Netra™ CT 820 Server System Administration Guide

Sun Microsystems, Inc.
www.sun.com

Part No. 817-2647-12
July 2004, Revision A

Submit comments about this document at: `http://www.sun.com/hwdocs/feedback`

Adobe PostScript™

Contents

Figures

Tables

Preface

The *Netra CT 820 Server System Administration Guide* contains configuration and administration information for system administrators of the Netra™ CT 820 server. This manual assumes you are familiar with UNIX® commands and networks.

How This Book Is Organized

Chapter 1 contains an introduction to the Netra CT software.

Chapter 2 contains information on configuring your system.

Chapter 3 describes how to administer your system.

Appendix A contains information on third-party node boards.

Appendix B contains information on error messages.

Using UNIX Commands

This document might not contain information on basic UNIX commands and procedures such as shutting down the system, booting the system, and configuring devices. See the following for this information:

- Software documentation that you received with your system
- Solaris™ Operating System (Solaris OS) documentation, which is at:

  `http://docs.sun.com`

Shell Prompts

Shell	Prompt
C shell	*machine-name*`%`
C shell superuser	*machine-name*`#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell superuser	`#`

Typographic Conventions

Typeface[*]	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`

`AaBbCc123`	What you type, when contrasted with on-screen computer output	`%` **`su`**
		`Password:`

AaBbCc123	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the *User's Guide.* These are called *class* options. You *must* be superuser to do this. To delete a file, type `rm` *filename*.

[*] The settings on your browser might differ from these settings.

Related Documentation

The Netra CT 820 server documentation is listed in the following table.

Title	Part Number
Netra CT 820 Server Product Overview	817-2643
Netra CT 820 Server Installation Guide	817-2641
Netra CT 820 Server Service Manual	817-2642
Netra CT 820 Server System Administration Guide	817-2647
Netra CT 820 Server Safety and Compliance Manual	817-2645
Netra CT 820 Server Software Developer's Guide	817-2648
Netra CT 820 Server Documentation Note	817-1907
Netra CT 820 Server Release Notes	817-2646

You might want to refer to documentation on the following products for additional information: the Solaris OS, the ChorusOS™ software, OpenBoot™ PROM firmware, and the Netra CP2300 cPSB board.

Accessing Sun Documentation Online

You can view, print, or purchase a broad selection of Sun documentation, including localized versions, at:

`http://www.sun.com/documentation`

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in this document, go to:

`http://www.sun.com/service/contacting`

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to:

`http://www.sun.com/hwdocs/feedback`

Please include the title and part number of your document with your feedback:

Netra CT 820 Server System Administration Guide, part number 817-2647-12.

CHAPTER 1

Introduction

This chapter includes the following sections:

- "Overview of Netra CT Server Software and Hardware" on page 1
- "System Administration Tasks" on page 5

Overview of Netra CT Server Software and Hardware

The Netra CT 820 server software can be categorized as follows:

- Operating systems and applications
- Firmware
- Network support

The software is described in TABLE 1-1 and represented logically, with the hardware, in FIGURE 1-1.

TABLE 1-1 Netra CT Server Software for System Administrators

Category	Name	Description
Operating Systems and Applications	Solaris Operating System (Solaris OS)	The Solaris OS runs on Sun supported compact packet-switched backplane (cPSB)-only node boards, such as the Netra CP2300 cPSB board. It is installed by the user.
	ChorusOS software	The ChorusOS software runs on the distributed management cards. It is factory-installed.
	Command-line interface (CLI)	The CLI is the primary user interface to the distributed management cards.

TABLE 1-1 Netra CT Server Software for System Administrators *(Continued)*

Category	Name	Description
	Managed Object Hierarchy (MOH)	Management application that monitors and manages the field-replaceable units (FRUs) in your system. It provides support for high-availability services and applications.
	Processor Management Service (PMS)	Management application that provides support for high-availability services and applications.
Firmware	OpenBoot PROM firmware	Firmware on a Sun supported cPSB-only node board, such as the Netra CP2300 cPSB board, that controls booting. It includes diagnostics.
	Boot control firmware (BCF)	Firmware on the distributed management cards that performs power-on self-test (POST) and controls booting of the distributed management card software.
	Baseboard management controller (BMC) firmware	Baseboard management controller firmware enables communication over the Intelligent Platform Management Interface (IPMI) on the distributed management cards.
	System management controller (SMC) firmware	System management controller firmware enables communication over the IPMI controller on a Sun supported cPSB-only node board, such as the Netra CP2300 cPSB board.
Network Interfaces	Internal Ethernet networks	The two internal Ethernet networks make up the cPSB bus.
	System management network	The system management network is a communication channel over the cPSB bus. It is used by the Netra CT 820 management software to communicate between the distributed management cards, Sun supported cPSB-only node boards, such as the Netra CP2300 cPSB board, and the switching fabric boards.

The Netra CT 820 system has two distributed management cards. You use the *active* distributed management card (when you power on the system, by default, the top card in slot 1A) for system-level configuration, administration, and management of most of the components connected to the midplane. The *standby* distributed management card (when you power on the system, by default, the bottom card in slot 1B) provides redundancy and failover capability for the active distributed management card.

The switching fabric boards connect the distributed management card and the node boards internally, and have Ethernet ports on the rear for external connectivity.

Sun supported cPSB-only node boards, such as the Netra CP2300 cPSB board, accept and own peripherals, such as disks. The node boards also run user applications. In a Netra CT 820 server, each node board runs its own copy of an operating system, and each is therefore considered a server. The distributed management cards, the node boards, the switching fabric boards, and the other system FRUs make up a system.

Note – In this manual, the use of the term *node board* refers to a Sun supported cPSB-only board, such as the Netra CP2300 cPSB board, unless otherwise specified.

Third-party cPSB-only node boards that are PICMG 2.16-compliant may be used in the Netra CT 820 server. These boards do not necessarily run the Solaris Operating System, and they do not run the Netra CT 820 server system management software, such as MOH. Because of this, they cannot be managed to the same extent as the Netra CP2300 cPSB board. Refer to Appendix A, "Third-Party Node Boards" on page 87 for information on these boards.

TABLE 1-2 contains a summary of how you can access the various boards. The distributed management card supports 22 sessions (Tip and Telnet connections) at once.

TABLE 1-2 Netra CT 820 System Board Access Methods

Board	Access Methods
Distributed management card (slot 1A and slot 1B)	• 1 rear serial port (console) for Tip or ASCII terminal connection • 1 front serial port (console) for Tip or ASCII terminal connection Note that either the rear or front serial port can be used, but not both at the same time. If you connect a cable to both ports, only the front port is active. • 1 rear external Ethernet port for Telnet connection • 1 internal Ethernet port for Telnet connection through a switching fabric board • Remote shell from a node board, using the `rsh` command
Switching fabric board (slots 2 and 21)	• Multiple rear Ethernet ports for Telnet connection
Node board (Sun supported cPSB-only boards) (slots 3 through 20)	For the Netra CP2300 cPSB board: • 2 rear serial ports (console) for Tip or ASCII terminal connection • 1 front serial port (console) for Tip or ASCII terminal connection • Console command from the distributed management card CLI • Telnet connection through the switching fabric board, specifying the IP address of the node board
Third-party cPSB-only node boards (slots 3 through 20)	Third-party board dependent.

The hardware interfaces include the Intelligent Platform Management Interface (IPMI), the compact packet-switched backplane (cPSB) bus, and the network interface on the distributed management cards, the node boards, and the switching fabric boards.

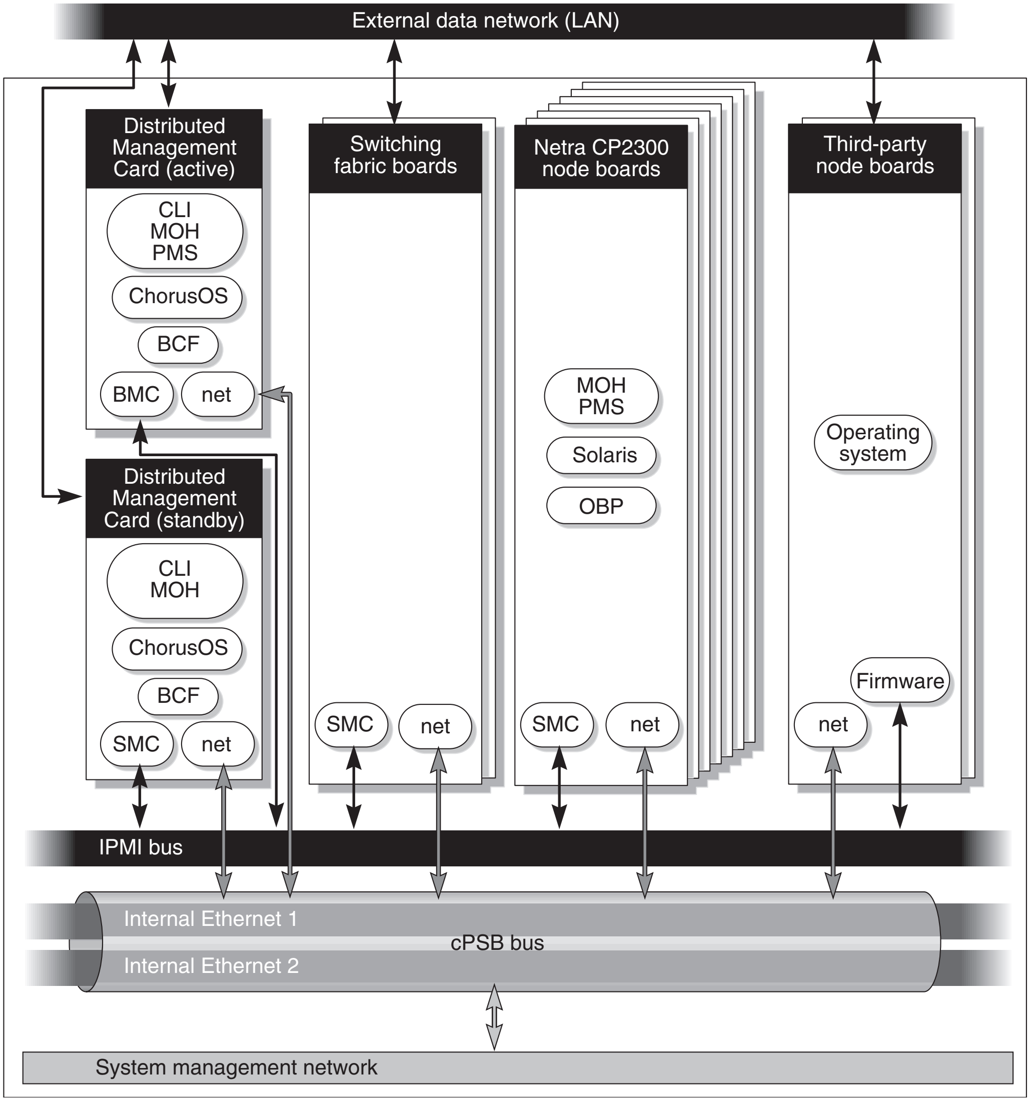

FIGURE 1-1 Logical Representation of Software and Hardware Interfaces in a Netra CT Server

System Administration Tasks

Netra CT 820 server system administration typically includes installation, configuration, and administration tasks.

Solaris administration on the Netra CT 820 server, including adding Solaris user accounts, is performed by logging into the node board. Netra CT 820 server administration is performed by logging into the distributed management card and using the distributed management card CLI. The distributed management card can be used as the single point of entry in the Netra CT system for configuration and administration purposes.

System administration tasks are described in the following chapters.

CHAPTER 2

Configuring Your System

This chapter assumes you have already installed the Solaris Operating System and the required patches on your Netra CT 820 node boards.

You configure the Netra CT 820 system primarily through the active distributed management card command-line interface (CLI). The active distributed management card CLI enables system-level configuration, administration, and management that includes the node boards, the switching fabric boards, the distributed management card, power supplies, and fan trays. The distributed management card CLI interface can be used both locally and remotely.

You configure the distributed management cards first, then the node boards, then the system-wide applications.

This chapter includes the following sections:

Accessing the Distributed Management Cards

When you initially access either distributed management card, you must do so over the serial port (console), using an ASCII terminal or the Tip program. When you first access the distributed management card, log in with the default user account of `netract` and the password `suncli1`. This account is set to full authorization (permissions). This account can not be deleted. However, you should change the password on this account for security purposes, before your Netra CT 820 server is operational.

The following sections provide information on configuring the distributed management cards' Ethernet ports and setting up user accounts and passwords using the distributed management card CLI. For more information on using the distributed management card CLI, refer to Chapter 3.

Each distributed management card supports 22 sessions (Tip and Telnet connections) at once. The active distributed management card is identified by the prompt *hostname* (Active *slot*#) cli> and the standby distributed management card is identified by the prompt *hostname* (Standby *slot*#) cli>.

Note – The term *distributed management card* as used in this manual refers to either the active or standby distributed management card unless otherwise specified. In this manual, the prompt for both is shortened to *hostname* cli>.

Configuring the Distributed Management Cards' Ethernet Ports

Each distributed management card has one external Ethernet port on the rear transition card, labeled SRVC LAN, and one internal Ethernet port. If you configure these ports, you can access the distributed management cards using a Telnet connection to the external Ethernet port or using a Telnet connection through the switching fabric board to the internal Ethernet port.

To configure the Ethernet ports, you must be logged in to the distributed management card with a user account that has full permissions. You configure the ports with CLI commands, and then reset the distributed management card for the changes to take effect. Use the following procedure for each distributed management card.

Note – The external Ethernet interface on the distributed management card and the external Ethernet interface on the switching fabric board must be connected to different subnets. If they are configured on the same subnet, `arp` messages are displayed on the distributed management card console.

To Configure the Distributed Management Cards' Ethernet Ports

1. **Log in to the distributed management card.**

2. **Set the IP mode:**

```
hostname cli> setipmode -b port_num rarp|config|none
```

   where *port_num* is 1 for the external Ethernet port or 2 for the internal Ethernet port. Choose the IP mode according to the services available in the network (`rarp`, `config`, or `none`). The default is `none`.

   If you set the IP mode to `rarp`, skip to Step 5.

3. **Set the IP address:**

```
hostname cli> setipaddr -b port_num addr
```

   where *port_num* is 1 for the external Ethernet port or 2 for the internal Ethernet port. Set the IP address of the distributed management card. The default is 0.0.0.0. This command is only used if the `ipmode` is set to `config`.

4. **Set the IP netmask:**

```
hostname cli> setipnetmask -b port_num addr
```

   where *port_num* is 1 for the external Ethernet port or 2 for the internal Ethernet port. Set the IP netmask of the distributed management card. The default is 255.255.255.0. This command is only used if the `ipmode` is set to `config`.

5. **Set the IP gateway:**

```
hostname cli> setipgateway addr
```

Set the IP gateway of the distributed management card to access the system from outside the subnet. The default is 0.0.0.0.

6. **Reset the distributed management card:**

```
hostname cli> reset dmc
```

Setting Up User Accounts on the Distributed Management Card

User accounts are set up using the distributed management card CLI. The default user account is `netract` and the password is `suncli1`. This account is set to full authorization (permissions). This account can not be deleted. However, you should change the password on this account for security purposes, before your Netra CT 820 server is operational.

User information is entered on the active distributed management card, and immediately *mirrored*, or shared, on the standby distributed management card. The distributed management card supports 16 accounts with passwords.

▼ To Set Up a User Account

1. **Log in to the active distributed management card.**

2. **Add a user:**

```
hostname cli> useradd username
```

3. **Add a password for that user:**

```
hostname cli> userpassword username
```

By default, new accounts are created with read-only permission. Permission levels can be changed using the `userperm` command. Refer to "CLI Commands" on page 52 for more information about permissions and the `userperm` command.

Username Restrictions

The username field has a maximum length of 16 characters. It must contain at least one lowercase alphabetic character, and the first character must be alphabetic.

Valid characters for *username* include:

- Alphabetic characters
- Numeric characters
- Period (.)
- Underscore (_)
- Hyphen (-)

Password Restrictions

Passwords have the following restrictions:

- They must contain at least six characters but not more than eight characters. Only the first eight characters are considered if the password is longer than eight characters.
- They must contain at least two alphabetic characters and at least one numeric or special character. Alphabetic characters can be both uppercase and lowercase. Special characters include printable characters, such as ! @ # $ % ^ & and *.
- They must differ from the user's login name and any reverse or circular shift of that login name. For comparison purposes, uppercase and lowercase letters are equivalent.
- The new password must differ from the old by at least three characters. For comparison purposes, uppercase and lowercase letters are equivalent.

Specifying Netra CT Server FRU ID Information

A field-replaceable unit (FRU) is a module or component that can typically be replaced in its entirety as part of a field service repair operation.

The Netra CT system FRUs include:

- Node boards
- Distributed management cards
- Switching fabric boards
- Power supplies
- Fan trays
- Midplane

All FRUs except power supplies contain *FRU ID* (identification) information that includes FRU manufacturing and configuration data. This information can be displayed through the distributed management card CLI (see TABLE 2-2). The Netra CT 820 system supports two FRU ID formats:

- The Sun FRU ID format, available on the distributed management cards and on the node board (Netra CP2300 cPSB board)
- The industry standard IPMI format, available on the midplane, switching fabric boards, fan trays, and third-party node boards conforming to the PICMG 2.9 specification

In addition, you can enter certain FRU ID information through the active distributed management card CLI, which is stored in the midplane. Note that you can also enter FRU ID information through the MOH application; refer to the *Netra CT Server Developer's Guide* for instructions. FRU ID information includes:

- Allowable plug-in boards (a default exists) and boot devices (a default exists in OpenBoot PROM) for the slots
- The system management network configuration (a default exists)
- System location information, customer data information, and user label information (there are no defaults; these are optional entries)

Some of this information is used by the MOH application to audit board insertions and prevent misconfigurations, and to display information; some is used by the system management network.

The format of the information to be specified is:

```
hostname cli> setfru fru_name instance fru_property value
```

The FRU instance is a logical number; it matches the slot number only for the slot FRU name. The FRU property is case-insensitive.

TABLE 2-1 shows the FRU ID information that can be specified with the CLI `setfru` command.

TABLE 2-1 FRU ID Information Specified Using the `setfru` Command

FRU Name	Instance	FRU Property	Value	Description
midplane	1	SysmgtbusIPSubnet	*IP subnet address (hexadecimal)*	Specify the IP subnet address for the system management network. The default is `0xc0a80d00` (192.168.13).
midplane	1	SysmgtbusIPSubnetMask	*IP subnet mask (hexadecimal)*	Specify the IP subnet mask for the system management network. The default is `0xffffffe0` (255.255.255.224).
midplane	1	Location	*text description*	A description of the location (for example, the number on the chassis label) of the Netra CT system. This description is used in the MOH application. The text can be up to 80 characters in length.
midplane	1	User_Label	*text description*	Any customer-supplied information. The text can be up to 10 characters in length.
dmc	1 or 2	Cust_data	*text description*	Any customer-supplied information. The text can be up to 80 characters in length. FRU instance 1 is the DMC in slot 1A; FRU instance 2 is the DMC in slot 1B.
slot	2 to 21	Acceptable_Fru_Types	*vendor:partnumber*	First, specify the chassis slot number to be configured. (Slots are numbered starting from the left.) Second, specify the allowable plug-in board(s) for that slot, where the value is the vendor name and part number (separated by a colon) of the board. Use the `showfru` command to display this information. Multiple boards may be specified, separated by a semi-colon (;). The default is to power on all Sun supported cPSB-only boards.

TABLE 2-1 FRU ID Information Specified Using the `setfru` Command *(Continued)*

FRU Name	Instance	FRU Property	Value	Description
slot	3 to 20	Acceptable_Fru_Types	`nonsun:picmg2.16`	This information applies to third-party node boards only. First, specify the chassis slot number to be configured. (Slots are numbered starting from the left.) Second, specify the value `nonsun:picmg2.16`, which indicates that a third-party node board is allowed in this slot.
slot	3 to 20	Boot_Devices	*boot_device_list*	First, specify the chassis slot number to be configured (Slots are numbered starting from the left.) Second, specify the alias(es) listing the devices and/or full device path names the board in this slot will boot from. The *boot_device_list* can be up to 16 characters in length. When the board in this slot is powered up, this information overwrites the entry in the OpenBoot PROM `boot-device` NVRAM configuration variable. Specifying "" (the null string) defaults to the OpenBoot PROM NVRAM setting.
slot	3 to 20; all	Boot_Mask	`true` or `false`	First, specify the chassis slot number to be configured (slots are numbered starting from the left) or `all` to refer to all configurable slots. Second, specify whether the board in this slot is a boot server for the system. The default is `false`, which means that the board is not a boot server. Refer to "Configuring a Node Board as a Boot Server" on page 20 for instructions on setting the boot mask for a slot.
slot	3 to 20	Cust_Data	*text description*	First, specify the chassis slot number to be configured (slots are numbered starting from the left). Second, specify any customer-supplied information. The text can be up to 80 characters in length.

Changes to FRU ID fields through the CLI setfru command require you to completely power the system off and on for the changes to take effect. It is recommended that you enter all necessary FRU ID information, then power the system off and on.

Displaying Netra CT Server FRU ID Information

FRU ID information entered during the manufacturing process and through the active distributed management card CLI setfru command can be displayed using the showfru command.

TABLE 2-2 shows the FRU ID information that can be displayed with the CLI showfru command. Use the FRU property to specify the information you want; the FRU property is case-insensitive.

TABLE 2-2 FRU ID Information Displayed Using the showfru Command

FRU Name	Instance	FRU Property	Description
midplane	1	Sun_Part_No	Display the part number for the midplane.
midplane	1	Sun_Serial_No	Display the serial number for the midplane.
midplane	1	SysmgtbusIPSubnet	Display the system management network IP subnet address in hexadecimal format for this system.
midplane	1	SysmgtbusIPSubnetMask	Display the system management network IP subnet mask in hexadecimal format for this system.
midplane	1	Vendor_Name	Display the vendor name for the midplane.
midplane	1	Fru_Shortname	Display the FRU short name for the midplane.
midplane	1	Location	Display any customer-supplied text specified for the Location of this system.
midplane	1	User_Label	Display any customer-supplied text for this field.
dmc	1 or 2	Sun_Part_No	Display the part number for the distributed management card in a particular slot. FRU instance 1 is the DMC in slot 1A; FRU instance 2 is the DMC in slot 1B.

TABLE 2-2 FRU ID Information Displayed Using the `showfru` Command *(Continued)*

FRU Name	Instance	FRU Property	Description
dmc	1 or 2	Sun_Serial_No	Display the serial number for the distributed management card in a particular slot. FRU instance 1 is the DMC in slot 1A; FRU instance 2 is the DMC in slot 1B.
dmc	1 or 2	Vendor_Name	Display the vendor name for the distributed management card in a particular slot. FRU instance 1 is the DMC in slot 1A; FRU instance 2 is the DMC in slot 1B.
dmc	1 or 2	Fru_Shortname	Display the FRU short name for the distributed management card in a particular slot. FRU instance 1 is the DMC in slot 1A; FRU instance 2 is the DMC in slot 1B.
dmc	1 or 2	Initial_HW_Dash_Level	Display the initial hardware dash level of the distributed management card in a particular slot. FRU instance 1 is the DMC in slot 1A; FRU instance 2 is the DMC in slot 1B.
dmc	1 or 2	Initial_HW_Rev_Level	Display the initial hardware revision level of the distributed management card in a particular slot. FRU instance 1 is the DMC in slot 1A; FRU instance 2 is the DMC in slot 1B.
dmc	1 or 2	Cust_Data	Display any customer-supplied text for this field for the distributed management card in a particular slot. FRU instance 1 is the DMC in slot 1A; FRU instance 2 is the DMC in slot 1B.
slot	3 to 20	Sun_Part_No	Display the part number for the board in a particular slot.
slot	3 to 20	Part_No	Display the part number for the third-party node board in a particular slot.
slot	3 to 20	Sun_Serial_No	Display the serial number for the board in a particular slot.
slot	3 to 20	Serial_No	Display the serial number for the third-party node board in a particular slot.
slot	2 to 21	Acceptable_Fru_Types	Display the allowable plug-in boards for a particular slot.
slot	3 to 20	Boot_Devices	Display the boot devices for a particular slot.
slot	3 to 20	Boot_Mask	Display whether or not the board in a particular slot is a boot server for the system.
slot	3 to 20	Vendor_Name	Display the vendor name for the board in a particular slot.

TABLE 2-2 FRU ID Information Displayed Using the `showfru` Command *(Continued)*

FRU Name	Instance	FRU Property	Description
slot	3 to 20	Fru_Shortname	Display the FRU short name for the board in a particular slot.
slot	3 to 20	Initial_HW_Dash_Level	Display the initial hardware dash level of the board in a particular slot.
slot	3 to 20	Initial_HW_Rev_Level	Display the initial hardware revision level of the board in a particular slot.
slot	3 to 20	Cust_Data	Display any customer-supplied text for this field for the board in a particular slot.
switch	1 or 2	Sun_Part_No	Display the part number for the specified switching fabric board. FRU instance 1 is the switch in slot 2; FRU instance 2 is the switch in slot 21.
switch	1 or 2	Sun_Serial_No	Display the serial number for the specified switching fabric board. FRU instance 1 is the switch in slot 2; FRU instance 2 is the switch in slot 21.
switch	1 or 2	Vendor_Name	Display the vendor name for the specified switching fabric board. FRU instance 1 is the switch in slot 2; FRU instance 2 is the switch in slot 21.
switch	1 or 2	Fru_Shortname	Display the FRU short name for the specified switching fabric board. FRU instance 1 is the switch in slot 2; FRU instance 2 is the switch in slot 21.
fantray	1 to 3	Sun_Part_No	Display the part number for the specified fan tray.
fantray	1 to 3	Sun_Serial_No	Display the serial number for the specified fan tray.
fantray	1 to 3	Vendor_Name	Display the vendor name for the specified fan tray.
fantray	1 to 3	Fru_Shortname	Display the FRU short name for the specified fan tray.

▼ To Display FRU ID Information

1. **Log in to the distributed management card.**

2. **Enter the** `showfru` **command:**

```
hostname cli> showfru fru_name instance fru_property
```

Refer to TABLE 2-2 for allowable information for each variable. For example, if you want to display the part number FRU ID information for fan tray 1, enter the following:

```
hostname cli> showfru fantray 1 Sun_Part_No
```

Use the FRU target "slot" to display information for the node boards. For example, to display part number FRU ID information for a board in slot 8, enter the following:

```
hostname cli> showfru slot 8 Sun_Part_No
```

The next several sections describe the configurations you can set by entering FRU ID information.

Configuring a Chassis Slot for a Board

You can specify the type of board that is allowed in a given chassis slot using the active distributed management card CLI. The slot usage information is used by the distributed management card software to audit board insertions and prevent misconfigurations. You can also specify the boot device for the slot, that is, the path to the device the board in the slot boots from. When the board is powered on, the FRU boot device information overwrites the entry in the OpenBoot PROM `boot-device` NVRAM configuration variable on that board. The chassis slot information can be changed at any time using the active distributed management card CLI.

By default, slots are configured to accept Sun supported cPSB-only board FRUs unless you specifically set an allowable plug-in for a specific slot. The exceptions are: for a Netra CT 820 server, the distributed management cards must be in slot 1A and 1B and the switching fabric boards must be in slots 2 and 21.

To set allowable plug-ins for a particular slot, you need the vendor name and the part number of the board. This FRU ID information can be displayed using the CLI `showfru` command. See "Displaying Netra CT Server FRU ID Information" on page 15 for more information.

▼ To Configure a Chassis Slot for a Board

1. **Log in to the active distributed management card.**

2. **Set the acceptable FRUs for the slot:**

```
hostname cli> setfru fru_name instance fru_property value
```

Refer to TABLE 2-1 for allowable information for each variable. For example, if you want to set chassis slot 5 to allow only a Sun Microsystems (vendor 003E) particular CPU board (part number 595-5769-03), enter the following:

```
hostname cli> setfru slot 5 Acceptable_Fru_Types 003E:595-5769-03
```

Multiple boards can be specified for one slot. Separate the boards with a semi-colon. You can also use the asterisk (*) as a wild card in the part number to allow multiple boards. For example, if you want to set chassis slot 4 to allow only boards from three particular vendors, with multiple board part numbers from one vendor, enter the following:

```
hostname cli> setfru slot 4 Acceptable_Fru_Types 003E:*;0004:1234-
5678-1;0001:8796541-02
```

3. **Set the boot device for the slot:**

```
hostname cli> setfru fru_name instance fru_property value
```

Refer to TABLE 2-1 for allowable information for each variable. For example, if you want to set chassis slot 5 to boot from a device on the network, enter the following:

```
hostname cli> setfru slot 5 Boot_Devices boot_device_list
```

where *boot_device_list* is the alias or aliases specifying the boot devices for example, `disk net`. The *boot_device_list* is limited to 25 bytes.

4. **Completely power off and on the system by locating the power switch at the rear of the Netra CT 820 server; press it to the Off (O) position, then press it to the On (|) position.**

Configuring a Node Board as a Boot Server

You can configure a node board (Sun supported cPSB-only boards) to be a boot server for the Netra CT 820 system. To do this, you use the Boot_Mask field in the midplane FRU ID. When the system is powered on, the distributed management card looks at the Boot_Mask field; if a boot server has been specified, the distributed management card powers on that node board first. There can be any number of boot servers per Netra CT 820 system. If multiple boot servers are specified, all boot servers are powered on simultaneously.

▼ To Configure a Node Board as a Boot Server

1. **Log in to the active distributed management card.**

2. **Specify which slot contains a node board boot server by setting the Boot_Mask:**

```
hostname cli> setfru fru_name instance fru_property value
```

Refer to TABLE 2-1 for allowable information for each variable. For example, if you want to specify chassis slot 3 as a node board boot server, enter the following:

```
hostname cli> setfru slot 3 Boot_Mask true
```

To specify all slots (3 to 20) as boot servers, enter the following:

```
hostname cli> setfru slot all Boot_Mask true
```

To clear all slots (3 to 20) as boot servers, enter the following:

```
hostname cli> setfru slot all Boot_Mask false
```

3. **Completely power off and on the system by locating the power switch at the rear of the Netra CT 820 server; press it to the Off (O) position, then press it to the On (|) position.**

Configuring the System Management Network

The system management network provides a communication channel over the midplane. It is used to communicate between the distributed management cards, the node boards, and the switching fabric boards. FIGURE 2-1 shows the physical connections between boards over the midplane in the Netra CT 820 system.

The network appears as any other generic Ethernet port in the Solaris Operating System, and is configured by default on Solaris OS and on the distributed management cards. The system management network is used by the applications and features, such as MOH, PMS, and console connections from the distributed management cards to node boards.

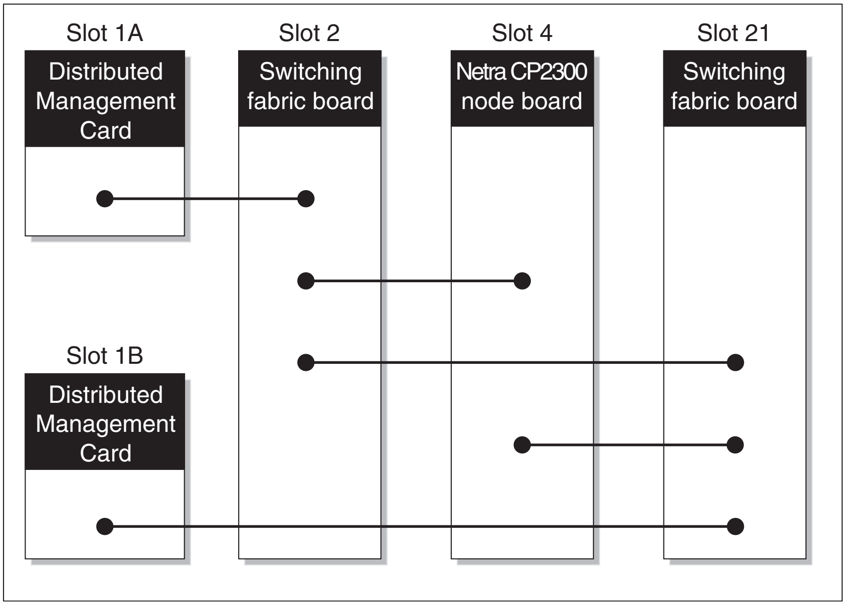

FIGURE 2-1 System Management Network Physical Connectivity over the cPSB Bus

The system management network consists of two virtual local area networks (VLANs) running over the two internal Ethernet interfaces, and a logical Carrier Grade Transport Protocol (CGTP) interface. The two VLANs and the logical CGTP interface allow distributed management card and switching fabric board redundancy. FIGURE 2-2 shows the VLAN traffic over the physical connectivity shown in FIGURE 2-1.

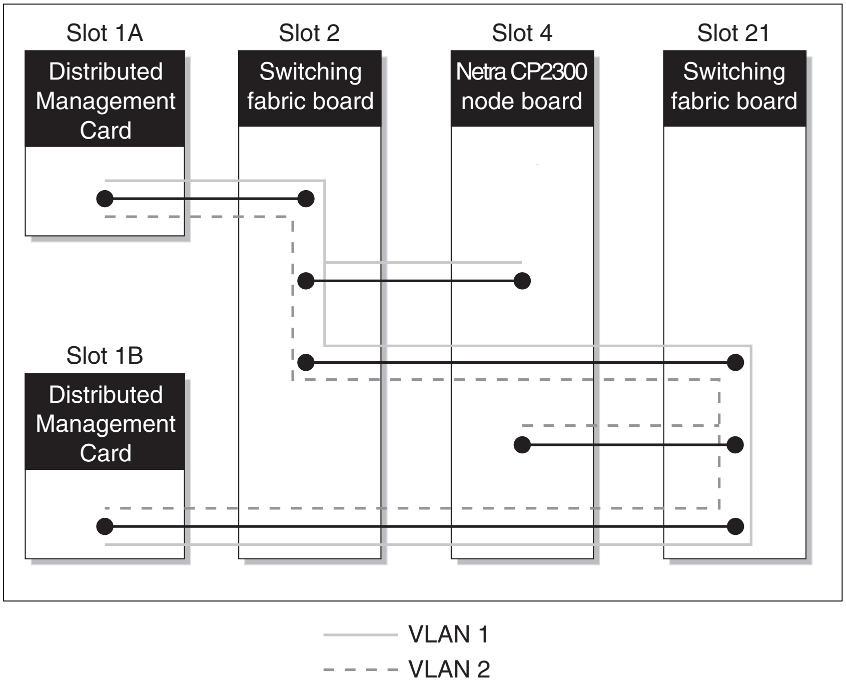

FIGURE 2-2 System Management Network VLAN Traffic over the cPSB Bus

On each node board, internal Ethernet ports dmfe33000 (VLAN tag 33) and dmfe44001 (VLAN tag 44) use CGTP to provide redundancy in case of failure of one of the ports or a failure of the switching fabric board connected to one of the ports. The interfaces are configured on each node board using a Solaris startup script. To verify that CGTP is installed on each node board, use the `pkginfo` command:

```
# pkginfo -l SUNWnhtp8 SUNWnhtu8
```

System management network traffic on the VLANs must always be contained within the chassis. Do not use VLAN tag 33 or 44.

IP Addressing for the System Management Network

The IP address of the system management network on the top (slot 1A) distributed management card is always the midplane FRU ID field `SysmgtbusIPSubnet` value *IP_subnet_address*.22; for the bottom (slot 1B) distributed management card it is always the midplane FRU ID field `SysmgtbusIPSubnet` value *IP_subnet_address*.23. The IP alias address for the system management network on the active distributed management card is the midplane FRU ID field `SysmgtbusIPSubnet` value *IP_subnet_address*.25. The IP alias address (the CGTP interface) provides packet redundancy.

The IP address of the system management network on the node boards is formed as follows. The midplane FRU ID field `SysmgtbusIPSubnet` contains the value *IP_subnet_address.slot_number*. The default IP subnet address is `c0a80d00` (192.168.13.00) and the default IP subnet mask is `0xffffffe0` (255.255. 255.224). When you power on the Netra CT server, and if you have not made any changes for the system management network in the midplane FRU ID, the IP address of a board installed in slot 3 is configured to 192.168.13.3; if you then move that board to slot 4, the IP address for that board is configured to 192.168.13.4.

TABLE 2-3 shows the system management network interfaces with the IP address defaults for the distributed management cards and a node board in slot 4.

TABLE 2-3 System Management Network Interface IP Address Defaults

Board	CGTP Interface Address	VLAN 1 Address	VLAN 2 Address
Distributed management card 1A	192.168.13.22	192.168.13.54	192.168.13.86
Distributed management card 1B	192.168.13.23	192.168.13.55	192.168.13.87
Active distributed management card	192.168.13.25 (alias)	192.168.13.57 (alias)	192.168.13.89 (alias)
Node board in slot 4	192.168.13.4	192.168.13.36	192.168.13.68

▼ To Configure the System Management Network

1. **Log in to the active distributed management card.**

2. **Set the FRU ID for the system management network:**

   ```
   hostname cli> setfru fru_name instance fru_property value
   ```

   Refer to TABLE 2-1 for allowable information for each variable. You must set both the system management network IP subnet address and the subnet mask in hexadecimal format. For example, to set the subnet address to 192.168.16.00 and the subnet mask to 255.255.255.224, enter the following:

   ```
   hostname cli> setfru midplane 1 SysmgtbusIPSubnet c0a81000
   hostname cli> setfru midplane 1 SysmgtbusSubnetMask ffffffe0
   ```

3. **Completely power off and on the system by locating the power switch at the rear of the Netra CT 820 server; press it to the Off (O) position, then press it to the On (|) position.**

Checking the System Management Network Configuration for the Solaris OS

After you boot the Solaris OS, you can check to see that the system management network has been configured by using the `ifconfig -a` command. You should see output for the `dmfe33000` interface (VLAN 1), the `dmfe44001` interface (VLAN 2), and the `cgtp1` interface similar to the following:

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4>
mtu 8232 index 1
  inet 127.0.0.1 netmask ff000000
dmfe0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 2
  inet 10.4.72.146 netmask ffffff00 broadcast 10.4.72.255
  ether 0:3:ba:2f:37:1a
dmfe33000: flags=
1008843<UP,BROADCAST,RUNNING,MULTICAST,PRIVATE,IPv4>
mtu 1500 index 3
  inet 192.168.13.35 netmask ffffffe0 broadcast 192.168.13.255
  ether 0:3:ba:2f:37:1a
dmfe44001: flags=
1008843<UP,BROADCAST,RUNNING,MULTICAST,PRIVATE,IPv4>
mtu 1500 index 4
  inet 192.168.13.67 netmask ffffffe0 broadcast 192.168.13.255
  ether 0:3:ba:2f:37:1b
cgtp1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 5
  inet 192.168.13.3 netmask ffffffe0 broadcast 192.168.13.255
  ether 0:0:0:0:0:0
```

To test for actual communication, use the `ping -s` command. You should see output similar to the following:

```
# ping -s 192.168.13.25
PING 192.168.13.25: 56 data bytes
64 bytes from 192.168.13.25:icmp_seq=0,time=1,ms
64 bytes from 192.168.13.25:icmp_seq=1,time=0,ms
64 bytes from 192.168.13.25:icmp_seq=2,time=0,ms
...
----192.168.13.25 PING statistics----
14 packets transmitted, 14 packets received, 0% packet loss
round-trip (ms) min/avg/max=0/0/1
```

The `cgtp1` interface should be plumbed and have a valid IP address assigned to it.

Note – This is a required interface. Never unplumb or unconfigure the system management network.

Checking the System Management Network Configuration on the Distributed Management Card

After you configure the system management network, you can check to see that it has been configured by using the CLI `shownetwork` command. You should see output similar to the following:

```
hostname cli> shownetwork
Netract network configuration is:

External Ethernet Interface : SRVC LAN
ip_addr : 10.4.72.170
ip_netmask : 0xffffff00
ip_alias : 10.4.72.195
ip_alias_netmask : 0xffffff00
mac_address : 00:03:ba:44:51:d0

System Management Interface :
ip_addr : 192.168.13.23
ip_alias : 192.168.13.25
ip_netmask : 0xffffffe0
 hostname cli>
```

Specifying Other FRU ID Information

You can use the FRU properties `Location`, `Cust_Data`, and `User_Label` to enter any customer-specific information about your system. These are optional entries; by default, there is no information stored in these fields. Information entered for the `Location` property is displayed through the MOH application.

You might want to use the `Location` FRU property to enter specific, physical location information for your system. For example, you might enter the number on the chassis label to indicate the location of the system.

▼ To Specify Other FRU ID Information

1. **Log in to the active distributed management card.**
2. **Specify other FRU ID information for the Netra CT server:**

```
hostname cli> setfru fru_name instance fru_property value
```

Refer to TABLE 2-1 for allowable information for each variable. For example, if you want to set the location information to reflect a chassis label that reads 12345-10-20, enter the following:

```
hostname cli> setfru midplane 1 Location 12345-10-20
```

3. **Completely power off and on the system by locating the power switch at the rear of the Netra CT 820 server; press it to the Off (O) position, then press it to the On (|) position.**

Configuring the Distributed Management Cards for Failover

The Netra CT 820 server provides distributed management card failover from the active distributed management card to the standby distributed management card for certain hardware and software events.

Failover includes moving all services provided by the active distributed management card to the standby distributed management card, which then becomes the newly active card.

This section describes the distributed management cards' failover and redundancy capabilities and provides procedures to:

- Configure whether failover occurs for certain events
- Configure whether recovery is tried after a failover
- Configure whether services, such as Telnet and MOH, are used with *static* IP addresses to the top or to the bottom distributed management cards, or with an *alias* IP address, which allows the services to stay connected to whichever distributed management card is active.

When you use CLI commands to enter information and set variables on the active distributed management card, this data is immediately *mirrored* on the standby distributed management card so that it is ready for a failover. Mirrored information includes user names, passwords, and permissions; configuration information, such as ntp server, alias IP address, SNMP interface, and MOH security information; and failover information.

If a failover occurs, services are started on the newly active card that are normally not running on the standby distributed management card, such as the PMS application. The failover is complete when the newly active distributed management card can provide services for CLI, MOH, and PMS.

Certain events always cause a failover. Other events cause a failover only if the `setfailover` mode is set to `on` (by default, failover mode is `off`). Failover mode can be turned on using the distributed management card CLI, a remote shell (`rsh` command), or MOH. Refer to the *Netra CT 820 Server Software Developer's Guide* for instructions on setting failover mode through MOH.

Failover Causes

Each distributed management card monitors itself (local monitoring) and the other distributed management card (remote monitoring). During this monitoring, a problem in any of the areas being monitored on the active distributed management card could cause a failover.

A failover always occurs with:

- A hot-swap of the active distributed management card
- A reset of the active distributed management card
- A user-initiated failover with the CLI command `setfailover force`

A failover occurs with any of the following events if the `setfailover` mode is set to `on`:

- A failure in the local IPMI controller (detected by a missed *heartbeat*).
- A failure in the 2.16 internal Ethernet interface (sends link down/link up status).
- Failure of the external Ethernet interface if the CLI command `setetherfailover` is set to `enable.`
- Failure of a critical daemon or service, such as MOH.
- A hardware failure of the switching fabric board connected to the active distributed management card.
- A hot-swap of the switching fabric board connected to the active distributed management card.

- A missed heartbeat during remote monitoring. Each distributed management card regularly sends `ping` packets over two internal serial interfaces to the other distributed management card. If the active distributed management card is hung, or is in a reset loop, for example, this could cause missed heartbeats.
- An *unhealthy* signal during remote monitoring. The #healthy line is a hardware internal link between the two distributed management cards. A hardware fault could trigger an unhealthy signal.

FIGURE 2-3 shows the internal hardware signals and interfaces that support distributed management card failover, and TABLE 2-4 describes the signals and interfaces.

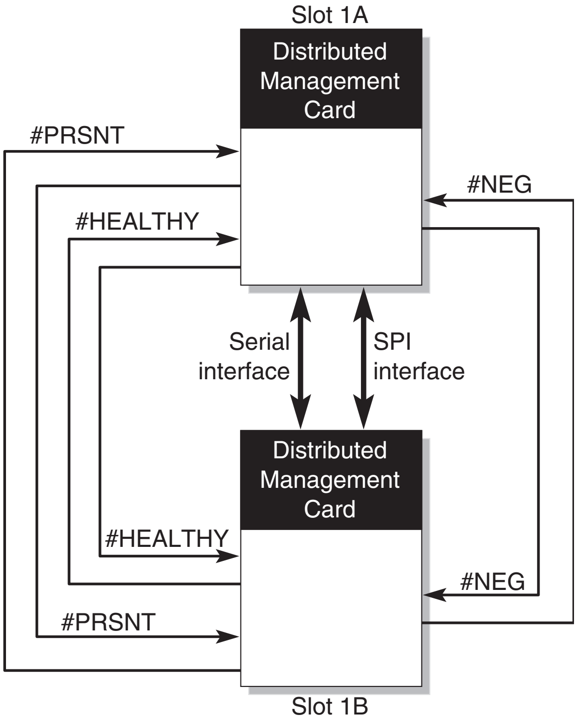

FIGURE 2-3 Hardware Signals and Interfaces Supporting Failover

TABLE 2-4 Hardware Signals and Interfaces Supporting Failover

Hardware	Description
Serial interface	The primary interface between the distributed management cards; it is used to send heartbeats and state synchronization information. Both distributed management cards must view the same field-replaceable unit (FRU), such as a particular fantray or a node board in a certain slot, in the same state, for example, powered on
SPI interface	The redundant interface for the serial interface; if the serial interface fails, the SPI interfaces takes over sending the heartbeat and state synchronization information
#PRSNT	This signal indicates the presence of a distributed management card
#NEG	This signal indicates which is the active distributed management card
#HEALTHY	This signal indicates the overall health of the distributed management card, including both the hardware and software

The external alarm port on the active distributed management card is not failed over to the standby distributed management card on a failover event. Refer to the *Netra CT 820 Server Installation Guide* for information on connecting alarm ports, and to the *Netra CT 820 Server Software Developer's Guide* for information on reloading the alarm severity profile on the newly active distributed management card.

TABLE 2-5 shows the relationship of services and IP addresses to failover. When specifying the alias IP address for an Ethernet port, use the alias IP address for the port you configured, that is, the external or the internal Ethernet port.

TABLE 2-5 Services and IP Connections on Failover

Service	IP Address Use	Failover Impact
CLI: Telnet	Static IP address for Ethernet ports on both top and bottom distributed management cards	Continue to communicate with the newly active distributed management card on failover.
	Alias IP address for Ethernet port on active distributed management card	Lose Telnet connection to active distributed management card on failover; must reconnect.
MOH: RMI application	Static IP address for Ethernet ports on both top and bottom distributed management cards	Keep RMI connection with the newly active distributed management card on failover. Notification is sent from the newly active distributed management card. See the *Netra CT 820 Server Software Developer's Guide* for information on how to manage this in your RMI application.

TABLE 2-5 Services and IP Connections on Failover *(Continued)*

Service	IP Address Use	Failover Impact
	Alias IP address for Ethernet port on active distributed management card	Lose RMI connection to active distributed management card on failover; must reconnect. No notification is sent.
MOH: SNMP application	Static IP address for Ethernet ports on both top and bottom distributed management cards	Continue to communicate with the newly active distributed management card. The management agent sends a trap indicating a change in the distributed management card standby status.
	Alias IP address for Ethernet port on active distributed management card	If the failover is caused by the `setfailover force` command, continue to communicate with the newly active distributed management card on failover; the management agent sends a trap indicating a change in the distributed management card standby status. If the failover is caused by a failover event, continue to communicate with the newly active distributed management card on failover; the management agent does not send a trap.
PMS application	Alias IP address for external Ethernet port on active distributed management card	The PMS client library reconnects to the newly active distributed management card PMS daemon on failover. This alias IP address is used for basic PMS connectivity and for partner lists (`slotrndaddressadd` command) for a remote system.
	Alias IP address for system management interface	The PMS client library reconnects to the newly active distributed management card PMS daemon on failover. This alias IP address is used for partner lists (`slotrndaddressadd` command) within the same system.
rsh	Static IP address for system management interface on top or bottom distributed management card	Lose connection to the failed distributed management card.
	Alias IP address for system management interface	Continue to communicate with the newly active distributed management card on failover.

Signs of a Failover

Signs of a failover from an active to a standby distributed management card include:

- The standby distributed management card CLI prompt changes from standby status to active status.
- The standby distributed management card Ready LED changes from blinking green to solid green.
- An MOH trap and/or alert is sent.
- A change in certain services. For example, a Telnet connection to the failed distributed management card could be lost.

If recovery is enabled (CLI `setdmcrecovery on` command), the active distributed management card tries a hard reset on the failed distributed management card. If the reset succeeds, the reset distributed management card comes online as the standby distributed management card. The reset is tried three times. An unsuccessful reset after the third try may indicate a serious hardware problem. By default, recovery mode is off.

▼ To Enable Failover Using the CLI

1. **Log in to the active distributed management card.**
2. **Set the failover mode to on:**

```
hostname cli> setfailover on
```

▼ To Enable Recovery

1. **Log in to the active distributed management card.**
2. **Set the recovery mode to on:**

```
hostname cli> setdmcrecovery on
```

Configuring Distributed Management Card Failover for External Ethernet Port Failure

You can configure the active distributed management card to fail over to the standby distributed management card if its external Ethernet port fails by using the CLI `setetherfailover` command.

To Configure the Active Distributed Management Card to Failover if its External Ethernet Port Fails

1. **Log in to the active distributed management card.**
2. **Verify that the failover mode is on:**

```
hostname cli> showfailover
DMC failover is turned: ON
```

3. **To enable failover if the external Ethernet interface fails, enter the following:**

```
hostname cli> setetherfailover -b 1 enable
```

Configuring Distributed Management Card Alias IP Addresses for the Ethernet Ports

You can configure an alias IP address for each Ethernet port on the active distributed management card. Using an alias IP address allows you to stay connected to whichever card is the active distributed management card in the event of a failover. The alias IP address must be in the same subnet as the IP address configured for that port. If you do not configure an alias IP address, you must connect to the static IP address.

Note – For the alias IP addresses to take effect, you must either reset the active distributed management card or force a failover.

▼ To Configure Alias IP Addresses for the Ethernet Ports

1. **Log in to the active distributed management card.**
2. **Verify that the failover mode is on:**

```
hostname cli> showfailover
DMC failover is turned: ON
```

3. **To configure an alias IP address for the internal Ethernet port, enter the following:**

```
hostname cli> setipalias -b 2 addr
hostname cli> setipaliasnetmask -b 2 addr
```

where port number 2 indicates the internal Ethernet port to the switching fabric board, and *addr* is the alias IP address and alias IP netmask for that port.

4. **To configure an alias IP address for the external Ethernet port, enter the following:**

```
hostname cli> setipalias -b 1 addr
hostname cli> setipaliasnetmask -b 1 addr
```

where port number 1 indicates the external Ethernet port to the network, and *addr* is the alias IP address and alias IP netmask for that port.

5. **Reset the active distributed management card or force a failover.**

Setting the Date and Time on the Distributed Management Cards

The distributed management card does not support battery backup time-of-day because battery life cannot be monitored to predict end of life, and drift in system clocks can be common. To provide a consistent system time, set the date and time on the distributed management card using one of these methods:

- Manually, using the CLI `setdate` command. You must set the date and time on both distributed management cards. The date and time must be reset after any power cycle.
- Configuring the distributed management card to be an NTP client, and optionally an NTP server, using the CLI `setntpserver` command. The Network Time Protocol (NTP) provides the correct timestamp for all systems on a network by synchronizing the clocks of all the systems. A Solaris server, called `xntp`, sets and maintains the timestamp. Refer to the online `man` pages for the `xntpd`, `ntpq`, and `ntpdate` commands for more information about NTP.

You can also set the time zone on the distributed management card. Daylight savings time is not supported.

To Set the Distributed Management Card Date and Time Manually

1. **Log in to the distributed management card.**
2. **Set the date and time manually:**

```
hostname cli> setdate [mmdd][HHMM]][ccyy][:ss]
```

where *mm* is the current month; *dd* is the current day of the month; *HH* is the current hour of the day; *MM* is the current minutes past the hour; *cc* is the current century minus one; *yy* is the current year; and :*ss* is the current second number.

Set the date and time on both distributed management cards.

To Set the Distributed Management Card Date and Time as an NTP Client (and optionally, as an NTP Server)

1. **Log in to the active distributed management card.**
2. **Set the distributed management card date and time as an NTP client:**

```
hostname cli> setntpserver addr
```

where *addr* is the IP address of the NTP server. This information is synchronized between the two distributed management cards.

3. **Reset the active distributed management card.**

   You can now configure the node boards to use the distributed management card as an NTP server, if desired. The recommended NTP client configuration is to use the NTP servers on both distributed management cards, with their respective system management network IP addresses (by default, 192.168.13.22 and 192.168.13.23).

▼ To Set the Time Zone on the Distributed Management Card

1. **Log in to the distributed management card.**
2. **Set the time zone with the** `settimezone` **command:**

```
hostname cli> settimezone time_zone +|- offset
```

   where *time_zone* is a valid three-character time zone, + or - indicates whether the time zone is west (+) or east (-) of Greenwich Mean Time (GMT), and *offset* is the number of hours (and optionally, minutes and seconds) the time zone is west or east of GMT.

   The offset has the form of *hh*[:*mm*[:*ss*]]. The minutes (*mm*) and seconds (*ss*) are optional. The hour (*hh*) is required and may be a single digit. The hour must be between 0 and 24, and the minutes and seconds (if present) between 0 and 59.

   For example, to set the local time zone to Pacific Standard Time, enter the following:

```
hostname cli> settimezone PST+9
```

   Daylight savings time is not supported.

3. **Reset the active distributed management card.**

Configuring the Node Boards

Verify that you can log in to the node boards. Complete any Solaris configuration needed for your environment, such as modifying OpenBoot PROM variables. Refer to the Solaris documentation, the OpenBoot PROM documentation, or to the specific node board documentation if you need additional information. Chapter 3 contains additional information on node boards.

Enabling the Managed Object Hierarchy Application

The Managed Object Hierarchy (MOH) is an application that runs on the distributed management cards and the node boards. It monitors the field-replaceable units (FRUs) in your system.

Software Required

The MOH application requires the Solaris 8 2/02 or compatible operating system, and additional Netra CT platform-specific Solaris patches that contain packages shown in TABLE 2-6.

TABLE 2-6 Solaris Packages for the MOH Application

Package	Description
`SUNW2jdrt`	Java™ Runtime Java Dynamic Management Kit (JDMK) package
`SUNWctmgx`	Netra CT management agent package
`SUNWctac`	Distributed management card firmware package that includes the Netra CT management agent

Download Solaris patch updates from the web site: `http://www.sunsolve.sun.com`. For current patch information, refer to the *Netra CT Server Release Notes*.

Install the patch updates using the `patchadd` command. After these packages are installed, they reside in the default installation directory, `/opt/SUNWnetract/mgmt3.0/`. To verify the packages are installed, use the `pkginfo` command:

```
# pkginfo -l SUNW2jdrt SUNWctmgx SUNWctac
...
PKGINST: SUNW2jdrt
...
```

Once the MOH application is running, MOH agents on the distributed management cards and on node boards interface with your Simple Network Management Protocol (SNMP) or Remote Method Invocation (RMI) application to *discover* network elements, monitor the system, and provide status messages.

Refer to the *Netra CT Server Software Developer's Guide* for information on writing applications to interface with the MOH application.

Starting the MOH Application

The MOH application is started automatically on the distributed management cards.

You must start the MOH application as root on the node boards using the `ctmgx start` command:

```
# cd /opt/SUNWnetract/mgmt3.0/bin
# ./ctmgx start [options]
```

If you installed the Solaris patches in a directory other than the default directory, specify that path instead.

TABLE 2-7 lists the options that can be specified with `ctmgx start` when you start the MOH application.

TABLE 2-7 `ctmgx` Options

Option	Description
`-rmiport` *portnum*	Specify the RMI port number. The default is 1099.
`-snmpport` *portnum*	Specify the SNMP port number. The default is 9161.
`-snmpacl` *filename*	Specify the SNMP ACL file to be used. The full path to *filename* must be specified.
`-showversion`	Print the system version number.

By default, SNMP and RMI applications have read-write access to MOH agents on the distributed management cards and on node boards. The following sections describe how to configure MOH to control SNMP and RMI access on the distributed management cards and on node boards.

MOH Configuration and SNMP

By default, SNMP applications have read-write access to the Netra CT 820 server MOH agents. If you want to control which applications communicate with the MOH agents, you must configure the distributed management card and node board SNMP interfaces. This configuration provides additional security by controlling who has access to the agent.

The SNMP interface uses an SNMP access control list (ACL) to control:

- The SNMP management applications that can access the information maintained by the MOH application, and the permissions. The control is based on the IP address and the *community* of the host on which the management application is running. Access can be either read-write or read-only.
- The IP addresses that can receive SNMP traps, or event notifications, from the MOH agent. There are several types of SNMP traps. MOH uses the ACL to determine where to send *coldStart* (initial) traps. A coldStart trap is sent to the system when MOH starts. For other types of traps or notifications, such as hardware status changes, MOH maintains a table which specifies where traps should be sent.

An SNMP *community* is a group of IP addresses of devices supporting SNMP. It helps define where information is sent. The community name identifies the group. An SNMP device or agent may belong to more than one SNMP community. An SNMP device or agent does not respond to requests originating from IP addresses that do not belong to one of its communities.

SNMP Applications and Failover

If a distributed management card failover occurs, an SNMP application responds as follows:

- If you use static IP addresses for both Ethernet ports on both the top and bottom distributed management cards, you continue to communicate with the newly active distributed management card. The management agent sends a trap indicating a change in the distributed management card standby status. You must manage both communication channels.
- If you use the alias IP address for the Ethernet port on the active distributed management card:
  - If the failover is caused by the `setfailover force` command, you continue to communicate with the newly active distributed management card; the management agent sends a trap indicating a change in the distributed management card standby status.
  - If the failover is caused by a failover event, you continue to communicate with the newly active distributed management card; the management agent does not send a trap.

Using the alias IP address for the Ethernet port is a simpler model to manage than using the static IP addresses for both Ethernet ports; it also ensures the failover is transparent.

Distributed Management Card SNMP Interface

On the active distributed management card, you enter ACL information using the CLI `snmpconfig` command. A limit of 20 communities can be specified. For each community, a limit of 5 IP addresses can be specified. The ACL information is stored in the distributed management card flash memory.

▼ To Configure the Distributed Management Card SNMP Interface

1. **Log in to the active distributed management card.**
2. **Enter SNMP ACL information with the `snmpconfig` command:**

```
hostname cli> snmpconfig add|del|show access|trap community [readonly|readwrite] [ip_addr]
```

where *community* is the name of a group that the MOH agent on the distributed management card supports, and *ip_addr* is the IP address of a device supporting an SNMP management application. For example, to add read-only access (the default) for the community `trees`, to add read-write access for the community `birds`, and to add a trap for the community `lakes`, enter the following:

```
hostname cli> snmpconfig add access trees ip_addr ip_addr ip_addr
hostname cli> snmpconfig add access birds readwrite ip_addr
hostname cli> snmpconfig add trap lakes ip_addr
```

3. **Reset the active distributed management card.**

You can use the `snmpconfig` command to show or delete existing ACL information. For example, to show the ACL access and trap information entered in Step 2 above, enter the following:

```
hostname cli> snmpconfig show access *
Community    Permissions    Hosts
trees        read-only      ip_addr ip_addr ip_addr
birds        read-write     ip_addr
hostname cli> snmpconfig show trap *
Community    Hosts
lakes        ip_addr
hostname cli>
```

Node Board SNMP Interface

On node boards, ACL information is stored in a configuration file in the Solaris OS.

The format of this file is specified in the JDMK documentation. An ACL file template that is part of the JDMK package is installed by default in `/opt/SUNWjdmk/jdmk4.2/1.2/etc/conf/template.acl`.

An example of a configuration file is:

```
acl = {
 {
 communities = trees
 access = read-only
 managers = oak, elm
 }
 {
 communities = birds
 access = read-write
 managers = robin
 }
}

trap = {
  {
  trap-community = lakes
  hosts = michigan, mead
  }
}
```

In this example, `oak`, `elm`, `robin`, `michigan`, and `mead` are hostnames. If this is the ACL file specified, when the MOH starts, a coldStart trap is sent to `michigan` and `mead`. Management applications running on `oak` and `elm` can read (get) information from MOH, but they cannot write (set) information. Management applications running on `robin` can read (get) and write (set) information from MOH.

The ACL file can be stored anywhere on your system. When you start the MOH application and you want to use an ACL file you created, you specify the complete path to the file.

Refer to the JDMK documentation (`http://www.sun.com/documentation`) for more information on ACL file format.

▼ To Configure a Node Board SNMP Interface

1. Log in to the server.

2. **Create a configuration file in the format of a JDMK ACL configuration file.**
3. **As root, start the MOH application.**

```
# cd /opt/SUNWnetract/mgmt3.0/bin
# ./ctmgx start [options]
```

If you installed the Solaris patches in a directory other than the default directory, specify that path instead.

The MOH application starts and reads the configuration file using one of these methods, in this order:

a. **If the command** `ctmgx start -snmpacl` *filename* **is used, MOH uses the specified file as the ACL file.**

b. **If the file** `/opt/SUNWjdmk/jdmk4.2/1.2/etc/conf/jdmk.acl` **exists, MOH uses that file as the ACL file when the command** `ctmgx start` **is used.**

If the ACL cannot be determined after these steps, SNMP applications have read-write access and MOH sends the coldStart trap to the local host only.

MOH Configuration and RMI

By default, RMI applications have read-write access to the Netra CT 820 server MOH agents. If you want to control which applications communicate with the MOH agents, you must configure the distributed management card interfaces for RMI. This configuration provides additional security by authenticating who has access to the agent.

To authenticate which RMI applications can access the MOH agents on the distributed management card, the following configuration is needed:

- The RMI application program(s) must contain a valid distributed management card user name and password to be authenticated. For information on adding this information to RMI programs, refer to the *Netra CT 820 Server Software Developer's Guide*. If the user name and password cannot be authenticated, a security exception occurs; no access is given.
- The types of requests a program can make depend on the user permissions associated with the particular authenticated user name making the request. If the user permissions are set to `c`, `u`, `a`, and `r`, the RMI permission is read-write; if the user permissions are set to anything less than `c`, `u`, `a`, and `r`, the RMI permission is read-only.
- The CLI `setmohsecurity` option must be set to `true`. The default is `false`, that is, all RMI applications can access the MOH agents on the distributed management card with read-write access.

If MOH security for RMI was enabled but becomes disabled on the distributed management card (for example, if the distributed management card is being reset or hot-swapped), security is disabled, a security exception occurs, and no access is given.

RMI Applications and Failover

If a distributed management card failover occurs, an RMI application responds as follows:

- If you use static IP addresses for the Ethernet ports on both top and bottom distributed management cards, you keep the connection with the active distributed management card on failover. Notification is sent from the newly active distributed management card. See the *Netra CT 820 Server Software Developer's Guide* for information on how to manage this in your RMI application.
- If you use the alias IP address for the Ethernet port on the active distributed management card, you lose the RMI connection to the active distributed management card on failover, and must reconnect. No notification is sent.

▼ To Configure the Distributed Management Card RMI Interface

1. **Verify that RMI programs you want to access the distributed management card MOH agent contain a valid distributed management card user name and password, with appropriate permissions.**
2. **Log in to the active distributed management card.**
3. **Set the `setmohsecurity` option to true:**

```
hostname cli> setmohsecurity true
```

4. **Reset the active distributed management card.**

   The RMI authentication takes effect immediately. Any modification to the distributed management card user names and passwords also takes effect immediately.

Enabling the Processor Management Service Application

The Processor Management Service (PMS) is a management application that provides support for high-availability services and applications. It provides both local and remote monitoring and control of a cluster of node boards. It monitors the health of node boards, takes recovery actions, and notifies partner nodes if so configured. It provides the state of the resources, such as hardware, operating system, and applications.

This section describes:

- Starting and stopping the PMS application on node boards.
- Stopping and restarting the PMS application on the distributed management card. The application starts automatically but can be restarted manually with various options.
- Setting the IP address by which the distributed management card monitors and controls a node board in a particular slot in the same system.
- Adding IP addresses by which a local node board monitors and controls node boards in local or remote systems.

You use the distributed management card PMS CLI commands to configure PMS services, such as fault detection/notification, and fault recovery. The recovery administration is described in "Using the PMS Application for Recovery and Control of Node Boards" on page 79.

You can also use the PMS API to configure partner lists. Partner lists are tables of distributed management card and node board information relating to connectivity and addressing. Refer to the `pms` API man pages, installed by default in `/opt/SUNWnetract/mgmt3.0/man`, for more information on partner lists.

Note that the PMS daemon runs only on the active distributed management card. Because of this, you can not use static IP addresses for the distributed management card with the PMS application. You must use alias IP addresses so that PMS daemons continue to run on the active distributed management card in case of a failover, as follows:

- Use the alias IP address for the external Ethernet port on the active distributed management card for basic PMS connectivity. If a failover occurs, the PMS client library reconnects to the newly active distributed management card PMS daemon. The PMS application receives notification of this reconnection.
- Use the alias IP address for the external Ethernet port on the active distributed management card for partner lists (`slotrndaddressadd` command) for a remote system. If a failover occurs, the PMS client library reconnects to the newly active

distributed management card PMS daemon. The PMS application receives notification of this reconnection. Refer to the *Netra CT 820 Server Software Developer's Guide* for information on adding IP addresses to your PMS client application.

- Use the alias IP address for the system management interface for partner lists (`slotrndaddressadd` command) within the same system. If a failover occurs, the PMS client library reconnects to the newly active distributed management card PMS daemon.The PMS application receives notification of this reconnection.

▼ To Start or Stop the PMS Application on a Node Board

1. **Log in as root to the server that has the Solaris patches installed (see "Software Required" on page 37).**

2. **Create a Solaris script to start, stop, and restart PMS, as follows:**

```
#!/sbin/sh
# Start/stop/restart processes required for PMS

case "$1" in
'start')
       /opt/SUNWnetract/mgmt3.0/bin/pmsd start -e force_avail
       ;;
'stop')
       /opt/SUNWnetract/mgmt3.0/bin/pmsd stop
       ;;
'restart')
       /opt/SUNWnetract/mgmt3.0/bin/pmsd stop
       /opt/SUNWnetract/mgmt3.0/bin/pmsd start -e force_avail
       ;;
*)
       echo "Usage: $0 {start | stop | restart }"
       exit 1
       ;;
esac
exit 0
```

3. **Save the script to a file.**

4. **Start, stop, or restart the PMS application by typing one of the following:**

   - *filename* **`start`**
   - *filename* **`stop`**

- *filename* **restart**

where *filename* is the name of the file in which you saved the script.

You can also save this script in the /etc/rc* directory of your choice to have PMS automatically start at boot time.

Stopping and Restarting the PMS Daemon on the Distributed Management Card

The PMS daemon (pmsd) starts automatically on the active distributed management card. However, you can manually stop and restart the PMS daemon on the active distributed management card.

Note – Stopping the PMS daemon on the active distributed management card forces a failover if the setfailover mode is set to on. If you do not want a failover to occur, set the setfailover mode to off, stop PMS, then re-enable failover. When you stop PMS, the healthy state of the active distributed management card becomes not healthy and you must reset the distributed management card to recover.

These optional parameters can be specified:

- The port number pmsd listens on for servicing clients (default is port 10300).
- The *state* pmsd is started in, either available or unavailable. The default is to start in the unavailable state, unless a previous and different operating state exists in persistent storage.
- Whether to reset persistent storage to the default values on the distributed management card. The default is to use existing persistent storage.

You specify the port number for pmsd using the parameter *port_num*.

You specify the state in which to start pmsd using the parameter *server_admin_state*. This parameter may be set to force_unavail (force pmsd to start in the unavailable state); force_avail (force pmsd to start in the available state); or vote_avail (start pmsd in the available state, but only if all conditions have been met to make it available; if all the conditions have not been met, pmsd will not become available).

You specify whether to reset persistent storage to the default values on the distributed management card using the -d option. Data in persistent storage remains across reboots or power on and off cycles. If you do not specify -d, pmsd is started using its existing persistent storage configuration; if you specify -d, the persistent storage configuration is reset to the defaults for pmsd. The -d option would typically be specified only to perform a bulk reset of persistent storage during initial system bring up or if corruption occurred.

To Manually Stop the Processor Management Service on the Distributed Management Card

1. **Log in to the active distributed management card.**
2. **Stop the PMS daemon with the** `stop` **command:**

```
hostname cli> pmsd stop [-p port_num]
```

where *port_num* is the port number of the currently running `pmsd` you want to stop. The default is port 10300. Note that stopping PMS on the active distributed management card forces a failover if the `setfailover` mode is set to `on`.

To Manually Start the Processor Management Service on the Distributed Management Card

1. **Log in to the active distributed management card.**
2. **Start the PMS daemon with the** `start` **command:**

```
hostname cli> pmsd start [-p port_num] [-e server_admin_state] [-d]
```

where *port_num* is the port number for `pmsd` to listen on, *server_admin_state* can be `force_unavail`, `force_avail`, or `vote_avail`, and `-d` resets the persistent storage to the defaults for `pmsd`.

Setting the IP Address for the Distributed Management Card to Control Node Boards in the Same System

The `pmsd slotaddressset` command is used to set the IP address by which the distributed management card controls and monitors a node board in a particular slot. The command establishes the connection between `pmsd` running on the distributed management card and `pmsd` running on a node board. The distributed management card and the node board must be in the same system.

You specify the slot number of the node board and the IP address to be configured. The default IP address for all slots is 0.0.0.0. Therefore, control is initially disabled.

▼ To Set the IP Address for the Distributed Management Card to Control Node Boards in the Same System

1. **Log in to the active distributed management card.**
2. **Set the IP address with the** `slotaddressset` **command:**

```
hostname cli> pmsd slotaddressset -s slot_num -i ip_addr
```

where *slot_num* can be a slot number from 3 to 20, and *ip_addr* is the IP address to be configured.

Printing IP Address Information

The `pmsd slotaddressshow -s` *slot_num* | `all` command can be used to print IP address information for the specified slot or all slots. If the IP address information is not 0.0.0.0 for a given slot, PMS is configured to manage the node board in this slot using this IP address.

Adding Address Information for a Local Node Board to Control Node Boards in Local or Remote Systems

You can use the PMS CLI application to enable local node boards to remotely monitor and control node boards in the same system or in other Netra CT systems. One use for this capability is in a high availability environment. For example, if a high availability application fails on a controlled node board, PMS notifies the controlling node board of the failure, and the controlling node board (through a customer application) notifies another controlled node board to start the same high availability application.

The pmsd slotrndaddressadd command is used to configure a local node board to control and monitor another node board by specifying the IP addresses and slot information for the node board to be controlled, using the parameters shown in TABLE 2-8.

TABLE 2-8 pmsd slotrndaddressadd Parameters

Parameter	Description
-s *slot_num* | all	Specifies the slot number of the node board that is being configured in the local system to monitor or control other local or remote node boards
-n *ip_addr*	Specifies the IP address of the node board in the local or remote system to be monitored or controlled by the local node board
-d *ip_addr*	Specifies the IP address of the distributed management card in the same local or remote system of the node board to be monitored or controlled by the local node board. If the distributed management card is in the local system, use the alias IP address for the system management interface (default is 192.168.13.25); if the distributed management card is in a remote system, use the alias IP address for the external Ethernet port on the active distributed management card.
-r *slot_num*	Specifies the slot number of the node board in the local or remote system to be monitored or controlled by the local node board

Each local node board can control and monitor 16 local or remote node boards. Each local node board being managed must have already had its IP address set using the pmsd slotaddressset command.

▼ To Add Address Information for a Local Node Board to Control Node Boards in Local or Remote Systems

1. **Log in to the active distributed management card.**
2. **Add the address information with the slotrndaddressadd command:**

```
hostname cli> pmsd slotrndaddressadd -s slot_num|all -n ip_addr -d ip_addr -r slot_num
```

where -s *slot_num* is the slot number in the same system of the local node board you want to use to control other local or remote node boards, and all specifies all slots containing node boards in the local system; -n *ip_addr* is the IP address of the node board to be controlled; -d *ip_addr* is either the alias IP address of the system management interface if the active distributed management card is in the system of the node board to be controlled *or* the alias IP address of the external Ethernet port of the active distributed management card if that card is in a remote system; and `-r` *slot_num* is the slot number of the node board to be controlled.

When you add address information with the `slotrndaddressadd` command, an index number is automatically assigned to the information. You can see index numbers by using the `slotrndaddressshow` command and use the index numbers to delete address information with the `slotrndaddressdelete` command.

Deleting Address Information

The `pmsd slotrndaddressdelete -s` *slot_num*|`all -i` *index_num*|`all` command can be used to delete address information from the controlling node board. The `-s` *slot_num*|`all` parameter specifies whether the address information is deleted on a single slot number or on all slots containing node boards in the local system. The `-i` *index_num*|`all` parameter specifies whether the address information will be deleted for a single address entry or for all address entries; *index_num* can be 1 to 16. Before using this command, it is advisable to print the current address information using the `pmsd slotrndaddressshow` command, so you know the index number to use.

Printing Address Information

The `pmsd slotrndaddressshow -s` *slot_num*|`all -i` *index_num*|`all` command can be used to print address information. The `-s` *slot_num*|`all` parameter specifies whether the address information is printed for a single slot number or for all slots containing node boards in the local system; *index_num* can be 1 to 16. The `-i` *index_num*|`all` parameter specifies whether the address information is printed for a single address entry or for all address entries.

CHAPTER 3

Administering Your System

You administer your system using the distributed management card command-line interface (CLI), and through the MOH and PMS applications.

The distributed management card CLI works with the MOH and PMS applications, and supports Simple Network Management Protocol (SNMP) and Remote Method Invocation (RMI) interfaces. MOH provides the SNMP and RMI interfaces to manage the system and send out events and alerts. CLI provides an overlapping subset of commands with MOH and also provides commands for the distributed management card itself. Sending out events and alerts is not a function of the CLI.

This chapter contains the following sections:

Using the Distributed Management Card CLI

The distributed management card CLI provides commands to control power of the system, control the node boards, administer the system, show status, and set configuration information. See "Accessing the Distributed Management Cards" on page 8 for information on how to access the distributed management card.

All CLI commands can be used on the active distributed management card. A subset of CLI commands can be used on the standby distributed management card.

CLI Commands

TABLE 3-1 lists the active distributed management card command-line interface commands by type, command name, default permission required to use the command, and command description. TABLE 3-2 lists the subset of the CLI commands available for the standby distributed management card.

Default permission levels are:

- c (console permission; authorized to connect to other server console)
- u (user administration permission; authorized to use commands that can add, delete, and change permission of users)
- a (administration permission; authorized to change the state of the CLI configuration variables)
- r (reset/poweron/poweroff permissions; authorized to reset, poweron, and poweroff any of the node boards)
- blank (permission not required).

The permission level for a user can be changed with the `userperm` command.

A `-h` option with a command indicates that help is available for that command.

TABLE 3-1 Active Distributed Management Card Command-Line Interface Commands

Command Type	Command	Permission	Description
Configuration	`setipmode -b` *port_num* `rarp|config|none`	a	Set the IP mode of the specified Ethernet port. Choose the IP mode according to the services available in the network (`rarp`, `config`, or `none`). The default for the external Ethernet port (`1`) is `none`; the default for the internal Ethernet port (`2`) is `none`, that is, no services are available on these ports. You must reset the distributed management card for the changes to take effect.
	`setipaddr -b` *port_num addr*	a	Set the IP address of the specified Ethernet port. The default is 0.0.0.0. This command is only used if the `ipmode` is set to `config`. You must reset the distributed management card for the changes to take effect.
	`setipnetmask -b` *port_num mask*	a	Set the IP netmask of the specified Ethernet port. The default is 255.255.255.0. This command is only used if the `ipmode` is set to `config`. You must reset the distributed management card for the changes to take effect.
	`setipgateway` *addr*	a	Set the IP gateway of the distributed management card. The default is 0.0.0.0. You must reset the distributed management card for the changes to take effect.
	`sethostname` *hostname*	a	Set the hostname to be used in the CLI prompt. The default is `netract`. The maximum length is 32 characters.
	`setservicemode true|false`	a	Set whether MOH and PMS services are started automatically on the distributed management card after a reboot. The default is `false`, meaning that these services are automatically started.
	`setmohsecurity [-h] true|false`	a	Set whether authentication is required on the distributed management card RMI interface. The default is `false`. Refer to "MOH Configuration and RMI" on page 42 for more information.
	`showmohsecurity [-h]`		Display the value for `setmohsecurity`.
	`setdmcrecovery [-h] on|off`	a	Set whether the active distributed management card should try to reset the other, failed distributed management card. The default is `off` (a reset is not tried).

TABLE 3-1 Active Distributed Management Card Command-Line Interface Commands *(Continued)*

Command Type	Command	Permis-sion	Description
	`showdmcrecovery` [`-h`]		Display the value for `setdmcrecovery`.
	`setdefaults`	a	Initialize the distributed management card system configuration variables, for example, the Ethernet variables and the hostname, to the defaults. To return all configuration information to the defaults, use the `setdefaults` command on both distributed management cards, and then press the Reset button on both distributed management cards at the same time or power cycle the system.
	`showipmode` `-b` *port_num*		Display the value of `ip_mode` for the specified port number.
	`showipaddr` `-b` *port_num*		Display the value of `ip_addr` for the specified port number.
	`showipnetmask` `-b` *port_num*		Display the value of `ip_netmask` for the specified port number.
	`showipgateway`		Display the value of `ip_gateway` for the distributed management card.
	`showhostname`		Display the value of the hostname used in the CLI prompt.
	`showservicemode`		Display the value of the distributed management card service mode.
	`setntpserver` *addr* | `none`	a	Configure the distributed management card to be an NTP client, and optionally an NTP server. The default is `none`.
	`showntpserver`		Display the IP address of the NTP server.
	`setfailover` `on|off|force`	a	Enable (`on`), disable (`off`), or force (`force`) a failover from the active distributed management card to the standby distributed management card. The default is `off`. Refer to "Configuring the Distributed Management Cards for Failover" on page 27 for more information.
	`showfailover`		Display the distributed management card failover mode.
	`setetherfailover -b 1` `enable|disable`	a	Set the external Ethernet interface of the distributed management card to enable or disable failover if the external Ethernet interface fails. The default is `disable`.
	`showetherfailover -b 1`		Display the distributed management card external Ethernet interface failover mode.

TABLE 3-1 Active Distributed Management Card Command-Line Interface Commands *(Continued)*

Command Type	Command	Permis-sion	Description
	`setipalias -b` *port_num addr*	a	Set the alias IP address for the specified Ethernet port. For the external Ethernet port, set the *port_num* to `1`. For the internal Ethernet port, set the *port_num* to `2`. The default is 0.0.0.0. You must reset the active distributed management card for the changes to take effect.
	`showipalias -b` *port_num*		Display the alias IP address for the specified Ethernet port.
	`setipaliasnetmask -b` *port_num mask*	a	Set the IP alias netmask for the specified Ethernet port. The default is 255.255.255.0.
	`showipaliasnetmask -b` *port_num*		Display the IP alias netmask for the specified Ethernet port.
	`settimezone` *time_zone*	a	Set the time zone value for the date. Refer to "Setting the Date and Time on the Distributed Management Cards" on page 34 for more information.
	`showtimezone`		Display the current time zone value.
Power control	`poweroff` *cpu_node*	r	Power off the specified node slot, where *cpu_node* can be 2 through 21. This command is also supported on third-party node boards.
	`poweron` *cpu_node*	r	Power on the specified node slot, where *cpu_node* can be 2 through 21. This command is also supported on third-party node boards.
	`powersupply` *n* `on|off`	r	Switch the specified power supply unit on or off, where *n* can be 1 through 8.
CPU control	`reset` [`-h`] [dmc|1A|1B|*cpu_node*`]` `[-x` *cpu_node*`]`	r	Reset (reboot) a specified node. `reset` [dmc|1A|1B|*cpu_node*`]` produces a soft reset (reboots the operating system), where `dmc` is the distributed management card the command is issued on; `1A` is the top distributed management card; `1B` is the bottom distributed management card; and *cpu_node* can be 3 through 20. `reset -x` produces a hard reset (resets the board), where *cpu_node* can be 2 through 21. `reset -x` is also supported on third-party node boards.

TABLE 3-1 Active Distributed Management Card Command-Line Interface Commands *(Continued)*

Command Type	Command	Permis- sion	Description
	`pmsd`	a	Display help information on starting, stopping, and controlling the PMS daemon on the distributed management card. See also the PMS daemon control commands at the end of TABLE 3-1; "Enabling the Processor Management Service Application" on page 44; and "Using the PMS Application for Recovery and Control of Node Boards" on page 79 for more information.
	`console` *cpu_node*	c	Enter console mode and connect to the specified node board, where *cpu_node* can be 3 through 20.
	`break` *cpu_node*	c	Put the server in debug mode, where *cpu_node* can be 3 through 20.
	`showhealth` [-b *cpu_node*]		Show the healthy information of a node, where *cpu_node* can be 0 through 21.
	`showcpustate`		Display the board type, power state, and boot state for each slot in the system. Refer to "Displaying Board State Information" on page 68 for more information. This command is also supported on third-party node boards.
System status	`showenvironment`		Display a summary of current environmental information, such as fantray and power supply status.
	`shownetwork`		Display the current network configuration of the distributed management card.
	`showdate`		Display the system date.
Administration	`setpanicdump` `[all`|*cpu_node*`]` `[true`|`false]`	a	Set whether a panic dump is generated when a node is reset, where `all` means all nodes 3 through 20, and *cpu_node* can be a specific node 3 through 20.
	`setescapechar` *value*		Set the escape character to end a console session. The default is a tilde (~).
	`useradd` [-h] *username*	u	Add a user account. The default user account is `netract`. The distributed management card supports 16 accounts.
	`userdel` [-h] *username*	u	Delete a user account.
	`usershow` [-h] [*username*]		Show user accounts.
	`userpassword` [-h] *username*	u	Set or change the password of a specified user account.
	`userperm` [-h] *username* `[c`|`u`|`a`|`r]`	u	Set or change the permission levels for a specified user account.

TABLE 3-1 Active Distributed Management Card Command-Line Interface Commands *(Continued)*

Command Type	Command	Permis-sion	Description
	`showusers`		Show the number of users logged in to the distributed management card.
	`logout`		Log out of the current session.
	`password [-h]`	u	Change the existing password.
	`flashupdate -d cmsw|bcfw|bmcfw|rpdf|scdf -f` *path*	a	Flash update the distributed management card software, where `cmsw` represents the chassis management software;. `bcfw` represents the boot control firmware; `bmcfw` represents the BMC firmware; `rpdf` represents the system configuration repository; and `scdf` initializes the system configuration variables to their defaults. Refer to "Updating the Distributed Management Cards' Flash Images" on page 63 for more information.
	`help`		Display a list of supported commands.
	`setdate [-h]` [*mmdd*][*HHMM*][*ccyy*][:*ss*]	a	Set the current date.
	`setfru [-h]` *fru_name instance fru_property value*	a	Set FRU ID information. Refer to "Specifying Netra CT Server FRU ID Information" on page 11 for more information. This command is also supported on third-party node boards. Refer to "To Configure a Chassis Slot for a Third-Party Node Board" on page 87 for more information.
	`showfru` *fru_name instance fru_property*		Display FRU ID information. Refer to "Displaying Netra CT Server FRU ID Information" on page 15 for more information. This command is also supported on third-party node boards. Refer to "To Display FRU ID Information for a Third-Party Node Board" on page 88 for more information.
	`showescapechar`	a	Show the escape character used to end a console session.
	`showpanicdump [all|`*cpu_node*`]`		Show whether or not a panic dump has been set for all nodes 3 through 20 or for a specific node 3 through 20.
	`version`		Display the versions of various software and firmware.

TABLE 3-1 Active Distributed Management Card Command-Line Interface Commands *(Continued)*

Command Type	Command	Permission	Description
	`snmpconfig [-h] add|del|show access|trap` *community* `[readonly|readwrite]` [*addr*]	a	Configure the distributed management card SNMP interface. Refer to "MOH Configuration and SNMP" on page 38 for more information.
PMS daemon control	`pmsd start [-p` *port_num*`] [-e` *server_admin_state*`] [-d]`	a	Start PMS on the distributed management card.
	`pmsd stop [-p` *port_num*`]`	a	Stop PMS on the distributed management card.
	`pmsd slotaddressset -s` *slot_num* `-i` *ip_addr*	a	Set the IP address for the distributed management card to control and monitor a node board.
	`pmsd slotaddressshow -s` *slot_num*`|all`	a	Print the IP address set with the `pmsd slotaddressset` command.
	`pmsd slotrndaddressadd -s` *slot_num*`|all -n` *ip_addr* `-d` *ip_addr* `-r` *slot_num*	a	Add address information for a node board to control other node boards.
	`pmsd slotrndaddressdelete -s` *slot_num*`|all -i` *index_num*`|all`	a	Delete address information added with the `pmsd slotrndaddressadd` command.
	`pmsd slotrndaddressshow -s` *slot_num*`|all -i` *index_num*`|all`	a	Print address information added with the `pmsd slotrndaddressadd` command.
	`pmsd operset -s` *slot_num*`|all -o maint_config|oper_config|none_config|graceful_reboot`	a	Enable automatic recovery of a node board.
	`pmsd infoshow -s` *slot_num*`|all`	a	Print PMS system information.
	`pmsd historyshow -s` *slot_num*`|all`	a	Print a log of PMS system events and time stamps.
	`pmsd recoveryoperset -s` *slot_num*`|all -o pc|rst|rstpc|pd|rb`	a	Manually recover a board in case of fault.

TABLE 3-1 Active Distributed Management Card Command-Line Interface Commands *(Continued)*

Command Type	Command	Permis-sion	Description
	pmsd recoveryautooperset -s *slot_num*|all -o pc|rst|rstpc|pd|rb|rbpc|none|trg [-d *startup_delay*] [-f on|off] [-r *retries*] [-n *inter_op_delay*] [-p *reset_power-cycle_delay*]	a	Automatically recover a board in case of fault.
	pmsd recoveryautoinfoshow -s *slot_num*|all	a	Print the configuration information affected by the recoveryautooperset command.
	pmsd hwoperset -s *slot_num*|all -o powerdown|powerup|reset|mon_enable|mon_disable [-f]	a	Perform operations on a node board hardware.
	pmsd hwinfoshow -s *slot_num*|all	a	Print PMS system information on the hardware.
	pmsd hwhistoryshow -s *slot_num*|all	a	Print a log of PMS hardware events and time stamps.
	pmsd osoperset -s *slot_num*|all -o reboot|mon_enable|mon_disable [-f]	a	Perform operations on a node board operating system.
	pmsd osinfoshow -s *slot_num*|all	a	Print PMS system information on the operating system.
	pmsd oshistoryshow -s *slot_num*|all	a	Print a log of PMS operating system events and time stamps.
	pmsd appoperset -s *slot_num*|all -o force_offline|vote_active|force_active	a	Perform operations on node board applications.
	pmsd appinfoshow -s *slot_num*|all	a	Print PMS system information on the applications.
	pmsd apphistoryshow -s *slot_num*|all	a	Print a log of PMS application events and time stamps.
	pmsd version	a	Print the PMS version.
	pmsd usage	a	Print a synopsis of the pmsd commands.

Information on configuring distributed management card ports, setting up user accounts, specifying FRU ID information, and starting the PMS daemon using the distributed management card CLI is provided in Chapter 2. The PMS daemon commands are described in "Using the PMS Application for Recovery and Control of Node Boards" on page 79.

TABLE 3-2 lists the commands valid on the standby distributed management card.

TABLE 3-2 Standby Distributed Management Card Command-Line Interface Commands

Command Type	Command	Permis-sion	Description
Configuration	`setipmode -b` *port_num* `rarp|config|none`	a	Set the IP mode of the specified Ethernet port. Choose the IP mode according to the services available in the network (`rarp`, `config`, or `none`). The default for the external Ethernet port (`1`) is `none`; the default for the internal Ethernet port (`2`) is `none`, that is, no services are available on these ports. You must reset the distributed management card for the changes to take effect.
	`setipaddr -b` *port_num addr*	a	Set the IP address of the specified Ethernet port. The default is 0.0.0.0. This command is only used if the `ipmode` is set to `config`. You must reset the distributed management card for the changes to take effect.
	`setipnetmask -b` *port_num mask*	a	Set the IP netmask of the specified Ethernet port. The default is 255.255.255.0. This command is only used if the `ipmode` is set to `config`. You must reset the distributed management card for the changes to take effect.
	`setipgateway` *addr*	a	Set the IP gateway of the distributed management card. The default is 0.0.0.0. You must reset the distributed management card for the changes to take effect.
	`sethostname` *hostname*	a	Set the hostname to be used in the CLI prompt. The default is `netract`. The maximum length is 32 characters.
	`setservicemode true|false`	a	Set whether MOH and PMS services are started automatically on the distributed management card after a reboot. The default is `false`, meaning that these services are automatically started.
	`showmohsecurity [-h]`		Display the value for `setmohsecurity.`
	`showdmcrecovery [-h]`		Display the value for `setdmcrecovery`.

TABLE 3-2 Standby Distributed Management Card Command-Line Interface Commands *(Continued)*

Command Type	Command	Permis-sion	Description
	`setdefaults`	a	Initialize the distributed management card system configuration variables, for example, the Ethernet variables and the hostname, to the defaults. To return all configuration information to the defaults, use the `setdefaults` command on both distributed management cards, and then press the Reset button on both distributed management cards at the same time or power cycle the system.
	`showipmode -b` *port_num*		Display the value of `ip_mode` for the specified port number.
	`showipaddr -b` *port_num*		Display the value of `ip_addr` for the specified port number.
	`showipnetmask -b` *port_num*		Display the value of `ip_netmask` for the specified port number.
	`showipgateway`		Display the value of `ip_gateway` for the distributed management card.
	`showhostname`		Display the value of the hostname used in the CLI prompt.
	`showservicemode`		Display the value of the distributed management card service mode.
	`showntpserver`		Display the IP address of the NTP server.
	`showfailover`		Display the distributed management card failover mode.
	`showetherfailover -b 1`		Display the distributed management card external Ethernet interface failover mode.
	`showipalias -b` *port_num*		Display the alias IP address for the specified Ethernet port.
	`showipaliasnetmask -b` *port_num*		Display the alias IP netmask for the specified Ethernet port.
	`showtimezone`		Display the current time zone value.
CPU control	`reset` [-h] [dmc|1A|1B]	r	Reset (reboot) a distributed management card. `reset` [dmc|1A|1B`]`produces a soft reset (reboots the operating system), where `dmc` is the distributed management card the command is issued on; `1A` is the top distributed management card; and `1B` is the bottom distributed management card.
	`showhealth` [-b *cpu_node*]		Show the healthy information of a node, where *cpu_node* can be 0 through 21.

TABLE 3-2 Standby Distributed Management Card Command-Line Interface Commands *(Continued)*

Command Type	Command	Permis-sion	Description
	`showcpustate`		Display the board type, power state, and boot state for each slot in the system. Refer to "Displaying Board State Information" on page 68 for more information. This command is also supported on third-party node boards.
System status	`showenvironment`		Display a summary of current environmental information, such as fantray and power supply status.
	`shownetwork`		Display the current network configuration of the distributed management card.
	`showdate`		Display the system date.
Administration	`usershow` [`-h`] [*username*]		Show user accounts.
	`showusers`		Show the number of users logged in to the distributed management card.
	`logout`		Log out of the current session.
	`password` [`-h`]	u	Change the existing password.
	`flashupdate -d cmsw`|`bcfw`|`bmcfw`|`rpdf`|`scdf -f` *path*	a	Flash update the distributed management card software, where `cmsw` represents the chassis management software;. `bcfw` represents the boot control firmware; `bmcfw` represents the BMC firmware; `rpdf` represents the system configuration repository; and `scdf` initializes the system configuration variables to their defaults. Refer to "Updating the Distributed Management Cards' Flash Images" on page 63 for more information.
	`help`		Display a list of supported commands.
	`setdate` [`-h`] [*mmdd*][*HHMM*][*ccyy*][:*ss*]	a	Set the current date.
	`showfru` *fru_name instance fru_property*		Display FRU ID information. Refer to "Displaying Netra CT Server FRU ID Information" on page 15 for more information. This command is also supported on third-party node boards. Refer to "To Display FRU ID Information for a Third-Party Node Board" on page 88 for more information.
	`showescapechar`	a	Show the escape character used to end a console session.

TABLE 3-2 Standby Distributed Management Card Command-Line Interface Commands *(Continued)*

Command Type	Command	Permis-sion	Description
	`showpanicdump [all|`*cpu_node*`]`		Show whether or not a panic dump has been set for all nodes 3 through 20 or for a specific node 3 through 20.
	`version`		Display the versions of various software and firmware.
	`snmpconfig [-h] show access|trap`	a	Display SNMP access or trap information for the distributed management card SNMP interface. Refer to "MOH Configuration and SNMP" on page 38 for more information.

Security

A remote command-line session or a console session automatically disconnects after 10 minutes of inactivity.

Security is also provided through the permission levels and passwords set for each account.

Updating the Distributed Management Cards' Flash Images

The primary boot device for the distributed management card is always the flash. You can update the distributed management card flash images over the network using `nfs` or `tftp`. TABLE 3-3 shows the distributed management card flash options.

TABLE 3-3 Distributed Management Card Flash Options

Option	Description
`cmsw`	Updates the chassis management software, which includes the Chorus software, the MOH application, and the PMS application.
`bcfw`	Updates the boot control firmware.

TABLE 3-3 Distributed Management Card Flash Options *(Continued)*

Option	Description
`bmcfw`	Updates the BMC firmware.
`rpdf`	Updates the system configuration repository, which contains information used internally by the CLI in the flash, reinitializes it to a default minimum, and resets the distributed management card.
`scdf`	(Optional) Initializes the system configuration variables, for example, the Ethernet variables and the hostname, to the defaults.

You must flash update the standby distributed management card first, then fail over from the active to the standby distributed management card, then flash update the new standby distributed management card. There is no required sequence for flashing each distributed management card, although the following order is recommended: `cmsw`, `bcfw`, `bmcfw`, and `rpdf`. You can update individual images if desired.

During a flash update of the BMC firmware, the BMC is not able to respond to communication requests, and the following messages may display on the console:

```
ysif_xfer_msg: kcs driver xfermsg returns -1
read_evt_buffer: sysif_xfer_msg returns -1
poll_evt_handler: read_evt_buffer returns -1
listner_thread: poll_evt_handler returns -1
```

These messages can be safely ignored during a flash update.

▼ To Update All the Distributed Management Cards’ Flash Images

1. **Log in to the standby distributed management card.**

2. **Flash update the standby distributed management card images:**

Note – The scdf option is not mandatory. Use it only if you want to initialize the system configuration variables to the defaults.

```
hostname cli> flashupdate -d cmsw -f path
hostname cli> flashupdate -d bcfw -f path
hostname cli> flashupdate -d bmcfw -f path
hostname cli> flashupdate -d scdf
hostname cli> flashupdate -d rpdf -f path
```

where *path* can be nfs://*nfs.server.ip.address/directory/filename* or tftp://*tftp.server.ip.address/directory/filename* where the software to use in the flash is installed. If you are using the NFS option, make sure that the path is a shared NFS mount.

After you update rpdf, the distributed management card resets itself. If you do not update rpdf, you must reset the distributed management card manually, with the reset dmc command.

3. **Log in to the active distributed management card.**

4. **Force a failover from the active distributed management card to the standby distributed management card:**

```
hostname cli> setfailover force
```

The active distributed management card becomes the new standby distributed management card.

5. **Repeat the instructions in Step 2 to flash update the new standby distributed management card.**

▼ To Update an Individual Distributed Management Card Flash Image

1. **Log in to the standby distributed management card.**

2. **Flash update a distributed management card image:**

```
hostname cli> flashupdate -d option
hostname cli> reset dmc
```

where *option* can be `cmsw -f` *path*, `bcfw -f` *path*, `bmcfw -f` *path, or* `scdf`, and *path* can be `nfs://`*nfs.server.ip.address/directory/filename* or `tftp://`*tftp.server.ip.address/directory/filename* where the software to use in the flash is installed. If you are using the NFS option, make sure that the path is a shared NFS mount.

If you update `rpdf`, the distributed management card resets itself after finishing the `rpdf` update.

3. **Log in to the active distributed management card.**

4. **Force a failover from the active distributed management card to the standby distributed management card:**

```
hostname cli> setfailover force
```

The active distributed management card becomes the new standby distributed management card.

5. **Repeat the instructions in Step 2 to flash update the new standby distributed management card.**

Running Scripts on the Distributed Management Cards

This section describes the Netra CT server distributed management card scripting feature.

Using Scripting

Normally, the distributed management card cannot execute batch commands. The distributed management card scripting feature allows you to write scripts to execute distributed management card CLI commands in batch mode, similar to using scripting in the Solaris OS. You run the scripts from a node board in the same system as the distributed management card.

As an example, using the scripting feature, you can write a script to configure an Ethernet port on the distributed management card, and then check to be sure it is configured the way you want. This sample script runs the `version` command, and the `setipmode`, `setipaddr`, `showipmode`, and `showipaddr` commands for Ethernet port 2 on the distributed management card:

```
rsh DMC_SysMgmt_ipaddress version
rsh DMC_SysMgmt_ipaddress setipmode -b 2 config
rsh DMC_SysMgmt_ipaddress setipaddr -b 2 addr
rsh DMC_SysMgmt_ipaddress showipmode -b 2
rsh DMC_SysMgmt_ipaddress showipaddr -b 2
```

The script includes the `rsh` command, the distributed management card System Management Network IP address, and the CLI command(s) to run. You can use the IP address for the distributed management card in slot 1A, the IP address for the distributed management card in slot 1B, or the alias IP address to stay connected to the active distributed management card. For information on the System Management Network IP address, refer to "Configuring the System Management Network" on page 21. For information on the CLI commands, refer to TABLE 3-1.

Scripting Limitations

All the active distributed management card CLI commands in TABLE 3-1 are supported in a script *except* the interactive commands `userpassword`, `password`, `console`, and `break`, and the command `logout`.

You can not use scripting on the standby distributed management card.

For security reasons, you must be a root user on a node board in the same system as the distributed management card. The commands can only be run over the System Management Network interface.

▼ To Run a Script on a Distributed Management Card

1. **Log in to the server.**
2. **Create a script:**

```
rsh DMC_SysMgmt_ipaddress  CLI_command
rsh DMC_SysMgmt_ipaddress  CLI_command
rsh DMC_SysMgmt_ipaddress  CLI_command
rsh DMC_SysMgmt_ipaddress  CLI_command
...
```

where *DMC_SysMgmt_ipaddress* is the System Management Network IP address of the distributed management card, and *CLI_command* is the CLI command you want to run.

3. **Save the script to a file.**
4. **As root, run the script:**

```
# /path/filename
```

where *path* is the path to the script and *filename* is the name of the script.

Before executing the commands in the script, the distributed management card verifies that the commands are being run by a root user on a node board in the same system as the distributed management card, and that the commands have been received over the System Management Network.

Displaying Board State Information

Board information, including type of board, power state of the board, and boot state of the board can be displayed for each slot in the system using the CLI `showcpustate` command.

Sample output from this command is:

```
hostname cli> showcpustate
-------------------------------------------------------
   Slot No   : Board Type   : Power_State : Boot_State
-------------------------------------------------------
      1A     :    DMC Board :          On :      Ready
      1B     :    DMC Board :          On :      Ready
       2     : SWITCH Board :          On :      Ready
       3     :    CPU Board :         Off :
       4     :        Empty :             :
       5     :    CPU Board :          On :      Ready
       6     :    CPU Board :          On :    Offline
       7     :    CPU Board :          On :    Offline
       8     :    CPU Board :          On :      Ready
       9     :    CPU Board :          On :    Offline
      10     :    CPU Board :          On :    Unknown

Press 'q' + Return to quit, hit Return to continue
```

TABLE 3-4 contains the various state descriptions.

TABLE 3-4 Board State Information

State	Value	Description
Power _State	On	The slot is powered on
Power _State	Off	The slot is powered off
Boot_State	Online	The board boot sequence has started
Boot_State	Ready	The board boot sequence has finished, and the board is ready to use
Boot_State	Offline	The board may be running its power-on self-test (POST), the board may be at the OpenBoot PROM level, or the boot may have failed
Boot_State	Unknown	The distributed management card can not determine the current boot state of the board

For third-party node boards, the `showcpustate` command returns a state of `unknown`.

Booting Node Boards

Node boards can boot from a local disk or over the network.

Board Power-on Sequence

When you power on the Netra CT 820 system by pressing the power switch on the rear of the system to the On (|) position, the boards power on in this sequence:

1. The two distributed management cards are powered on. The top card in slot 1A is designated as the active distributed management card, and the bottom card in slot 1B is designated as the standby distributed management card. Once the cards have booted and are ready for use, the Ready LED is solid green on the active card and blinking green on the standby card.

2. The active distributed management card powers on the switching fabric boards in slots 2 and 21. While the switching fabric boards are powering on, the blue LED state is solid. Once the boards have booted and are ready for use, the blue LED turns off.

3. The active distributed management card looks at the `Boot_Mask` field in the midplane FRU ID for boot servers and performs one of the following actions:

   - If one or more boot servers are designated in the `Boot_Mask` field, the active distributed management card powers on the boot servers first; once the boards have booted and are ready for use, the active distributed management card powers on the rest of the node boards together; these boards boot from the boot servers. The method of booting depends first on the value in the `Boot_Devices` field in the midplane FRU ID or secondly on the value in the OpenBoot PROM NVRAM `boot_device` configuration variable. After a board has booted and is ready for use, the Ready LED is solid green and the blue LED is off.

   or

   - If no boot server is designated in the `Boot_Mask` field, the active distributed management card powers on all the node slots together. Once the node boards are powered on, the method of booting depends first on the value in the `Boot_Devices` field in the midplane FRU ID or secondly on the value in the OpenBoot PROM NVRAM `boot_device` configuration variable. After a board has booted and is ready for use, the Ready LED is solid green and the blue LED is off.

A midplane FRU ID fault is a system fault, and no boards can be powered on.

Boot Device Variables

By default, the OpenBoot PROM NVRAM `boot-device` configuration variable is set to `disk net`, `disk` being an alias for the path to the local disk, and `net` being an alias for the path of the primary network. You can set the boot device for node boards through the distributed management card CLI `setfru` command. Refer to "Configuring a Chassis Slot for a Board" on page 18 for information on using the `setfru` command to specify a boot device for a board.

For example, you might want to change the node board in slot 3 to boot first from its PMC disk. To do this, check the current OpenBoot PROM `boot-device` variable:

```
ok printenv boot-device
boot-device =     disk net
ok
```

On the active distributed management card, check and change the `boot_devices` setting:

```
hostname cli> showfru slot 3 boot_devices
showfru: Boot_Devices:
hostname cli> setfru slot 3 boot_devices pmc0/disk net
hostname cli> showfru slot 3 boot_devices
showfru: Boot_Devices: pmc0/disk net
hostname cli>
```

After you power cycle the system, check the OpenBoot PROM `boot-device` variable:

```
ok printenv boot-device
boot-device =     pmc0/disk net
ok
```

When a node board is hot-swapped, power cycled, rebooted, or reset, the OpenBoot PROM firmware checks with the distributed management card for a boot device for that slot. The distributed management card sends the value from the `Boot_Devices` field in FRU ID to the OpenBoot PROM firmware; the value is either the boot device list for that slot you set using the `setfru` command or a null string if you did not set a boot device list for that slot. The value overwrites the NVRAM `boot-device` value. The board boots from the value in the `boot-device` variable if its `diag-switch?` variable is set to `false` (the default). The board boots from the value in the `diag-device` variable (the default is `net`) if its `diag-switch?` variable is set to `true.`

Booting with a DHCP Server

You can configure Netra CT node boards to boot over DHCP. This process includes setting the node board boot device for DHCP, forming the node board DHCP *client ID*, and configuring the DHCP server.

On the Netra CT system, the DHCP client ID is a combination of the system's midplane Sun part number (7 bytes), the system's midplane Sun serial number (6 bytes), and the board's geographical address (slot number) (2 bytes). The parts are separated by a colon (:).

▼ To Configure a Node Board to Boot Over DHCP

1. **Log in to the active distributed management card.**
2. **Set the boot device for the board to dhcp with the `setfru` command:**

```
hostname cli> setfru slot instance Boot_Devices network_devicename:dhcp
```

where *instance* is the slot number of the board to be configured for DHCP and *network_devicename* is a path or alias to a network device. For example, to set the boot device to dhcp for the node board in slot 4, enter the following:

```
hostname cli> setfru slot 4 Boot_Devices net:dhcp
```

3. **Get the Netra CT system part number and the system serial number with the `showfru` command:**

```
hostname cli> showfru midplane 1 Sun_Part_No
...
hostname cli> showfru midplane 1 Sun_Serial_No
...
```

4. **Form the three-part client ID by using the system part number, the system serial number, and the slot number, separated by colons. Then, convert the client ID to ASCII.**

For example, if the output from the `showfru` commands in Step 3 is 375-4335 (Sun part number) and 000001 (Sun serial number), and you want to form the client ID for the node board in slot 4, the client ID is 3754335:000001:04.

Translate the client ID to its ASCII equivalent. For example:

Client ID part	ASCII Representation
3754335	33 37 35 34 33 33 35
:	3A
000001	30 30 30 30 30 31
:	3A
04	30 34

Thus, the example client ID in ASCII is:

33 37 35 34 33 33 35 3A 30 30 30 30 30 31 3A 30 34.

5. **Configure the DHCP server.**

   Refer to the *Solaris DHCP Administration Guide* on the web site `http://docs.sun.com` for information on how to configure the DHCP server for remote boot and diskless boot clients.

   The client ID is retained across a node board power cycle, reboot, or reset. The distributed management card updates the client ID during a first-time power on or a hot-swap of a node board.

Connecting to Node Board Consoles from the Distributed Management Card

The Netra CT system provides the capability to connect to node boards and open console sessions from the active distributed management card.

You begin by logging in to the distributed management card through either the serial port or the Ethernet port. Once a console session with a node board is established, you can run Solaris system administration commands, such as `passwd`, read status and error messages, or halt the board in that particular slot.

Configuring Your System for Multiple Console Use

To enable your system to use multiple consoles, you set several variables, either at the Solaris level or at the OpenBoot PROM level. Set these variables on each node board to enable console use.

▼ To Configure Your System for Multiple Consoles

1. **Log in as root to the node board, using the on-board console port `ttya`.**
2. **Enter either set of the following commands to enable multiple consoles:**
   - From the Solaris level:

```
# eeprom “multiplexer-output-devices=ttya ssp-serial”
# eeprom “multiplexer-input-devices=ttya ssp-serial”
# eeprom input-device=input-mux
# eeprom output-device=output-mux
# reboot
```

or

- From the OpenBoot PROM level:

```
ok setenv multiplexer-output-devices ttya ssp-serial
ok setenv multiplexer-input-devices ttya ssp-serial
ok setenv input-device input-mux
ok setenv output-device output-mux
ok reset-all
```

Establishing Console Sessions Between the Distributed Management Card and Node Boards

Once you have configured your system for multiple console use, you can log in to the active distributed management card and open a console for a slot. The Netra CT system allows four console users per node board slot.

TABLE 3-5 shows the distributed management card CLI console-related commands that can be executed from the current login session on the distributed management card.

TABLE 3-5 Distributed Management Card CLI Console-Related Commands

Command	Description
`console` *cpu_node*	Enter console mode and connect to a specified node board, where *cpu_node* can be 3 through 20.
`break` *cpu_node*	Put the specified node board in debug mode, where *cpu_node* can be 3 through 20. Debug mode uses the OpenBoot PROM level.
`setescapechar` *value*	Set the escape character to be used in all future console sessions. The default is tilde (~). Refer to TABLE 3-6 for escape character use.
`showescapechar`	Show the current escape character.

Most node board consoles use the system management bus, but a board at the OpenBoot PROM level connects over the IPMI bus. There can be only one console user on the IPMI bus at any one time.

For example, if the board in slot 4 is at the OpenBoot PROM level, the user opening a console session connects to it over the IPMI bus. This causes the IPMI bus to be fully occupied and no other users can connect over that bus. If they try, an error message displays. However, other users can connect to boards in other slots over the system management bus. The system management bus is faster than the IPMI bus, while the IPMI bus is typically a more stable communication channel than the system management bus.

Once you have a console connection with a node board, you can issue normal Solaris commands. There are several escape character sequences to control the current session. TABLE 3-6 shows these sequences.

TABLE 3-6 Node Board Console-Related Escape Character Sequences

Sequence	Description
`~b`	Break from the Solaris level and enter the OpenBoot PROM (debug) level.
`~.`	End the console session.
`~g`	Determine the status (system management bus or IPMI) of the current console.
`~t`	Toggle between system management bus and IPMI.

▼ To Start a Console Session From the Distributed Management Card

1. **Log in to the active distributed management card.**

   You can log in to the distributed management card through a terminal attached to either the serial port connection or the Ethernet port connection.

2. **Open a console session to a board in a slot:**

```
hostname cli> console cpu_node
```

where *cpu_node* is 3 through 20. For example, to open a console to the board in slot 4, enter the following:

```
hostname cli> console 4
```

You now have access to the board in slot 4. Depending on the state of the board in that particular slot, and whether the previous user logged out of the shell, you see one of several prompts:

- `console login%` (Solaris level)
- `#` (Solaris level, previous user logged in as root, and did not log out before disconnecting from the console)
- `ok` (OpenBoot PROM level, previous user did not log out before disconnecting from the console)

▼ To Determine the Status of the Current Console

- Enter the escape sequence `~g` at the start of a new line:

```
~g
```

A message displays, indicating the current state of the console connection. The message is either:

```
Console mode is IPMI
```

This means the console is in Solaris mode or OpenBoot PROM mode.

Or the message might be:

```
Console mode is NET
```

This means the console is in Solaris mode.

To Toggle Between the System Management Bus and IPMI

Toggling between the system management bus and IPMI could be useful for troubleshooting. For example, if the console stops working for some reason, you could try toggling to IPMI (the more reliable communication channel).

1. **If the node board is in Solaris mode, enter the escape sequence `~t`:**

```
# ~t
New console mode is IPMI
#
```

The console switches between the system management bus and IPMI mode. The console now fully occupies the IPMI bus. No other console may be at the OpenBoot PROM level at the same time. If another user attempts to access a board that is occupying the IPMI bus, the console connection fails.

2. **To return to the system management bus mode, enter `~t` again:**

```
# ~t
New console mode is NET
#
```

To Break Into OpenBoot PROM From the Console

- At the Solaris prompt, enter the escape sequence ~b:

```
# ~b
```

The console mode switches to IPMI:

```
New console mode is IPMI
Type ‘go’ to resume
ok
```

You can now debug from the OpenBoot PROM level.

To End the Console Session

1. **(Optional) Log out of the Solaris shell.**
2. **At the prompt, disconnect from the console by entering the escape sequence ~. (tilde period):**

```
prompt ~.
hostname cli>
```

Disconnecting from the console does not automatically log you out from the remote host. Unless you log out from the remote host, the next console user who connects to that board sees the shell prompt of your previous session.

To Show the Current Escape Character

- At the distributed management card prompt, enter the following command:

```
hostname cli> showescapechar
```

The current escape character is displayed:

```
hostname cli> escape_char: value
```

▼ To Change the Default Escape Character

- At the active distributed management card prompt, enter the following command:

```
hostname cli> setescapechar value
```

where *value* is any printable character. For example, to change the default escape character from tilde (~) to pound sign (#), enter the following:

```
hostname cli> setescapechar #
```

The pound sign is now the escape character for all future console sessions.

Using the PMS Application for Recovery and Control of Node Boards

This section describes specifying recovery operations and controlling node boards through the active distributed management card PMS CLI commands.

Recovery Configuration of a Node Board From the Distributed Management Card

You specify the recovery configuration of a node board by using the command `pmsd operset -s` *slot_num*`|all` (a single slot number or all slots in the Netra CT system containing a node board) and the recovery mode for the specified slot(s).

The recovery configuration can be maintenance mode, operational mode, or none mode. *Maintenance mode* means the distributed management card's automatic recovery of a node board is disabled, and PMS applications are started in an offline state, so that you can use manual maintenance operations. *Operational mode* means the distributed management card's automatic recovery of a node board is enabled; the distributed management card recovers the node board in the event of a monitoring fault, and starts PMS applications in an active state. *None mode* means the distributed management card's automatic recovery mode may be manually enabled or disabled; PMS application states are not enforced.

The mode is stored in persistent storage. You specify the operation to be performed on the specified slot by using the option `-o` with one of the following parameters: `maint_config` (set the hardware, operating system, and applications into maintenance mode), `oper_config` (set the hardware, operating system, and applications into operational mode), `none_config` (set the hardware, operating system, and applications into no enforcement mode), or `graceful_reboot` (bring the applications offline if needed and then reboot the operating system).

▼ To Specify the Recovery Configuration of a Node Board

1. **Log in to the active distributed management card.**

2. **Configure the automatic recovery mode with the `operset` command:**

```
hostname cli> pmsd operset -s slot_num|all -o
maint_config|oper_config|none_config|gracefulreboot
```

where *slot_num* can be a slot number from 3 to 20, and `all` specifies all slots containing node boards. For example, to make PMS recovery operational for the entire Netra CT server, enter:

```
hostname cli> pmsd operset -s all -o oper_config
```

Printing PMS Recovery Configuration Information

The `pmsd infoshow -s` *slot_num*|`all` command can be used to print the recovery configuration and alarm status for the recovery configuration.

The `pmsd historyshow -s` *slot_num*|`all` command can be used to print a recovery configuration and runtime message log. The log is printed to the ChorusOS terminal performing the operation.

Detailed Recovery of a Board in Case of Fault

You can perform detailed, manual recovery operations on a board or instruct PMS to perform detailed, automatic recovery operations on a board using the CLI. The operations are performed across the hardware, the operating system, and the applications.

For manual recovery, use the `pmsd recoveryoperset -s` *slot_num*|`all` command. This command can only be run when the board is in *maintenance mode* or *none mode* (PMS applications are offline). You specify the recovery operation to be performed on the specified slot by using the option `-o` with one of the following parameters: `pc` (power cycle), `rst` (reset), `rstpc` (reset, then power cycle), `pd` (power down), or `rb` (reboot).

For automatic recovery, use the `recoveryautooperset -s` *slot_num*|`all` command. This command sets how PMS responds to a fault when the board is in *operational mode* (PMS applications are active).

You specify the automatic recovery operation to be performed on the specified slot by using the option `-o` with one of the following parameters: `pc` (power cycle), `rst` (reset), `rstpc` (reset, then power cycle), `pd` (power down), or `rb` (reboot), `rbpc` (reboot, then power cycle), `none` (no recovery), or `trg` (manually simulate a fault to trigger a recovery). Optional parameters for automatic recovery include: `-d startup` *delay* (the time in deciseconds between a fault occurrence and the start of a recovery operation; default is 0 deciseconds), `-f off|on` (whether a power down operation will occur if the recovery operation fails; `on` specifies power down will occur and `off` specifies that power down will not occur; the default is `off`), `-r` *retries* (the number of times a recovery operation can occur and fail before it is terminated; the default is one try), `-n` *inter_op_delay* (the time in deciseconds between one and the next operation for an operation with multiple retries; the default is 0 deciseconds [1 decisecond equals 10 milliseconds]; you should change the default to a number other than 0, for example, 4000 [equals 40 seconds], to allow time between the operations), and `-p` *reset_power-cycle_delay* (the time in deciseconds to be waited between the reset and power cycle portions of the recovery operation before a failed reset is declared and the power cycle portion of the operation starts; default is 0 deciseconds).

To Manually Recover a Board

1. **Log in to the active distributed management card.**
2. **Perform manual recovery operations on a board with the `recoveryoperset` command:**

```
hostname cli> pmsd recoveryoperset -s slot_num|all -o pc|rst|rstpc|pd|rb
```

where *slot_num* can be a slot number from 3 to 20, and `all` specifies all slots containing node boards. For example, to instruct PMS to reboot slot 5 after a fault, enter the following:

```
hostname cli> pmsd recoveryoperset -s 5 -o rb
```

▼ To Automatically Recover a Board

1. **Log in to the active distributed management card.**

2. **Perform automatic recovery operations on a board with the** recoveryoperset **command:**

```
hostname cli> pmsd recoveryautooperset -s slot_num|all -o
pc|rst|rstpc|pd|rb|rbpc|none|trg [-d startup delay][-f on|off][-r retries][-n
inter_op_delay][-p reset_power-cycle_delay]
```

where *slot_num* can be a slot number from 3 to 20, and `all` specifies all slots containing node boards. For example, to instruct PMS to automatically reboot slot 5 after a fault, with the default delays, retries, and failure power state, enter the following:

```
hostname cli> pmsd recoveryautooperset -s 5 -o rb
```

Printing PMS Automatic Recovery Information

The `pmsd recoveryautoinfoshow -s` *slot_num*|`all` command can be used to print information showing the configuration information affected by the `recoveryautooperset` command.

Monitoring and Controlling a Node Board's Resources From the Distributed Management Card

PMS can perform operations on a board's hardware, the operating system, and applications. You can specify that PMS performs operations on one of these, rather than all.

Hardware Operations

The `pmsd hwoperset -s` *slot_num*|`all` command performs operations on the hardware. The operations can only be performed in maintenance or none mode unless the optional `-f` parameter is used. You specify the operation to be performed on the specified slot by using the option `-o` with one of the following parameters: `powerdown` (set the hardware to the power-off state), `powerup` (set the hardware to the power-on state), `reset` (reset the hardware), `mon_enable` (enable health monitoring of the hardware), or `mon_disable` (disable health monitoring of the hardware). The optional `-f` parameter can be used to perform the operation even if applications are in the active state, and the slot is in operational mode.

The `pmsd hwinfoshow -s` *slot_num* | `all` command can be used to print PMS system information on the hardware state, monitoring status, and alarm status (whether an alarm was generated).

The `pmsd hwhistoryshow -s` *slot_num* | `all` command can be used to print a short log (one-line descriptions) of messages pertaining to changes in the hardware's operation. The log is printed to the ChorusOS terminal performing the operation.

Operating System Operations

The `pmsd osoperset -s` *slot_num* | `all` command performs operations on the operating system. The operations can only be performed in maintenance or none mode unless the optional `-f` parameter is used. You specify the operation to be performed on the specified slot by using the option `-o` with one of the following parameters: `reboot` (reboot the operating system), `mon_enable` (enable health monitoring of the operating system), or `mon_disable` (disable health monitoring of the operating system). The optional `-f` parameter can be used to perform the operation even if applications are in the active state, and the slot is in operational mode.

The `pmsd osinfoshow -s` *slot_num* | `all` command can be used to print PMS system information on the operating system state, monitoring status, and alarm status (whether an alarm was generated).

The `pmsd oshistoryshow -s` *slot_num* | `all` command can be used to print a short log (one-line descriptions) of messages pertaining to changes in the operating system's operation. The log is printed to the ChorusOS terminal performing the operation.

Application Operations

The `pmsd appoperset -s` *slot_num* | `all` command performs operations on the applications. The operations can only be performed in the none mode. You specify the operation to be performed on the specified slot by using the option `-o` with one of the following parameters: `force_offline` (force the applications to an offline state), `vote_active` (move the group of applications to the active state only if all of the applications agree to be moved), or `force_active` (force the applications to the active state).

The pmsd appinfoshow -s *slot_num* | all command can be used to print PMS system information on the applications' state and alarm status (whether an alarm was generated).

The pmsd apphistoryshow -s *slot_num* | all command can be used to print a short log (one-line descriptions) of messages pertaining to changes in the applications' operation. The log is printed to the ChorusOS terminal performing the operation.

Printing Other PMS Information

The pmsd version command prints the current version of pmsd.

The pmsd usage command prints a synopsis of the pmsd commands.

Monitoring Your System

This section describes various ways to monitor your system.

CLI Information

The distributed management card CLI provides many commands to display system status. Refer to the distributed management card CLI commands in the section, "Using the Distributed Management Card CLI" on page 51, in particular the show commands, to view system status.

The MOH Application

The MOH collects information about individual field replaceable units (FRUs) in your system and monitors their operational status. MOH can also monitor certain daemons. For example, if you installed the Netra High Availability Suite, MOH monitors daemons through that application.

Starting and Stopping MOH

If you installed the Solaris patches for MOH in a directory other than the default directory, specify that path instead. You must start the MOH application as root.

```
# cd /opt/SUNWnetract/mgmt3.0/bin
# ./ctmgx start [option]
```

Refer to TABLE 2-7 for the options available with `ctmgx start`.

```
# cd /opt/SUNWnetract/mgmt3.0/bin
# ./ctmgx stop
```

Once MOH is running, it interfaces with your SNMP or RMI application to discover network elements, monitor the system, and provide status messages. Refer to the *Netra CT Server Software Developer's Guide* for information on writing applications to interface with the MOH application.

Additional Troubleshooting Information

In the event of an active distributed management card fault, hot-swapping is not supported.

For additional troubleshooting information, refer to the *Netra CT 820 Server Service Manual*.

APPENDIX **A**

Third-Party Node Boards

Third-party cPSB-only node boards that are PICMG 2.16-compliant may be used in the Netra CT 820 server. These boards do not necessarily run the Solaris OS, and they do not run the Netra CT 820 server system management software, such as MOH. Because of this, they cannot be managed to the same extent as the Netra CP2300 cPSB board.

This appendix contains information on:

- "Third-Party Node Board FRU ID Information" on page 87
- "CLI Commands Supported on a Third-Party Node Board" on page 89

Third-Party Node Board FRU ID Information

You can set and display certain FRU ID information for third-party node boards using the CLI `setfru` and `showfru` commands.

▼ To Configure a Chassis Slot for a Third-Party Node Board

1. **Log in to the active distributed management card.**

2. **Set the acceptable FRU for the slot:**

```
hostname cli> setfru slot instance Acceptable_Fru_Types nonsun:picmg2.16
```

where *instance* is the chassis slot number to be configured; valid values are 3 to 20. For example, if you want to set chassis slot 6 to allow a third-party node board, enter the following:

```
hostname cli> setfru slot 6 Acceptable_Fru_Types nonsun:picmg2.16
```

3. **Insert the third-party node board into the configured slot.**

4. **Completely power off and on the system by locating the power switch at the rear of the Netra CT 820 server; press it to the Off (O) position, then press it to the On (|) position.**

   The slot is now configured to accept a third-party cPSB-only node board.

To Display FRU ID Information for a Third-Party Node Board

1. **Log in to the active distributed management card.**

2. **Enter the** `showfru` **command:**

```
hostname cli> showfru slot instance fru_property
```

where *instance* is a chassis slot number from 3 to 20, and *fru_property* can be `Part_No` (part number) or `Serial_No` (serial number). For example, to display part number FRU ID information for a third-party node board in slot 6, enter the following:

```
hostname cli> showfru slot 6 Part_No
```

For more information on FRU ID, refer to "Specifying Netra CT Server FRU ID Information" on page 11 and "Displaying Netra CT Server FRU ID Information" on page 15.

CLI Commands Supported on a Third-Party Node Board

A limited number of CLI commands from the active distributed management card support third-party node boards. TABLE A-1 lists and describes these commands.

TABLE A-1 CLI Commands Supported on a Third-Party Node Board

Command	Permission	Description
`poweron` *cpu_node*	r	Power on the specified node slot, where *cpu_node* can be 3 through 20.
`poweroff` *cpu_node*	r	Power off the specified node slot, where *cpu_node* can be 3 through 20.
`reset -x` *cpu_node*	r	Reset (reboot) a specified node. `reset -x` produces a hard reset (resets the board), where *cpu_node* can be 3 through 20.
`setfru [-h]` *fru_name instance fru_property value*	a	Set FRU ID information. See "Third-Party Node Board FRU ID Information" on page 87 for more information.
`showfru` *fru_name instance fru_property*	a	Display FRU ID information. See "Third-Party Node Board FRU ID Information" on page 87 for more information.
`showcpustate`		Display the board type, power state, and boot state for each slot in the system. For third-party node boards, this command returns a boot state of `unknown`.

For more information on using the CLI, refer to "Using the Distributed Management Card CLI" on page 51.

APPENDIX **B**

Software Error Messages

This appendix contains information on Netra CT 820 platform-specific software error messages. Messages are produced by software and firmware running on the distributed management card, and by software and firmware running on the Netra CT 820 system, including: the Solaris OS, OpenBoot PROM firmware, the MOH application, and the PMS application.

For Netra CT 820 platform-specific hardware error messages, refer to the *Netra CT 820 Server Service Manual*.

For additional information on software error messages not specific to the Netra CT 820 system, refer to:

- The web site `http://docs.sun.com` for the Solaris OS, OpenBoot, DHCP, and ChorusOS documentation
- The web site `http://www.sun.com/products-n-solutions/hardware/docs` for Netra High Availability Suite documentation
- Third-party board documentation for any third-party node boards you are using

This appendix includes the following sections:

- "Overview" on page 91
- "Messages" on page 93

Overview

This appendix lists error messages in alphabetical order, with the format:

`Message`, **Cause**, **Action**

Distributed Management Card Messages. Error messages originate from software and firmware on the distributed management card itself, such as ChorusOS, BMC, and the CLI. In addition, messages from other software, such as the PMS application and the OpenBoot PROM firmware, might be displayed on the distributed management card console.

Distributed management card error messages are displayed on the distributed management card console. They are not saved to a log.

Solaris OS Messages. Messages are displayed on the Netra CP2300 cPSB Board console. They are saved to a log in `/var/adm/messages`.

OpenBoot PROM Firmware Messages. Messages from OpenBoot PROM are displayed through a Netra CP2300 cPSB Board console. They can be displayed on the node board console itself or on the distributed management card console if you are logged in remotely using the CLI `console` command.

OpenBoot PROM error and warning messages are not saved to a log on either the distributed management card or on a node board.

MOH Application Messages. These messages are displayed on a node board console, on the distributed management card console, or on both. They are not saved to a log.

PMS Application Messages. PMS is a high-level application. Thus, faults in various places in the software and hardware underlying this application can result in PMS error messages. For example, a fault could occur on the midplane or on a disk. This situation might make it difficult to isolate where a specific fault is occurring. A solution to many PMS error messages is to reset the distributed management card.

PMS error messages are printed to the console you are using to execute the `pmsd` CLI command; they are not saved to a log on either the distributed management card or on a node board.

Messages

```
!!! ALERT !!! Crossing Critical temperature threshold
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
!!! ALERT!!! Crossing Shutdown temperature threshold
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
An attempt to start the "protocol" communication server failed,
will retry.
The problem could be because of a misconfigured primary network
interface, or possibly another instance of the agent is running
```

Cause: A network configuration problem, another MOH agent instance, or another application or process using the MOH port has resulted in the MOH agent’s inability to start the RMI server.

Action: If this message occurs on the distributed management card, check the network interfaces (for example, make sure the ifeth0 interface has a valid IP address). If this message occurs on a node board, check the network interfaces;

check to see if an MOH agent is already running, with the command `pgrep -fl java`; try stopping and restarting the agent; check to see which ports are in use, with the command `netstat -a`.

```
An attempt to start the SnmpView failed, will retry.
Check the network configuration
```

Cause: The MOH agent could not start the SNMP view, because of a network configuration problem or because another application or process is using the SNMP port.

Action: This message could occur on the distributed management card or on a node board. (1) Check the network configuration. (2) Check to see which ports are in use, with the command `netstat -a`.

```
Board RG0 resource state must be OFFLINE to perform operation
on this slot
```

Cause: Resource Group 0 (RG0), the group of applications on a node board that PMS manages, must be offline before you can run certain commands from the distributed management card.

Action: Change the RG0 state from active to offline. For example, use the command `pmsd appoperset -o force_offline`.

```
Can't reset: Standby is not healthy
```

Cause: You tried to reset the active distributed management card, but the standby distributed management card is not in a healthy state.

Action: Verify the status of the standby distributed management card with the `showhealth` command. Reset the standby distributed management card if necessary.

```
CLI: unknown command: use help for valid commands
```

Cause: (1) You used a distributed management card CLI command that is not a valid command. (2) On the standby distributed management card, you used an active distributed management card CLI command that is not valid on the standby distributed management card.

Action: For a list of valid CLI commands, either use the CLI `help` command or refer to TABLE 3-1 and TABLE 3-2.

```
Configuration Download Error: Node
card in slot number failed to poweron
```

Cause: The midplane FRU ID is corrupted. The distributed management card is unable to communicate with the node board over IPMI.

Action: Contact SunService[SM].

`console:  All console sessions busy to slot` *number*

Cause: From the distributed management card, you tried to open a console session to a node board, but the maximum four console sessions for that node board are already open.

Action: Either retry connecting later or free up a session to that node board.

`console: failed to connect to console in slot` *number*

Cause: This message on the distributed management card console could indicate an IPMI bus problem or a node board configuration problem after you try to open a console connection to a node board.

Action: Try opening a console connection to a different slot. If this fails, reset the distributed management card and try reconnecting to the same slot.

`DM board or switch board slot not managed by PMS Daemon`

Cause: Many `pmsd` CLI commands can generate this message.

Action: PMS does not manage the distributed management card or the switching fabric boards.

`DMC is in Standby Mode: pmsd operations not available`

Cause: You issued a `pmsd` CLI command on the standby distributed management card; `pmsd` operations are not available on the standby distributed management card.

Action: Use the active distributed management card for `pmsd` commands.

`Error Disabling CPU Sensor`

Cause: The `reset-all` OpenBoot PROM command could generate this message. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists, the board might need to be returned to SunService.

`Error Disabling Temperature Sensor`

Cause: This message could occur at power on of the node board, after POST has completed, but before the OpenBoot PROM prompt is displayed. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists at power on, the board might need to be returned to SunService.

`Error Disabling the Watchdog`

Cause: The following OpenBoot PROM commands could generate this message: `reset-all`, `flash-update`, `delete-dropin`, `add-dropin`, `flat-update`, `flash-from-rombo`. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists, the board might need to be returned to SunService.

`Error Enabling Temperature Sensor`

Cause: This message could occur at power on of the node board, after POST has completed, but before the OpenBoot PROM prompt is displayed. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists at power on, the board might need to be returned to SunService.

`Error Setting Temperature Threshold`

Cause: This message could occur at power on of the node board, after POST has completed, but before the OpenBoot PROM prompt is displayed. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists at power on, the board might need to be returned to SunService.

`Failover Manager:Partner DMC at` *state* `state, partner is not ready to take over ACTIVE role`

Cause: This message results from using either the `setfailover force` command or the `reset dmc` command on the active distributed management card when the standby distributed management card is not ready to become the active card.

Action: Wait for the standby distributed management card to be ready before using the `setfailover force` or `reset dmc` commands.

`Failover Manager:Partner DMC is unhealthier, local DMC will remain ACTIVE`

Cause: This message results from using either the `setfailover force` command or the `reset dmc` command on the active distributed management card when the standby distributed management card is not in a healthy state and able to become the active card.

Action: Wait for the standby distributed management card to be healthy before using the `setfailover force` or `reset dmc` commands.

`Failover Manager:Partner event - A service failure`

Cause: A service, such as ntp or MOH, on either distributed management card has failed.

Action: If the `setdmcrecovery` mode is `on`, the newly active distributed management card will try to recover the failed distributed management card; if the `setdmcrecovery` mode is `off`, try to reset the failed distributed management card from the newly active distributed management card.

`Failover Manager:Partner event - DMC Initialization failure`

Cause: After a reset, the distributed management card could not come to a "ready" state.

Action: If the `setdmcrecovery` mode is `on`, the newly active distributed management card will try to recover the failed distributed management card; if the `setdmcrecovery` mode is `off`, try to reset the failed distributed management card from the newly active distributed management card.

`Failover Manager:Partner event - External ethernet interface down`

Cause: This message occurs if either distributed management card's external Ethernet link goes down, and the `setetherfailover` mode is set to `enable`.

Action: Check the cable connection on the external Ethernet port.

`Failover Manager:Partner event - KCS interface failure`

Cause: BMC firmware on either distributed management card is not responding.

Action: If the `setdmcrecovery` mode is `on`, the newly active distributed management card will try to recover the failed distributed management card; if the `setdmcrecovery` mode is `off`, try to reset the failed distributed management card from the newly active distributed management card.

`Failover Manager:Partner event - SysBus Interface down`

Cause: A distributed management card's internal link to the switching fabric board failed.

Action: Check the corresponding switching fabric board state; reset the switching fabric board. If the error still occurs, contact SunService.

`Failover Manager:Partner event - Unexpected reset of local BMC`

Cause: The BMC on the active distributed management card reset itself.

Action: If the `setdmcrecovery` mode is `on`, the newly active distributed management card will try to recover the failed distributed management card; if the `setdmcrecovery` mode is `off`, try to reset the failed distributed management card from the newly active distributed management card.

`Failover Manager:Partner failed - Partner boot failure`

Cause: This message occurs if either distributed management card doesn't boot after a reset or recovery within the expected boot-up time.

Action: Contact SunService.

`Failover Manager:Partner failed - Partner Healthy# down`

Cause: The active distributed management card's #HEALTHY signal is down due to a panic, a watchdog timer generated reset, or a hardware fault.

Action: If the `setdmcrecovery` mode is `on`, the distributed management card should recover. If it does not, contact SunService.

`Failover Manager:Partner failed - Partner Heartbeat down`

Cause: The internal heartbeat mechanism between the two distributed management cards detected a heartbeat failure.

Action: Contact SunService.

`Failover Manager:Recovering the partner for` *number* `time`

Cause: This message occurs during attempted recovery of a failed distributed management card.

Action: None needed. If the failed distributed management card does not recover, contact SunService.

`Failover Manager:Recovery attempts are exhausted. No more recovery of the failed partner`

Cause: Three successive recovery attempts have failed on the failed distributed management card.

Action: Contact SunService.

`Invalid cpu_node number:` *number*

Cause: You entered an invalid node board number for a console connection from the distributed management card.

Action: Enter a valid node number, 3 through 30.

```
Invalid IP mode
```

Cause: You specified an invalid syntax for the CLI command `setipmode`.

Action: The `setipmode` usage is: `setipmode -b` *port_num* `rarp|config|none`. Refer to "Configuring the Distributed Management Cards' Ethernet Ports" on page 8 for more information.

```
Invalid slot number
```

Cause: You specified an invalid slot number for a CLI command that accepts a slot number option.

Action: Refer to TABLE 3-1 for the correct syntax for that particular command.

```
IP address for the system management bus interface not found -
For distributed agent functionality
Please check the following interface configuration : interface
```

Cause: The MOH application needs an IP address for the system management network to be able to communicate between the distributed management card and the node boards. This message displays if the distributed management card or a node board does not have an IP address for the system management network interface, or if either of these interfaces failed to initialize.

Action: Configure the specified interface and restart the MOH application.

```
Lower Critical - going high
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
Lower Critical - going low
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
Lower Non-critical - going high
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
Lower Non-critical - going low
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
Lower Non-recoverable - going high
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
Lower Non-recoverable - going low
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
NFS Portmap: RPC: Rpcbind failure - RPC: Timed out
```

Cause: Using the CLI `flashupdate` command with the NFS option might cause NFS timeouts.

Action: (1) Make sure the NFS path is a shared NFS mount. (2) If the shared NFS server is on a different network, make sure that the gateway is properly configured. (3) Check the distributed management card network configuration.

```
OS is not up on this CPU node
```

Cause: From the distributed management card, you tried to reset a node board, which would reboot the node board under the Solaris OS. However, the node board is at the OpenBoot PROM prompt.

Action: Either use the `reset -x` command to force a hard reboot of the node board or bring up the operating system on the node board and then reboot it.

```
Permission denied
```

Cause: You used a distributed management card CLI command for which you do not have the correct user permissions.

Action: For information on CLI command user permissions, either use the CLI `help` command or refer to "CLI Commands" on page 52.

```
Recovery of failed active is a must for failover, ignoring
dmcrecovery flag
```

Cause: This message occurs if the `setfailover` mode is `on`, but the `setdmcrecovery` mode is `off`, and a failover event occurs that requires the standby distributed management card to recover the failed, active distributed management card before the standby distributed management card can become active.

Action: If the standby distributed management card can not recover the failed, active distributed management card, contact SunService.

```
showfru: failed to get the FRU property
```

Cause: The CLI `showfru` command may generate this message. It indicates either (1) A FRU ID (midplane, node board, or third-party node board) is not programmed; or (2) An IPMI bus problem occurred.

Action: (1) Make sure your hardware has the FRU ID programmed, for example, check to see if you can read a different FRU property. (2) Reset the distributed management card. (3) Power cycle the system. (4) If the error still occurs, contact SunService.

```
Slot not configured to be managed by PMS Daemon
```

Cause: Many `pmsd` CLI commands can generate this message.

Action: Use the `pmsd slotaddressset` command to set the IP address for the slot.

```
Slot/powersupply is already in powered off/on state
```

Cause: You tried to power off or power on a slot or a power supply that is already powered off or powered on.

Action: No action needed.

`SMD is re-booting the DMC because of the failure of Service` *service*

Cause: A particular service on the distributed management card has failed, and the service monitoring daemon will reset the distributed management card.

Action: Reset the distributed management card. If the error still occurs, contact SunService.

`smd:startup:run_ntpdate: Not Valid NTP Server`

Cause: The service monitoring daemon on the distributed management card has detected that the NTP server address is either 0.0.0.0 or 256.256.256.256, which are invalid NTP server addresses.

Action: Configure the NTP server using the `setntpserver` command. Refer to "Setting the Date and Time on the Distributed Management Cards" on page 34 for more information.

`SUNW_envmond: current temperature (`*temp*`) exceeds upper warning temperature (`*temp*`)`

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the temperature threshold settings (`prtpicl -v -c temperature-sensor`) to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information).

`SUNW_envmond: current temperature (`*temp*`) exceeds upper critical temperature (`*temp*`)`

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the temperature threshold settings (`prtpicl -v -c temperature-sensor`) to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information).

`SUNW_envmond: current temperature (`*`temp`*`) is below lower warning temperature (`*`temp`*`)`

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the temperature threshold settings (`prtpicl -v -c temperature-sensor`) to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information).

`SUNW_envmond: current temperature (`*`temp`*`) is below lower critical temperature (`*`temp`*`)`

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the temperature threshold settings (`prtpicl -v -c temperature-sensor`) to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information).

`SUNW_picl_watchdog: Error in opening SMC drv`

Cause: The watchdog timer failed to access the Netra CT system management controller (SMC) driver.

Action: (1) Check whether your watchdog timer application is accessing the watchdog correctly (refer to the *Netra CP2300 cPSB Board Programming Guide* or to the *Netra CT 820 Server Software Developer's Guide* for more information). (2) Reboot the node board.

`SUNW_picl_watchdog: Error in patting the watchdog`

Cause: The watchdog timer failed to access the Netra CT system management controller (SMC) driver.

Action: (1) Check whether your watchdog timer application is accessing the watchdog correctly (refer to the *Netra CP2300 cPSB Board Programming Guide* and/or to the *Netra CT 820 Server Software Developer's Guide* for more information). (2) Reboot the node board.

`SUNW_picl_watchdog: Error in writing to SMC`

Cause: The watchdog timer failed to access the Netra CT system management controller (SMC) driver.

Action: (1) Check whether your watchdog timer application is accessing the watchdog correctly (refer to the *Netra CP2300 cPSB Board Programming Guide* and/or to the *Netra CT 820 Server Software Developer's Guide* for more information). (2) Reboot the node board.

`Unable to communicate with CPU board PMS Daemon`

Cause: Several `pmsd` CLI commands can generate this message.

Action: (1) Check network connectivity. (2) Check to see if the `ping` command works between the distributed management card and the node board. (3) Check the status of the node board with the `pmsd slotrndaddersshow` command, and modify if appropriate, with the `pmsd slotrndaddersssadd` command.

`Unable to communicate with DM board PMS Daemon`

Cause: Many `pmsd` CLI commands can generate this message. The distributed management card CPU might be temporarily overloaded.

Action: (1) Retry the command after waiting 15 seconds or more. (2) Reset the distributed management card.

`Unable to connect to CPU board PMS Daemon`

Cause: Several `pmsd` CLI commands can generate this message.

Action: (1) Check network connectivity. (2) Check to see if the `ping` command works between the distributed management card and the node board. (3) Check the status of the node board with the `pmsd slotrndaddersshow` command, and modify if appropriate, with the `pmsd slotrndaddersssadd` command.

`Unable to connect to DM board COSL`

Cause: Many `pmsd` CLI commands can generate this message. The message usually results from PMS being unable to monitor or control the hardware. PMS cannot get the information it needs from the lower-level common operating system library (COSL) hardware interface.

Action: Reset the distributed management card.

`Unable to connect to DM board PMS Daemon`

Cause: Many `pmsd` CLI commands can generate this message. The distributed management card CPU might be temporarily overloaded.

Action: (1) Retry the command after waiting 15 seconds or more. (2) Reset the distributed management card.

`Unable to connect to the ctmgx agent`

Cause: This message occurs if the `ctmgx stop` command is issued on a node board, and the MOH agent can't be contacted.

Action: Check to see whether the MOH application is running on the node board using the command `pgrep -fl java`. If it is running, kill the process with the command `kill` *process_id*.

`Unable to disconnect from DM board COSL`

Cause: Many `pmsd` CLI commands can generate this message. The message usually results from PMS being unable to monitor or control the hardware. PMS cannot get the information it needs from the lower-level common operating system library (COSL) hardware interface.

Action: Reset the distributed management card.

`Unable to fetch valid data from DM board COSL`

Cause: Many `pmsd` CLI commands can generate this message. The message usually results from PMS being unable to monitor or control the hardware. PMS cannot get the information it needs from the lower-level common operating system library (COSL) hardware interface.

Action: Reset the distributed management card.

`Unable to get valid data for this slot`

Cause: Many `pmsd` CLI commands can generate this message. The most probable cause is that PMS is having trouble communicating with the hardware or the node boards.

Action: (1) Check network connectivity. (2) Reset the distributed management card. (3) Reboot the node boards.

`Unable to perform operation on empty slot/entry`

Cause: Many `pmsd` CLI commands can generate this message.

Action: If you want to use PMS to control the slot, put a board in the slot.

`Unable to perform operation on this slot`

Cause: Many `pmsd` CLI commands can generate this message. For example, if you used the `pmsd hardware -o reset` command on a slot that was empty or not powered on, this message would display.

Action: Compare the command issued and the state of the slot.

```
Unable to perform operation on this slot/entry
```

Cause: Several `pmsd` CLI commands can generate this message.

Action: (1) Make sure the remote distributed management card and the remote node board are operational. (2) Check network connectivity. (3) Check the status of the node board with the `pmsd slotrndaddersssshow` command, and modify if appropriate, with the `pmsd slotrndaddersssadd` command.

```
Unable to start CPU board PMS Daemon
```

Cause: The PMS daemon can’t be started on a node board.

Action: (1) Check to see if a PMS daemon is already running on the node board; if there is, stop the daemon and try restarting it. (2) Reboot the node board and try restarting the daemon.

```
Unable to start DM board PMS Daemon
```

Cause: May occur after the CLI `pmsd start` command is used. The PMS daemon can’t be started on the distributed management card.

Action: (1) Check to see if a PMS daemon is already running; if there is, stop the daemon and try restarting it. (2) Reset the distributed management card.

```
Unable to stop CPU board PMS Daemon
```

Cause: The PMS daemon can’t be stopped on the node board.

Action: (1) Check to see if a PMS daemon is already running; if there is, stop the daemon with the command `kill` *process number*. (2) Reboot the node board.

```
Unable to stop DM board PMS Daemon
```

Cause: This error might occur after the CLI `pmsd stop` command is used. The PMS daemon can’t be stopped on the distributed management card.

Action: (1) Check to see if a PMS daemon is already running; if there is, stop the daemon with the Chorus command `akill` *process number*. (2) Reset the distributed management card.

```
Unable to write default data to DM board COSL
```

Cause: Many `pmsd` CLI commands can generate this message. The message usually results from PMS being unable to monitor or control the hardware. PMS cannot get the information it needs from the lower-level common operating system library (COSL) hardware interface.

Action: Reset the distributed management card.

```
Unrecognized property name
failed to get the FRU property
```

Cause: The CLI `showfru` command might generate this message. You entered an invalid FRU property.

Action: Refer to TABLE 2-2 for valid syntax.

```
Upper Critical - going low
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
Upper Non-critical - going low
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
Upper Non-recoverable - going low
The current threshold setting is: number degreeC
The current temperature is : number degreeC
```

Cause: A temperature problem, either in the chassis environment (for example, a fan failure) or as configured on the node board (for example, a user misconfiguration of a temperature setting), causes this message.

Action: (1) Check the fans to make sure they are working properly; replace if necessary. (2) Check the room environment for proper cooling and adjust if necessary. (3) Check the OpenBoot PROM environment variables `warning-temperature`, `critical-temperature`, and `shutdown-temperature` to make sure they are within range of the chassis environment (refer to the *Netra CP2300 cPSB Board Programming Guide* for more information) and adjust the environment variables as necessary.

```
WARNING: Could not check healthy line status!
```

Cause: This message could occur while the operating system is being halted or while breaking from the operating system to go to the OpenBoot PROM prompt. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists, the board might need to be returned to SunService.

```
WARNING: Could not get current execution state!
```

Cause: This message could occur while the operating system is being halted or while breaking from the operating system to go to the OpenBoot PROM prompt. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists, the board might need to be returned to SunService.

```
WARNING: Could not set previous execution state!
```

Cause: This message could occur while the operating system is being halted or while breaking from the operating system to go to the OpenBoot PROM prompt. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists, the board might need to be returned to SunService.

```
WARNING: Could not set state break!
```

Cause: This message could occur while the operating system is being halted or while breaking from the operating system to go to the OpenBoot PROM prompt. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists, the board might need to be returned to SunService.

```
WARNING: Could not set state offline!
```

Cause: This message could occur while the operating system is being halted or while breaking from the operating system to go to the OpenBoot PROM prompt. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists, the board might need to be returned to SunService.

```
WARNING: Could not set state online!
```

Cause: This message could occur while the operating system is being halted or while breaking from the operating system to go to the OpenBoot PROM prompt. The node board could be in an unknown state or could have a hardware problem.

Action: Hot-swap the node board. If the problem still exists, the board might need to be returned to SunService.

```
ysif_xfer_msg: kcs driver xfermsg returns -1
read_evt_buffer: sysif_xfer_msg returns -1
poll_evt_handler: read_evt_buffer returns -1
listner_thread: poll_evt_handler returns -1
```

Cause: The distributed management card BMC firmware might generate this message during a flash update of the distributed management card or during normal operation. It means that the BMC firmware is unable to respond to communication requests.

Action: If the message occurs during a flash update, the message can be safely ignored. After the flash update is complete, reset the distributed management card and the message will not be repeated. If the message occurs during normal operation, the distributed management card fails over and clears the fault by resetting the BMC.

Index