



Netra™ CT Server System Administration Guide

For the Netra CT 810 Server and Netra CT 410 Server

Sun Microsystems, Inc.
www.sun.com

Part No. 816-2483-12
April 2004, Revision A

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

Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, Netra, ChorusOS, OpenBoot, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and in other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, Californie 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuels relatants à la technologie qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuels peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y ena.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, Netra, ChorusOS, OpenBoot, Java, et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits protant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciées de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Please
Recycle



Adobe PostScript

Contents

Preface xi

1. Introduction 1

Overview of Netra CT Server Software 1

System Administration Tasks 3

2. Configuring Your System 5

Accessing the Alarm Card 6

Configuring the Alarm Card Serial Ports 6

Configuring the Alarm Card Ethernet Ports 8

Setting Up User Accounts on the Alarm Card 10

 Username Restrictions 10

 Password Restrictions 11

Specifying Netra CT Server FRU ID Information 11

Specifying the Netra CT Server Functional Configuration 14

Configuring a Chassis Slot for a Board 15

Configuring the MCNet Interface 17

 Choosing the IP Address for the MCNet 17

 Checking the MCNet Configuration for the Solaris Environment 19

 Checking the MCNet Configuration on the Alarm Card 20

Specifying Other FRU ID Information	20
Displaying Netra CT Server FRU ID Information	21
Configuring the CPU Boards	25
Enabling the Managed Object Hierarchy Application	25
Software Required	25
Starting the MOH Application	26
MOH Configuration and SNMP	27
MOH Configuration and RMI	31
Enabling the Processor Management Service Application	32
Stopping and Restarting the PMS Daemon on the Alarm Card	34
Setting the IP Address for the Alarm Card to Control CPU Boards in the Same System	36
Adding Address Information for a Local CPU Board to Control CPU Boards in Local or Remote Systems	37
3. Administering Your System	39
Using the Alarm Card Command-Line Interface	40
CLI Commands	40
Security Provided	47
Updating the Alarm Card Flash Images	48
Setting the Date and Time on the Alarm Card	50
Running Scripts on the Alarm Card	51
Using Scripting	51
Scripting Limitations	51
Viewing Alarm Card Logs	52
Console Logs	53
Event Logs	53
Booting CPU Boards	54
Boot Device Variables	54
Booting with a DHCP Server	55

Connecting to CPU Board Consoles from the Alarm Card	57
Configuring Your System for Multiple Console Use	57
Establishing Console Sessions Between the Alarm Card and CPU Boards	58
Using the PMS Application for Recovery and Control of CPU Boards	63
Recovery Configuration of a CPU Board From the Alarm Card	63
Detailed Recovery of a Board in Case of Fault	64
Monitoring and Controlling a CPU Board's Resources From the Alarm Card	66
Using the Netra High Availability Suite With the Netra CT Server Applications	68
Monitoring Your System	69
Command-line Interface Information	69
LED Information	69
The MOH Application	75
Additional Troubleshooting Information	75
Hot Swap on the Netra CT Server	76
How High Availability Hot Swap Works	77
Hot Swap With Boards That Don't Support Full Hot Swap	77
Index	81

Figures

- [FIGURE 1-1](#) Logical Representation of Software and the Hardware Interfaces in a Netra CT Server 3
- [FIGURE 3-1](#) System Status Panel (Netra CT 810 Server) 69
- [FIGURE 3-2](#) System Status Panel (Netra CT 410 Server) 70
- [FIGURE 3-3](#) Power and Okay to Remove LEDs 71
- [FIGURE 3-4](#) Power and Fault LEDs 71

Tables

TABLE 1-1	Netra CT Server Software for System Administrators	1
TABLE 2-1	FRU ID Information Specified Using the <code>setfru</code> Command	12
TABLE 2-2	Netra CT Server Functional Configurations	14
TABLE 2-3	FRU ID Information Displayed Using the <code>showfru</code> Command	22
TABLE 2-4	Solaris Packages for the MOH Application	25
TABLE 2-5	<code>ctmgx</code> Options	27
TABLE 2-6	<code>pmsd slotrndaddressadd</code> Parameters	37
TABLE 3-1	Alarm Card Command-Line Interface Commands	40
TABLE 3-2	Alarm Card Flash Options	48
TABLE 3-3	Alarm Card CLI Console-Related Commands	59
TABLE 3-4	CPU Board Console-Related Escape Character Sequences	59
TABLE 3-5	System Status Panel LEDs for the Netra CT 810 Server	70
TABLE 3-6	System Status Panel LEDs for the Netra CT 410 Server	71
TABLE 3-7	CompactPCI Board LED States and Meanings	72
TABLE 3-8	Meanings of Power and Okay to Remove LEDs	73
TABLE 3-9	Meanings of Power and Fault LEDs	74
TABLE 3-10	Netra CT System Hot-Swap Modes	76

Preface

The *Netra CT Server System Administration Guide* contains configuration and administration information for system administrators of the Netra™ CT 810 and 410 servers. 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.

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 environment documentation, which is at <http://docs.sun.com>

Typographic Conventions

Typeface*	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. % You have mail.
AaBbCc123	What you type, when contrasted with on-screen computer output	% su Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this. To delete a file, type <code>rm filename</code> .

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

Shell Prompts

Shell	Prompt
C shell	<i>machine-name%</i>
C shell superuser	<i>machine-name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Related Documentation

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

Title	Part Number
<i>Netra CT Server Start Here</i>	816-2479
<i>Netra CT Server Product Overview</i>	816-2480
<i>Netra CT Server Installation Guide</i>	816-2481
<i>Netra CT Server Service Manual</i>	816-2482
<i>Netra CT Server System Administration Guide</i>	816-2483
<i>Netra CT Server Safety and Compliance Manual</i>	816-2484
<i>Netra CT Server Software Developer's Guide</i>	816-2486
<i>Netra CT Server Product Note</i>	816-2488

You might want to refer to documentation on the following products for additional information: the Solaris™ operating environment, the ChorusOS™ environment, OpenBoot™ PROM firmware, the Netra High Availability (HA) Suite, and the Netra CP2140 CompactPCI 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 Server System Administration Guide, part number 816-2483-12

Introduction

This chapter includes the following sections:

- [Overview of Netra CT Server Software](#)
- [System Administration Tasks](#)



Overview of Netra CT Server Software

The Netra CT server software can be categorized as follows:

- Operating environments 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
<i>Operating Environments and Applications</i>	Solaris operating environment	The Solaris operating environment runs on the host and satellite CPU boards. It is installed by the user.
	ChorusOS operating environment	The ChorusOS operating environment runs on the alarm card. It manages the Netra CT server, that is, most components connected to the midplane. It is factory-installed.
	Command-line Interface (CLI)	The CLI is the primary user interface to the alarm card.

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, such as the Netra High Availability (HA) Suite.
<i>Firmware</i>	OpenBoot PROM firmware	Firmware on the host and satellite CPU boards that controls booting. It includes diagnostics.
	Boot control firmware (BCF)	Firmware on the alarm card that performs power-on self-test (POST) and controls booting of the alarm card software.
	Baseboard Management Controller (BMC) firmware	Baseboard Management Controller firmware enables communication over the Intelligent Platform Management Interface (IPMI) controller on the alarm card.
	System Management Controller (SMC) firmware	System Management Controller firmware enables communication over the IPMI controller on CPU boards.
<i>Network Support</i>	MCNet	MCNet is a communication channel over the cPCI midplane. It can be used to communicate between the alarm card, the host CPU board, and any MCNet-capable satellite CPU boards for exchanging system management information.

Note – The Netra High Availability (HA) Suite may be used to provide enhanced services for customer high-availability applications. It is required to use certain monitoring capabilities of the MOH application, such as monitoring `nfs` and `tftp` daemons. When installed, it runs on the host and satellite CPU boards. The Netra HA Suite is ordered and shipped separately from the Netra CT server.

In the Netra CT server, the alarm card manages most of the components connected to the midplane. The host CPU board accepts and owns peripherals, such as I/O boards or disks; it runs user applications and distributes tasks within a system. In a Netra CT server, each CPU board (including satellite and host CPU boards) runs its own copy of the Solaris operating environment, and each is therefore considered a server; the alarm card, plus the CPU boards and the other system FRUs, make up a system. There can be several systems in one chassis.

The hardware interfaces include the Intelligent Platform Management Interface (IPMI), the CompactPCI (cPCI) bus, and the PCI interface (PCI i/f) on the alarm card, host CPU boards, and satellite CPU boards.

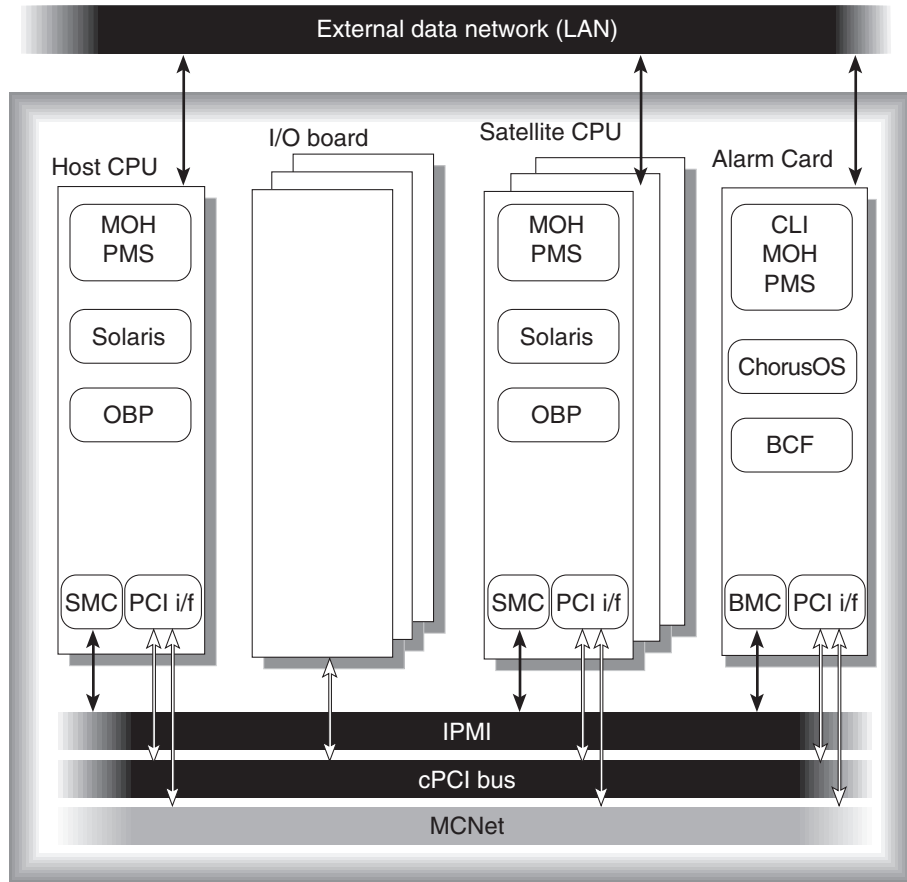


FIGURE 1-1 Logical Representation of Software and the Hardware Interfaces in a Netra CT Server

System Administration Tasks

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

Solaris administration on the Netra CT server, including adding Solaris user accounts, is performed by logging into the host or satellite CPU board. Netra CT server administration is performed by logging into the alarm card and using the alarm card command-line interface. The alarm 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.

Configuring Your System

This chapter assumes you have already installed the Solaris operating environment and the required patches on your Netra CT system.

You configure the Netra CT system primarily through the alarm card command-line interface (CLI). The alarm card CLI enables system-level configuration, administration, and management that includes the CPU nodes, I/O boards, the alarm card, power supplies, and fan trays. The alarm card CLI interface can be used both locally and remotely.

You configure the alarm card first, then the CPU boards, then the system-wide applications.

This chapter includes the following sections:

- [Accessing the Alarm Card](#)
- [Configuring the Alarm Card Serial Ports](#)
- [Configuring the Alarm Card Ethernet Ports](#)
- [Setting Up User Accounts on the Alarm Card](#)
- [Specifying Netra CT Server FRU ID Information](#)
- [Specifying the Netra CT Server Functional Configuration](#)
- [Configuring a Chassis Slot for a Board](#)
- [Configuring the MCNet Interface](#)
- [Specifying Other FRU ID Information](#)
- [Displaying Netra CT Server FRU ID Information](#)
- [Configuring the CPU Boards](#)
- [Enabling the Managed Object Hierarchy Application](#)
- [Enabling the Processor Management Service Application](#)

Accessing the Alarm Card

When you initially access the alarm card, you must do so over serial port COM1 (console), using an ASCII terminal or the `tip` program.

When you first access the alarm 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 server is operational.

The next sections provide information on configuring the alarm card serial and Ethernet ports, and setting up user accounts and passwords using the alarm card command-line interface. For more information on using the alarm card command-line interface, refer to [Chapter 3](#).

After you configure the serial and Ethernet ports, you can access and configure the alarm card over:

- The Ethernet port, using `telnet`
- The serial port (console), using an ASCII terminal or the `tip` program.

If you have a rear-access Netra CT server, to use the console, a cable should be connected to the rear serial port on the alarm card (the front ports are disabled on a rear-access system).

Configuring the Alarm Card Serial Ports

The alarm card has two serial ports, COM1 and COM2. COM1 is configured for the console; you can not change this port. You can configure COM2 using the following CLI commands:

- `setserialmode`
- `setserialbaud`
- `setserialparity`
- `setserialstop`
- `setserialhwhandshake`

You must be logged in to the alarm card with a user account that has full permissions.

When you specify the port number (*port_num*), use 2 to reference serial port COM2.

▼ To Configure the Alarm Card Serial Ports

1. Log in to the alarm card.

2. Set the serial mode:

```
hostname cli> setserialmode -b port_num tty|none
```

Set the mode of the specified serial port to `tty` or `none`. The default for COM2 is `none`, that is, no services are available on this port.

3. Set the serial baud rate:

```
hostname cli> setserialbaud -b port_num baudrate
```

Valid values for the baud rate are 1200, 4800, 9600, 19200, 38400, and 56000. The default is 9600.

4. Set the serial parity:

```
hostname cli> setserialparity -b port_num none|odd|even
```

Valid values for the parity bit are `none`, `odd`, or `even`. The default is `odd`.

5. Set the serial stop bit:

```
hostname cli> setserialstop -b port_num none|odd|even
```

Valid values for the stop bit are 1 or 2. The default is 1.

6. Set the serial data bit number:

```
hostname cli> setserialdata -b port_num 7|8
```

Valid values for the number of data bits are 7 or 8. The default is 8.

7. Set the serial hardware handshake:

```
hostname cli> setserialhwhandshake -b port_num true|false
```

Valid values for the hardware handshake are `true` or `false`. The default is `false`.

Configuring the Alarm Card Ethernet Ports

The alarm card has two Ethernet ports, ENET1 and ENET2. You configure these ports using the following CLI commands:

- `setipmode`
- `setipaddr`
- `setipnetmask`
- `setipgateway`

You must be logged in to the alarm card with a user account that has full permissions.

When you specify the port number (*port_num*), use 1 or 2, depending on which port you are referencing.

Any one of the Ethernet ports can be configured for failover to the other port. Refer to “Set the IP mode,” below, for instructions.

You must reset the alarm card for any changes to take effect.

▼ To Configure the Alarm Card Ethernet Ports

1. **Log in to the alarm card.**
2. **Set the IP mode:**

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

Choose the IP mode according to the services available in the network (*rarp* or *config*) or to configure the port for failover (*standby*). The default for ENET1 is *rarp*; the default for ENET2 is *none*, that is, no services are available on this port. You must reset the alarm card for the changes to take effect.

Any one of the Ethernet ports can be configured for failover. To do this, set the IP mode to `standby` on one port, and set the IP mode to `rarp` or `config` on the other port. If the port configured for `rarp` or `config` fails, the network traffic will be switched over to the port configured for `standby`. For example:

```
hostname cli> setipmode -b 1 rarp  
hostname cli> setipmode -b 2 standby  
hostname cli> reset ac
```

In this example, ENET2 is set to `standby`. If ENET1 fails, all network traffic is switched over to ENET2.

3. Set the IP address:

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

The default is 0.0.0.0. This command is only used if the `ipmode` is set to `config`. You must reset the alarm card for the changes to take effect.

4. Set the IP netmask:

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

The default is 0.0.0.0. This command is only used if the `ipmode` is set to `config`. You must reset the alarm card for the changes to take effect.

5. Set the IP gateway:

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

Set the IP gateway of Ethernet port 1. The default is 0.0.0.0. You must reset the alarm card for the changes to take effect.

6. Reset the alarm card.

Setting Up User Accounts on the Alarm Card

User accounts are set up using the alarm card command-line interface. 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 server is operational.

The alarm card supports 16 accounts with passwords.

▼ To Set Up a User Account

1. Log in to the alarm 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 40](#) 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.
- 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:

- Host CPU board
- Alarm card
- System controller board (SCB)
- Power supply 1
- Power supply 2 (only on the Netra CT 810 server)
- Fan tray 1
- Fan tray 2
- Satellite CPU boards
- I/O boards
- Midplane

All FRUs contain *FRU ID* (identification) information that includes FRU manufacturing and configuration data. This information can be displayed through the alarm card CLI (see [TABLE 2-3](#)).

In addition, you enter certain FRU ID information, through the alarm card CLI, that 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:

- The functional configuration of the system (there is no default)
- Allowable plug-in boards (a default exists) and boot devices (a default exists in OpenBoot PROM) for the cPCI slots
- The MCNet 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 MCNet interface.

The format of the information to be specified is:

```
hostname cli> setfru fru_target fru_instance fru_field value
```

FRU ID information can be displayed using the CLI `showfru` command; see [“Displaying Netra CT Server FRU ID Information” on page 21](#) for more information.

[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 Target	FRU Instance	FRU Field	Value	Description
midplane	1	Drawer_Cfg	1 or 2	Set the Netra CT functional configuration to 1 (satellite only) or 2 (hosted or mixed)
midplane	1	MCNetIPSubnet	<i>IP subnet address (hexadecimal)</i>	Specify the IP subnet address for the MCNet. The default is 0xc0a80d (192.168.13).
midplane	1	MCNetIPSubnetMask	<i>IP subnet mask (hexadecimal)</i>	Specify the IP subnet mask for the MCNet. The default is 0xffffffff00 (255.255.255.0).
midplane	1	Location	<i>text description</i>	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.

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

FRU Target	FRU Instance	FRU Field	Value	Description
midplane	1	Cust_Data	<i>text description</i>	Any customer-supplied information. The text can be up to 80 characters in length.
midplane	1	User_Label	<i>text description</i>	Any customer-supplied information. The text can be up to 10 characters in length.
slot	1 to 7	Acceptable_Fru_Types	<i>vendor:partnumber</i>	First, specify the chassis slot number to be configured, where the FRU Instance can be 1, 2, 3, 4, 5, 6, or 7 for a Netra CT 810 and 2, 3, 4, or 5 for a Netra CT 410. (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 <code>showfru</code> command to display this information. Multiple boards may be specified, separated by a semi-colon (;). The default is to allow all cPCI boards.
slot	1 to 7	Boot_Devices	<i>boot_device_list</i>	First, specify the chassis slot number to be configured, where the FRU Instance can be 1, 2, 3, 4, 5, 6, or 7 for a Netra CT 810 and 2, 3, 4, or 5 for a Netra CT 410. (Slots are numbered starting from the left.) Second, specify the alias(es) listing the devices the board in this slot will boot from. When the board in this slot is powered up, this FRU information overwrites the entry in the OpenBoot PROM <code>boot-device</code> NVRAM configuration variable.
slot	1 to 7	Cust_Data	<i>text description</i>	First, specify the chassis slot number to be configured, where the FRU Instance can be 1, 2, 3, 4, 5, 6, or 7 for a Netra CT 810 and 2, 3, 4, or 5 for a Netra CT 410. (Slots are numbered starting from the left.) Second, specify any customer-supplied information. The text can be up to 80 characters in length.

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

FRU Target	FRU Instance	FRU Field	Value	Description
fan	1 or 2	Cust_Data	<i>text description</i>	Any customer-supplied information. The text can be up to 80 characters in length.
ps	1 or 2	Cust_Data	<i>text description</i>	Any customer-supplied information. The text can be up to 80 characters in length.
scb	1	Cust_Data	<i>text description</i>	Any customer-supplied information. The text can be up to 80 characters in length.

Changes to FRU 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.

Note – You must have the host CPU board, the alarm card, and the system controller board installed in the Netra CT server before powering it on. The system will not power on properly if all of these components are not installed.

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

Specifying the Netra CT Server Functional Configuration

Netra CT server base configurations of front or rear access with diskful or diskless access are set at the factory. Each of these base configurations supports any one of the functional configurations shown in [TABLE 2-2](#).

TABLE 2-2 Netra CT Server Functional Configurations

Configuration	Description
Hosted	Host CPU board and I/O boards
Satellite	Host CPU board and satellite CPU boards
Mixed	Host CPU board, satellite CPU boards, and I/O boards

There is no default functional configuration on the Netra CT server; you set the functional configuration using the alarm card CLI. The functional configuration information is used by the MOH application to audit board insertions and prevent misconfigurations. The functional configuration can be changed at any time if desired using the alarm card CLI.

▼ To Specify the Netra CT Server Functional Configuration

1. Log in to the alarm card.
2. Set the functional configuration for the Netra CT server:

```
hostname cli> setfru fru_target fru_instance fru_field value
```

Refer to [TABLE 2-1](#) for allowable information for each variable. For example, if you want to set the functional configuration to “hosted,” enter the following:

```
hostname cli> setfru midplane 1 Drawer_Cfg 2
```

3. Completely power off and on the system:
 - a. Press the system power button on the system status panel and release it to go through a graceful soft power-down; wait for the system power LED to go off.
 - b. Push the locking mechanism on the power supplies up (unlocked) to power down; wait for the green LEDs on the power supplies to go off; then push the locking mechanism on the power supplies down (locked) to power up. Note: on the Netra CT 810 server, push the locking mechanism on *both* power supplies up and then down at the same time.
 - c. Press the system power button on the system status panel and release it to power on the server.

Configuring a Chassis Slot for a Board

You can specify the type of board that is allowed in a given chassis slot using the alarm card CLI. The slot usage information is used by the MOH application 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 will boot 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 if desired using the alarm card CLI.

By default, slots are configured to accept any cPCI FRU unless you specifically set an allowable plug-in for a specific slot. The exceptions are: for a Netra CT 810 server, the alarm card must be in slot 8 and the host CPU must be in slot 1; for a Netra CT 410 server, the alarm card must be in slot 1 and the host CPU must be in slot 3.

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 21](#) for more information.

▼ To Configure a Chassis Slot for a Board

1. Log in to the alarm card.
2. Set the acceptable FRUs for the slot:

```
hostname cli> setfru fru_target fru_instance fru_field 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:595-5*;0004:1234-5678-1;0001:8796541-02
```

3. Set the boot device for the slot:

```
hostname cli> setfru fru_target fru_instance fru_field 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(es) specifying the boot devices (limit is 25 bytes), for example, `disk net`.

4. Completely power off and on the system:

- a. Press the system power button on the system status panel and release it to go through a graceful soft power-down; wait for the system power LED to go off.
- b. Push the locking mechanism on the power supplies up (unlocked) to power down; wait for the green LEDs on the power supplies to go off; then push the locking mechanism on the power supplies down (locked) to power up. Note: on the Netra CT 810 server, push the locking mechanism on *both* power supplies up and then down at the same time.
- c. Press the system power button on the system status panel and release it to power on the server.

Configuring the MCNet Interface

MCNet provides a communication channel over the cPCI midplane. It can be used to communicate between the alarm card, the host CPU board, and satellite CPU boards. It appears as any other generic Ethernet port in the Solaris operating environment. MCNet is configured by default on Solaris (host CPU and satellite CPUs) and on the alarm card. MCNet is used by the MOH and PMS applications.

Choosing the IP Address for the MCNet

The IP address of the MCNet interfaces on the CPU boards is formed as follows: the midplane FRU ID field `MCNetIPSubnet` contains the value `IP_subnet_address.slot_number`. The default IP subnet address is `0xc0a80d` (192.168.13) and the default IP subnet mask is `0xfffffff00` (255.255. 255.0). When you power on

the Netra CT server, and if you have not made any changes for the MCNet interface in the midplane FRU ID, the IP address of a board installed in slot 2 will be configured to 192.168.13.2; if you then move that board to slot 4, the IP address for that board will be configured to 192.168.13.4.

The IP address of the MCNet interface on the alarm card is always the midplane FRU ID field MCNetIPSubnet value *IP_subnet_address*.8. This is the case for the alarm card in the Netra CT 810 server and in the Netra CT 410 server.

▼ To Configure the MCNet Interface

1. Log in to the alarm card.
2. Set the FRU ID for the MCNet interface:

```
hostname cli> setfru fru_target fru_instance fru_field value
```

Refer to [TABLE 2-1](#) for allowable information for each variable. You must set both the MCNet IP subnet address and the subnet mask in hexadecimal format. For example, to set the subnet address to 192.168.16 and the subnet mask to 255.255.255.0, enter the following:

```
hostname cli> setfru midplane 1 MCNetIPSubnet 0xc0a810  
hostname cli> setfru midplane 1 MCNetIPSubnetMask 0xffffffff00
```

3. Completely power off and on the system:
 - a. Press the system power button on the system status panel and release it to go through a graceful soft power-down; wait for the system power LED to go off.
 - b. Push the locking mechanism on the power supplies up (unlocked) to power down; wait for the green LEDs on the power supplies to go off; then push the locking mechanism on the power supplies down (locked) to power up. Note: on the Netra CT 810 server, push the locking mechanism on *both* power supplies up and then down at the same time.
 - c. Press the system power button on the system status panel and release it to power on the server.

Checking the MCNet Configuration for the Solaris Environment

After you boot the Solaris operating environment, you can check to see that MCNet has been configured by using the `ifconfig -a` command. You should see output for the `mcn0` interface similar to the following:

```
# ifconfig -a
...
eri0: flags=10000843<UP, BROADCAST, RUNNING, MULTICAST, IPv4>mtu 1500
index 1
  inet 192.168.207.64 netmask ffffffff broadcast 192.168.207.255
  ether 8:0:20:a9:4d:1d
lo0: flags=1000849<UP, LOOPBACK, RUNNING, MULTICAST, IPv4>mtu 1500
index 2
  inet 127.0.0.1 netmask ff000000
mcn0: flags=10000843<UP, BROADCAST, RUNNING, MULTICAST, IPv4>mtu 1500
index 3
  inet 192.168.16.1 netmask ffffffff broadcast 192.168.16.255
  ether 8:0:20:a9:4d:1d
```

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

```
# ping -s 192.168.16.3
PING 192.168.13.3: 56 data bytes
64 bytes from 192.168.16.3:icmp_seq=0,time=1,ms
64 bytes from 192.168.16.3:icmp_seq=1,time=0,ms
64 bytes from 192.168.16.3:icmp_seq=2,time=0,ms
...
----192.168.16.3 PING statistics----
14 packets transmitted, 14 packets received, 0% packet loss
round-trip (ms) min/avg/max=0/0/1
```

Checking the MCNet Configuration on the Alarm Card

After you configure the MCNet interface, 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
Netrtract network configuration is:

ethernet ports
ip_addr  :192.168.207.130
ip_netmask : 0xffffffff00
mac_address : 00:03:ba:13:c4:dd

ip_addr  :192.168.13.8
ip_netmask : 0xffffffff00
mac_address : 00:03:ba:13:c4:dd
hostname cli>
```

Specifying Other FRU ID Information

You can use the FRU fields `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 in the `Location` field is displayed through the MOH application.

You might want to use the `Location` FRU field 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 alarm card.

2. Specify other FRU ID information for the Netra CT server:

```
hostname cli> setfru fru_target fru_instance fru_field 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:

- a. Press the system power button on the system status panel and release it to go through a graceful soft power-down; wait for the system power LED to go off.
- b. Push the locking mechanism on the power supplies up (unlocked) to power down; wait for the green LEDs on the power supplies to go off; then push the locking mechanism on the power supplies down (locked) to power up. Note: on the Netra CT 810 server, push the locking mechanism on *both* power supplies up and then down at the same time.
- c. Press the system power button on the system status panel and release it to power on the server.

Displaying Netra CT Server FRU ID Information

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

TABLE 2-3 shows the FRU ID information that can be displayed with the CLI `showfru` command. Use the FRU field to specify the information you want.

TABLE 2-3 FRU ID Information Displayed Using the `showfru` Command

FRU Target	FRU Instance	FRU Field	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	Drawer_Cfg	Display the functional configuration (satellite, hosted, or mixed) for this system.
midplane	1	MCNetIPSubnet	Display the MCNet IP subnet address in hexadecimal format for this system.
midplane	1	MCNetIPSubnetMask	Display the MCNet 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	Initial_HW_Dash_Level	Display the initial hardware dash level of the midplane
midplane	1	Initial_HW_Rev_Level	Display the initial hardware revision level of 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.
midplane	1	Cust_Data	Display any customer-supplied text for this field.
slot	1 to 8	Sun_Part_No	Display the part number for the board in a particular slot.
slot	1 to 8	Sun_Serial_No	Display the serial number for the board in a particular slot.
slot	1 to 8	Acceptable_Fru_Types	Display the allowable plug-in boards for a particular slot.
slot	1 to 8	Boot_Devices	Display the boot devices for a particular slot.
slot	1 to 8	Vendor_Name	Display the vendor name for the board in a particular slot.
slot	1 to 8	Fru_Shortname	Display the FRU short name for the board in a particular slot.
slot	1 to 8	Initial_HW_Dash_Level	Display the initial hardware dash level of the board in a particular slot.

TABLE 2-3 FRU ID Information Displayed Using the `showfru` Command (Continued)

FRU Target	FRU Instance	FRU Field	Description
slot	1 to 8	Initial_HW_Rev_Level	Display the initial hardware revision level of the board in a particular slot.
slot	1 to 8	Cust_Data	Display any customer-supplied text for this field for the board in a particular slot.
fan	1 or 2	Sun_Part_No	Display the part number for fan tray 1 or 2.
fan	1 or 2	Sun_Serial_No	Display the serial number for fan tray 1 or 2.
fan	1 or 2	Vendor_Name	Display the vendor name for fan tray 1 or 2.
fan	1 or 2	Fru_Shortname	Display the FRU short name for fan tray 1 or 2.
fan	1 or 2	Initial_HW_Dash_Level	Display the initial hardware dash level of fan tray 1 or 2.
fan	1 or 2	Initial_HW_Rev_Level	Display the initial hardware revision level of fan tray 1 or 2.
fan	1 or 2	Cust_Data	Display any customer-supplied text for this field for fan tray 1 or 2.
ps	1 or 2	Sun_Part_No	Display the part number for power supply unit 1 or 2.
ps	1 or 2	Sun_Serial_No	Display the serial number for power supply unit 1 or 2.
ps	1 or 2	Vendor_Name	Display the vendor name for power supply unit 1 or 2.
ps	1 or 2	Fru_Shortname	Display the FRU short name for power supply unit 1 or 2.
ps	1 or 2	Initial_HW_Dash_Level	Display the initial hardware dash level of power supply unit 1 or 2.
ps	1 or 2	Initial_HW_Rev_Level	Display the initial hardware revision level of power supply unit 1 or 2.
ps	1 or 2	Cust_Data	Display any customer-supplied text for this field for power supply unit 1 or 2.
scb	1	Sun_Part_No	Display the part number for the system controller board.
scb	1	Sun_Serial_No	Display the serial number for the system controller board.
scb	1	Vendor_Name	Display the vendor name for the system controller board.
scb	1	Fru_Shortname	Display the FRU short name for the system controller board.

TABLE 2-3 FRU ID Information Displayed Using the `showfru` Command (Continued)

FRU Target	FRU Instance	FRU Field	Description
scb	1	Initial_HW_Dash_Level	Display the initial hardware dash level of the system controller board.
scb	1	Initial_HW_Rev_Level	Display the initial hardware revision level of the system controller board.
scb	1	Cust_Data	Display any customer-supplied text for this field for the system controller board.

▼ To Display FRU ID Information

1. Log in to the alarm card.
2. Enter the `showfru` command:

```
hostname cli> showfru fru_target fru_instance fru_field
```

Refer to [TABLE 2-3](#) 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 fan 1 Sun_Part_No
```

Use the FRU target “slot” to display information for the alarm card, the CPU boards, and the I/O boards; the FRU slot instance can be 1, 2, 3, 4, 5, 6, 7, or 8 for a Netra CT 810 and 1, 2, 3, 4, or 5 for a Netra CT 410 (slots are numbered starting from the left). For example, to display part number FRU ID information for the alarm card in a Netra CT 810 server, enter the following:

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

Configuring the CPU Boards

You should verify that you can log in to the CPU boards. Any Solaris configuration needed for your environment should be done, such as modifying OpenBoot PROM variables. Refer to the Solaris documentation, the OpenBoot PROM documentation, or to the specific CPU board documentation if you need additional information.

Note that if the alarm card is not present in the system or is in the process of resetting, you can not reboot the host CPU board.

Enabling the Managed Object Hierarchy Application

The Managed Object Hierarchy (MOH) is an application that runs on the alarm card, the host CPU, and satellite CPUs. It monitors the field-replaceable units (FRUs) in your system.

Software Required

The MOH application requires the Solaris 9 operating environment, and additional Netra CT platform-specific Solaris patches that contain packages shown in [TABLE 2-4](#).

TABLE 2-4 Solaris Packages for the MOH Application

Package	Description
SUNW2jdrct	Java™ Runtime Java Dynamic Management Kit (JDMK) package
SUNWctmgx	Netra CT management agent package
SUNWctac	Alarm 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/mgmt2.0/bin`. To verify the packages are installed, use the `pkginfo` command:

```
# pkginfo -l SUNW2jdrct SUNWctmgx SUNWctac
...
PKGINST: SUNW2jdrct
...
```

The Netra High Availability Suite may be used to provide enhanced services for customer high-availability applications. It is required to use certain monitoring capabilities of the MOH application, such as monitoring `nfs` and `tftp` daemons. The Netra HA Suite is ordered and shipped separately from the Netra CT server.

Once the MOH application is running, MOH agents on the alarm card and on CPU 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 alarm card.

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

```
# cd /opt/SUNWnetract/mgmt2.0/bin
# ./ctmgx start [options]
```

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

Options that can be specified with `ctmgx start` when you start the MOH application include:

TABLE 2-5 `ctmgx` Options

Option	Description
<code>-drawerview</code>	This option is required if you want to authenticate RMI application programs. It is valid only on the host CPU board. Specify that the MOH application model (represent) all components in the chassis. If the <code>-drawerview</code> option is not specified, MOH models only the host CPU board and any other cPCI boards in the chassis.
<code>-rmiport portnum</code>	Specify the RMI port number. The default is 1099.
<code>-snmpport portnum</code>	Specify the SNMP port number. The default is 9161.
<code>-snmpacl filename</code>	Specify the SNMP access control list (ACL) file to be used. The full path to <i>filename</i> must be specified.
<code>-showversion</code>	Print the system version number.

By default, SNMP and RMI applications have read-write access to MOH agents on the alarm card and on CPU boards. The next sections describe how to configure MOH to control SNMP and RMI access on the alarm card and CPU boards.

MOH Configuration and SNMP

By default, SNMP applications have read-write access to the Netra CT server MOH agents. If you want to control which applications communicate with the MOH agents, you must configure the alarm card and CPU 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 will not respond to requests originating from IP addresses that do not belong to one of its communities.

Alarm Card SNMP Interface

On the alarm 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 alarm card flash memory.

▼ To Configure the Alarm Card SNMP Interface

1. Log in to the alarm 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 alarm 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 alarm 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>
```

CPU Board SNMP Interface

On CPU boards, ACL information is stored in a configuration file in the Solaris operating environment.

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 will be 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 CPU 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/mgmt2.0/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 will have read-write access and MOH will send the coldStart trap to the local host only.

MOH Configuration and RMI

By default, RMI applications have read-write access to the Netra CT server MOH agents. If you want to control which applications communicate with the MOH agents, you must configure the alarm card and *host* CPU board 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 alarm card and on the host CPU board, the following configuration is needed:

- The RMI application program(s) must contain a valid alarm card user name and password to be authenticated (for information on adding this information to RMI programs, refer to the *Netra CT Server Software Developer's Guide*). This information is communicated between the MOH agent running on the alarm card and the MOH agent running on the host CPU board. 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 depends 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 will be read and write; if the user permissions are set to anything less than *c*, *u*, *a*, and *r*, the RMI permission will be 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 alarm card and host CPU board with read-write access.)
- You must start the MOH application on the host CPU board with the `-drawerview` option.

If MOH security for RMI was enabled but becomes disabled on the alarm card (for example, if the alarm card is being reset or hot swapped), security will be disabled on the host CPU board as well; a security exception occurs and no access is given.

▼ To Configure the Alarm Card RMI Interface

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

```
hostname cli> setmohsecurity true
```

4. Reset the alarm card.

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

▼ To Configure the Host CPU Board RMI Interface

1. Verify the following:
 - a. The RMI programs you want to access the host CPU board MOH agent contain a valid alarm card user name and password, with appropriate permissions.
 - b. The CLI `setmohsecurity` option on the alarm card is set to `true`.
2. Log in to the host CPU board.
3. As root, start the MOH application.

```
# cd /opt/SUNWnetract/mgmt2.0/bin
# ./ctmgx start -drawerview
```

Enabling the Processor Management Service Application

The Processor Management Service (PMS) is a management application that provides support for high-availability services and applications, such as the Netra High Availability Suite. It provides both local and remote monitoring and control of a cluster of CPU boards.

This section describes:

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

You use the alarm card PMS CLI commands to control 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 CPU Boards” on page 63](#). You can also use the PMS API to configure partner lists (tables of alarm card and CPU board information relating to connectivity and addressing; the alarm card and the CPU boards in a partner list must be in the same system). Refer to the `pms` API man pages, installed by default in `/opt/SUNWnetract/mgmt2.0/man`, for more information on partner lists.

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

1. Log in as root to the server that has the Solaris patches installed (see [“Software Required” on page 25](#)).
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/mgmt2.0/bin/pmsd start -e force_avail
    ;;
'stop')
    /opt/SUNWnetract/mgmt2.0/bin/pmsd stop
    ;;
'restart')
    /opt/SUNWnetract/mgmt2.0/bin/pmsd stop
    /opt/SUNWnetract/mgmt2.0/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.

This script starts PMS in the available *state* (`start -e force_avail`).

On CPU boards, PMS's internal timer service uses a default interval of 0.1 seconds as the time *tick interval*. You can adjust the tick interval to a number from 0.1 seconds to 2.0 seconds by using the `-t` option. For example, to start PMS with a tick interval of 1.0 seconds, use the command `pmsd start -e force_avail -t 1`.

Keeping the default interval value may lead to a large number of voluntary context switches in the system. You can check the effect of increasing the `-t` option to various intervals by looking at the output from the `prstat` command; the column labeled VCX contains the number of voluntary context switches received by the operating system from an application. An example of `prstat` output, with the `-t` option set to 1, is:

```
# prstat -v -c -L -p 'pgrep pmsd' 10
  PID USERNAME  USR  SYS  TRP  TFL  DFL  LCK  SLP  LAT  VCX  ICX  SCL  SIG  PROCESS/LWPID
  868 root        0.0  0.0  -    -    -    -    100  -    30   0   20   0   pmsd/6
  868 root        0.0  0.0  -    -    -    -    100  -    10   0   13   0   pmsd/5
  868 root        0.0  0.0  -    -    -    -    100  -     0   0   0    0   pmsd/4
  868 root        0.0  0.0  -    -    -    -    100  -     4   0   12   0   pmsd/3
  868 root        0.0  0.0  -    -    -    -    100  -     6   0   9    0   pmsd/2
  868 root        0.0  0.0  -    -    -    -    100  -     1   0   2    0   pmsd/1
Total: 1 processes, 6 lwps, load averages: 0.02, 0.02, 0.02
...
```

Stopping and Restarting the PMS Daemon on the Alarm Card

The PMS daemon (`pmsd`) starts automatically on the alarm card. However, you can manually stop and restart the PMS daemon on the alarm card, specifying these optional parameters:

- The port number `pmsd` listens on for servicing clients (default is port 10300).
- The *state* `pmsd` will be started in: available or unavailable (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 alarm card (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 alarm 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 Alarm Card

1. Log in to the alarm 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.

▼ To Manually Start the Processor Management Service on the Alarm Card

1. Log in to the alarm 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 Alarm Card to Control CPU Boards in the Same System

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

You specify the slot number of the CPU 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 Alarm Card to Control CPU Boards in the Same System

1. Log in to the alarm card.
2. Set the IP address with the `slotaddressesset` command:

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

where *slot_num* can be a slot number from 1 to 8, and *ip_addr* is the IP address to be configured.

Printing IP Address Information

The `pmsd slotaddresssshow -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 CPU board in this slot using this IP address.

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

You can use the PMS CLI application to enable local CPU boards to remotely monitor and control CPU 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 CPU board, PMS notifies the controlling CPU board of the failure, and the controlling CPU board (through a customer application) notifies another controlled CPU board to start the same high availability application.

The `pmsd slotrndaddressadd` command is used to configure a local CPU board to control and monitor another CPU board by specifying the IP addresses and slot information for the CPU board to be controlled, using the parameters shown in [TABLE 2-6](#).

TABLE 2-6 `pmsd slotrndaddressadd` Parameters

Parameter	Description
<code>-s slot_num all</code>	Specifies the slot number of the CPU that is being configured in the local system to monitor or control other local or remote CPUs
<code>-n ip_addr</code>	Specifies the IP address of the CPU board in the local or remote system to be monitored or controlled by the local CPU
<code>-d ip_addr</code>	Specifies the IP address of the alarm card in the same local or remote system of the CPU board to be monitored or controlled by the local CPU
<code>-r slot_num</code>	Specifies the slot number of the CPU board in the local or remote system to be monitored or controlled by the local CPU

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

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

1. Log in to the alarm 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 CPU board you want to use to control other local or remote CPU boards, and `all` specifies all slots containing CPU boards in the local system; `-n ip_addr` is the IP address of the CPU board to be controlled; `-d ip_addr` is the IP address of the alarm card in the system of the CPU board to be controlled; and `-r slot_num` is the slot number of the CPU 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, described below.

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 CPU board. The `-s slot_num |all` parameter specifies whether the address information will be deleted on a single slot number or on all slots containing CPU 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 will be printed for a single slot number or for all slots containing CPU boards in the local system; `index_num` can be 1 to 16. The `-i index_num |all` parameter specifies whether the address information will be printed for a single address entry or for all address entries.

Administering Your System

You administer your system using the alarm card command-line interface and through the MOH application.

The alarm 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 alarm card itself; sending out events and alerts is not a function of the CLI.

This chapter contains the following sections:

- [Using the Alarm Card Command-Line Interface](#)
- [Updating the Alarm Card Flash Images](#)
- [Setting the Date and Time on the Alarm Card](#)
- [Running Scripts on the Alarm Card](#)
- [Viewing Alarm Card Logs](#)
- [Booting CPU Boards](#)
- [Connecting to CPU Board Consoles from the Alarm Card](#)
- [Using the PMS Application for Recovery and Control of CPU Boards](#)
- [Using the Netra High Availability Suite With the Netra CT Server Applications](#)
- [Monitoring Your System](#)
- [Hot Swap on the Netra CT Server](#)

Using the Alarm Card Command-Line Interface

The alarm card command-line interface provides commands to control power of the system, control the CPU nodes, administer the system, show status, and set configuration variables. (See “[Accessing the Alarm Card](#)” on page 6 for information on how to access the alarm card.)

CLI Commands

TABLE 3-1 lists the alarm card command-line interface commands by type, command name, default permission required to use the command, and command description. A `-h` option with a command indicates that help is available for that command.

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 CPU boards)
- blank (permission not required).

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

TABLE 3-1 Alarm Card Command-Line Interface Commands

Command Type	Command	Permission	Description
Status	<code>showenvironment</code>		Display a summary of current environmental information, such as fan and power supply status.
	<code>shownetwork</code>		Display the current network configuration of the alarm card.
	<code>showserialmode</code> <code>-b port_num</code>		Display the value of <code>serial_mode</code> for the specified port number.
	<code>showserialbaud</code> <code>-b port_num</code>		Display the value of <code>serial_baud</code> for the specified port number.

TABLE 3-1 Alarm Card Command-Line Interface Commands (Continued)

Command Type	Command	Permission	Description
	showserialparity -b <i>port_num</i>		Display the value of <code>serial_parity</code> for the specified port number.
	showserialstop -b <i>port_num</i>		Display the value of <code>serial_stop</code> for the specified port number.
	showserialdata -b <i>port_num</i>		Display the value of <code>serial_data</code> for the specified port number.
	showserialhwhandshake -b <i>port_num</i>		Display the value of <code>serial_hwhandshake</code> for the specified port number.
	showipmode -b <i>port_num</i>		Display the value of <code>ip_mode</code> for the specified port number.
	showipaddr -b <i>port_num</i>		Display the value of <code>ip_addr</code> for the specified port number.
	showipnetmask -b <i>port_num</i>		Display the value of <code>ip_netmask</code> for the specified port number.
	showipgateway -b <i>port_num</i>		Display the value of <code>ip_gateway</code> for the specified port number.
	showdate		Display the system date.
	showntpserver		Display the IP address of the NTP server.
	showfru <i>target instance field</i>		Display FRU ID information. Refer to “Displaying Netra CT Server FRU ID Information” on page 21 for more information.
	showhostname		Display the value of the hostname used in the CLI prompt.
	showservicemode		Display the value of the alarm card flash update service mode.
	showcpustate		Display the board type, power state, and boot state for each CPU board in the system.
	showmohsecurity		Display the value of the alarm card MOH security mode.
Power control	poweroff <i>[cpu_node]</i>	r	Power off the specified CPU node slot, where <i>cpu_node</i> can be 1 to 8 on a Netra CT 810 or 1 to 5 on a Netra CT 410; if no node is specified, power off the whole system.
	poweron <i>[cpu_node]</i>	r	Power on the specified CPU node slot, where <i>cpu_node</i> can be 1 to 8 on a Netra CT 810 or 1 to 5 on a Netra CT 410; if no node is specified, power on the whole system.

TABLE 3-1 Alarm Card Command-Line Interface Commands (Continued)

Command Type	Command	Permission	Description
	<code>powersupply n on off</code>	r	Switch on or off the specified power supply unit.
CPU control	<code>console <i>cpu_node</i></code>	c	Enter console mode and connect to the specified CPU node, where <i>cpu_node</i> can be 1 to 8 on a Netra CT 810 or 1 to 5 on a Netra CT 410.
	<code>consolehistory [run orun] [index [+ -] n] [pause n]</code>	c	Display the contents of the alarm card console run log or orun log.
	<code>consolerestart</code>	a	Copy the alarm card console run log (run buffer) into the old log (orun buffer), overwriting the previous contents; then clear the run buffer.
	<code>break <i>cpu_node</i></code>	c	Put the server in debug mode, where <i>cpu_node</i> can be 1 to 8 on a Netra CT 810 or 1 to 5 on a Netra CT 410.
	<code>reset [-h] [<i>cpu_node</i>] [-x <i>cpu_node</i> ac host]</code>	r	Reset (reboot) a specified server, where <i>cpu_node</i> can be 1 to 8 on a Netra CT 810 or 1 to 5 on a Netra CT 410; ac is the alarm card; host is the host CPU board. <code>reset <i>cpu_node</i></code> produces a soft reset (reboots the operating system); <code>reset -x</code> produces a hard reset (reboots the board).
	<code>setpanicdump <i>cpu_node</i> [true false]</code>	a	Set whether a panic dump is generated when a CPU node is reset.
	<code>showpanicdump <i>cpu_node</i></code>		Show whether or not a panic dump has been set for a specific CPU node.
	<code>setescapechar <i>value</i></code>	a	Set the escape character to end a console session. The default is a ~ (tilde).
	<code>showhealth [-b <i>cpu_node</i>]</code>		Show the healthy information of a CPU node, where <i>cpu_node</i> can be 1 to 8 on a Netra CT 810 or 1 to 5 on a Netra CT 410.
	<code>pmsd help</code>	u	Display help information on starting, stopping, and controlling the PMS daemon on the alarm card. Refer to “Enabling the Processor Management Service Application” on page 32 and to “Using the PMS Application for Recovery and Control of CPU Boards” on page 63 for more information.
Administration	<code>useradd [-h] <i>username</i></code>	u	Add a user account. The default user account is <code>netract</code> . The alarm card supports 16 accounts.
	<code>userdel [-h] <i>username</i></code>	u	Delete a user account.
	<code>usershow [-h] [<i>username</i>]</code>	u	Show user accounts.

TABLE 3-1 Alarm Card Command-Line Interface Commands (Continued)

Command Type	Command	Permission	Description
	<code>userpassword [-h] <i>username</i></code>	u	Set or change the password of a specified user account.
	<code>userperm [-h] <i>username</i> [c u a r]</code>	u	Set or change the permission levels for a specified user account.
	<code>logout</code>		Log out of the current session.
	<code>password [-h]</code>	u	Change the existing password.
	<code>flashupdate -d <i>cmsw bcfw bmcfw rpdf scdf -f path</i></code>	a	Flash update the alarm card software, where <i>cmsw</i> represents the chassis management software; <i>bcfw</i> represents the boot control firmware; <i>bmcfw</i> represents the BMC firmware; <i>rpdf</i> represents the system configuration repository; and <i>scdf</i> initializes the system configuration variables to their defaults. Refer to “Updating the Alarm Card Flash Images” on page 48 for more information.
	<code>help</code>		Display a list of supported commands.
	<code>version</code>	u	Display the versions of various software and firmware.
	<code>setdate [-h] <i>mmddHHMMccyy</i></code>	a	Set the current date.
	<code>setsecondaryboot [-h] <i>rarp</i></code>	a	The primary boot device for the alarm card is always the flash. In case of flash failure, the secondary boot device is used. The default is <i>rarp</i> .
	<code>showsecondaryboot</code>		Display the secondary boot mode.
	<code>setntpserver <i>addr none</i></code>	a	Configure the alarm card to be an NTP client. The NTP server IP address must be on the same subnet as the alarm card. The default is <i>none</i> .
	<code>setfru [-h] <i>target instance field value</i></code>	a	Set FRU ID information. Refer to “Specifying Netra CT Server FRU ID Information” on page 11 for more information.
	<code>showescapechar</code>	a	Show the escape character used to end a console session.
	<code>setrecovery <i>pmsd moh true false</i></code>	a	Set whether the alarm card will reset itself if the PMS daemon and/or the MOH application exit. The default is <i>false</i> , that is, the alarm card will not reset itself.
	<code>showrecovery</code>	a	Show the value of the <i>setrecovery</i> action the alarm card takes if the PMS daemon and/or the MOH application exit.

TABLE 3-1 Alarm Card Command-Line Interface Commands (Continued)

Command Type	Command	Permission	Description
	loghistory [index [+ -]n] [pause n]	a	Display the contents of the alarm card event log
	snmpconfig add del show access trap <i>community</i> [readonly readwrite] [<i>ip_addr</i>]	a	Configure the alarm card SNMP interface for the MOH application. The default is <code>readonly</code> . Refer to “MOH Configuration and SNMP” on page 27 for more information.
	setmohsecurity true false	a	Configure the alarm card RMI interface for the MOH application. The default is <code>false</code> . Refer to “MOH Configuration and RMI” on page 31 for more information.
Configuration (serial ports)	setserialmode -b <i>port_num</i> tty none	a	Set the mode of the specified serial port to <code>tty</code> or <code>none</code> . The default for COM2 is <code>none</code> , that is, no services are available on this port.
	setserialbaud -b <i>port_num</i> <i>baudrate</i>	a	Set the baud rate of the specified serial port. The default is 9600. Valid values are: 1200, 4800, 9600, 19200, 38400, 56000.
	setserialparity -b <i>port_num</i> none odd even	a	Set the parity bit of the specified serial port. Valid values are <code>none</code> , <code>odd</code> , or <code>even</code> . The default is <code>odd</code> .
	setserialstop -b <i>port_num</i> 1 2	a	Set the stop bit of the specified serial port. Valid values are 1 or 2. The default is 1.
	setserialdata -b <i>port_num</i> 7 8	a	Set the number of data bits of the specified serial port. Valid values are 7 or 8. The default is 7.
	setserialhwhandshake -b <i>port_num</i> true false	a	Set the hardware handshake of the specified serial port. Valid values are <code>true</code> or <code>false</code> . The default is <code>false</code> .
Configuration (Ethernet ports)	setipmode -b <i>port_num</i> rarp config standby none	a	Set the IP mode of the specified Ethernet port. Choose the IP mode according to the services available in the network (<code>rarp</code> , <code>config</code>) or to configure the port for failover (<code>standby</code>). The default for ENET1 is <code>rarp</code> , the default for ENET2 is <code>none</code> , that is, no services are available on this port. You must reset the server for the changes to take effect.
	setipaddr -b <i>port_num</i> <i>addr</i>	a	Set the IP address of the specified Ethernet port. The default is 0.0.0.0. This command is only used if the <code>ipmode</code> is set to <code>config</code> . You must reset the server for the changes to take effect.

TABLE 3-1 Alarm Card Command-Line Interface Commands (Continued)

Command Type	Command	Permission	Description
	setipnetmask -b <i>port_num mask</i>	a	Set the IP netmask of the specified Ethernet port. The default is 0.0.0.0. This command is only used if the <i>ipmode</i> is set to <i>config</i> . You must reset the server for the changes to take effect.
	setipgateway -b <i>port_num addr</i>	a	Set the IP gateway of Ethernet port 1. The default is 0.0.0.0. You must reset the server for the changes to take effect.
Configuration (Other)	sethostname <i>hostname</i>	a	Set the hostname to be used in the CLI prompt. The default is <i>netract</i> . The maximum length is 32 characters.
	setservicemode true false	a	When the <i>servicemode</i> is set to <i>true</i> , MOH and PMS services are stopped for the alarm card flash update. Refer to “Updating the Alarm Card Flash Images” on page 48 for more information.
PMS daemon control	pmsd start [-p <i>port_num</i>] [-e <i>server_admin_state</i>] [-t <i>tick_interval</i>][-d]	a	Start PMS on the alarm card or a CPU board. The -t option can only be used on a CPU board.
	pmsd stop [-p <i>port_num</i>]	a	Stop PMS on the alarm card or a CPU board.
	pmsd slotaddressset -s <i>slot_num -i ip_addr</i>	a	Set the IP address for the alarm card to control and monitor a CPU board.
	pmsd slotaddressshow -s <i>slot_num</i> all	a	Print the IP address set with the <i>pmsd slotaddressset</i> command.
	pmsd slotrndaddressadd -s <i>slot_num</i> all -n <i>ip_addr</i> -d <i>ip_addr -r slot_num</i>	a	Add address information for a CPU board to control other CPU boards.
	pmsd slotrndaddressdelete -s <i>slot_num</i> all -i <i>index_num</i> all	a	Delete address information added with the <i>pmsd slotrndaddressadd</i> command.
	pmsd slotrndaddressshow -s <i>slot_num</i> all -i <i>index_num</i> all	a	Print address information added with the <i>pmsd slotrndaddressadd</i> command.
	pmsd operset -s <i>slot_num</i> all -o <i>maint_config</i> <i>oper_config</i> <i>none_config</i> <i>graceful_reboot</i>	a	Enable automatic recovery of a CPU board.

TABLE 3-1 Alarm Card Command-Line Interface Commands (Continued)

Command Type	Command	Permission	Description
	<code>pmsd infoshow -s slot_num all</code>	a	Print PMS system information.
	<code>pmsd historyshow -s slot_num all</code>	a	Print a log of PMS system events and time stamps.
	<code>pmsd recoveryoperset -s slot_num all -o pc rst rstpc pd rb</code>	a	Manually recover a board in case of fault.
	<code>pmsd recoveryautooperset -s slot_num all -o pc rst rstpc pd rb rbp none trg [-d startup_delay] [-f failure power on off] [-r retries] [-n inter-operation delay] [-p reset power cycle delay]</code>	a	Automatically recover a board in case of fault.
	<code>pmsd recoveryautoinfoshow -s slot_num all</code>	a	Print the configuration information affected by the <code>recoveryautooperset</code> command.
	<code>pmsd hwoperset -s slot_num all -o powerdown powerup reset mon_enable mon_disable [-f]</code>	a	Perform operations on a CPU board hardware.
	<code>pmsd hwinfoshow -s slot_num all</code>	a	Print PMS system information on the hardware.
	<code>pmsd hwhistoryshow -s slot_num all</code>	a	Print a log of PMS hardware events and time stamps.
	<code>pmsd osoperset -s slot_num all -o reboot mon_enable mon_disable [-f]</code>	a	Perform operations on a CPU board operating system.
	<code>pmsd osinfoshow -s slot_num all</code>	a	Print PMS system information on the operating system.
	<code>pmsd oshistoryshow -s slot_num all</code>	a	Print a log of PMS operating system events and time stamps.

TABLE 3-1 Alarm Card Command-Line Interface Commands *(Continued)*

Command Type	Command	Permission	Description
	<code>pmsd appoperset -s slot_num all -o force_offline vote_active force_active</code>	a	Perform operations on a CPU board applications.
	<code>pmsd appinfo show -s slot_num all</code>	a	Print PMS system information on the applications.
	<code>pmsd apphistory show -s slot_num all</code>	a	Print a log of PMS application events and time stamps.
	<code>pmsd version</code>	a	Print the PMS version.
	<code>pmsd usage</code>	a	Print a synopsis of the <code>pmsd</code> commands.

Information on configuring alarm card ports, setting up user accounts, specifying FRU ID information, and starting the PMS daemon using the alarm card CLI is provided in [Chapter 2](#). The PMS daemon commands are described in [“Using the PMS Application for Recovery and Control of CPU Boards”](#) on page 63.

Security Provided

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 Alarm Card Flash Images

You can update the alarm card flash images over the network. [TABLE 3-2](#) shows the alarm card flash options.

TABLE 3-2 Alarm Card Flash Options

Option	Description
<code>cmsw</code>	Updates the chassis management software, which includes the Chorus operating system, the MOH application, and the PMS application.
<code>bcfw</code>	Updates the boot control firmware.
<code>bmcfw</code>	Updates the BMC firmware.
<code>rpdf</code>	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 alarm card.
<code>sddf</code>	(Optional) Initializes the system configuration variables, for example, the serial port variables, to the defaults.

There is no required sequence for flashing the alarm card, although the following order is recommended: `cmsw`, `bcfw`, `bmcfw`, and `rpdf`. You can update individual images if you want.

▼ To Update All the Alarm Card Flash Images

1. Log in to the alarm card.
2. Set the `servicemode` to true by entering the following command:

```
hostname cli> setservicemode true
```

Setting the `servicemode` to true allows the alarm card to be flash updated; it also stops the MOH and PMS services on the alarm card.

Note – In [Step 3](#), the `sddf` option is not mandatory. Use it only if you want to initialize the system configuration variables to the defaults.

3. Flash update all the alarm card images, and complete the process by entering the following commands:

```
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> setservicemode false
hostname cli> flashupdate -d rpdf -f path
```

where *path* is `nfs://nfs.server.ip.address/directory/filename` where the software to use in the flash is installed.

After you update `rpdf`, the alarm card resets itself. If you do not update `rpdf`, you must reset the alarm card manually.

▼ To Update an Individual Alarm Card Flash Image

1. Log in to the alarm card.
2. Set the servicemode to true by entering the following command:

```
hostname cli> setservicemode true
```

Setting the `servicemode` to true allows the alarm card to be flash updated; it also stops the MOH and PMS services on the alarm card.

3. Flash update an alarm card image, and complete the process by entering the following commands:

```
hostname cli> flashupdate -d option
hostname cli> setservicemode false
hostname cli> reset ac
```

where *option* can be `cmsw -f path`, `bcfw -f path`, `bmcfw -f path`, or `scdf`, and *path* is `nfs://nfs.server.ip.address/directory/filename` where the software to use in the flash is installed. Note that if you want to update `rpdf`, you must set the `servicemode` to false before using the `flashupdate` command, and the alarm card will reset itself after finishing the `rpdf` update.

Setting the Date and Time on the Alarm Card

The alarm 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 alarm card using one of these methods:

- Manually, using the CLI `setdate` command. The date and time must be reset after any power cycle.
- Configuring the alarm card to be an NTP client, 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. The NTP server must be on the same subnet as the alarm card. Refer to the online man pages for the `xntpd`, `ntpq`, and `ntpdate` commands for more information about NTP.

▼ To Set the Alarm Card Date and Time Manually

1. Log in to the alarm card.
2. Set the date and time manually:

```
hostname cli> setdate mmddHHMMccyy
```

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; and *yy* is the current year.

▼ To Set the Alarm Card Date and Time as an NTP Client

1. Log in to the alarm card.
2. Set the date and time as an NTP client:

```
hostname cli> setntpserver addr
```


where *addr* is the IP address of the NTP server. The NTP server must be on the same subnet as the alarm card.

Running Scripts on the Alarm Card

This section describes the Netra CT server alarm card scripting feature.

Using Scripting

Normally, the alarm card cannot execute batch commands. The alarm card scripting feature allows you to write scripts to execute alarm card CLI commands in batch mode on the alarm card, similar to using scripting in the Solaris operating environment. You run the scripts from a host or satellite CPU board in the same system as the alarm card.

As an example, using the scripting feature, you can write a script to configure an Ethernet port on the alarm 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 alarm card:

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

The script includes the `rsh` command, the alarm card MCNet IP address, and the CLI command(s) to run. For information on the MCNet IP address, refer to [“Configuring the MCNet Interface” on page 17](#); for information on the CLI commands, refer to [TABLE 3-1](#).

Scripting Limitations

All the alarm card CLI commands in [TABLE 3-1](#) are supported in a script *except* for the following interactive commands: `userpassword`, `password`, `console`, and `break`.

For security reasons, you must be a root user on a host or satellite CPU board in the same system as the alarm card. The commands can be run only over the MCNet interface.

▼ To Run a Script on the Alarm Card

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

```
rsh alarm_card_MCNet_ipaddress CLI_command  
rsh alarm_card_MCNet_ipaddress CLI_command  
rsh alarm_card_MCNet_ipaddress CLI_command  
rsh alarm_card_MCNet_ipaddress CLI_command  
...
```

where *alarm_card_MCNet_ipaddress* is the MCNet IP address of the alarm 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 alarm card verifies that the commands are being run by a root user on a host or satellite CPU board in the same system as the alarm card, and that the commands have been received over the MCNet.

Viewing Alarm Card Logs

The alarm card keeps console logs and event logs. The logs are kept in buffers.

Console Logs

The alarm card console logs contain messages received from the host CPU board. There are two types of console logs:

- The `run` log contains the most recent data received from the host CPU operating system. The alarm card always writes to this log. When the `run` log is full, the alarm card overwrites old data in the `run` log.
- The `orun` log contains messages printed to the console (1) prior to a host CPU reboot or (2) when the `consolerestart` command is issued. When either of these events occur, the alarm card stores the contents of the `run` log in the `orun` log, and then clears the `run` log to store further host CPU operating system messages.

The `run` and `orun` logs together can contain up to 16 Kbytes of data.

▼ To View Console Logs

1. Log in to the alarm card.
2. View a console log with the `consolehistory` command:

```
hostname cli> consolehistory [run|orun] [index [+|-] n] [pause n]
```

where `index n` is the number of lines to display from either the oldest log entry forward (positive index) or the most recent log entry back (negative index); and `pause n` is the number of lines to display before pausing (default pause value is 10 lines). For example, to display the contents of the `run` log, pausing after 20 lines at a time, enter the following:

```
hostname cli> consolehistory run pause 20
```

If no options are specified, the `consolehistory` command prints out the entire contents of all non-empty console logs.

Event Logs

The alarm card event log contains event history, that is, all events that change the state of the system. The log entries are stored in the circular buffer of the alarm card RAM. The buffer holds up to 2,048 log entries; it is reset if the alarm card is reset.

A log entry includes the time of the event, a hostname, a unique event ID, and a description of the event. For example:

```
hostname cli> loghistory
Feb 3 02:38:10 netract: 0009: Alarm Card Booted
Feb 3 02:38:11 netract: 0004: ENET2 now DOWN
Feb 3 02:39:57 netract: 0022: User netract Logged on
...
```

▼ To View the Event Log

1. Log in to the alarm card.
2. View an event log with the `loghistory` command:

```
hostname cli> loghistory [index [+|-] n] [pause n]
```

where `index n` is the number of lines to display from either the oldest log entry forward (positive index) or the most recent log entry back (negative index); and `pause n` is the number of lines to display before pausing (the default is to display the entire log without pausing). For example, to display the last 30 lines of the event log, enter the following:

```
hostname cli> loghistory index -30
```

Booting CPU Boards

Host and satellite CPU boards can boot from a local disk or over the network.

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 CPU

boards through the alarm card CLI `setfru` command. Refer to [“Configuring a Chassis Slot for a Board” on page 15](#) for more information on using the `setfru` command to specify a boot device for a board.

When the alarm card powers on a board in a slot, the OpenBoot PROM firmware checks with the alarm card for a boot device for that slot. The alarm 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.

In the event of an alarm card fault, a CPU board hot swap, power cycle, reboot or reset will cause the OpenBoot PROM firmware to default to the value set in the `boot-device` variable.

Booting with a DHCP Server

You can configure Netra CT CPU boards to boot over DHCP. This process includes setting the CPU board boot device for DHCP, forming the CPU 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 CPU Board to Boot Over DHCP

1. Log in to the alarm card.
2. Set the boot device for the board to `dhcp` with the `setfru` command:

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

where `fru_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 CPU 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 CPU 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 `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 CPU board power cycle, reboot, or reset; the alarm card updates the client ID during a first-time power on or a hot swap of a CPU board. In the event of an alarm card fault, a CPU board reboot or reset will retrieve the previously written client ID.

Connecting to CPU Board Consoles from the Alarm Card

The Netra CT system provides the capability to connect to CPU boards and open console sessions from the alarm card.

You begin by logging in to the alarm card through either the serial port or the Ethernet port. Once a console session with a CPU 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 CPU board to enable console use.

▼ To Configure Your System for Multiple Consoles

1. **Log in as root to the CPU 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 Alarm Card and CPU Boards

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

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

TABLE 3-3 Alarm Card CLI Console-Related Commands

Command	Description
<code>console <i>cpu_node</i></code>	Enter console mode and connect to a specified CPU board, where <i>cpu_node</i> can be 1 to 8 on a Netra CT 810 or 1 to 5 on a Netra CT 410 server. If <i>cpu_node</i> is not specified, connect to the host CPU board.
<code>break <i>cpu_node</i></code>	Put the specified CPU board in debug mode, where <i>cpu_node</i> can be 1 to 8 on a Netra CT 810 or 1 to 5 on a Netra CT 410 server. Debug mode can use OpenBoot PROM or <code>kaadb</code> , depending on server configuration.
<code>setescapechar <i>value</i></code>	Set the escape character to be used in all future console sessions. The default is <code>~</code> (tilde). Refer to TABLE 3-4 for escape character use.
<code>showescapechar</code>	Show the current escape character.

Most CPU board consoles use the MCNet 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 will connect to it over the IPMI bus. This will cause 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 MCNet bus. The MCNet bus is faster than the IPMI bus, while the IMPI bus is typically a more stable communication channel than the MCNet bus.

Once you have a console connection with a CPU board, you can issue normal Solaris commands. There are several escape character sequences to control the current session. TABLE 3-4 shows these sequences.

TABLE 3-4 CPU Board Console-Related Escape Character Sequences

Sequence	Description
<code>~b</code>	Break from the Solaris level and enter the OpenBoot PROM (debug) level.
<code>~.</code>	End the console session.
<code>~g</code>	Determine the status (MCNet or IMPI) of the current console.
<code>~t</code>	Toggle between MCNet and IPMI.

▼ To Start a Console Session from the Alarm Card

1. Log in to the alarm card.

You can log in to the alarm 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 1 to 8 on a Netra CT 810 system or 1 to 5 on a Netra CT 410 system. 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 MCNET
```

This means the console is in Solaris mode.

▼ To Toggle Between MCNet and IPMI

Toggling between MCNet 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 CPU board is in Solaris mode, enter the escape sequence `~t`:**

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

The console switches between MCNet 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 will fail.

2. **To return to MCNet mode, enter `~t` again and press enter:**

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

▼ 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 alarm 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 alarm 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 CPU Boards

This section describes specifying recovery operations and controlling CPU boards through the alarm card PMS CLI commands.

Recovery Configuration of a CPU Board From the Alarm Card

You specify the recovery configuration of a CPU 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 CPU 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 alarm card's automatic recovery of a CPU board is disabled, and PMS applications are started in an offline state, so that you can use manual maintenance operations. *Operational mode* means the alarm card's automatic recovery of a CPU board is enabled; the alarm card will recover the CPU board in the event of a monitoring fault, and start PMS applications in an active state. *None mode* means the alarm 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 the parameter `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 CPU Board

1. Log in to the alarm 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 1 to 8, and `all` specifies all slots containing CPU 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 the 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 instructs PMS what to do in response 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 the parameters: `pc` (power cycle), `rst` (reset), `rstpc` (reset, then power cycle), `pd` (power down), `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 failure power 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-operation delay` (the time in deciseconds between one and the next operation for an operation with multiple retries; default is 0 deciseconds), 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 alarm 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 1 to 8, and `all` specifies all slots containing CPU 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 alarm 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 failure power on|off] [-r
retries] [-n inter-operation delay] [-p reset power cycle delay]
```

where `slot_num` can be a slot number from 1 to 8, and `all` specifies all slots containing CPU 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 CPU Board's Resources From the Alarm 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 the 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 the 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. You specify the operation to be performed on the specified slot by using the option `-o` with the 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.

Using the Netra High Availability Suite With the Netra CT Server Applications

The Netra High Availability (HA) Suite software provides enhanced services for customer high-availability applications. When installed, it runs on the host and satellite CPU boards. The Netra HA Suite provides reliable (redundant) services across CPU boards; you can fail over from one CPU board in one Netra CT system to another CPU board in another Netra CT system.

The MOH and PMS applications integrate with these Netra HA Suite foundation services: reliable NFS, reliable DHCP/boot server, and CGTP (Carrier-Grade Transport Protocol, providing IP packet services).

The MOH application has to manage these services, for example, monitoring the `nfs` and `tftp` daemons. It does this through the node manager agent (NMA). For example, if there is an NFS failure, the MOH application will detect this failure.

The points of interaction between the Netra CT server software and the Netra HA Suite are:

- MOH software modules interact with the Netra HA Suite Process Monitor Daemon (PMD) and NMA
- MOH I/O interfaces interact with the Netra HA Suite NMA and CGTP
- PMS interacts with the Netra HA Suite probe

The Netra HA Suite starts RNFS, RDHCP, and CGTP by default. If you want to change the Netra HA Suite services that are started by default, configure the Process Monitor Daemon (PMD). Refer to the Netra HA Suite documentation for more information on how to do this.

The Netra CT PMS probe brings together the PMS partner list and the Netra HA Suite master and vice-master cluster. Refer to the pms API man pages for more information on partner lists; the man pages are installed by default in `/opt/SUNWnetract/mgmt2.0/man`.

Refer to the Netra HA Suite documentation for more information on this application.

Monitoring Your System

This section describes various ways to monitor your system.

Command-line Interface Information

The alarm card CLI provides many commands to display system status. Refer to the alarm card CLI commands in the section, [“Using the Alarm Card Command-Line Interface” on page 40](#), in particular the `show` commands, to view system status. The alarm card also keeps several logs; refer to [“Viewing Alarm Card Logs” on page 52](#) for more information.

LED Information

The system status panel is a module designed to give feedback on the status of the key components within the Netra CT server. The system status panel has one set of LEDs for each component within that particular server. [FIGURE 3-1](#) shows the LEDs on the system status panel for the Netra CT 810 server, and [FIGURE 3-2](#) shows the LEDs on the system status panel for the Netra CT 410 server.

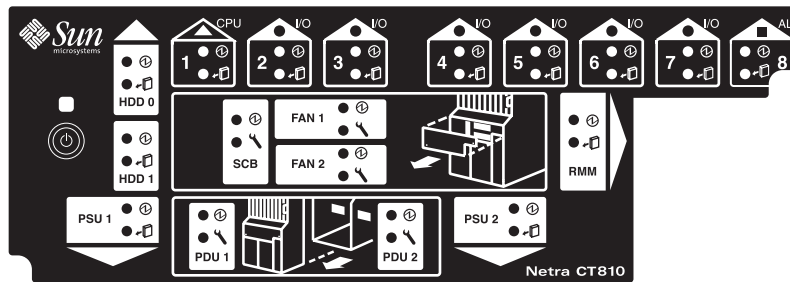


FIGURE 3-1 System Status Panel (Netra CT 810 Server)

TABLE 3-5 describes the system status panel LEDs for the Netra CT 810 server.

TABLE 3-5 System Status Panel LEDs for the Netra CT 810 Server

LED	LEDs Available	Component
HDD 0	Power and Okay to Remove	Upper hard disk drive
HDD 1	Power and Okay to Remove	Lower hard disk drive
Slot 1	Power and Okay to Remove	Host CPU board installed in slot 1
Slots 2-7	Power and Okay to Remove	I/O boards or satellite CPU boards installed in slots 2-7
Slot 8	Power and Okay to Remove	Alarm card installed in slot 8
SCB	Power and Fault	System controller board (behind the system status panel)
FAN 1	Power and Fault	Upper fan tray (behind the system status panel)
FAN 2	Power and Fault	Lower fan tray (behind the system status panel)
RMM	Power and Okay to Remove	Removable media module
PDU 1	Power and Fault (DC only)	Left power distribution unit (behind the server)
PDU 2	Power and Fault (DC only)	Right power distribution unit (behind the server)
PSU 1	Power and Okay to Remove	Left power supply unit
PSU 2	Power and Okay to Remove	Right power supply unit

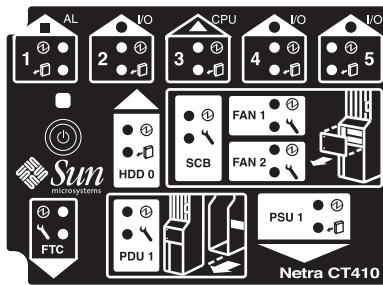


FIGURE 3-2 System Status Panel (Netra CT 410 Server)

TABLE 3-6 describes the system status panel LEDs for the Netra CT 810 server.

TABLE 3-6 System Status Panel LEDs for the Netra CT 410 Server

LED	LEDs Available	Component
Slot 1	Power and Okay to Remove	Alarm card installed in slot 1
Slot 2	Power and Okay to Remove	I/O board or satellite CPU board installed in slot 2
Slot 3	Power and Okay to Remove	Host CPU board installed in slot 3
Slot 4 and 5	Power and Okay to Remove	I/O boards or satellite CPU boards installed in slot 4 and 5
HDD 0	Power and Okay to Remove	Hard disk drive
SCB	Power and Fault	System controller board (behind the system status panel)
FAN 1	Power and Fault	Upper fan tray (behind the system status panel)
FAN 2	Power and Fault	Lower fan tray (behind the system status panel)
FTC	Power and Fault	Host CPU front transition card or host CPU front termination board
PDU 1	Power and Fault (DC only)	Power distribution unit (behind the server)
PSU 1	Power and Okay to Remove	Power supply

Each major component in the Netra CT 810 server or Netra CT 410 server has a set of LEDs on the system status panel that gives the status on that particular component. Each component has either the green Power and the amber Okay to Remove LEDs (FIGURE 3-3) or the green Power and amber Fault LEDs (FIGURE 3-4). Note that the components in the Netra CT servers all have the green Power LED, and they have either the amber Okay to Remove LED *or* the amber Fault LED, but not both.

Green Power LED



Amber Okay to Remove LED

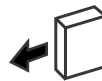


FIGURE 3-3 Power and Okay to Remove LEDs

Green Power LED



Amber Fault LED



FIGURE 3-4 Power and Fault LEDs

- [TABLE 3-7](#) gives the LED states and meanings for any *CompactPCI board* installed in a slot in the Netra CT 810 server or Netra CT 410 server.
- [TABLE 3-8](#) gives the LED states and meanings for any component other than a CompactPCI board that has the green Power and amber *Okay to Remove* LEDs.
- [TABLE 3-9](#) gives the LED states and meanings for any component other than a CompactPCI board that has the green Power and amber *Fault* LEDs.

TABLE 3-7 CompactPCI Board LED States and Meanings

Green Power LED state	Amber Okay to Remove LED state	Meaning	Action
Off	Off	The slot is empty or the system thinks that the slot is empty because the system didn't detect the board when it was inserted.	If there is a board installed in this slot, then one of the following components is faulty: <ul style="list-style-type: none"> • the board installed in the slot • the alarm card • the system controller board Remove and replace the failed component to clear this state.
Blinking	Off	The board is coming up or going down.	<i>Do not</i> remove the board in this state.
On	Off	The board is up and running.	<i>Do not</i> remove the board in this state.

TABLE 3-7 CompactPCI Board LED States and Meanings (Continued)

Green Power LED state	Amber Okay to Remove LED state	Meaning	Action
Off	On	The board is powered off.	You can remove the board in this state.
Blinking	On	The board is powered on, but it is offline for some reason (for example, a fault was detected on the board).	<p>Wait several seconds to see if the green Power LED stops blinking. If it does not stop blinking after several seconds, enter <code>cfgadm -a1</code> and verify that the board is in the unconfigured and disconnected state, then perform the necessary action, depending on the board:</p> <ul style="list-style-type: none"> • Alarm card—You can remove the alarm card in this state. • All other boards—Power off the slot through the alarm card software, then remove the board.
On	On	The board is powered on and is in use, but a fault has been detected on the board.	<p>Deactivate the board using one of the following methods:</p> <ul style="list-style-type: none"> • Use the <code>cfgadm -f -c unconfigure</code> command to deactivate the board. Note that in some cases, this may cause the system to panic, depending on the nature of the board hardware or software. • Halt the system and power off the slot through the alarm card software, then remove the board. The green Power LED will then give status information: <ul style="list-style-type: none"> • If the green Power LED goes off, then you can remove the board. • If the green Power LED remains on, then you must halt the system and power off the slot through the alarm card software.

TABLE 3-8 Meanings of Power and Okay to Remove LEDs



LED State	Power LED 	Okay to Remove LED 
On, Solid	Component is installed and configured.	Component is Okay to Remove. You can remove the component from the system, if necessary.
On, Flashing	Component is installed but is unconfigured or is going through the configuration process.	Not applicable.

TABLE 3-8 Meanings of Power and Okay to Remove LEDs (Continued)





LED State	Power LED 	Okay to Remove LED 
Off	Component was not recognized by the system or is not installed in the slot.	Component is <i>not</i> Okay to Remove. Do <i>not</i> remove the component while the system is running.

TABLE 3-9 Meanings of Power and Fault LEDs

LED State	Power LED 	Fault LED 
On, Solid	Component is installed and configured.	Component has failed. Replace the component.
On, Flashing	Component is installed but is unconfigured or is going through the configuration process.	Not applicable.
Off	Component was not recognized by the system or is not installed in the slot.	Component is functioning properly.

There is also a green system power LED and power on/off button located on the system status panel. When the system is off, the system power LED will be unlit. Pressing the system power button when the system is off will start the power-up sequence. Once the system is completely powered up, the system power LED remains on.

When the system is powered on, pressing the system power button for less than 4 seconds will start the orderly power-down sequence—in a manner that no persistent operating system data structures are corrupted—indicated by a blinking LED. In the orderly power-down, applications in service may be abnormally terminated and no further services will be invoked by the CPU. Once the CPU has reached a quiescent state (run level-0, as if `init 0` had been invoked), then the power supply(s) will turn off, indicated by the LED changing from a blinking state to the off state.

If the button is held down for 4 seconds or longer, the power supply(s) are turned off without any intervention of the CPU; that is, the “emergency” power-down sequence occurs.

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/mgmt2.0/bin
# ./ctmgx start [option]
```

Refer to [TABLE 2-5](#) for the options available with `ctmgx start`.

```
# cd /opt/SUNWnetract/mgmt2.0/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

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

Hot Swap on the Netra CT Server

Most FRUs in the Netra CT system are hot-swappable.¹ Hot swap, a key feature of the PICMG standard, means that a CompactPCI board that meets the PICMG standard can be reliably inserted into or extracted from a powered and operating CompactPCI platform without affecting the other functions of the platform.

The Netra CT system hot-swap modes are shown in [TABLE 3-10](#).

TABLE 3-10 Netra CT System Hot-Swap Modes

Type of Hot Swap	Description
Basic	The hardware connection/disconnection process is performed automatically by the hardware, while the software connection process requires user assistance through the <code>cfgadm (1M)</code> command
Full	Both the hardware and the software connection process are performed automatically
High Availability	High availability hot swap provides the ability to control the hardware connection process. This provides a higher degree of control than just indicating insertion and extraction of a board. The hardware connection process is controlled by software on high availability systems, such as the Netra CT server

The Netra CT system is configured for full hot swap by default. You can change the mode of the slot for the CPU boards and I/O boards to basic or full hot swap using the `cfgadm (1M)` command. You might want to change the hot-swap state of a slot to basic, for example, if you need to insert or remove a third-party I/O board that does not have full hot-swap support.

Note that whenever you reboot or power your system on and off, the hot-swap states revert back to the default full hot-swap state for all I/O slots.

Complete information on hot swapping FRUs is contained in the *Netra CT Server Service Manual*.

1. Exceptions include the single power supply and the single hard disk drive in the Netra CT 410 server; a single or lone remaining power supply and a single or lone remaining hard disk drive in the Netra CT 810 server; and the power distribution units.

How High Availability Hot Swap Works

By default, the Netra CT server is configured to accept any cPCI FRU unless you specifically set an allowable plug-in for a specific slot. (Refer to [“Configuring a Chassis Slot for a Board”](#) on page 15 for more information.)

When a board is inserted into the Netra CT server, the alarm card checks the midplane FRU ID information for allowable FRUs for that slot, then checks the inserted board’s FRU ID to make sure the board is allowed in the particular slot. If the board is allowed in the slot, the alarm card powers up the board. If the board is not allowed in the slot, the alarm card does not enable power to the slot.

If a host or satellite CPU board is in use, that is, has applications currently running, the alarm card CLI power commands, such as `poweron` or `poweroff`, will not work for that CPU board.

Hot Swap With Boards That Don’t Support Full Hot Swap

You might want to change the hot-swap state of a slot from full to basic if you need to insert or remove a third-party I/O board that does not have full hot-swap support.

To determine the current hot-swap state of a slot, you use the `prtconf(1M)` command. To enable or disable a type of hot swap on a slot, you use the `cfgadm(1M)` command. For many `cfgadm` commands, you must know the attachment point ID for the I/O slot that you will be working on.

▼ To Determine the Current Hot-Swap State of a Slot

- As root on the server, enter the command:

```
# prtconf -v -P
```

For a Netra CT 810 server, the output is similar to the following:

```
cphsc, instance #0
  System properties:
    name='instance' type=int items=1
    value=00000000
    name='default-hotswap-mode' type=string items=1
    value='full'
  Driver properties:
    name='AL-8-autoconfig' type=string items=1 dev=none
    value='enabled'
    name='IO-7-autoconfig' type=string items=1 dev=none
    value='enabled'
    name='IO-6-autoconfig' type=string items=1 dev=none
    value='enabled'
    name='IO-5-autoconfig' type=string items=1 dev=none
    value='enabled'
    name='IO-4-autoconfig' type=string items=1 dev=none
    value='enabled'
    name='IO-3-autoconfig' type=string items=1 dev=none
    value='enabled'
    name='IO-2-autoconfig' type=string items=1 dev=none
    value='enabled'
    name='CPU-autoconfig' type=string items=1 dev=none
    value='enabled'
    name='hotswap-mode' type=string items=1 dev=none
    value='full'
```

- If you see value `'basic'` underneath the `default-hotswap-mode` line, then *all* of the I/O slots in the Netra CT server have been set to *basic* hot swap. You should see value `'disabled'` for every I/O slot in the system in this situation.
- If you see value `'full'` underneath the `default-hotswap-mode` line, then *at least one* of the I/O slots in the Netra CT server has been set to *full* hot swap. You must look at the entries for individual I/O slots to determine if they have been set to basic or full hot swap mode in this situation:

- If you see value 'enabled' underneath an autoconfig line, then that slot is set to *full* hot swap.
- If you see value 'disabled' underneath an autoconfig line, then that slot is set to *basic* hot swap.

▼ To List Attachment Point IDs for I/O Slots

- As root on the server, enter the command:

```
# cfigadm
```

For a Netra CT 810 server, the output is similar to the following:

Ap_Id	Type	Receptacle	Occupant	Condition
AL-8	mcd/fhs	connected	configured	ok
CPU	bridge/fhs	connected	configured	ok
IO-2	stpcipci/fhs	connected	configured	ok
IO-3	unknown	empty	unconfigured	unknown
IO-4	stpcipci/fhs	connected	configured	ok
IO-5	unknown	empty	unconfigured	unknown
IO-6	unknown	empty	unconfigured	unknown
IO-7	unknown	empty	unconfigured	unknown

where the attachment point ID is shown in the first column of the readout; for example, the attachment point ID for I/O slot 2 in a Netra CT 810 server would be IO-2.

For a Netra CT 410 server, the output is similar to the following:

Ap_Id	Type	Receptacle	Occupant	Condition
AL-1	mcd/fhs	connected	configured	ok
CPU	bridge/fhs	connected	configured	ok
IO-2	unknown	empty	unconfigured	unknown
IO-4	stpcipci/fhs	connected	configured	ok
IO-5	stpcipci/fhs	connected	configured	ok

where the attachment point ID is shown in the first column of the readout; for example, the attachment point ID for I/O slot 4 in a Netra CT 410 server would be IO-4.

▼ To Disable Full Hot Swap and Enable Basic Hot Swap

- As root on the server, enter the command:

```
# cfgadm -x disable_autoconfig ap_id
```

where *ap_id* is the attachment point ID in the server that you want to have basic hot swap enabled on.

▼ To Re-Enable Full Hot Swap

- As root on the server, enter the command:

```
# cfgadm -x enable_autoconfig ap_id
```

where *ap_id* is the attachment point ID in the server that you want to have full hot swap enabled on.

Index

A

- ACL information, 27 to 30
- alarm card
 - accessing, 6
 - and MOH application, 27 to 29, 31 to 32
 - and PMS application, 33 to 38, 63 to 68
 - command-line interface, 1, 3, 40 to 47
 - console, 6, 42, 47, 57 to 63
 - date and time, 50
 - description of, 2, 3
 - Ethernet ports, 8
 - flashing, 48
 - logs, 52 to 54
 - permission levels, 10, 31, 40
 - scripts, 51 to 52
 - serial ports, 6
 - user account, 6, 10 to 11

B

- BCF firmware, 2, 3, 48
- BMC firmware, 2, 3, 48
- boot device, 13, 43, 54
- booting, 54 to 56

C

- ChorusOS operating environment, 1, 3, 48
- cluster, 69
- command-line interface, 1, 3, 40 to 47

commands

- break, 42, 51, 59
- cfgadm, 73, 76, 77, 79, 80
- console, 42, 51, 59, 60
- consolehistory, 42, 53
- consolerestart, 42, 53
- ctmgx start, 26, 30, 32, 75
- ctmgx stop, 75
- eeprom, 58
- flashupdate, 43, 48 to 49
- help, 43
- ifconfig, 19
- loghistory, 44, 54
- logout, 43
- password, 43, 51
- patchadd, 26
- ping, 19
- pkginfo, 26
- pmsd apphistoryshow, 47, 68
- pmsd appinfoshow, 47, 67
- pmsd appoperset, 47, 67
- pmsd help, 42
- pmsd historyshow, 46, 64
- pmsd hwhistoryshow, 46, 67
- pmsd hwinfoshow, 46, 67
- pmsd hwoperset, 46, 66
- pmsd infoshow, 46, 64
- pmsd operset, 45, 63, 64
- pmsd oshistoryshow, 46, 67
- pmsd osinfoshow, 46, 67
- pmsd osoperset, 46, 67
- pmsd recoveryautoinfoshow, 46, 66
- pmsd recoveryautooperset, 46, 65, 66
- pmsd recoveryoperset, 46, 65

- pmsd slotaddressset, 36, 37, 45
- pmsd slotaddressshow, 36, 45
- pmsd slotrndaddressadd, 37, 45
- pmsd slotrndaddressdelete, 38, 45
- pmsd slotrndaddressshow, 38, 45
- pmsd start, 34, 36, 45
- pmsd stop, 35, 45
- pmsd usage, 47
- pmsd version, 47
- poweroff, 41, 77
- poweron, 41, 77
- powersupply, 42
- prstat, 34
- prtconf, 77, 78
- reset, 42
- rsh, 51
- setdate, 43, 50
- setenv, 58
- setescapechar, 42, 59, 63
- setfru, 12 to 18, 21, 43
- sethostname, 45
- setipaddr, 9, 44
- setipgateway, 9, 45
- setipmode, 8, 9, 44
- setipnetmask, 9, 45
- setmohsecurity, 31, 44
- setntpserver, 43, 50
- setpanicdump, 42
- setrecovery, 43
- setsecondaryboot, 43
- setserialbaud, 7, 44
- setserialdata, 7, 44
- setserialhwhandshake, 7, 44
- setserialmode, 7, 44
- setserialparity, 7, 44
- setserialstop, 7, 44
- set servicemode, 45, 48, 49
- showcpustate, 41
- showdate, 41
- showenvironment, 40
- showescapechar, 43, 59, 62
- showfru, 22 to 24, 41
- showhealth, 42
- showhostname, 41
- showipaddr, 41
- showipgateway, 41
- showipmode, 41
- showipnetmask, 41
- showmohsecurity, 41

- shownetwork, 20, 40
- showntpserver, 41
- showpanicdump, 42
- showrecovery, 43
- showsecondaryboot, 43
- showserialbaud, 40
- showserialdata, 41
- showserialhwhandshake, 41
- showserialmode, 40
- showserialparity, 41
- showserialstop, 41
- showservicemode, 41
- snmpconfig, 28, 44
- useradd, 10, 42
- userdel, 42
- userpassword, 10, 43, 51
- userperm, 10, 40, 43
- usershow, 42
- version, 43

- console, 6, 42, 47, 57 to 63

- context switch, 34

- cPCI bus, 2, 3

CPU board

- booting, 54 to 56

- configuring chassis slot for, 15 to 17

- configuring MOH for SNMP, 27 to 30

- console, 57 to 63

- controlling other CPU boards, 37 to 38

- description of, 2, 3

- enabling PMS application on, 33

- hot swap, 76 to 77

- in Netra CT configuration, 15

- LEDs, 72 to 73

- recovery, 63 to 66

- scripts, 51 to 52

- starting MOH application on, 26

D

- date and time, 50

- DHCP, 55 to 56, 68

- documentation, xiii

E

- Ethernet ports, 8, 17

F

failover, Ethernet ports, 9
flash, alarm card, 43, 48
FRU ID, 11 to 24, 77
FRU, system, 11

H

host CPU board
 booting, 25, 54 to 56
 configuring chassis slot for, 15 to 17
 configuring MOH for RMI, 31 to 32
 configuring MOH for SNMP, 27 to 30
 console, 57 to 63
 controlling other CPU boards, 37 to 38
 description of, 2, 3
 enabling PMS application on, 33
 hot swap, 76 to 77
 in Netra CT configuration, 15
 LEDs, 72 to 73
 recovery, 63 to 66
 scripts, 51 to 52
 starting MOH application on, 26
hot swap, 76 to 80

I

I/O board
 configuring chassis slot for, 15 to 17
 description of, 3
 hot swap, 76 to 77
 in Netra CT configuration, 15
 LEDs, 72 to 73
IPMI, 2, 3, 59

J

JDMK, 29

L

LEDs, 69 to 74
logs, 52 to 54

M

Managed Object Hierarchy application, *see* MOH application
MCNet, 2, 3, 17 to 20, 51, 59
MOH application, 2, 3, 25 to 32

N

Netra High Availability Suite, 2, 68 to 69
NFS, 68
NTP, 50

O

OpenBoot PROM firmware, 2, 3, 13, 16, 25, 54, 62

P

panic dump, 42
partner list, 33, 69
password, 6, 11
patches, Solaris, 25
permission levels, alarm card, 10, 31, 40
PMS application, 2, 3, 32 to 38, 63 to 68
POST, 2
power on/off server, 14, 21, 74, 76, 77
Processor Management Service application, *see* PMS application

R

recovery, CPU boards, 63 to 66
RMI, 26, 27, 31 to 32

S

satellite CPU board
 booting, 54 to 56
 configuring chassis slot for, 15 to 17
 configuring MOH for SNMP, 27 to 30
 console, 57 to 63
 controlling other CPU boards, 37 to 38

- description of, 2, 3
- enabling PMS application on, 33
- hot swap, 76 to 77
- in Netra CT configuration, 15
- LEDs, 72 to 73
- recovery, 63 to 66
- scripts, 51 to 52
- starting MOH application on, 26
- scripts, alarm card, 51 to 52
- security, 27, 31, 47, 52
- serial ports, 6
- slot, 13, 15 to 17, 77 to 80
- SMC firmware, 2, 3
- SNMP, 26, 27 to 30
- Solaris operating environment, 1, 3, 25
- system status panel, 69

T

- telnet, 6
- tip program, 6

U

- user account, 3, 6, 10

V

- variables, system configuration, 40, 48