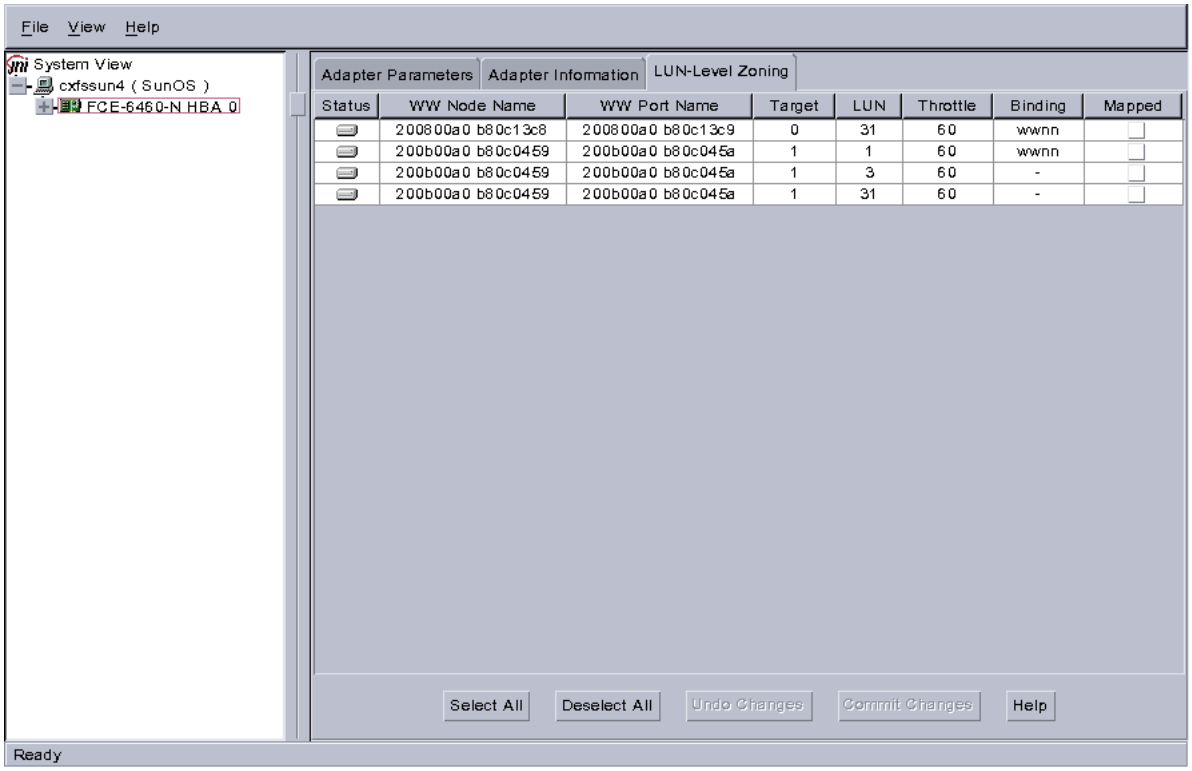

Figure 8-7 After Clicking on LUN-Level Zoning

5. 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 8-8. 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 TP9400. This LUN must not be used for other purposes; do not map it or use it for XVM volumes.

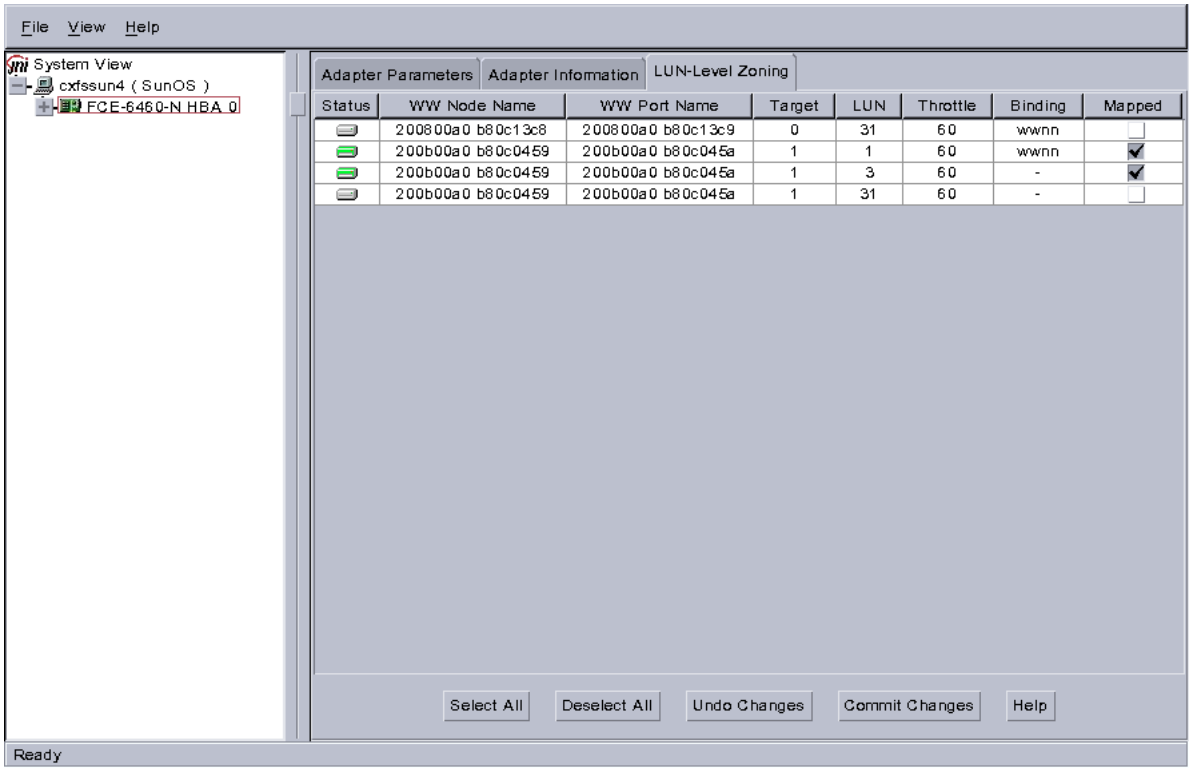

Figure 8-8 Example: After Mapping the LUNs and Committing the Changes

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

```
solaris# init 6
```

Verifying the JNI 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.

If you are having trouble with the verification steps, see "Common HBA Problems" on page 179.

Preinstallation Steps for Solaris

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.

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

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

Note: A private network is **required** for use with CXFS. Only the private network is used by CXFS for heartbeat/control messages.

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 every node in the cluster and their private interfaces as well.

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 "Hostname Resolution and Network Configuration Rules for All Platforms" on page 12.

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 Steps 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 `README` file for the Solaris platform to learn about any late-breaking changes in the installation procedure.
2. Verify that the node has been upgraded to Solaris 8 (also known as SunOS 5.8) or Solaris 9 (also known as SunOS 5.9) 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 5.8 or 5.9.
3. Insert the *CXFS MultiOS Client 3.1* CD-ROM.

4. Read the already inserted CD-ROM as follows:

   - Solaris 8:

     ```
     solaris# pkgadd -d /cdrom/cdrom01/solaris/SGIcxfs-sol8.pkg
     ```

   - Solaris 9

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

   For example, installing `SGIcxfs-sol8.pkg` under Solaris 8 will display at least the following output, although the exact version numbers may differ:

```
solaris# pkgadd -d /cdrom/cdrom01/solaris/SGIcxfs-sol8.pkg
The following packages are available:
  1 SGIcxfs     SGI CXFS client software
                (sparc) release 2.4

Select package(s) you wish to process (or ²all² to process
all packages). (default: all) [?,??,q]:

Processing package instance <SGIcxfs> from </cdrom/solaris/SGIcxfs-sol8.pkg>
. . .
```

5. Verify that the CXFS license key has been installed. See Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29.

   For example:

   ```
   solaris# /usr/cxfs_cluster/bin/cxfslicense -d
   CXFS license granted.
   ```

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.
```

Manual CXFS Startup/Shutdown 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 processes required to run CXFS.

To start up CXFS processes manually on your Solaris node, enter the following:

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

To stop CXFS processes manually, enter the following:

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

Software Maintenance for Solaris

This section contains the following:

- "Upgrading the CXFS Software on a Solaris System"
- "Modifying the CXFS Software on a Solaris System" on page 114

Upgrading the CXFS Software on a Solaris System

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 on a Solaris System

You can modify the CXFS client service (`/usr/cxfs_cluster/bin/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.

The first line in the `cxfs_client.options` file must contain the options you want `cxfs_client` to process; you cannot include a comment as the first line.

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

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

Windows Platforms

CXFS supports a client-only node running the Windows 2000 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 **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 Specific to Windows

In addition to the items listed in "Requirements" on page 6, using a Windows node to support CXFS requires the insertion of a Windows host with at least the following:

- An Intel Pentium or compatible processor
- Minimum RAM requirements (more will improve performance):
  - Windows 2000: 128 MB
  - Windows XP: 256 MB
- A minimum of 10 MB of free disk space
- A QLogic 2200, QLogic 2310, or QLogic 2342 host bus adapter (HBA)
- The following QLogic software from the http://www.qlogic.com website:
  - QLA2200 (Bios 1.76):
    - Windows 2000: v8.1.5.15
    - Windows XP: v8.1.5.12
  - QLA2310 (Bios 1.34):
    - Windows 2000: v8.2.2.10
    - Windows XP: v8.1.5.12
  - QLA2342 (Bios 1.34):
    - Windows 2000: v8.2.2.10
    - Windows XP: v8.1.5.12

  - QLDirect v8.01.07 and SANBlade Manager 2.0.15 or later are appropriate for all HBAs and versions of Windows

  You should install the documentation associated with the software. See the SANblade Manager `README` for the default password. Follow the QLogic instructions to install the driver, the SANblade NT Agent, and the SANblade Manager software. If you do not have the correct QLogic BIOS version installed, see the procedure in "Upgrading the QLogic BIOS" on page 140.

- If two QLogic HBAs are installed, you should also install the QLdirect Filter (8.1.3 or later) 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 3 or Service Pack 4
  - Windows XP Service Pack 1

CXFS Commands Installed on Windows

A single CXFS 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 `C:\Program Files\CXFS\cxfslicense` assists with license validation; see "Configuring the FLEXlm License for Windows" on page 150.

The command `C:\Program Files\CXFS\cxfs_info` displays the current state of the node in the cluster in a graphical user interface; see "Windows Log Files and Cluster Status" and "Verifying the Cluster" on page 165.

Windows Log Files and Cluster Status

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 Infinite Storage*. The CXFS Client service will also log important information to the following file:

```
C:\Program Files\CXFS\log\cxfs_client.log
```

This log file is not automatically rotated. To enable log rotation, add the `-z` *bytes* option to the CXFS Client service additional arguments during installation (see Figure 9-3 on page 145), or update the arguments (see "Modifying the CXFS for Windows Software" on page 155).

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

Start
> Programs
> CXFS
> CXFS Info

To start the GUI so that it is minimized in the notification tray, right-click on the icon in the **Start** menu, select properties, and add the `-m` option to the shortcut target. This is particularly useful if you are starting the command from the Windows **Startup** folder in the **Start** menu.

Figure 9-1 shows an example of the **CXFS Info** window.

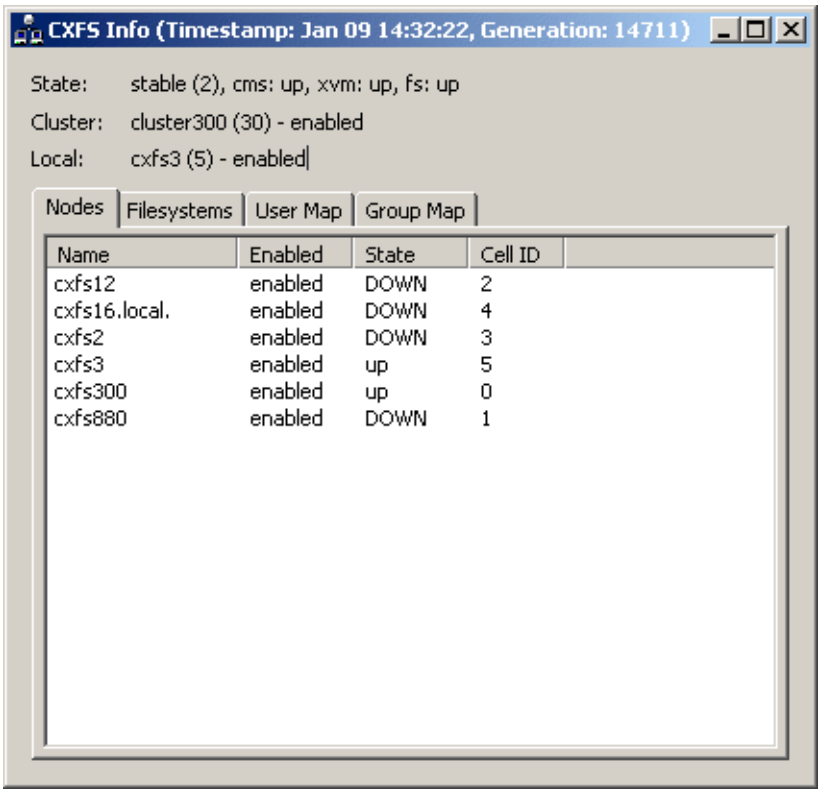

Figure 9-1 CXFS Info Window

Functional Limitations Specific to Windows

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

UNIX Perspective of CXFS on a Windows Node

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. However, a CXFS mount point or directory beneath a mount point can be mapped to another drive letter using the `subst` command from a command shell to which the application can write.

- 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 on a Windows Node

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:

  ```
  C:\Program Files\CXFS\passwd
  C:\Program Files\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 host.
- The volume properties window 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 a Windows Node

SGI recommends that you enable the forced unmount feature on CXFS filesystems. See "Recommendations" on page 9 and "Forced Unmount of CXFS Filesystems" on page 168.

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.

Memory Mapping Large Files

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 several 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.

Maximum CXFS Filesystem Size and Offset Within a File on Windows

The maximum size of a CXFS filesystem on Windows is 2^{64} bytes (about 18 million TB). The maximum offset within a file is 2^{63-1} bytes. An attempt to write beyond this limit will result in an `Invalid argument` or a `File too large` error.

Performance Considerations on a CXFS Windows Node

The following are performance considerations on a CXFS Windows node, in addition to the limitations described in "Performance Considerations" on page 5:

- 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 CXFS 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 to export CXFS filesystems to other nodes that are not running CXFS.
- 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 host, 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

Access Controls on a Windows Node

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.

User Identification on a Windows Node

The CXFS software supports several user identification mechanisms, which are described in "User Identification Mapping Methods" on page 126. 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.

For example, suppose a CXFS Windows node was configured with the following `passwd` and `group` files:

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

User Identification Mapping Methods

User identification can be performed by one of the following methods:

- **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" on page 151.
- **ldap_activedir:** Windows Active Directory server with Services for UNIX installed, which uses lightweight directory access protocol (LDAP).

  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` Cannot Map Users other than `Administrator` on a Windows Node" on page 184.
- **ldap_generic:** Generic LDAP lookup for UNIX users and groups from another LDAP server.

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 for Windows Software" on page 155.

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, in Figure 9-3 on page 145, 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.

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.

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

- Microsoft Windows 2000 Server or Windows 2000 Advanced Server
- Windows 2000 Active Directory
- Microsoft Windows Services for UNIX (SFU) Version 2 or 3, including the NFS Server option

---

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

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

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.

User Identification Map Updates

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.

The minimum update interval may be configured by editing the registry using `regedit`:

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.

Enforcing Access to Files and Directories

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 super user privileges on a UNIX node.

The following example is a directory listing on the 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 9-1. (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 9-1.

Table 9-1 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 will 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)

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

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 on Window nodes

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 on a Windows node

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`. This mask is defined in the registry of the CXFS driver and may be configured to other values (typically `000` or `002`):

```
HKEY_LOCAL_MACHINE->SYSTEM->CurrentControlSet->Services->CXFS->Parameters->DefaultUmask
```

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.

Host Bus Adapter 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

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 Two HBAs for Failover Operation on Windows 2000

This procedure assumes that the CXFS driver is already installed and working properly with one host bus adapter (HBA) for a Windows 2000 node.

Note: Two HBAs and/or more than one port of a dual-port HBA are not support in Windows XP.

QLogic only supports failover and 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 failover operation under Windows 2000, do the following:

1. Install the QLdirect driver v8.01.07 by following all the default settings for the installation and verify that the CXFS client still operates normally.
2. Disable fencing for this node. You can do this using the CXFS GUI or the `cmgr`(1M) command.
3. Determine the world wide port name (WWPN) of the current adapter:
   a. Install SANsurfer Qlogic SANblade Agent v2.0.15.
   b. Install SANsurfer Qlogic SANblade Manager v2.0.15.
   c. Run SANsurfer to determine the WWPN.
   d. Record the WWPN on paper.
4. Shut down Windows 2000.
5. Install the second HBA and start Windows 2000.
6. If the second HBA is a different model from the original one, install its mini port driver (for example, `ql2300.sys`).
7. Start the Qlogic SANblade 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 world wide node name (WWNN) but different WWPNs. The original HBA can be recognized by its WWPN recorded in step 3.

8. Verify the driver parameters for the Qlogic HBAs. Run `regedit` and go to the following key:

   HKEY_LOCAL_MACHINE
   > SYSTEM
   > CurrentControlSet
   > Services
   > ql2200 ***(or ql2300)***
   > Parameters
   > Device

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

   ```
   Buschange=0;FixupInquriy=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.

9. Configure the HBA port (click on **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 on **OK** to continue.

   The HBA0 devices are automatically set to be visible for Windows 2000 applications (notice the open eye) and HBA1 devices are set to be invisible (notice the closed eye).

10. 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 LANs traffic load associated with this device between the two HBAs.

Repeat this step for each of the other HBA devices.

11. Click on **Apply** to save the new configuration.
12. Update the switch port information using the CXFS GUI or the `cmgr` command. Enable fencing.
13. Reboot Windows.

For more information about using the CXFS GUI or the `cmgr` command to perform these tasks, see *CXFS Administration Guide for SGI Infinite Storage*.

Preinstallation Steps for Windows

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.

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:

Upgrading the QLogic BIOS

Note: If CXFS is already installed and running, stop the CXFS Client service and set it to manual, as described in "Manual CXFS Startup/Shutdown for Windows" on page 154, and then restart the machine. You can then perform the following procedure.

If you need to upgrade the QLogic BIOS, do the following:

1. Run the QLogic SANsurfer software and connect to the machine:

   Start
   > Programs
   > QLogic Management Suite
   > SANsurfer
   > Connect

   If you are unable to connect to the machine, you may not have installed the QLogic NT Agent software, which is another option in the SANsurfer software installation.

2. Enable the BIOS on the HBA by selecting the following:

   Adapter 2*xxx*
   > NVRAM Settings
   > Enable Host Adaptor BIOS

3. Update the BIOS by selecting the following:

   Adapter 2*xxx*
   > Utilities
   > Update Flash

   Select the BIOS update file.

4. Mark the CXFS Client service to automatically start.

5. Reboot the machine.

Adding a Private Network for Windows Nodes

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

Note: A private network is **required** for use with CXFS. Only the private network is used by CXFS for heartbeat/control messages.

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

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 on the private network adapter and click on **Properties**. Uncheck all check boxes except `Internet Protocol (TCP/IP)`
4. Select `Internet Protocol (TCP/IP)` and then click on **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.

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. 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:

```
> 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 . . . . . . . . . :
```

Client Software Installation Steps 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 `C:\Program Files\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`.
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 3.1* CD-ROM into the Windows host. Normally, the setup program will automatically run, otherwise run the following program from the CD-ROM:

   ```
   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 9-2. If the software is to be installed in a nondefault directory, click on **Browse** to select another directory. Click on **Next** when finished.

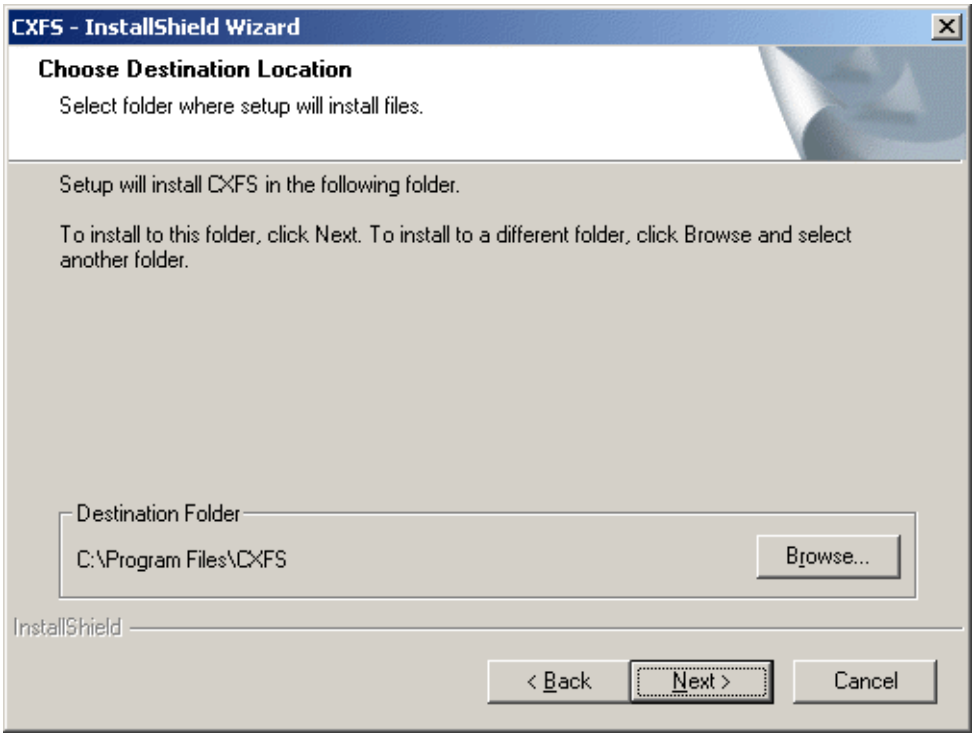

Figure 9-2 Choose Destination Location

7. Enter details for the following fields as shown in Figure 9-3 and click on **Next** when finished:

   - **Drive letter for CXFS volumes to be mounted on:** specify the **drive letter** under which all CXFS filesystems will be mounted. You cannot select a drive letter that is currently in use.
   - **User ID mapping lookup sequence:** choose the appropriate primary and (optionally) secondary method. See "User Identification Mapping Methods" on page 126.
   - **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 `C:\Program Files\CXFS`.

- **IP address of the heartbeat network adapter:** specify the IP address of the private network adapter on the Windows node.
- **Additional arguments**: enter arguments that may be passed to the CXFS Client service. For most configurations, this can be left empty. See "Modifying the CXFS Software on a Solaris System" on page 114.

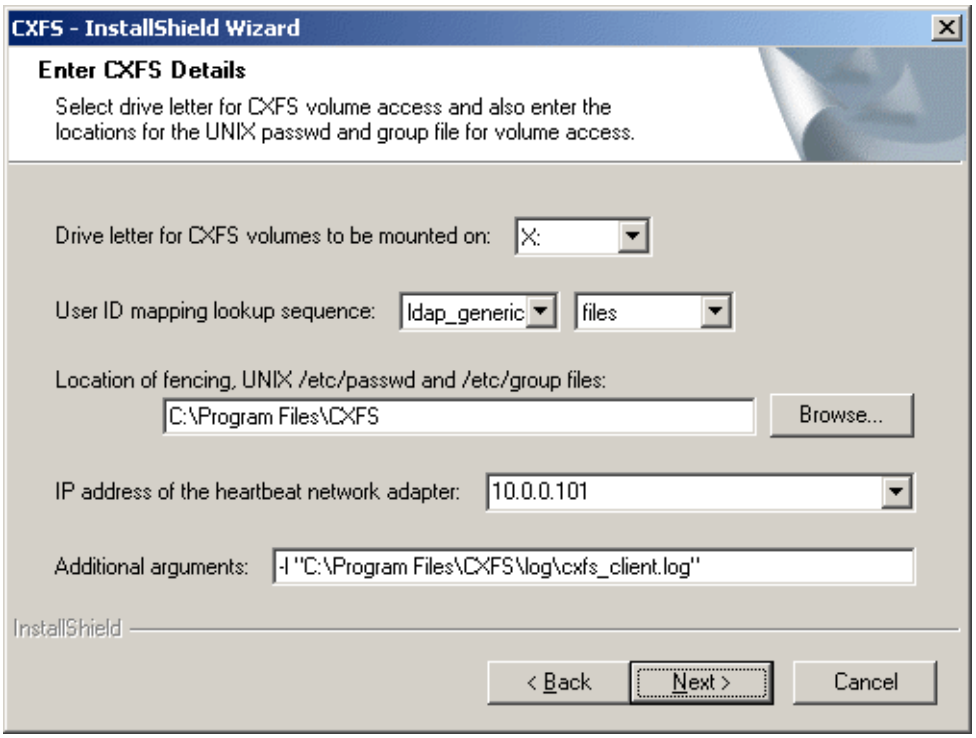

Figure 9-3 Enter CXFS Details

8. If you select **ldap_activedir** as the user ID mapping method, the dialog in Figure 9-4 is displayed after you click **Next**.

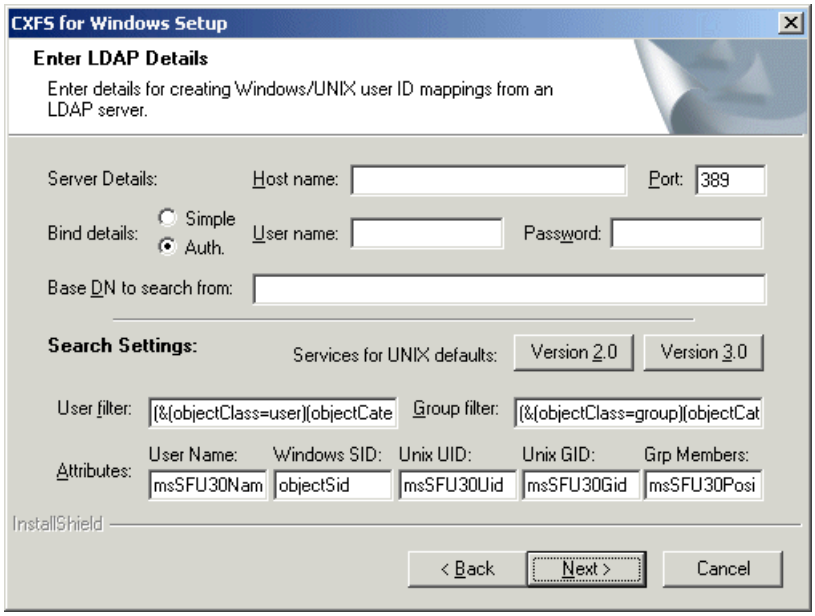

Figure 9-4 Active Directory Details

If you have a standard Active Directory configuration with Windows Services for UNIX (SFU), you need only to select the version of SFU and ensure that **Authenticated** binding is selected; 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 9-5 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**.

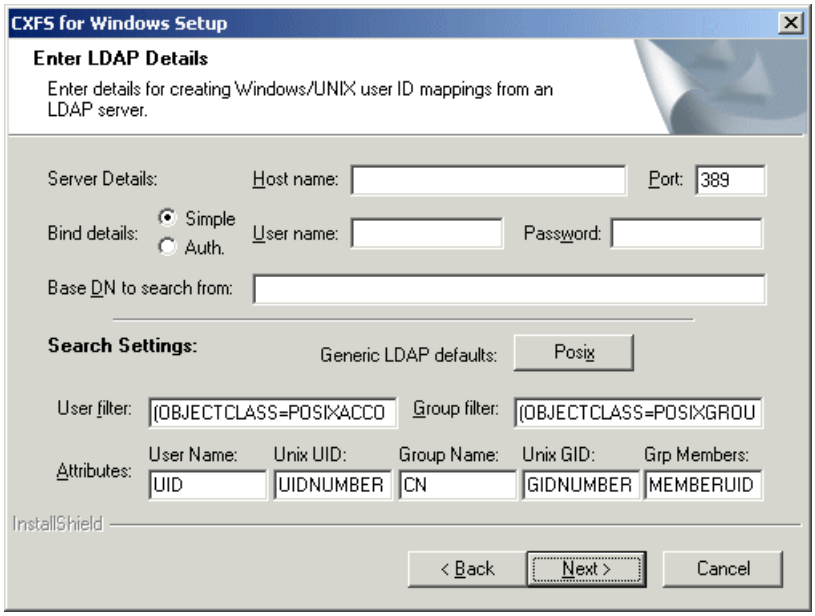

Figure 9-5 Generic LDAP Details

10. Review the settings, as shown in Figure 9-6. If they appear as you intended, click on **Next**. If you need to make corrections, click on **Back**.

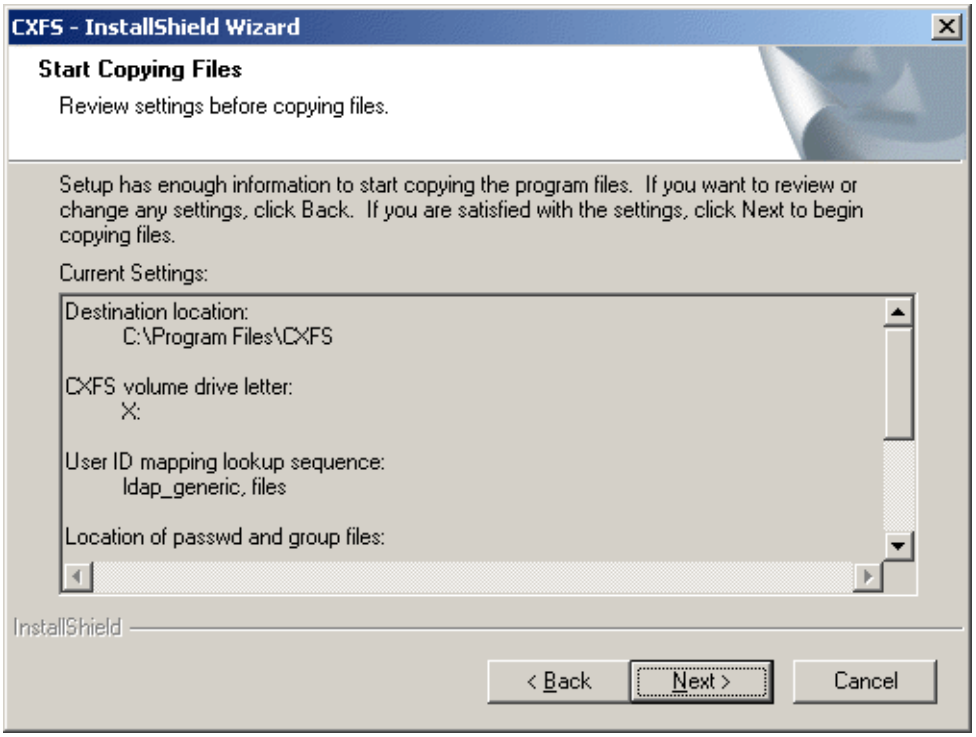

Figure 9-6 Review the Settings

After you click on **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 9-7.

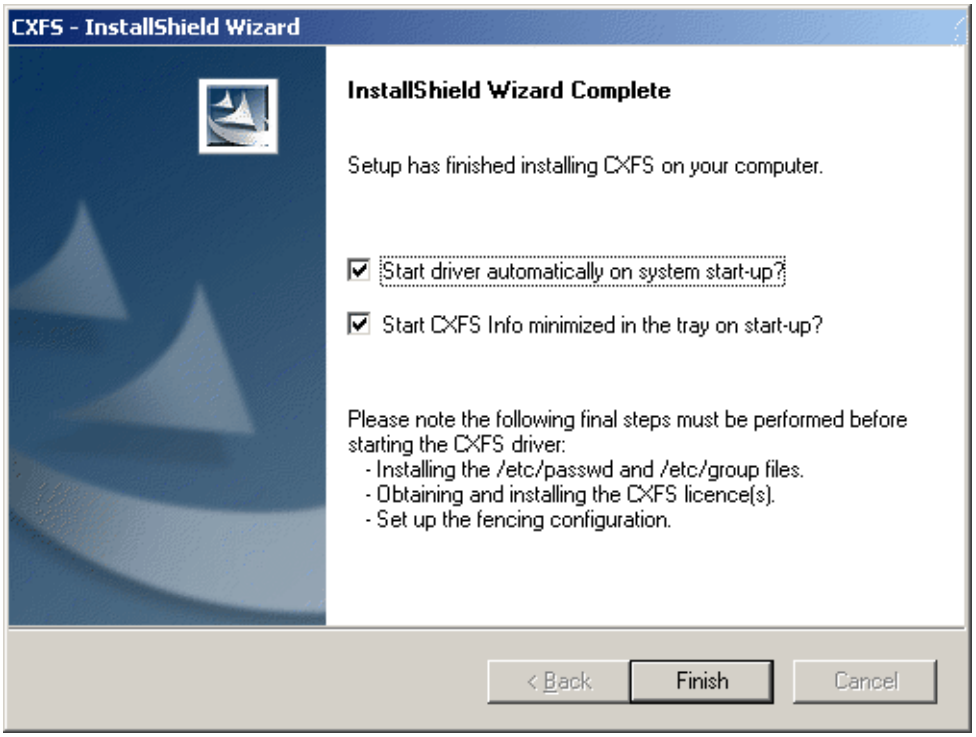

Figure 9-7 Start CXFS Driver

Because there are some important postinstallation steps, do not start the CXFS driver now. Click **Start driver automatically on system start-up** and click on **Finish**.

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:

- "Configuring the FLEXlm License for Windows"

Configuring the FLEXlm License for Windows

Note: Windows 2000 licenses cannot be used under Windows XP, and vice versa. If you are upgrading a Windows node, you must obtain a new license.

You must configure a FLEXlm license before you restart the Windows node by following these steps:

1. Add the mandatory CXFS license and the optional XVM license to the following file:

   ```
   C:\Program Files\CXFS\lib\license.dat
   ```

   For more information, see Chapter 4, "Obtaining CXFS and XVM FLEXlm Licenses" on page 29.

2. Validate these licenses by running 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:\Program Files\CXFS\cxfslicense.exe
   ```

   If a valid license has been correctly specified, the following will be displayed:

   ```
   Found valid license for feature CXFS_W2K version 2.000
   The CPU count specified in the license is OK.
   ```

If the Windows node has the optional XVM mirroring license, you will also see the following:

```
Found valid license for feature XVM_W2K version 3.000
The CPU count specified in the license is OK.
```

Note: Licenses for Windows 2000 have the feature names `CXFS_W2K` and `XVM_W2K`. Licenses for Windows XP have the feature names `CXFS_WXP` and `XVM_WXP`.

Performing User Configuration

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 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 the CXFS node is restarted, 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, which defaults to the following:

- `/etc/passwd` as `C:\Program Files\CXFS\passwd`
- `/etc/group` as `C:\Program Files\CXFS\group`

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

Select a user and then select:

Properties
> UNIX Attributes

The CXFS software will check for changes to the LDAP database every 5 minutes.

After the CXFS software has started, you can use the `cxfs_info` command to confirm the user configuration, regardless of the user ID mapping method chosen. See "User Identification on a Windows Node" on page 125.

If only the `Administrator` user is mapped, see "`cxfs_client` Cannot Map Users other than `Administrator` on a Windows Node" on page 184.

Checking Permissions on the Password and Group Files

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 file names 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/GUI at will and even gain superuser access to the files on the CXFS filesystem.

Creating a New Hardware Profile

It is strongly recommended that you create a new hardware profile and that you disable the CXFS software in the current hardware profile, in order to have a backup profile available. If the CXFS software causes the host to crash on startup, you can easily switch back to the original hardware profile and successfully return to the configuration before the CXFS software was installed.

To create a new hardware profile, right-click the **My Computer** icon and select the following:

Properties
> Hardware
> Hardware Profiles
> Copy

This action copies the current hardware profile, most likely called **Profile 1 (Windows 2000 and Windows XP)**. You should call this new profile **CXFS Configuration** to distinguish it from other profiles. You can make the **CXFS Configuration** the default profile chosen on startup by selecting the up arrow button and moving the **CXFS Configuration** profile to the top of the list.

To remove the CXFS driver from the current hardware profile, which should be the original profile, select the following:

Start
> Settings
> Control Panel
> Administrative Tools
> Computer Management
> System Tools
> Device Manager

To show the CXFS driver, select the following:

Non-Plug and Play Devices
> CXFS
> Properties
> Device Usage
> Do not use this device in the current hardware profile

You should also disable the CXFS Client service for the current profile by selecting the following:

Start
> Settings
> Control Panel
> Administrative Tools
> Services
> CXFS Client
> Properties
> Log On
> Disable

When the Windows host boots, you may choose **CXFS Configuration** to automatically start CXFS or choose the previous profile (most likely **Original Configuration**) to start without CXFS.

Manual CXFS Startup/Shutdown for Windows

The CXFS processes are 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, so 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.

Software Maintenance for Windows

This section contains the following:

Modifying the CXFS for Windows Software

To change the location of the software and other configuration settings that were requested in "Client Software Installation Steps for Windows" on page 142, perform the following steps:

1. Select the following:

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

   Figure 9-8 shows the screen that lets you modify the software.

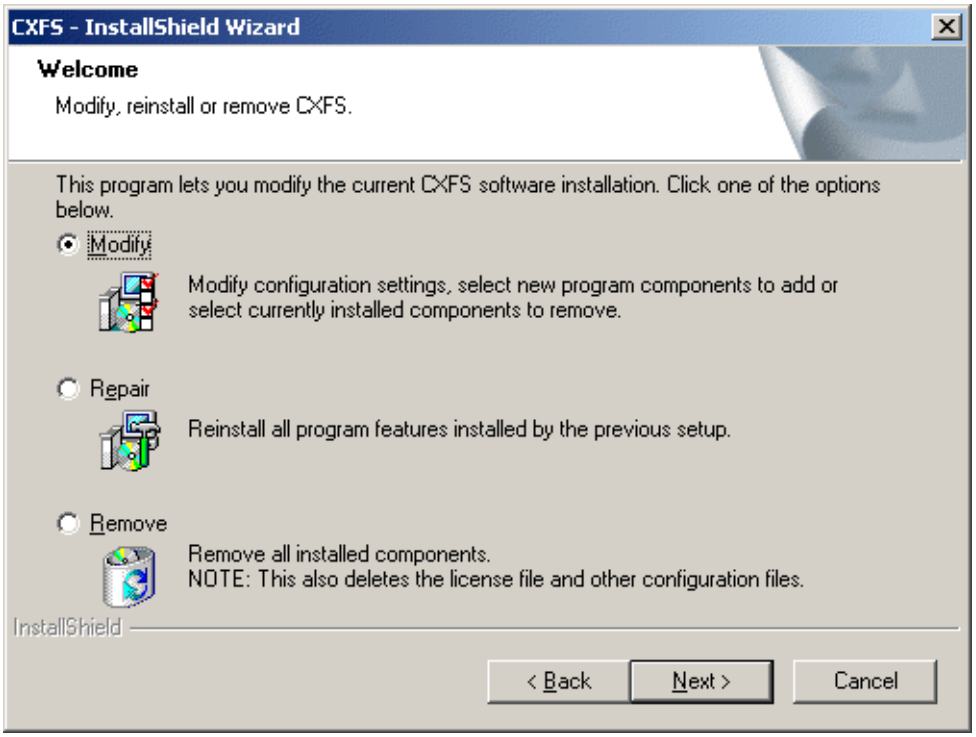

Figure 9-8 Modify the CXFS for Windows

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 DOS command shell as follows:

   - Windows 2000:

     ```
     > C:\Winnt\system32\cxfs_client.exe -h
     ```

   - Windows XP:

     ```
     > C:\Windows\system32\cxfs_client.exe -h
     ```

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

Upgrading the CXFS Software on a Windows System

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

1. Insert the CD-ROM containing the upgraded software to run the setup program. If the setup program does not automatically start, run `winnt/setup.exe` from the CD-ROM.
2. A welcome screen will appear that displays the version you are upgrading from and the version you are upgrading to. Figure 9-9 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 Steps for Windows" on page 142.

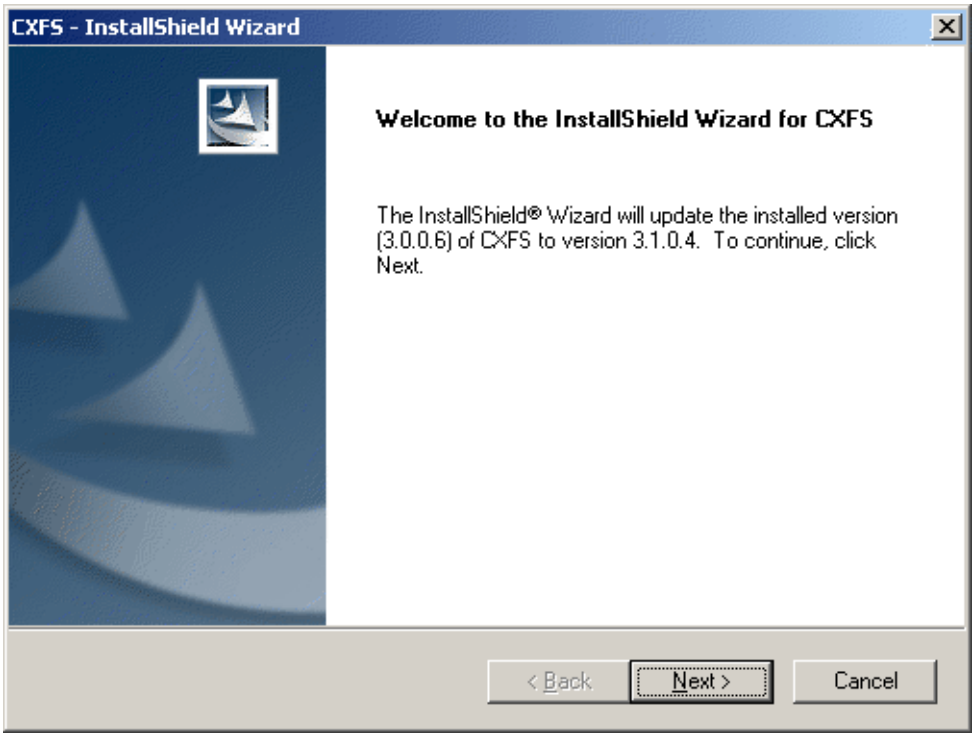

Figure 9-9 Upgrading the Windows Software

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

Removing the CXFS Software from a Windows System

To remove the CXFS for Windows software, first ensure that no applications on this host 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 9-8 on page 156 shows the screen that lets you remove the software.

Note: The `passwd` and `group` files will be removed.

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

Downgrading the CXFS Software on a Windows System

To downgrade the CXFS software, follow the instructions to remove the software in "Removing the CXFS Software from a Windows System" on page 158 and then install the older version of the software as directed in "Client Software Installation Steps for Windows" on page 142.

Note: The removal process may remove the configuration and license files. You should back up these files before removing the CXFS software so that you can easily restore them after installing the downgrade.

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 on a CXFS administration node, using either the CXFS GUI (invoked by the `cxfsmgr` command) or the `cmgr` command. The GUI also provides a guided configuration for defining a cluster.

This section discusses the following tasks in cluster configuration:

For detailed configuration instructions, see the *CXFS Administration Guide for SGI Infinite Storage*.

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. You can do this on a CXFS administration node using the CXFS GUI or `cmgr` command.

Do the following to determine the value for the hostname field in the GUI:

- AIX: use the value displayed by `/usr/bin/hostname`, which must match the node's primary hostname in the `/etc/hosts` file; that is, the first field after the node's IP address in `/etc/hosts`. This field can be either the hostname or the fully qualified name.
- Linux 32-bit: use the value displayed by `/bin/hostname`.
- Mac OS X: use the value displayed by `/bin/hostname`.
- Solaris: use the value displayed by `/etc/nodename`, which must match the node's primary hostname in the `/etc/inet/hosts` (or `/etc/hosts`) file; that is, the first field after the node's IP address in `/etc/inet/hosts` (or `/etc/hosts`). This field can be either the hostname or the fully qualified domain name.
- 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` 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
```

For details about these commands, see the "Define a Node" sections of the GUI or `cmgr` reference chapters in the *CXFS Administration Guide for SGI Infinite Storage*.

Adding the Client-Only Nodes to the Cluster

After you define all of the client-only nodes, you must add them to the cluster using either the CXFS GUI or the `cmgr` command on a CXFS administration node.

For example, if you have already defined a cluster named `cxfscluster` 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
```

For details, see the "Modify a Cluster" sections of the GUI or `cmgr` reference chapters in the *CXFS Administration Guide for SGI Infinite Storage*.

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 164.

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 Brocade Fibre Channel switch. To define the switch for the cluster database, use either the CXFS GUI or the `cmgr` command on a CXFS administration node.

For example:

```
cmgr> define switch ptg-brocade username admin password password
```

After you have defined the switch, you must ensure that all of the Brocade 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 using the GUI or `cmgr`.

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

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
```

For details, see the "Define a Switch" and "Update Switch Port Information" sections of the GUI or `cmgr` reference chapters in the *CXFS Administration Guide for SGI Infinite Storage*.

Starting CXFS Services on the Client-Only Nodes

After adding the client-only nodes to the cluster, you must start CXFS services on them. You can do this using either the CXFS GUI or the `cmgr` command on a CXFS administration node.

For example:

```
cmgr> start cx_services on node solaris1 for cluster cxfscluster
cmgr> start cx_services on node solaris2 for cluster cxfscluster
```

For details, see the "Start CXFS Services" sections of the GUI or `cmgr` reference chapters in the *CXFS Administration Guide for SGI Infinite Storage*.

Verifying LUN Masking

You should verify that the HBA has logical unit (LUN) masking configured such that the LUNs are visible to all the machines in the cluster after you connect the HBA to the Brocade Fibre Channel 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, then you do not need to specifically mount the filesystems on the new client-only nodes that you added to the cluster.

Otherwise, 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. You can do this using the GUI or the `cmgr` command on a CXFS administration node.

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 "Recommendations" on page 9 and "Forced Unmount of CXFS Filesystems" on page 168.

For details, see the “Define a Filesystem” and “Mount a Filesystem” sections of the GUI or the `cmgr` reference chapters in the *CXFS Administration Guide for SGI Infinite Storage*.

Restarting the Windows Node

After completing the steps in "Postinstallation Steps for Windows" on page 149 and this chapter, you should restart the Windows node. This will automatically start the driver and the 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

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 `clconf_info` command, and and the `cluster_status` command on a CXFS administration node.

For example:

```
irix# /var/cluster/cmgr-scripts/cluster_status

+ Cluster=cxfscluster  FailSafe=Not Configured CXFS=ACTIVE              15:15:33
   Nodes =   cxfs6    cxfs7    cxfs8    solaris1    solaris2
FailSafe =
    CXFS =      UP       UP       UP          UP          UP

CXFS             DevName           MountPoint           MetaServer     Status
fs1         /dev/cxvm/fs1                /fs1                cxfs7        UP
fs2         /dev/cxvm/fs2                /fs2                cxfs6        UP
```

On client-only nodes, the `cxfs_info` command serves a similar purpose. The command path is as follows:

- AIX, IRIX, Linux 32-bit, and Solaris: `/usr/cxfs_cluster/bin/cxfs_info`
- Windows: `C:\program files\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 Jul 19 13:30:22 / generation 21604]

Cluster:
    zagato (1) - enabled
Local:
    thump (2) - enabled, state: stable, cms: up, xvm: up, fs: up
Nodes:
    leesa      enabled  up    0
    thump      enabled  up    2
    thunderbox enabled  up    1
Filesystems:
    bigstripe0 enabled  mounted          bigstripe0           /mnt/bigstripe0
    concat0    enabled  mounted          concat0              /mnt/concat0
    mirror0    enabled  mounted          mirror0              /mnt/mirror0
    r0lun0s0   enabled  mounted          r0lun0s0             /mnt/cxfs0
    r0lun0s1   enabled  mounted          r0lun0s1             /mnt/cxfs1
    r0lun0s2   enabled  mounted          r0lun0s2             /mnt/cxfs2
    stripe0    enabled  mounted          stripe0              /mnt/stripe0
```

The `Local` 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 info 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.

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. To enable forced unmount, define or modify the filesystem to unmount with force and then unmount the filesystem, using the following `cmgr` commands:

```
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 Infinite Storage*.

Troubleshooting

This chapter contains the following:

Identifying Problems

This section provides tips about identifying problems according to operating system.

Is the Client-Only Node in the Cluster?

To determine if the node is in the cluster, use the `cluster_status` command on a CXFS administration node, connect to the CXFS GUI on an administration node, or use the `cxfs_info` command on the client-only node. See "Verifying the Cluster" on page 165.

Are there Error Messages?

This section describes potential error messages for each platform.

AIX Error Messages

Look at the `/var/tmp/cxfs_client` log to see if there are any messages containing the words `ERROR` or `Warning`. Specific cases where these messages will occur include the following:

- The fencing file was not found, therefore the fencing configuration will not be updated on the server. For example:

```
cxfs_client: cis_get_hba_wwns warning: fencing configuration file "/etc/fencing.conf" not found
```

- A filesystem mount has failed and will be retried. For example:

  ```
  cxfs_client: op_failed ERROR : Mount failed for aixdisk0s0
  ```

Linux 32-bit Error Messages

Look at the `/var/log/cxfs_client` log to see if there are any messages containing the words `ERROR` or `Warning`. Specific cases in which these messages will occur include the following:

- The fencing file was not found, therefore the fencing configuration will not be updated on the server. For example:

```
cxfs_client: cis_get_hba_wwns warning: fencing configuration file "fencing.conf" not found
```

- A filesystem mount has failed and will be retried. For example:

  ```
  cxfs_client:op_failed ERROR: Mount failed for concat0
  ```

For more information about these files, see "Log Files on Linux 32-bit Platforms" on page 53. Also see the log files on the CXFS administration node; for more information, see the *CXFS Administration Guide for SGI Infinite Storage*.

Mac OS X Error Messages

Look at the `/var/log/cxfs_client` log to see if there are any messages containing the words `ERROR` or `Warning`. Specific cases in which these messages will occur include the following:

- The fencing file was not found, therefore the fencing configuration will not be updated on the server. For example:

```
cxfs_client: cis_get_hba_wwns warning: fencing configuration file "/etc/fencing.conf" not found
```

- A filesystem mount has failed and will be retried. For example:

  ```
  cxfs_client: op_failed ERROR : Mount failed for concat0
  ```

Solaris Error Messages

Look at the `/var/log/cxfs_client` log to see if there are any messages containing the words `ERROR` or `Warning`. Specific cases in which these messages will occur include the following:

- The fencing file was not found, therefore the fencing configuration will not be updated on the server. For example:

```
cxfs_client: cis_get_hba_wwns warning: fencing configuration file "fencing.conf" not found
```

- A filesystem mount has failed and will be retried. For example:

```
cxfs_client:op_failed ERROR: Mount failed for concat0
```

For more information about these files, see "Log Files on Solaris" on page 89. Also see the log files on the CXFS administration node; for more information, see the *CXFS Administration Guide for SGI Infinite Storage*.

Windows Error Messages

Look in the following file to see if there are any error or warning messages:

```
C:\Program Files\CXFS\log\cxfs_client.log
```

You can also view the **System Event** log by selecting the following:

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

Identifying Other Problems on Windows Nodes

The following sections will help you identify problems with Windows client-only nodes.

Is the CXFS Software Running Correctly on the Windows Node?

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

Windows Error Message Explanations

Following are typical Windows error messages and their meanings:

```
op_failed ERROR: Mount failed for concat0
```

A filesystem mount has failed and will be retried.

```
cis_generate_userid_map warning: could not open passwd file
```

The `passwd` file could not be found.

```
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 89 and "Windows Log Files and Cluster Status" on page 120. Also see the log files on the CXFS administration node; for more information, see the *CXFS Administration Guide for SGI Infinite Storage*.

```
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 for Windows Software" on page 155.

```
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" on page 152.

```
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.

```
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`.

```
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.

```
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.

```
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 Windows 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.

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.

Common Problems and Solutions

This section contains the following common problems and their solutions:

Incorrect Configuration

To avoid having trouble with the CXFS client-only node, ensure you have the correct configuration. See "Requirements" on page 6.

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 logged to the following files:

- `/var/log/cxfs_client` on Solaris and Linux 32-bit nodes
- `C:\Program Files\CXFS\cxfs_client.log` on Windows nodes

If no WWPNs are detected, you can manually specify the WWPNs in the fencing file for the following platforms:

- Linux 32-bit
- Mac OS X
- Solaris
- Windows

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

The fencing file is required on the AIX platform; see "Postinstallation Steps for AIX: Creating the I/O Fencing File" on page 46. However, the fencing file is not used on the IRIX platform.

The fencing file enumerates the worldwide port name 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, the fencing file must contain a simple list of WWPNs, one per line.

If you use the fencing file, you must update it whenever the HBA configuration changes, including the replacement of an HBA.

The file location varies by platform:

- `/etc/fencing.conf` on AIX, Linux, and Solaris nodes
- `C:\Program Files\CXFS\fencing.conf` on Windows nodes

Do the following:

Note: On Solaris nodes, you might be able to determine the HBA WWPN by running the EZ Fibre Configuration GUI: see "Installing and Running the EZ Fibre Configuration GUI" on page 95 and Figure 8-6 on page 102. If so, and you are **completely certain** that you can determine the correct WWPN of the HBA (and **not** that of any of the SAN targets), you can enter this value in the `/etc/fencing.conf` file.

---

1. Set up the Brocade Fibre Channel switch and HBA.
2. Follow the Fibre Channel cable on the back of the node to determine the port to which it is connected in the Brocade Fibre Channel 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 Brocade Fibre Channel switch and log in as user `admin` (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 the fencing file 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 Brocade Fibre Channel switch, you fencing file should contain the following:

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

6. After the node is added to the cluster (see Chapter 10, "Cluster Configuration" on page 159), enable the fencing feature by using the CXFS GUI or `cmgr` command on a server-capable administration node; for more information, see the *CXFS Administration Guide for SGI Infinite Storage*.

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. For more details, see the *CXFS Administration Guide for SGI Infinite Storage*.

Common HBA Problems

Consult the following checklist to help you identify the problem with a host bus adapter (HBA):

- Is the HBA firmly seated in its PCI slot?
- Are all cables undamaged and connected?
- Is power applied to all devices?
- Do the link lights illuminates on all units?
- Is the problem confined to just one unit? If so, check the cabling between the switch and the unit; if no units are being shown, suspect cabling from the HBA.
- Is the Brocade switch properly licensed?
- For a Solaris node, did you enable fabric mode? See step 4 in "Installing the AMCC JNI HBA" on page 93.
- For Windows node, check the QLogic management tool event and alarm logs. Select the following:

  Start
  > Programs
  > QLogic Management Suite
  > SANsurfer

For more information, see the HBA documentation listed in the preface.

Common AIX Problems

The `cxfs_client` Service is Not Started on an AIX Node

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

- If the following message is output to the console, it means that the workstation is in 32-bit kernel mode:

  ```
  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 following message is output to the `/var/tmp/cxfs_client` file, the license has expired:

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

  In this case, you must reinstall the license.

The Filesystem Does Not Mount on an AIX Node Due to Address

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.

The AIX Node Cannot Achieve UP State

The AIX node might not achieve UP state for the following reasons:

- If the cluster_status command is carried out and the mutual status is confirmed in the metadata server, but the AIX node and metadata server are out of synchronization.

  In this case, the status of the two nodes must be made to correspond by doing the following, as needed:

  1. Restart cluster daemons on the metadata server.
  2. If that does not solve the problem, reboot the metadata server.
  3. If that still does not solve the problem, reboot the AIX node.

- If the following message is displayed in the /var/tmp/cxfs_client file, the definitions of all hosts are not found in the /etc/hosts file:

```
cix_socket_recv warning: error reading socket: short read 0 !=1
```

  In this case, you must define all of the hosts in the /etc/hosts file and then reboot the AIX client.

- If the following message is displayed in the /var/tmp/cxfs_client file, then the node previously belonged to another cluster without first being rebooted:

```
error including cell node name : Invalid argument unable to start cms daemon: Invalid argument
```

  In this case, you must reboot the node.

Panic Occurs when Executing cxfs_cluster on an AIX Node

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 fileset.

A Memory Error Occurs with cp -p on an AIX Node

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 an AIX Node

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

Common Linux 32-bit Problems

The kernels provided for the Linux 32-bit 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

Common Mac OS X Problems

The `cxfs_client` Service is Not Started on a MAC OS X Node

The `cxfs_client` service might not start if the license 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 check fails:

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

The Mac OS X Node Does Not Mount Any Filesystems

The Mac OS X node might not mount any filesystems for the following reasons:

- The Mac OS X `cxfs_client` service may not be running. Verify that `cxfs_client` is running by executing the following:

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

- The cluster membership (`cms`), XVM, or the filesystems are not 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 in all these states, 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? See "Configuring Hostnames on Mac OS X" on page 69.
    - Has the node has been added to the cluster and enabled? See "Verifying the Cluster" on page 165.

  – 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.

Common Solaris Problems

If the filesystem does not mount on a Solaris node, verify that the LUNs used by this filesystem have been mapped in the EZ Fibre configuration utility. See "Installing and Running the EZ Fibre Configuration GUI" on page 95.

Common Windows Problems

This section contains the following common Windows problems:

- "`cxfs_client` Cannot Map Users other than `Administrator` on a Windows Node"
- "Filesystems Are Not Displayed on a Windows Node" on page 185

`cxfs_client` Cannot Map Users other than `Administrator` on a Windows Node

If `cxfs_client` cannot map any users other than `Administrator` and there are no LDAP errors in the `cxfs_client` log file, you must change the configuration to allow reading of the attributes. To do this, 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 on 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 a Windows Node

If the CXFS drive letter is visible in Windows Explorer but no filesystems are mounted, do the following:

- Run `C:\Program Files\CXFS\cxfs_info` to ensure that the filesystems have been configured for this node.
- Verify the filesystems that should be mounted by using the `cmgr` command on a CXFS administration node. For more information, see "Mounting Filesystems on the Client-Only Nodes" on page 164.
- Ensure that the CXFS metadata server is up and that the Windows node is in the cluster membership; see "Verifying the Cluster" on page 165.
- Check that the CXFS Client service has started. See "Is the CXFS Software Running Correctly on the Windows Node?" on page 172 and "Manual CXFS Startup/Shutdown for Windows" on page 154.
- Check the following file for warnings and errors regarding licenses or mounting filesystems:

  ```
  C:\Program Files\CXFS\log\cxfs_client.log
  ```

- 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:

```
C:\Program Files\CXFS\log\cxfs_client.log
```

This log file may become quite large over a period of time if the verbosity level is increased. The service does not perform any automatic log rotation, so the service must be stopped in order to move or truncate this file, then restarted. See "Manual CXFS Startup/Shutdown for Windows" on page 154 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 of the Windows Node

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 (line break added here for readability):

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/winxppro/reskit/prmd_stp_fztl.asp

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 186
- "Reporting Linux 32-bit Problems" on page 188
- "Reporting Mac OS X Problems" on page 189
- "Reporting Solaris Problems" on page 190
- "Reporting Windows Problems" on page 191

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 `/var/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.
- 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 an AIX node, it stores information in `/var/cxfsdump-data/`*nodename*`.tar.gz`

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

Reporting Linux 32-bit Problems

When reporting a problem about a Linux 32-bit node to SGI, you should retain the following information:

- The kernel you are running:

  ```
  [root@linux32 root]# uname -a
  ```

- The CXFS packages you are running:

  ```
  [root@linux32 root]# rpm -q cxfs_client cxfs-modules cxfs_utils xvm-cmds
  ```

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

  ```
  [root@linux32 root]# cat /proc/cpuinfo
  ```

- The hardware installed on your machine:

  ```
  [root@linux32 root]# /sbin/lspci
  ```

- Modules that are loaded on your machine:

  ```
  [root@linux32 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 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 32-bit 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@linux32 root]# echo 1 > /proc/sys/kernel/kdb
```

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

Reporting Mac OS X Problems

When reporting a problem about a CXFS Mac OS X node to SGI, you should gather the following information:

- Panic log: `/Library/Logs/panic.log` (if there is a system panic)
- CXFS client log: `/var/log/cxfs_client`
- System log: `/var/log/system.log`
- Console log: `/var/tmp/console.log`

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`.

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

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, you should retain the following information:

- The configuration of the machine. Select the following:

  Start
  > Programs
  > Accessories
  > System Tools
  > System Information
  > Action
  > Save As System Information File

  This will create a file that describes all of the installed hardware and configured drivers on the machine.

  Alternatively, you could dump information about each item in the hardware tree to a text file by using the following selection:

  Action
  > Save As Text File

  However, you must repeat this action for each item.

- The build date and firmware versions. Using Windows Explorer, open the following directory:

  ```
  C:\Winnt\system32\drivers
  ```

Then do the following:

– Right-click on `cxfs.sys` and select the following:

Properties
> Version

Record the values of **BuildDate** and **Product Version**.

– Right click on `ql2200.sys` and select the following:

Properties
> Version

Record the values of **Firmware** and **Product Version**.

- The contents of the following file:

  `C:\Program Files\CXFS\log\cxfs_client.log`

  Compress this file with `winzip` if it is large.

- The contents of the crash dump if one was generated. Compress this file with `winzip`. For more information, see "Save Crash Dumps for Windows" on page 192.

- Output about the cluster obtained from the `cxfsdump` utility run 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. On Windows, 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 192), 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:** `l`

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

Operating System Path Differences

This appendix lists the location of CXFS-specific commands and files. For more information, see the `cxfs_client` man page.

Table A-1 AIX Paths

Component	Path
Client service:	`/usr/cxfs_cluster/bin/cxfs_client`
Command that normally invokes the client service:	`/etc/init.d/cxfs_client`
Log file:	`/var/tmp/cxfs_client`
Options file:	`/usr/cxfs_cluster/bin/cxfs_client.options`
CXFS status:	`/usr/cxfs_cluster/bin/cxfs_info`

Table A-2 Linux 32-bit Paths

Component	Path
Client service:	`/usr/cluster/bin/cxfs_client`
Command that normally invokes the client service:	`/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`

Table A-3 Mac OS X Paths

Component	Path
Client service:	`/usr/cluster/bin/cxfs_client`
Command that normally invokes the client service:	`/Library/StartupItems/cxfs/cxfs`
Log file:	`/var/log/cxfs_client`
Options file:	`/usr/cluster/bin/cxfs_client.options`
CXFS status:	`/usr/cluster/bin/cxfs_info`

Table A-4 Solaris Paths

Component	Path
Client service:	`/usr/cxfs_cluster/bin/cxfs_client`
Command that normally invokes the client service:	`/etc/init.d/cxfs_client`
Log file:	`/var/log/cxfs_client`
Options file:	`/usr/cxfs_cluster/bin/cxfs_client.options`
CXFS status:	`/usr/cxfs_cluster/bin/cxfs_info`

Table A-5 Windows Paths

Component	Path
Client service:	
Windows 2000:	`C:\Winnt\system32\cxfs_client.exe`
Windows XP:	`C:\Windows\system32\cxfs_client.exe`
Command that normally invokes the client service:	See "Manual CXFS Startup/Shutdown for Windows" on page 154
Log file:	`C:\Program Files\CXFS\log\cxfs_client.log`
Options file:	"Modifying the CXFS for Windows Software" on page 155
CXFS status:	`C:\Program Files\CXFS\cxfs_info`

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 Version 2 Software Installation and Administration Guide*.

- 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 *SGI InfiniteStorage CXFS Administration Guide* that support CXFS 3.0 for SGI ProPack 2.3 for Linux.

Glossary

active metadata server

A server-capable administration node chosen from the list of potential metadata servers. There can be multiple active metadata servers, one for each 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.

CIFS

Microsoft's Common Internet File System, which is used for exporting filesystems to other Windows hosts

client

See *CXFS client*.

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.

cluster database

Contains configuration information about nodes, filesystems, and the cluster. The database is managed by the `fs2d` daemon and is stored on administration nodes.

cluster ID

A unique number within your network in the range 1 through 128. The cluster ID is used by the 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 that share a network must have unique names and IDs.

cluster node

A node that is defined as part of the cluster.

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.

A node's control networks should not be set to accept control messages if the node is not a dedicated CXFS node. Otherwise, end users who run other jobs on the machine can have their jobs killed unexpectedly when CXFS resets the node.

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.

client-only node

A node that is part of the cluster but is not server capable.

CXFS client

A node that is part of the cluster and is a potential metadata server, but is currently not acting as the active metadata server. See also and *client-only node*.

CXFS database

See *cluster database*.

CXFS 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 membership. Once accepted, the node can share the filesystems of the cluster. (Also known as *kernel-space membership*.)

database
See *cluster database*.

GUI
Graphical user interface.

heartbeat messages
Messages the cluster software sends between the nodes that indicate a node is operational. Heartbeat messages and control messages are sent through the node's network interfaces that have been attached to a control network.

I/O fencing
The failure action 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.

membership
See *CXFS 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 metadata on behalf of all nodes in a cluster. There can be multiple potential metadata servers, but only one is chosen to be the active metadata server for any one filesystem. See also *active metadata server* and *potential metadata server*.

multiOS cluster
A cluster that is running multiple operating systems, such as IRIX and Solaris.

node

A node is an operating system (OS) image, usually an individual computer. (This use of the term node does not have the same meaning as a node in an SGI Origin 3000 or SGI 2000 system.) A given node can be a member of only one pool and therefore only one cluster.

node membership

The list of nodes that are active (have CXFS membership) in a cluster.

pool

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

A pool is formed when you connect to a given node and define that node in the cluster database using the 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).

potential metadata server

A server-capable administration node that is listed in the metadata server list when defining a filesystem; there can be multiple potential metadata servers, but only one node in the list will be chosen as the active metadata server for one filesystem.

recovery

The process by which the metadata server moves from one node to another due to an interruption in services on the first node.

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.

SAN

Storage area network: a high-speed, scalable network of servers and storage devices that provides storage resource consolidation, enhanced data access/availability, and centralized storage management.

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.

tree view

The portion of the CXFS GUI window that displays components graphically.

quorum

The number of nodes required to form a cluster.

Index

C

D

K

L

M

Q

R

T

U

V