



# Netra™ CP32x0 10GbE Advanced Rear Transition Module, Dual Port User's Guide

---

Sun Microsystems, Inc.  
[www.sun.com](http://www.sun.com)

Part No. 820-3150-10  
September 2008, Revision 01

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

Copyright © 2008 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

This distribution may include materials developed by third parties.

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, Netra, Sun Ray, the Netra logo and the Solaris logo are trademarks or registered trademarks of Sun Microsystems, Inc., or its subsidiaries, in the U.S. and other countries.

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

Use of any spare or replacement CPUs is limited to repair or one-for-one replacement of CPUs in products exported in compliance with U.S. export laws. Use of CPUs as product upgrades unless authorized by the U.S. Government is strictly prohibited.

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 © 2008 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Cette distribution peut comprendre des composants développés par des tierces parties.

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, Netra, Sun Ray, le logo Netra et le logo Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc., ou ses filiales, 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 portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'utilisation de pieces detachees ou d'unités centrales de remplacement est limitee aux reparations ou a l'echange standard d'unités centrales pour les produits exportes, conformément a la legislation americaine en matiere d'exportation. Sauf autorisation par les autorites des Etats-Unis, l'utilisation d'unités centrales pour proceder a des mises a jour de produits est rigoureusement interdite.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" 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 CONTREFACON.



Please  
Recycle



Adobe PostScript





# Contents

---

## **Preface xv**

## **1. Overview 1-1**

- 1.1 Features 1-2
- 1.2 I/O PICMG Standards Compliance 1-3
- 1.3 I/O Interfaces 1-3
  - 1.3.1 External SFP Connections 1-4
    - 1.3.1.1 Features Common to all Ports 1-4
    - 1.3.1.2 1GbE Port Features 1-5
    - 1.3.1.3 10GbE Port Features 1-5
  - 1.3.2 Ethernet Management Port (RJ-45) 1-5
  - 1.3.3 Serial Console Port (RJ-45) 1-6
- 1.4 Back Panel LEDs 1-6
- 1.5 Part Number and Serial Number Labels 1-9
- 1.6 Software Support 1-10

## **2. Installing and Removing the ARTM 2-1**

- 2.1 Important Information 2-2
  - 2.1.1 Take Antistatic Precautions 2-2
  - 2.1.2 Unpack the Equipment Carefully 2-3

- 2.1.3 Use a Compatible Chassis and Slot 2-3
  - 2.1.3.1 Safety Statement 2-4
  - 2.1.3.2 Maximum Power Requirements 2-4
- 2.1.4 Use a Compatible Host Board 2-4
- 2.2 Before Installing or Removing the ARTM 2-5
  - 2.2.1 Observe ESD Precautions 2-5
  - 2.2.2 Inspect for Bent PINs or Other Damage 2-5
  - 2.2.3 Preserve EMI Compliance 2-6
  - 2.2.4 Understand Hotswap 2-6
  - 2.2.5 Verify Slot Usage 2-6
- 2.3 Installing the ARTM 2-7
- 2.4 Upgrading the Firmware 2-10
- 2.5 Removing the ARTM 2-11
- 2.6 Verifying the Hardware Installation 2-12

### **3. Driver Installation and Configuration 3-1**

- 3.1 Installing the nxge Driver 3-2
  - 3.1.1 Download and Install for Solaris Operating Systems 3-3
  - 3.1.2 Download and Install for Linux Operating Systems 3-4
- 3.2 Configuring the nxge Driver 3-6
  - 3.2.1 Configuring nxge Driver Parameters on a Solaris Platform 3-6
    - 3.2.1.1 Modifying Parameters Using the ndd Utility 3-7
    - 3.2.1.2 Modifying Parameters Using the nxge.conf File 3-9
  - 3.2.2 Configuring Driver Parameters on a Linux Platform 3-15
    - 3.2.2.1 Modifying Parameters Using the ethtool Utility 3-15
    - 3.2.2.2 Modifying Parameters Using configtool Utility 3-19
- 3.3 Tuning Performance on a Linux Platform 3-20
- 3.4 Enabling Host Board PCI-Express Communication 3-21

3.5	Removing the nxge Driver	3-22
<b>4.</b>	<b>Network Configuration</b>	<b>4-1</b>
4.1	Configuring Network Host Files	4-2
4.2	Configuring Jumbo Frames	4-5
4.2.1	Displaying Layer Configuration in a Solaris Environment	4-5
4.2.2	Displaying Driver Statistics	4-6
4.2.3	Enabling Jumbo Frames	4-8
4.2.3.1	Enabling Jumbo Frames in a Solaris Environment	4-8
4.2.3.2	Enabling Jumbo Frames in a Linux Environment	4-9
4.3	Configuring Link Aggregation	4-10
4.3.1	Configuring Link Aggregation in a Solaris Environment	4-10
4.3.2	Configuring Link Aggregation in a Linux Environment	4-12
4.4	Configuring VLANs	4-13
4.4.1	Configuring VLANs in a Solaris Environment	4-14
4.4.2	Configuring VLANs in a Linux Environment	4-16
<b>A.</b>	<b>MMC and IPMI Functions</b>	<b>A-1</b>
A.1	MMC and IPMI Functions	A-2
A.2	Sensor Data Records	A-3
A.3	FRU Information Records	A-3
A.4	ARTM e-Keying Port Assignments	A-5
A.5	IPMI Commands	A-6
<b>B.</b>	<b>Specifications and Connectors</b>	<b>B-1</b>
B.1	Specifications	B-2
B.1.1	Physical Dimensions	B-2
B.1.2	Power Requirements	B-2
B.1.3	Environmental Specifications and Compliance	B-2
B.1.4	Pressure Drop Versus Airflow	B-3

B.1.5	NEBS Compliance	B-5
B.1.6	Electromagnetic Compliance	B-5
B.2	Connectors and PIN Assignments	B-6
B.2.1	1GbE SFP Ethernet Ports	B-8
B.2.2	10GbE SFP+ Ethernet Ports	B-8
B.2.3	10/100/1000 Mb Management Port	B-8
B.2.4	Serial Console Management Port	B-9
B.2.5	Zone 3 Connectors	B-10
B.2.5.1	Connector P31, 10GbE XAUI PIN Assignments	B-12
B.2.5.2	Connector P32, Management Infrastructure PIN Assignments	B-13
B.2.5.3	Connector P33, Management Infrastructure PIN Assignments	B-14
B.2.6	Connector P30, ARTM Power PIN Assignment	B-15

**Index   Index-1**

# Figures

---

FIGURE 1-1	Netra CP32x0 ARTM-10GbE Functional Interconnect	1–4
FIGURE 1-2	Ethernet Management Port	1–5
FIGURE 1-3	Netra CP32x0 ARTM-10GbE Back Panel LEDs	1–7
FIGURE 1-4	Netra CP32x0 ARTM-10G Identification Labels	1–9
FIGURE 2-1	Installing a Node Board and ARTM Into an ATCA Chassis	2–8
FIGURE 2-2	Injector/Ejector Latch and Locking Screw	2–9
FIGURE B-1	Pressure Drop Simulation Model	B–4
FIGURE B-2	Netra CP32x0 ARTM-10G Connectors	B–7
FIGURE B-3	10/100/1000 Mb Management Port PIN Location	B–9
FIGURE B-4	Serial Console Management Port PIN Locations	B–10
FIGURE B-5	Zone 3 Connector Port PIN Locations	B–11
FIGURE B-6	J30/P30 Receptacle and Connector	B–16



# Tables

---

TABLE 1-1	I/O Ports on ARTM-10GbE Faceplate	1–3
TABLE 1-2	LEDs and Their Meanings	1–8
TABLE 1-3	Netra CP32x0 ARTM-10G Identification Labels	1–10
TABLE 2-1	Compatible Sun Host Boards	2–4
TABLE 4-1	Ethernet Tag Header Format	4–14
TABLE A-1	Sensor Data Records	A–3
TABLE A-2	Standard FRU Product Data Records	A–3
TABLE A-3	Standard FRU Board Data Records	A–4
TABLE A-4	ARTM e-Keying Port Assignments	A–5
TABLE A-5	IPMI Commands	A–6
TABLE B-1	Environmental Specifications	B–2
TABLE B-2	Pressure Drop Versus Inlet Air Speed	B–4
TABLE B-3	EMC Emission Compliance	B–5
TABLE B-4	Connector Port Identification and Location	B–7
TABLE B-5	1GbE SFP Connector Characteristics	B–8
TABLE B-6	10GbE SFP+ Connector Characteristics	B–8
TABLE B-7	10/100/1000 Management Port PIN Locations	B–9
TABLE B-8	Serial Console Management Port PIN Locations	B–10
TABLE B-9	Connector P31, 10GbE XAUI PIN Assignments	B–12
TABLE B-10	Connector P31, 10GbE XAUI Signal Descriptions	B–12

<a href="#">TABLE B-11</a>	Connector P32, Management Infrastructure PIN Assignments	B-13
<a href="#">TABLE B-12</a>	Connector P32, Management Infrastructure Signal Descriptions	B-13
<a href="#">TABLE B-13</a>	Connector P33, PCIe and Miscellaneous ARTM PIN Assignments	B-14
<a href="#">TABLE B-14</a>	Connector P33, PCIe and Miscellaneous ARTM Signal Descriptions	B-14
<a href="#">TABLE B-15</a>	P30, ARTM Power PIN Header Assignment	B-16
<a href="#">TABLE B-16</a>	Connector P30, ARTM Power PIN Signal Descriptions	B-17

# Code Examples

---

- [CODE EXAMPLE 3-1](#) Driver Parameters Modifiable via `nxge.conf` File 3-11
- [CODE EXAMPLE 3-2](#) Listing Modifiable Driver Parameters on a Linux Platform 3-15
- [CODE EXAMPLE 3-3](#) Common Parameters That are Modifiable on a Linux Platform 3-17
- [CODE EXAMPLE 3-4](#) Changing RX Coalesce Parameters 3-18
- [CODE EXAMPLE 3-5](#) Obtaining Status of L4 Hardware 3-18
- [CODE EXAMPLE 3-6](#) Obtaining a List of Tunable Driver Parameters on a Linux Platform 3-19
- [CODE EXAMPLE 3-7](#) Sample `.conf` File for Tuning Ethernet Ports on a Linux Platform 3-20



# Preface

---

This user's guide explains how to unpack, install, remove, and use the Netra CP32x0 10GbE Advanced Rear Transition Module, Dual Port (ARTM-10G) and related software and firmware.

---

# Using UNIX Commands

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

- Software documentation that you received with your system
- Solaris™ Operating System 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.
<b>AaBbCc123</b>	What you type, when contrasted with on-screen computer output	% <b>su</b> 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 file name</code> .

---

**Note** – Characters display differently depending on browser settings. If characters do not display correctly, change the character encoding in your browser to Unicode UTF-8.

---

---

## Related Documentation

The following table lists the documentation for this product. Except for the *Important Safety Information for Sun Hardware Systems*, all the documents listed are available online at:

<http://docs.sun.com/app/docs/prod/cp32x0.10gbee?l=en#hic>

Application	Title	Part Number	Format	Location
Getting Started	<i>Netra CP32x0 10GbE Ethernet Advanced Rear Transition Module, Dual Port Getting Started Guide</i>	820-3151-xx	Printed PDF	Shipping kit Online
Latest information, Upgrades	<i>Netra CP32x0 Advanced Rear Transition Module Product Notes</i>	820-3261-xx	PDF	Online
Usage	<i>Netra CP32x0 10GbE Ethernet Advanced Rear Transition Module, Dual Port User's Guide (this manual)</i>	820-3150-xx	PDF HTML	Online Online
Safety and Compliance	<i>Netra CP32x0 Advanced Rear Transition Module Safety and Compliance Manual</i>	820-3506	PDF	Online
Safety	<i>Important Safety Information for Sun Hardware Systems</i>	816-7190	Printed	Shipping kit

The following table lists the documentation that is related to this product.

Application	Title	Part Number	Format	Location
Latest information	<i>Netra CP3220 Board Product Notes</i>	820-0455	PDF	Online at:  <a href="http://docs.sun.com/app/docs/prod/cp3220.brd#hic">http://docs.sun.com/app/docs/prod/cp3220.brd#hic</a>
Latest information	<i>Netra CP3260 Board Product Notes</i>	820-0455	PDF	Online at:  <a href="http://docs.sun.com/app/docs/prod/cp3260.brd#hic">http://docs.sun.com/app/docs/prod/cp3260.brd#hic</a>

---

# Documentation, Support, and Training

Sun Function	URL
Documentation	<a href="http://www.sun.com/documentation">http://www.sun.com/documentation</a>
Support	<a href="http://www.sun.com/support/">http://www.sun.com/support/</a>
Training	<a href="http://www.sun.com/training/">http://www.sun.com/training/</a>

---

## Third-Party Web Sites

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

---

## 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 CP32x0 10GbE Advanced Rear Transition Module, Dual Port User's Guide*, part number 820-3150-10.

# Overview

---

This chapter provides an overview of the Netra CP32x0 10GbE Advanced Rear Transition Module, Dual Port (ARTM-10G).

This chapter contains the following topics:

- [Section 1.1, “Features” on page 1-2](#)
- [Section 1.2, “I/O PICMG Standards Compliance” on page 1-3](#)
- [Section 1.4, “Back Panel LEDs” on page 1-6](#)
- [Section 1.5, “Part Number and Serial Number Labels” on page 1-9](#)
- [Section 1.6, “Software Support” on page 1-10](#)

---

## 1.1 Features

The Netra CP32x0 ARTM-10G is an advanced rear transition module (ARTM) for AdvancedTCA systems. It provides up to four 10GbE SFP+ and two 1GbE SFP (small form-factor pluggable) connectors on the ARTM faceplate that together provide the mechanical and electrical means to conduct high-speed data transfers with an ATCA node board.

The Netra CP32x0 ARTM-10G extends CPU and OS parallelism with its support for hardware-based flow classification and multiple DMAs. Using CPU thread affinity to bind a given flow to a specific CPU thread, the Netra CP32x0 ARTM-10G enables a one-to-one correlation of RX and TX packets across the same TCP connection. This functionality can help eliminate cross-calls and context switching to deliver greater performance while reducing the need for CPU resources to support I/O processing. The Netra CP32x0 ARTM-10G uses a Sun™ MAC controller to map the 10-Gigabit XAUI interface onto the ARTM form factor.

ATCA cards with proper Zone 3 connector pin-outs can route traffic to the ARTM interface resources via x8 PCIe and/or 10GbE XAUI channels. The Netra CP32x0 ARTM-10G includes two R-J45 management ports (RS-232 and 1GbE) and a Module Management Controller (MMC).

The Netra CP32x0 ARTM-10G is a 6U x 80mm single slot ARTM that supplies several high speed I/O connections. The prominent digital components include:

- Sun ASIC that converts PCIe to 10GbE XAUI and 1GbE
- 10GbE transceiver
- 1GbE transceiver
- MMC, with hotswap and LED control
- FRU ID serial EERPORM (the FRU ID is not a separate part but is embedded in the MMC controller)
- Temperature sensors
- Voltage sensors
- SPI flash memory

---

## 1.2 I/O PICMG Standards Compliance

The Netra CP32x0 ARTM-10G expansion ARTM is fully compliant with the PCI Industrial Computer Manufacturers Group (PICMG) specifications, PICMG 3.1.

---

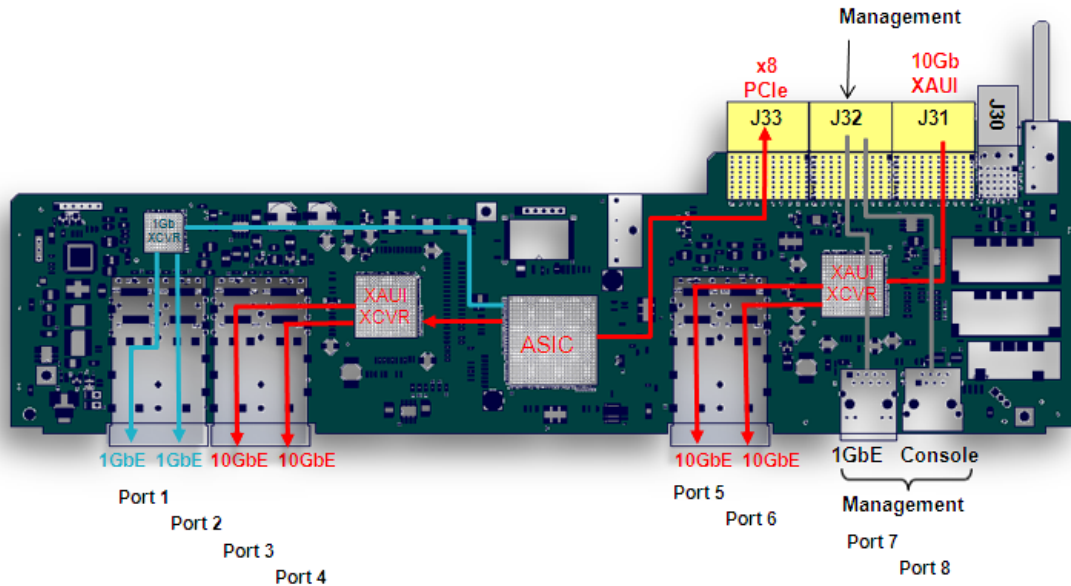
## 1.3 I/O Interfaces

The Netra CP32x0 ARTM-10G mates to ATCA carrier blades via Zone 3 connectors in a fashion conforming to PICMG 3.0 ATCA specification, revision 2. Internally the Netra CP32x0 ARTM-10G provides circuitry to “bridge” between Zone 3 and the ARTM faceplate as follows in [TABLE 1-1](#) and [FIGURE 1-1](#).

**TABLE 1-1** I/O Ports on ARTM-10GbE Faceplate

RTM Zone-3 Function	RTM Faceplate
PCIe, x8 (2.5 GHz)	Port 1 - 1GbE SFP
	Port 2 - 1GbE SFP
	Port 3 - 10GbE SFP+
	Port 4 - 10GbE SFP +
10GbE -XAUI	Port 5 - 10GbE SFP+
	Port 6 - 10GbE SFP+
Management port 1	Port 7 - 1GbE R-J45
Management port 2	Port 8 - RS-232 R-J45
IPMI to MMC device on ARTM	NA
Power	NA

**FIGURE 1-1** Netra CP32x0 ARTM-10GbE Functional Interconnect



## 1.3.1 External SFP Connections

The Netra CP32x0 ARTM-10G expansion supports up to six optical small form-factor pluggable (SFP) connections. These connections exchange high-speed Ethernet packet traffic with external equipment. See [Appendix B](#) for a list of compatible SFP modules.

### 1.3.1.1 Features Common to all Ports

- IEEE 802.3ae 2002-compliant
- Networking I/O virtualization supporting Solaris LDOMS 1.0 software
- Hardware-based flow classification for extending parallelism and virtualization to networking
- Up to 16 receive DMA channels and up to 24 transmit DMA channels, multiple receive and transmit descriptor rings, and dedicated networking hardware resources (DMA, interrupts, buffer, and more) for each thread or strand
- CPU/thread affinity and CPU load balancing at L1, L2, L3, and L4

### 1.3.1.2 1GbE Port Features

Both of the 1-Gigabit Ethernet ports (ports 1 and 2) are full duplex and use a SFP optical transceiver with a duplex high-level data link control (HDLC) fiber connector. Functional features of these ports:

- TCP and UDP checksum off load and CRC32C support
- L1 to L4 IPv4/v6 header parsing
- Jumbo frames support (up to 9216B)

### 1.3.1.3 10GbE Port Features

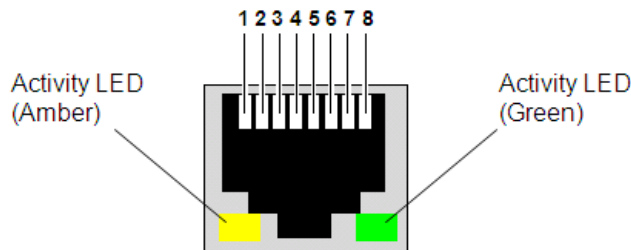
All of the 10-Gigabit Ethernet ports (ports 3, 4, 5, 6) are full duplex and use a small form-factor pluggable (SFP+) optical transceiver with duplex LC fiber connector. Functional features of these ports:

- TCP and UDP checksum off load and CRC32C support
- L1 to L4 IPv4/v6 header parsing
- Jumbo frames support (up to 9216B)
- 16 unique MAC addresses per port

## 1.3.2 Ethernet Management Port (RJ-45)

An Ethernet management port is provided on the Netra CP32x0 ARTM-10G (see [FIGURE 1-1](#) and [FIGURE 1-2](#)). The port uses an RJ-45 connector, with integrated LED for Activity (Amber) and Link (Green). The port auto negotiates to 10/100/1000BASE-T.

**FIGURE 1-2** Ethernet Management Port



### 1.3.3 Serial Console Port (RJ-45)

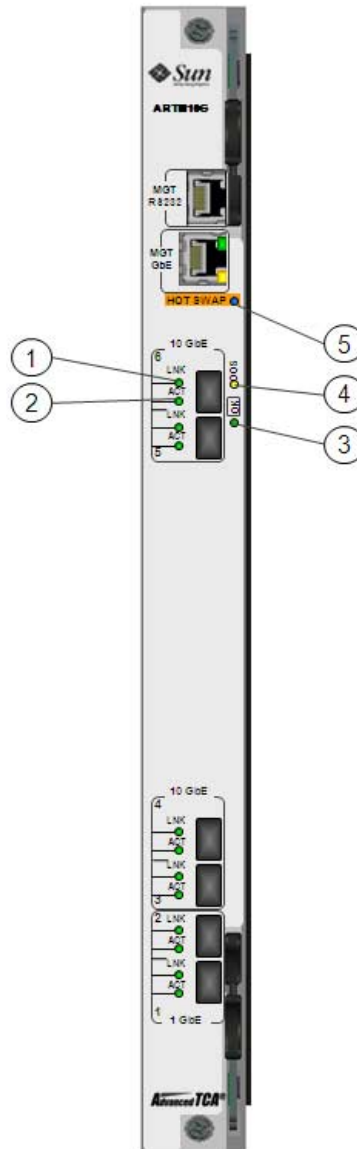
An RS-232 serial console port is provided on the Netra CP32x0 ARTM-10G ([FIGURE 1-1](#)). The port uses an RJ-45 connector.

---

## 1.4 Back Panel LEDs

Several LEDs are located on the back panel of the Netra CP32x0 ARTM-10G, as shown in [FIGURE 1-3](#) and described in [TABLE 1-2](#).

**FIGURE 1-3** Netra CP32x0 ARTM-10GbE Back Panel LEDs



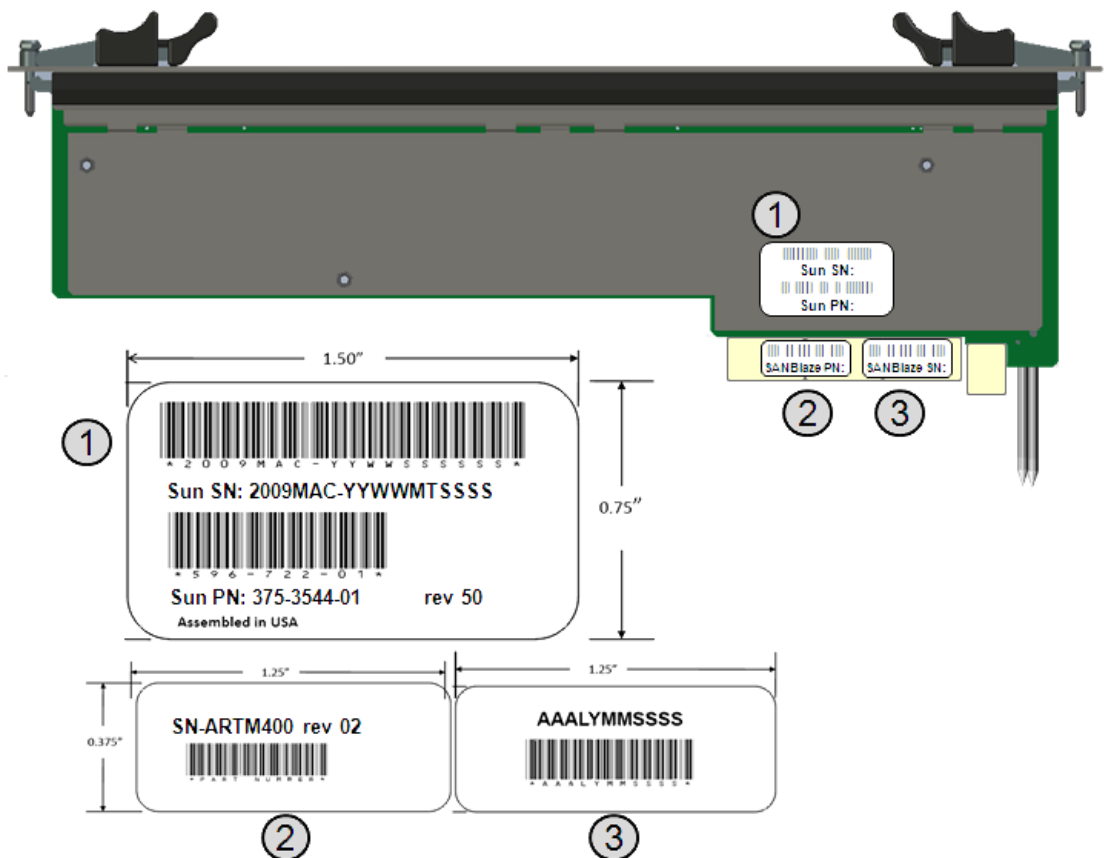
**TABLE 1-2** LEDs and Their Meanings

Figure Callout	LED	Color	Usage Summary	State	Status
1	LNK	GREEN	Ethernet Link If LED is off, it usually means there is a bad connection or cable.	On	Link Active
				Off	Link Inactive
2	ACT	GREEN	Ethernet Activity LED blinks to show data network activity.	Off	No network activity
				Blinking	Network activity
3	OK	GREEN	Module is powered	On	Chassis is powered on
				Off	Chassis is powered off
4	OOS	AMBER	Fault or “Out of Service”	On	Module fault
				Off	No module fault
5	HS	BLUE	Hotswap indicator	On	Management power available to the module, and the module can safely be extracted
				Off	The module is operational and is unsafe to extract
				Slow Blinking	Delay before module is activated
				Fast Blinking	Delay before module is deactivated

## 1.5 Part Number and Serial Number Labels

At manufacturing time, three barcode labels consisting of part number and serial number are affixed to the Netra CP32x0 ARTM-10G as shown in [FIGURE 1-4](#) and described in [TABLE 1-3](#). For proper identification of the ARTM, use these labels to determine the module identity. The barcode labels provide the following information.

**FIGURE 1-4** Netra CP32x0 ARTM-10G Identification Labels



**TABLE 1-3** Netra CP32x0 ARTM-10G Identification Labels

Label	Description
Label 1: Sun Label	SUN S/N Format: YY = year of final assembly WW = calendar week of final assembly NT = multiuse code SSSSSS = sequence number (does NOT reset each week or year) Sun P/N: 375-3544-xx rev xx
Label 2: Contract manufacturer part format	P/N = Assembly part number Rev = Assembly revision (Refer to Bill Of Material)
Label 3: Contract manufacturer serial number	S/N Format: AAA = Assembly number (400) L =Location of manufacturer (S) Y = Calendar year of manufacturer (2008 = 8) MM = Calendar month of manufacturer (March = 03) SSSS = Sequence number (resets each month) (1234)

## 1.6 Software Support

The Netra CP32x0 ARTM-10G supports the following ATCA node boards and operating systems:

- Solaris 10 (SPARC®) Operating System
- Solaris 10 (x86) Operating system
- Wind-River Linux 4.0

Refer to the following documentation for software support information:

- *Netra CP3220 Board Product Notes* (820-1980)
- *Netra CP3220 Board User's Guide* (820-1982)
- *Netra CP3260 Board User's Guide* (820-0457)
- *Netra CP3260 Board Product Notes* (820-0455)

# Installing and Removing the ARTM

---

This chapter contains procedures for unpacking, installing, and removing the Netra CP32x0 10GbE Advanced Rear Transition Module, Dual Port (ARTM-10G).

This chapter contains the following topics:

- [Section 2.1, “Important Information” on page 2-2](#)
- [Section 2.2, “Before Installing or Removing the ARTM” on page 2-5](#)
- [Section 2.3, “Installing the ARTM” on page 2-7](#)
- [Section 2.4, “Upgrading the Firmware” on page 2-10](#)
- [Section 2.5, “Removing the ARTM” on page 2-11](#)
- [Section 2.6, “Verifying the Hardware Installation” on page 2-12](#)

---

## 2.1 Important Information

Before installing the Netra CP32x0 ARTM-10G, check the ARTM's part number to ensure that the correct ARTM is being installed into the system. For information on identifying the ARTM, see [Section 1.5, "Part Number and Serial Number Labels" on page 1-9](#).

The following paragraphs provide important information to ensure compatibility and safety of the ARTMs, other components, and the system.

### 2.1.1 Take Antistatic Precautions



---

**Caution** – When a system is plugged in, energy hazards are present on the midplane. Do not reach into the enclosure.

---



---

**Caution** – Electrostatic discharge (ESD) can damage electronic components. Use an antistatic wrist strap and a conductive foam pad when installing or upgrading any components.

---

After removing the component from its protective wrapper or from the system, place the component flat on a grounded, static-free surface (and, in the case of a board, component side up). Do not slide the component over any surface.

If an ESD station is not available, you can avoid damage resulting from ESD by wearing an antistatic wrist strap (available at electronics stores) that is attached to an active electrical ground. Note that a system chassis might not be grounded if it is unplugged.



---

**Caution** – Dangerous voltages, capable of causing death, are present in this equipment. Use extreme caution when handling, testing, and adjusting equipment.

---

## 2.1.2 Unpack the Equipment Carefully

If the shipping carton is damaged upon receipt, request that the carrier's agent be present during the unpacking and inspection of the equipment.



---

**Caution** – Sun Microsystems has designed special packing material to protect the component during shipping. It is critical that you save the packing material. Shipping the component without the original packing material might void the warranty. Replacement packing material can be purchased from Sun Microsystems.

---

1. Unpack the equipment from the shipping carton.
2. Refer to the packing list and verify that all items are present.
3. Save the packing material for storing and reshipping of equipment.
4. Avoid touching areas of integrated circuitry.  
Static discharge can damage circuits.
5. After removing the product from the packaging, check for any obvious physical damage.
6. Disconnect the chassis from the main power supply before you install the ARTM.

## 2.1.3 Use a Compatible Chassis and Slot

The Netra CP32x0 ARTM-10G can be installed into an ATCA shelf (chassis) with a midplane made for front- and rear-board installations. The ARTM must be installed in the slot directly behind the host ATCA node board. These back-to-back slots have common PINs to enable passing of signals via the Zone 3 connector complex.



---

**Caution** – Connectors on earlier generation node boards do not mate with connectors on the Netra CP32x0 ARTM-10G. Use the Netra CP32x0 ARTM-10G only with compatible node boards. Attempts to install the ARTM with an incompatible node board might damage the node board and/or the ARTM.

---

### 2.1.3.1 Safety Statement

The Netra CP32x0 ARTM-10G is designed to comply with UL60950-1, and is intended to be used with similarly tested AdvancedTCA products that have user documentation detailing installation of an Advanced Mezzanine Card (AMC) module accessories.

### 2.1.3.2 Maximum Power Requirements

Be sure to validate that the chassis and the ARTM slot are cable of providing the ARTM the maximum power requirements of 25W current draw.

## 2.1.4 Use a Compatible Host Board

The Netra CP32x0 ARTM-10G was tested and qualified to operate with the following Sun Netra™ ATCA host boards:

**TABLE 2-1** Compatible Sun Host Boards

ATCA Blade	Description	FW revision
Sun Netra CP3220	10GbE Opteron blade.	Rev. 0.16 or greater
Sun Netra CP3260	UltraSPARC T2 ATCA Blade	OpenBoot™ 4.27.8 or greater

The Netra CP32x0 ARTM-10G is designed to operate with similarly tested AdvancedTCA products.

---

## 2.2 Before Installing or Removing the ARTM

ARTMs can be damaged when improperly handled. Please read and follow the guidelines in this section to protect your equipment.

### 2.2.1 Observe ESD Precautions

Installation persons are urged to use an antistatic wrist strap and a conductive foam pad when installing or upgrading a system. See [Section 2.1.1, “Take Antistatic Precautions”](#) on page 2-2.

### 2.2.2 Inspect for Bent PINs or Other Damage

Bent PINs or loose components can cause damage to the ARTM, host board, the backplane, or other system components. Carefully inspect your ARTM and the backplane for both PIN and component integrity before installation. Our manufacturers take significant steps to ensure there are no bent PINs on the backplane or connector damage to the products prior to leaving our factory. Bent PINs caused by improper installation or by products with damaged connectors could void the warranty for the backplane, boards, and modules.

If a system contains one or more crushed PINs, power off the system and contact your local sales representative to schedule delivery of a replacement chassis assembly.



---

**Caution** – When first installing boards in an empty chassis or onto a carrier card, start at the left of the card cage and work to the right. When inserting or removing a board in a slot adjacent to other boards, use extra caution to avoid damage to the PINs and components located on the primary or secondary sides of the boards.

---

## 2.2.3 Preserve EMI Compliance

During operation, to preserve compliance with applicable standards and regulations for electromagnetic interference (EMI), all front and rear openings on the chassis or board faceplates must be filled with an appropriate card or covered with a filler panel. If the EMI barrier is open, devices might cause or be susceptible to excessive interference.

## 2.2.4 Understand Hotswap

Your ARTM is electrically designed for hot-swap insertion within a fully powered chassis. However hot-swap removal is not supported.



---

**Caution** – Powering down or removing a board before the operating system or other software running on the board has been properly shut down can cause corruption of data or file systems.

---

Prior to board removal, terminate applications or operating systems running on the board. When this task is complete, use the shelf manager to manually shut down (activate/deactivate) the ATCA board, which in turn shuts down the ARTM.

There is a BLUE LED on the rear faceplate. This LED is under software control. When the BLUE LED:

- Is off, the module has been properly recognized.
- Is lit, the module can be safely removed.
- Is blinking, the module state is indeterminate, and the module should not be removed.

Hot-swap compliant components may be installed while the system is powered on. If a module is not hot-swap compliant, you should remove power to the slot or system before installing the component.

## 2.2.5 Verify Slot Usage

Prevent possible damage to components by verifying the proper slot usage for your configuration. See [Section 2.1.3, “Use a Compatible Chassis and Slot” on page 2-3](#).

In most cases, electronic keying (e-Keying) will prevent power on of a module in an incompatible slot. However, as an extra precaution, you should be familiar with the slot purpose.

---

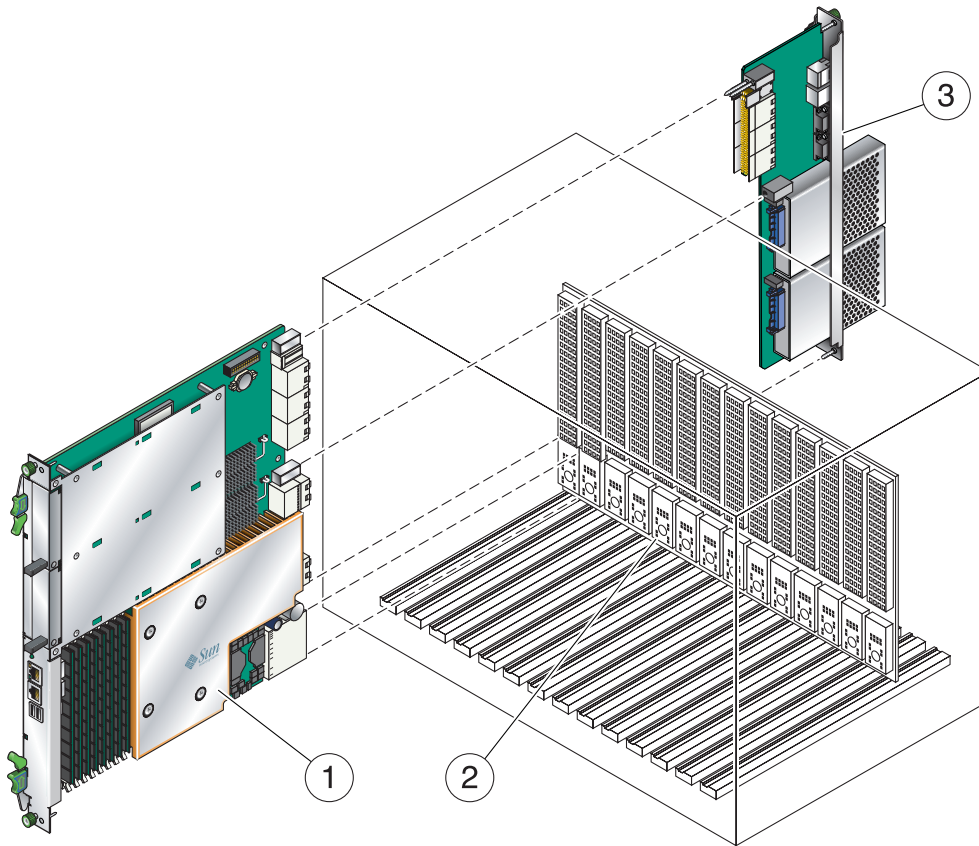
## 2.3 Installing the ARTM

This section describes the procedure for installing the ARTM in a chassis. See [FIGURE 2-1](#) for an example of installing a node board and ARTM into a chassis.

Before you install your ARTM, read all cautions, warnings, and instructions presented in the following sections:

- [Section 2.1.1, “Take Antistatic Precautions” on page 2-2](#)
- [Section 2.1.2, “Unpack the Equipment Carefully” on page 2-3](#)
- [Section 2.1.3, “Use a Compatible Chassis and Slot” on page 2-3](#)
- [Section 2.1.4, “Use a Compatible Host Board” on page 2-4](#)
- [Section 2.2.2, “Inspect for Bent PINs or Other Damage” on page 2-5](#)
- [Section 2.2.3, “Preserve EMI Compliance” on page 2-6](#)
- [Section 2.2.4, “Understand Hotswap” on page 2-6](#)

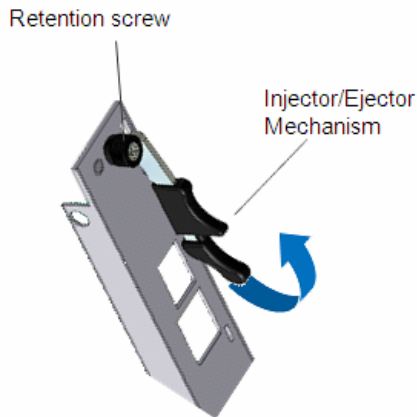
**FIGURE 2-1** Installing a Node Board and ARTM Into an ATCA Chassis



## ▼ To Install an ARTM

1. Take the necessary antistatic precautions while unpacking and handling the ARTM and system.
2. At the back of the system, choose a compatible slot for the ARTM.  
ARTMs must be installed in-line behind the accompanying host board. For example, if the accompanying host board is going to be installed in slot 3, its ARTM must be installed at the back of the system in slot 3.
3. Remove the slot filler panel from the selected host board slot, if necessary.
4. Prepare the module by loosening the locking screws and opening the injector/ejector latch at the top of the module as shown in [FIGURE 2-2](#).

**FIGURE 2-2** Injector/Ejector Latch and Locking Screw



5. Look into the enclosure to verify correct alignment of the rails in the guides.
6. Align the edges of the module with the card-cage rail guides in the appropriate slot.
7. Taking care to keep the module aligned in the guides, while holding the module handles, apply equal and steady pressure to slide the module in until the injector/ejector mechanism engages the retention bars.



---

**Caution** – Do not exert unnecessary pressure on the faceplate.

---

8. Position your thumbs at the top and bottom of the module; simultaneously push in the module and rotate the injector/ejector mechanisms inward to their closed position to seat and secure the ARTM.



---

**Caution** – Do not force the ARTM into the slot.

---

9. Tighten the two module retention screws to secure the module into the chassis.
10. Power on the system, if necessary.

Refer to your system manual for instructions on correctly powering on the system. Once power is applied to the chassis, the internal MMC controller runs a self-test that runs for approximately 10 seconds. Upon a successful power up self-test, the BLUE LED (hotswap) will blink then turn off, indicating that the module has been placed in operation.

---

## 2.4 Upgrading the Firmware

The Netra CP32x0 ARTM-10G is loaded with `fcode` and firmware prior to shipping.

All `fcode` and firmware code is stored in a single EEPROM Flash device located on the ARTM. The code provides the ATCA CPU instructions for operating the Sun Multithreaded 10GbE Networking Technology located on the ARTM.

When updates are available, new releases are offered online to improve the functionality or performance of the ARTM.

After you install the ARTM, determine if any updates are available, and upgrade the firmware on your ARTM.

For detailed upgrading instructions, refer to the *Netra CP32x0 Advanced Rear Transition Module Product Notes* (820-3261), available at the following web site:

<http://docs.sun.com/app/docs/prod/cp32x0.10gbee?l=en#hic>

---

## 2.5 Removing the ARTM

The Netra CP32x0 ARTM-10G ARTM is hot-swappable and can be removed from the chassis without powering down its associated host board. This section describes the hot-swap procedure for removing a module from a chassis.

Before you remove your module, read all cautions, warnings, and instructions presented in [Section 2.2.4, “Understand Hotswap”](#) on page 2-6.

If the chassis is not hot-swap compliant, remove power to the slot or system before removing the ARTM.

### ▼ To Remove the ARTM

1. Loosen the locking screws on the module.

2. Visually inspect the BLUE LED.

It will illuminate a steady, non-blinking BLUE color when the module is properly prepared for removal.

3. Rotate the top ejector handle to the half way (HW) position.



---

**Caution** – Do not remove the module yet. Powering down or removing a module before the operating system or other software running on the board has been properly shut down may cause corruption of data or file systems.

---

- If your host board is running hot-swap software, the action of rotating the ejector lever will start the shutdown process on the module. The software will slowly blink the BLUE LED, indicating the module is in the process of being de-activated.
- If your board or system is not running hot-swap software, the BLUE LED might illuminate without regard to software processes still running on the board. This scenario is likely the case when the BLUE LED continues to blink for more than 10-15 seconds after you rotate the ejector handle of the module. Be sure to manually shut down applications or operating systems running on the board prior to board removal.

4. If you need to manually shut down applications, execute the following command from the shelf manager.

```
clia deactivate xx 1
```

Where *xx* is the slot number.

When the module has been deactivated, the BLUE LED will illuminate steady.

5. When the BLUE LED is steady, extract the module by pulling on the module handles.
6. Carefully remove the module from the chassis.
7. If the slot is to remain empty, install a filler panel in the slot.

---

## 2.6 Verifying the Hardware Installation

This section provides instructions for verifying the installation of the Netra CP32x0 ARTM-10G.

### ▼ To Verify Hardware Installation With SPARC Host

1. Power on the system.
2. When the banner appears, press the Stop-A key sequence to interrupt the boot process and display the OpenBoot (ok) prompt.

### 3. List the network devices on your system, as in the following example.

```
ok show-nets
a) /pci@0/pci@0/pci@9/network@0,1
b) /pci@0/pci@0/pci@9/network@0
c) /pci@0/pci@0/pci@8/network@0,3
d) /pci@0/pci@0/pci@8/network@0,2
e) /pci@0/pci@0/pci@8/network@0,1
f) /pci@0/pci@0/pci@8/network@0
g) /pci@0/pci@0/pci@2/network@0,1
h) /pci@0/pci@0/pci@2/network@0
i) /pci@0/pci@0/pci@1/network@0,1
j) /pci@0/pci@0/pci@1/network@0
m) MORE SELECTIONS
q) NO SELECTIONS
ok cd /pci@0/pci@0/pci@8/network@0,3
```

Usually /pci@0/pci@0/pci@8 or /pci@0/pci@0/pci@9 corresponds to PCIe slots, so look at those devices first. Also, the ARTM is usually the only device that will display four ports.

---

**Note** – If you do not see the device listed, check that the ARTM is properly seated. If necessary, reinstall the ARTM.

---

### 4. Use the .properties command to display a list of device properties.

Checking the .properties output of a device is the surest way of identifying the ports on the ARTM. For this ARTM, your output will be similar to the following.

```
ok .properties
assigned-addresses      82060310 00000000 04000000 00000000
01000000
                        82060318 00000000 00430000 00000000 00008000
                        82060320 00000000 00438000 00000000 00008000
                        82060330 00000000 00800000 00000000 00100000
local-mac-address       00 14 4f 8c 0c db
phy-type                mif
reg                     00060300 00000000 00000000 00000000 00000000
                        03060310 00000000 00000000 00000000 01000000
                        03060318 00000000 00000000 00000000 00008000
                        03060320 00000000 00000000 00000000 00008000
                        02060330 00000000 00000000 00000000 00100000
version                 ARTM 10G/1G NIC PXE1.48 FCode 3.13 08/01/08
board-model             375-3544-01
model                  SUNW,pcie-artm
compatible              pciex108e,abcd.108e.0.1
                        pciex108e,abcd.108e.0
```

```

pciex108e,abcd.1
pciex108e,abcd
pciexclass,020000
pciexclass,0200
address-bits      00000030
max-frame-size    00002400
network-interface-type ethernet
device_type       network
name              network
fcode-rom-offset  00007600
interrupts        00000004
cache-line-size   00000010
class-code        00020000
subsystem-vendor-id 0000108e
revision-id       00000001
device-id         0000abcd
vendor-id 0000108e

```

5. Type the following when you are done looking at the .properties values:

```
ok device-end
```

## ▼ To Verify Hardware Installation With x86 Host

### 1. Inspect the BLUE hotswap LED on the ARTM.

If the module is installed correctly, the LED will not be illuminated.

### 2. Open a Linux console window from the host board.

```
# lspci | grep Ethernet
```

This command displays a list of PCI devices containing Ethernet in the description, such as the following example.

```

00:08.0 Bridge: nVidia Corporation MCP55 Ethernet (rev a3)
00:09.0 Bridge: nVidia Corporation MCP55 Ethernet (rev a3)
02:00.0 Ethernet controller: Sun Microsystems Computer Corp.
Unknown device abcd (rev 01)
02:00.1 Ethernet controller: Sun Microsystems Computer Corp.
Unknown device abcd (rev 01)
02:00.2 Ethernet controller: Sun Microsystems Computer Corp.
Unknown device abcd (rev 01)
02:00.3 Ethernet controller: Sun Microsystems Computer Corp.
Unknown device abcd (rev 01)
07:04.0 Ethernet controller: Broadcom Corporation NetXtreme
BCM5715 Gigabit Ethernet (rev a3)

```

```
07:04.1 Ethernet controller: Broadcom Corporation NetXtreme
BCM5715 Gigabit Ethernet (rev a3)
09:00.0 Ethernet controller: Sun Microsystems Computer Corp.
Unknown device abcd (rev 01)
09:00.1 Ethernet controller: Sun Microsystems Computer Corp.
Unknown device abcd (rev 01)
09:00.2 Ethernet controller: Sun Microsystems Computer Corp.
Unknown device abcd (rev 01)
09:00.3 Ethernet controller: Sun Microsystems Computer Corp.
Unknown device abcd (rev 01)
```

The host board and the ARTM use the same four-port Ethernet controller. If these are installed correctly, the output will show eight Sun Ethernet instances.

## ▼ To Reboot the System

After verifying the ARTM installation, perform a reconfiguration boot on your system.

- **Type the following**

```
ok boot -r
```



# Driver Installation and Configuration

---

This chapter explains how to download and install the `nxge` driver used to operate certain ports on the Netra CP32x0 10GbE Ethernet Advanced Rear Transition Module, Dual Port (ARTM-10G). More specifically, the `nxge` driver facilitates use of SFP ports 1, 2, 3 and 4.

This chapter contains the following topics:

- [Section 3.1, “Installing the `nxge` Driver” on page 3-2](#)
- [Section 3.2, “Configuring the `nxge` Driver” on page 3-6](#)
- [Section 3.3, “Tuning Performance on a Linux Platform” on page 3-20](#)
- [Section 3.4, “Enabling Host Board PCI-Express Communication” on page 3-21](#)
- [Section 3.5, “Removing the `nxge` Driver” on page 3-22](#)

---

## 3.1 Installing the nxge Driver

The `nxge` driver is the Gigabit Ethernet driver that operates the Netra CP32x0 ARTM-10G in a Solaris or Linux environment. The `nxge` driver is managed by the `dladm` command-line utility, which allows VLANs to be defined on top of `nxge` instances and for `nxge` instances to be aggregated. See the `dladm(1M)` man page for more details on configuring the data-link interfaces and link aggregations.

A driver must be loaded on your system to operate the ARTM-10G. All OS's use a variant of a driver design called `nxge`. New versions of this driver must be used to recognize and operate the new ARTM-10G hardware.

The `nxge` device driver automatically negotiates maximum link speed up to 10 Gbit/sec and conforms to the IEEE 802.3 Ethernet standard. Each interface is allocated eight receive DMA channels and twelve transmit DMA channels to enable parallel processing of the packets.

Ports 5 and 6 of the ARTM-10G are direct pass-through connections to the Zone 3 connector, and require no additional drivers for the ARTM installation. However, by default, these ports are off (disabled), and will not pass traffic unless the front ATCA blade is configured to redirect traffic to these ports.

---

**Note** – At this time, there is not a command or process available to route traffic to these ports. Refer to the *Netra CP32x0 Advanced Rear Transition Module Product Notes* (820-3261).

---

## 3.1.1 Download and Install for Solaris Operating Systems

The `nxge` device driver is *required* for systems using the SPARC-based or x86-based Solaris Operating System.

### ▼ To Download and Install the Driver for Solaris Operating Systems

1. **Locate and download the patch with the `nxge` device driver software from the following web site:**

<http://www.sun.com/download>

Refer to the README file and the *Netra CP32x0 Advanced Rear Transition Module Product Notes* (820-3261) for the most current information.

2. **Install the driver onto your system.**

- a. **Uncompress or unzip the patch into the target directory.**

- b. **Type the following commands to install the patch into the target directory:**

```
% cd <patch_dir>
% patchadd .
```

3. **Verify that the `nxge` driver is installed on the system.**

```
Sun x8 10G/1G Ethernet Adapter Driver(i386)
1.0,REV=2006.12.05.10.0 Copyright 2006 Sun Microsystems,
Inc. All rights reserved. Use is subject to license terms.
## Executing checkinstall script. Using as the package base
directory.
## Processing package information.
## Processing system information.
4 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs. This package contains
scripts which will be executed with super-user permission
during the process of installing this package.
Do you want to continue with the installation of [y,n,?] y
Installing Sun x8 10G/1G Ethernet Adapter Driver as
## Installing part 1 of 1.
```

```
/kernel/drv/amd64/nxge [ verifying class ]  
# Executing postinstall script.  
# Installation of was successful.
```

## 3.1.2 Download and Install for Linux Operating Systems

The nxge device driver is *required* for systems using the Linux operating system.

### ▼ To Download and Install the Driver for Linux Operating Systems

1. **Locate and download the patch with the nxge device driver software from the following web site:**

<http://www.sun.com/download>

Refer to the README file and the *Netra CP32x0 Advanced Rear Transition Module Product Notes* (820-3261) for the most current information.

2. **Choose the rpm that matches your OS distribution, for example:**

```
nxge-x.x-x.rpm
```

Where x.x-x is the version number.

3. **List the network interfaces that exist prior to adding the driver package:**

```
# ifconfig -a |grep eth  
eth0      Link encap:Ethernet  HWaddr 00:14:4F:91:94:6E  
eth1      Link encap:Ethernet  HWaddr 00:14:4F:91:94:6F  
eth2      Link encap:Ethernet  HWaddr 00:14:4F:91:94:70  
eth3      Link encap:Ethernet  HWaddr 00:14:4F:91:94:71
```

4. **Install the driver on Linux.**

Where x is the most current version available.

```
# rpm -ivh nxge-x.x-x.rpm  
Preparing... #####  
1:nxge-2.6.21-bigsmg #####
```

5. List the network interfaces again, and note the new network interface instances corresponding to the Netra CP32x0 ARTM-10G:

#	ifconfig -a	grep eth
eth0	Link encap:Ethernet	HWaddr 00:14:4F:91:94:6E
eth1	Link encap:Ethernet	HWaddr 00:14:4F:91:94:6F
eth2	Link encap:Ethernet	HWaddr 00:14:4F:91:94:70
eth3	Link encap:Ethernet	HWaddr 00:14:4F:91:94:71
eth4	Link encap:Ethernet	HWaddr 00:14:4F:91:94:72
eth5	Link encap:Ethernet	HWaddr 00:14:4F:91:94:73
eth6	Link encap:Ethernet	HWaddr 00:14:4F:91:94:74
eth7	Link encap:Ethernet	HWaddr 00:14:4F:91:94:75
eth9	Link encap:Ethernet	HWaddr 00:11:0D:D0:04:C0
eth10	Link encap:Ethernet	HWaddr 00:11:0D:D0:04:C2
eth11	Link encap:Ethernet	HWaddr 00:11:0D:D0:04:C3
eth12	Link encap:Ethernet	HWaddr 00:11:0D:D0:04:C1

The additional Netra CP32x0 ARTM-10G instances, eth9 through eth12, are shown. The MAC address of the ARTM resource is easily identified as 00:11:0d:xx:xx:xx.

6. Add the nxge interfaces to the `/etc/modules.conf` file to automatically load the driver after system reboot:

alias eth9 nxge
alias eth10 nxge
alias eth11 nxge
alias eth12 nxge

7. Check the parameter configurations that apply to the nxge driver.

# <b>ethtool -i eth9</b>
driver: nxge
version: 2.1.0
firmware-version: ARTM PXE1.4? FCode 3.14
bus-info: 0000:02:00.0
# <b>ethtool eth9</b>
Settings for eth9:
Supported ports: [ FIBRE ]
Supported link modes:
Supports auto-negotiation: No
Advertised link modes: Not reported
Advertised auto-negotiation: No
Speed: Unknown! (10000)
Duplex: Full
Port: FIBRE
PHYAD: 16
Transceiver: internal

```
Auto-negotiation: off
Current message level: 0x000000ff (255)
Link detected: yes
```

---

## 3.2 Configuring the nxge Driver

The Netra CP32x0 ARTM-10G includes six ports, which are controlled by the nxge driver:

- Two 1-Gigabit full duplex networking interfaces (ports 1 and 2)
- Two 10-Gigabit full-duplex network interfaces (ports 3 and 4)
- Two 10-Gigabit full-duplex network interfaces driven by ATCA front board resource (ports 5 and 6)

This section provides procedures for configuring nxge driver attributes and parameters, based on your OS.

- [Section 3.2.1, “Configuring nxge Driver Parameters on a Solaris Platform” on page 3-6](#)
- [Section 3.2.2, “Configuring Driver Parameters on a Linux Platform” on page 3-15](#)

### 3.2.1 Configuring nxge Driver Parameters on a Solaris Platform

You can configure and modify nxge device driver parameters in two ways:

- Using the `ndd` utility
- Using the `nxge.conf` file

If you use the `ndd` utility, the parameters are valid only until you reboot the system. This method is good for testing parameter settings.

### 3.2.1.1 Modifying Parameters Using the `ndd` Utility

Use the `ndd` utility to configure parameters that are valid until you reboot the system (non-persistent configurations).

The following sections describe how to use the `ndd` utility to modify (with the `-set` option) or display (without the `-set` option) parameters for each `nxge` device.

The `ndd` utility can be invoked in either of two modes:

- Interactive menu mode. In this mode, the utility displays a menu of commands, and prompts you for each input value. You can set and get multiple parameters before exiting the utility.
- Non-interactive mode. In this mode, invoke the utility to execute a single command. Once the command is executed, the utility exits and returns to the command prompt.

Refer to the `ndd(1M)` man page for more information.

#### ▼ To Specify Device Instances

Before you use the `ndd` utility to get or set a parameter for an `nxge` device, you must specify the device instance for the utility.

- **Check the `/etc/path_to_inst` file to identify the instance associated with a device.**

```
# grep nxge /etc/path_to_inst
"/pci@7c0/pci@0/pci@9/network@0" 0 "nxge"
"/pci@7c0/pci@0/pci@9/network@0,1" 1 "nxge"
"/pci@0/pci@0/pci@8/network@0" 2 "nxge"
"/pci@0/pci@0/pci@8/network@0,1" 3 "nxge"
"/pci@0/pci@0/pci@8/network@0,2" 4 "nxge"
"/pci@0/pci@0/pci@8/network@0,3" 5 "nxge"
"/pci@0/pci@0/pci@8/ethernet@0" 6 "nxge"
"/pci@0/pci@0/pci@8/ethernet@0,1" 7 "nxge"
"/pci@0/pci@0/pci@8/ethernet@0,2" 8 "nxge"
"/pci@0/pci@0/pci@8/ethernet@0,3" 9 "nxge"
"/niu@80/network@0" 0 "nxge"
"/niu@80/network@1" 1 "nxge"
```

## ▼ To Modify Parameters in Interactive Mode

1. Type **? to see all the parameters supported by the nxge driver.**

```
# ndd /dev/nxge0
name to get/set ? ?
? (read only)
function_number (read only)
fw_version (read only)
adv_autoneg_cap (read and write)
adv_10gfdx_cap (read and write)
adv_1000fdx_cap (read and write)
adv_100fdx_cap (read and write)
adv_10fdx_cap (read and write)
adv_pause_cap (read and write)
accept_jumbo (read and write)
rxdma_intr_time (read and write)
rxdma_intr_pkts (read and write)
class_opt_ipv4_tcp (read and write)
class_opt_ipv4_udp (read and write)
class_opt_ipv4_ah (read and write)
class_opt_ipv4_sctp (read and write)
class_opt_ipv6_tcp (read and write)
class_opt_ipv6_udp (read and write)
class_opt_ipv6_ah (read and write)
class_opt_ipv6_sctp (read and write)
```

2. Modify a parameter value by specifying **ndd /dev/nxge<x>**, such as the following example.

```
# ndd /dev/nxge0
name to get/set?
```

3. Enter the parameter name.

After you enter the parameter name, the ndd utility prompts you for the parameter value.

4. Enter the parameter value or ? for more information.

Refer to the ndd(1M) man page for more information.

## ▼ To Modify Parameter Values in Non-Interactive Mode

### 1. Invoke the `ndd` utility with or without the `-set` option.

- When you invoke the `ndd` utility with the `-set` option, the utility passes value, which must be specified, down to the named `/dev/nxgedriver-instance`, and assigns the value to the parameter:

```
# ndd -set /dev/nxge<x> <parameter-value>
```

Where *x* is the driver instance, for example `/dev/nxge0`, `/dev/nxge1`, and *parameter-value* is a valid input or option.

- When you omit the `-set` option, the utility queries the named driver instance, retrieves the value associated with the specified parameter, and prints the value:

```
# ndd /dev/nxge<x>
```

Where *x* is the driver instance.

### 2. Display the value of a parameter by specifying the parameter name and omitting the value.

### 3. Modify a parameter value.

Refer to the `ndd(1M)` man page for more information.

## 3.2.1.2 Modifying Parameters Using the `nxge.conf` File

Use an `nxge.conf` file when you want to establish persistent parameter values for a device in the system.

For more information, refer to the `prtconf(1M)` and `driver.conf(4)` man pages.

## ▼ To Modify Parameters Using `nxge.conf` File

1. Obtain the hardware path names for the `nxge` devices in the device tree.
2. Check the `/etc/driver_aliases` file to identify the name associated with a device.

```
# grep nxge /etc/driver_aliases
nxge "pciex108e,abcd"
```

3. Locate the path names and the associated instance numbers in the `/etc/path_to_inst` file.

```
# grep nxge/etc/path_to_inst
"/pci@780/pci@0/pci@8/network@0" 0 "nxge"
"/pci@780/pci@0/pci@8/network@0,1" 1 "nxge"
```

This command returns two `nxge` interfaces, and for each, includes three descriptive parameters:

- Instance pathname in quotes
- Instance Number (integer, no quotes)
- Instance driver name in quotes

The `nxge.conf` will likely contain information for several `nxge` interfaces. The output from steps 1 and 2 is used to specify the `nxge` interface, if needed. In this example:

```
name = "pciex108e,abcd"
parent = "/pci@780/pci@0/pci@8/network@0"
unit-address = "0"
```

4. Edit the `/kernel/drv/nxge.conf` file to set `nxge` parameters.

For an output of all the parameters you can modify, see [CODE EXAMPLE 3-1](#).

**5. To set persistence for all parameters, specify the driver parameter properties for each device by creating an `nxge.conf` file located in the following directory:**

- For SPARC-based host, `/platform/sunv4/kernel/drv/nxge.conf`
- For x86-based host, `/kernel/drv/nxge.conf`

Edits to `/kernel/drv/nxge.conf` file may be made with any ASCII, text editor.

To apply an edit to *all* ports, entries are listed with no specific node prefix. In this example, all modules are being set for load balancing RX traffic based on the IP source address. The default value is F80, indicating RX load balancing based on IP 5-tuple. Notice the semicolon at the end of the last parameter.

```
class-opt-ipv4-tcp = 100;  
class-opt-ipv4-udp = 100;
```

**6. To set persistence for a specific parameter, edit the `nxge.conf` file to add the specific parameter values.**

These parameters are reloaded at each reboot.

To apply an edit to specific ports, an entry is prefixed with the node/interface specific information as shown:

```
name = "pciex108e,abcd" parent = "/pci@780/pci@0/pci@8/"  
unitaddress = "0" class-opt-ipv4-tcp = 0x100;  
name = "pciex108e,abcd" parent = "/pci@7c0/pci@0/pci@9/"  
unitaddress = "0" class-opt-ipv4-tcp = 0x40;
```

**7. Save the `nxge.conf` file.**

The following [CODE EXAMPLE 3-1](#) provides an output of all the parameters modifiable via the `nxge.conf` file. See also [Chapter 4](#) for instructions on how to configure the network settings.

**CODE EXAMPLE 3-1** Driver Parameters Modifiable via `nxge.conf` File

```
#  
#-----Link Configuration -----  
# The link parameters depend on the type of the card  
# and the port.  
# 10-Gigabit related parameters ( i.e adv_10gfdx_cap)  
# apply only to 10gigabit ports.  
# Half duplex is not supported on any NIU card.  
#  
# adv-autoneg-cap  
# Advertise auto-negotiation capability.  
# default is 1  
# adv-autoneg-cap = 1;  
#  
# adv_10gfdx_cap
```

**CODE EXAMPLE 3-1** Driver Parameters Modifiable via `nxge.conf` File (Continued)

```
# Advertise 10gbps Full duplex capability.
# default is 1
# adv_10gfdx_cap = 1;
#
# adv_1000fdx_cap
# Advertise 1gbps Full duplex capability.
# default is 1
# adv_1000fdx_cap = 1;
#
# adv_100fdx_cap
# Advertise 100mbps Full duplex capability.
# default is 1
# adv_100fdx_cap = 1;
#
# adv_10fdx_cap
# Advertise 10mbps Full duplex capability.
# default is 1
# adv_10fdx_cap = 1;
#
# adv_asmpause_cap
# Advertise Asymmetric pause capability.
# default is 0
# adv_asmpause_cap = 0;
#
# adv_pause_cap
# Advertise pause capability.
# default is 1
# adv_pause_cap = 1;
#
#
#----- Jumbo frame support -----
# To enable jumbo support for all nxge interfaces,
# accept_jumbo = 1;
#
# To disable jumbo support for all nxge interfaces,
# accept_jumbo = 0;
#
# Default is 0. See the example at the end of this file for
# enabling or disabling jumbo for a particular nxge interface.
#
#
#----- Receive DMA Configuration -----
#
# rxdma-intr-time
# Interrupts after this number of NIU hardware ticks have
# elapsed since the last packet was received.
# A value of zero means no time blanking (Default = 8).
```

**CODE EXAMPLE 3-1** Driver Parameters Modifiable via `nxge.conf` File (*Continued*)

```
#
# rxdma-intr-pkts
# Interrupt after this number of packets have arrived since
# the last packet was serviced. A value of zero indicates
# no packet blanking (Default = 20).
#
# Default Interrupt Blanking parameters.
#
# rxdma-intr-time = 8;
# rxdma-intr-pkts = 20;
#
#
#----- Classification and Load Distribution Configuration -----
#
# class-opt-****-***
# These variables define how each IP class is configured.
# Configuration options range from whether TCAM lookup ie
# is enabled to flow hash generation.
# This parameters also control how the flow template is
# constructed and how packet is distributed within RDC
# groups.
#
# supported classes:
# class-opt-ipv4-tcp class-opt-ipv4-udp class-opt-ipv4-sctp
# class-opt-ipv4-ah class-opt-ipv6-tcp class-opt-ipv6-udp
# class-opt-ipv6-sctp class-opt-ipv6-ah
#
# Configuration bits (The following bits will be decoded
# by the driver as hex format).
#
# 0010: use MAC Port (for flow key)
# 0020: use L2DA (for flow key)
# 0040: use VLAN (for flow key)
# 0080: use proto (for flow key)
# 0100: use IP src addr (for flow key)
# 0200: use IP dest addr (for flow key)
# 0400: use Src Port (for flow key)
# 0800: use Dest Port (for flow key)
#
# class-opt-ipv4-tcp = fe0;
The following parameters operate on a per port basis and can be set
using the /kernel/drv/nxge.conf file.
#
# ----- How to set parameters for a particular interface -----
# The example below shows how to locate the device path and set a
# parameter for a particular nxge interface. (Using jumbo support
as
```

**CODE EXAMPLE 3-1** Driver Parameters Modifiable via `nxge.conf` File (*Continued*)

```
# an example)
#
# Use the following command to find out the device paths for nxge,
# more /etc/path_to_inst | grep nxge
#
# For example, if you see,
# "/pci@7c0/pci@0/pci@8/network@0" 0 "nxge"
# "/pci@7c0/pci@0/pci@8/network@0,1" 1 "nxge"
# "/pci@7c0/pci@0/pci@8/network@0,2" 2 "nxge"
# "/pci@7c0/pci@0/pci@8/network@0,3" 3 "nxge"
#
# then you can enable jumbo for ports 0 and 1 and disable jumbo for
ports 2
# and 3 as follows,
#
# name = "pciex108e,abcd" parent = "/pci@7c0/pci@0/pci@8/"
unit-address
= "0"
# accept_jumbo = 1;
# name = "pciex108e,abcd" parent = "/pci@7c0/pci@0/pci@8/"
unit-address
= "0,1"
# accept_jumbo = 1;
# name = "pciex108e,abcd" parent = "/pci@7c0/pci@0/pci@8/"
unit-address
= "0,2"
# accept_jumbo = 0;
# name = "pciex108e,abcd" parent = "/pci@7c0/pci@0/pci@8/"
unit-address
= "0,3"
# accept_jumbo = 0;
```

## 3.2.2 Configuring Driver Parameters on a Linux Platform

Use the `ethtool` utility or the `configtool` utility to modify `nxge` driver parameters on a Linux platform.

### 3.2.2.1 Modifying Parameters Using the `ethtool` Utility

This section describes `ethtool` commands for modifying driver parameters. Refer to the `ethtool(1M)` man page for more information and detailed usage examples.

#### ▼ To Determine Modifiable Parameters

- Use the `ethtool` utility to list all driver parameters that can be modified.

**CODE EXAMPLE 3-2** Listing Modifiable Driver Parameters on a Linux Platform

```
# ethtool -help eth4
ethtool version 3
Usage:
ethtool DEVNAME
ethtool -a DEVNAME
ethtool -A DEVNAME \
[ autoneg on|off ] \
[ rx on|off ] \
[ tx on|off ]
ethtool -c DEVNAME
ethtool -C DEVNAME \
[adaptive-rx on|off] \
[adaptive-tx on|off] \
[rx-usecs N] \
[rx-frames N] \
[rx-usecs-irq N] \
[rx-frames-irq N] \
[tx-usecs N] \
[tx-frames N] \
[tx-usecs-irq N] \
[tx-frames-irq N] \
[stats-block-usecs N] \
[pkt-rate-low N] \
[rx-usecs-low N] \
[rx-frames-low N] \
[tx-usecs-low N] \
[tx-frames-low N] \
[pkt-rate-high N] \
```

**CODE EXAMPLE 3-2** Listing Modifiable Driver Parameters on a Linux Platform

```
[rx-usecs-high N] \  
[rx-frames-high N] \  
[tx-usecs-high N] \  
[tx-frames-high N] \  
[sample-interval N]  
ethtool -g DEVNAME  
ethtool -G DEVNAME \  
[ rx N ] \  
[ rx-mini N ] \  
[ rx-jumbo N ] \  
[ tx N ]  
ethtool -i DEVNAME  
ethtool -d DEVNAME  
ethtool -e DEVNAME \  
[ raw on|off ] \  
[ offset N ] \  
[ length N ]  
ethtool -E DEVNAME \  
[ magic N ] \  
[ offset N ] \  
[ value N ]  
ethtool -k DEVNAME  
ethtool -K DEVNAME \  
[ rx on|off ] \  
[ tx on|off ] \  
[ sg on|off ] \  
[ tso on|off ]  
ethtool -r DEVNAME  
ethtool -p DEVNAME [ %d ]  
ethtool -t DEVNAME [online|(offline)]  
ethtool -s DEVNAME \  
[ speed 10|100|1000 ] \  
[ duplex half|full ] \  
[ port tp|au|bnc|mii|fibre ] \  
[ autoneg on|off ] \  
[ phyad %d ] \  
[ xcvr internal|external ] \  
[ wol p|u|m|b|a|g|s|d... ] \  
[ sopass %x:%x:%x:%x:%x ] \  
[ msglvl %d ]  
ethtool -S DEVNAME
```

Following are common parameters that can be changed:

**CODE EXAMPLE 3-3** Common Parameters That are Modifiable on a Linux Platform

```
# ethtool -c eth8
Coalesce parameters for eth8:
Adaptive RX: off TX: off
stats-block-usecs: 0
sample-interval: 0
pkt-rate-low: 0
pkt-rate-high: 0
rx-usecs: 8
rx-frames: 512
rx-usecs-irq: 0
rx-frames-irq: 512
tx-usecs: 0
tx-frames: 0
tx-usecs-irq: 0
tx-frames-irq: 0
rx-usecs-low: 0
rx-frame-low: 0
tx-usecs-low: 0
tx-frame-low: 0
rx-usecs-high: 0
rx-frame-high: 0
tx-usecs-high: 0
tx-frame--high: 0
```

The parameters `rx-usecs` and `rx-frames` control the RX interrupt rate per RX DMA channel. RX interrupt will be generated after `rx-frames` have been received or after `rxusecs` time interval, if fewer than `rx-frames` have been received within the interval.

- For low latency applications, set `rx-usecs` to a smaller value.
- For bulk traffic, use larger values of `rx-usecs` and control the rate with `rx-frames`.

The `rx-frames-irq` controls the maximum number of RX packets processed with a single RX interrupt.

## ▼ To Change RX Coalesce Parameters

- Use the **ethtool -C** command.

**CODE EXAMPLE 3-4** Changing RX Coalesce Parameters

```
# ethtool -C eth4 rx-usecs 20
# ethtool -c eth4
Coalesce parameters for eth4:
Adaptive RX: off TX: off
stats-block-usecs: 0
sample-interval: 0
pkt-rate-low: 0
pkt-rate-high: 0
rx-usecs: 20
rx-frames: 512
rx-usecs-irq: 0
rx-frames-irq: 512
tx-usecs: 0
tx-frames: 0
tx-usecs-irq: 0
tx-frames-irq: 0
rx-usecs-low: 0
rx-frame-low: 0
tx-usecs-low: 0
tx-frame-low: 0
rx-usecs-high: 0
rx-frame-high: 0
tx-usecs-high: 0
tx-frame-high: 0
```

## ▼ To Obtain the Status of L4 Hardware

- Use the **ethtool -k** command.

**CODE EXAMPLE 3-5** Obtaining Status of L4 Hardware

```
# ethtool -k eth4
Offload parameters for eth4:
Cannot get device tcp segmentation offload settings: Operation not
supported
rx-checksumming: on
tx-checksumming: on
scatter-gather: off
tcp segmentation offload: off
```

### 3.2.2.2 Modifying Parameters Using configtool Utility

This section describes how to use the configtool utility to modify driver parameters.

#### ▼ To Obtain a List of Modifiable Parameters

- Use the `nxge_config if-name get` command.

**CODE EXAMPLE 3-6** Obtaining a List of Tunable Driver Parameters on a Linux Platform

```
# /usr/local/bin/nxge_config eth4 get
The tunable parameters exported by this device are:
class_opt_ipv4_tcp Read-Write
class_opt_ipv4_udp Read-Write
class_opt_ipv4_ah Read-Write
class_opt_ipv4_sctp Read-Write
class_opt_ipv6_tcp Read-Write
class_opt_ipv6_udp Read-Write
class_opt_ipv6_ah Read-Write
class_opt_ipv6_sctp Read-Write
```

These classification variables define how each IP class is configured. These parameters also control how the flow template is constructed and how packets are distributed within RDC groups.

```
Configuration bits:
0x0010: use MAC Port (for flow key)
0x0020: use L2DA (for flow key)
0x0040: use VLAN (for flow key)
0x0080: use proto (for flow key)
0x0100: use IP src addr (for flow key)
0x0200: use IP dest addr (for flow key)
0x0400: use Src Port (for flow key)
0x0800: use Dest Port (for flow key)
```

---

**Note** – The classification variables are modified on an ARTM basis. That is, if any of these variables is modified for one port, the change carries over to all other ports of the ARTM.

---

### ▼ To Display a Parameter

- Use the `nxge_config <if-name> get param-name>` command.

```
# /usr/local/bin/nxge_config eth4 get class_opt_ipv4_udp
class_opt_ipv4_udp 0xfe3
```

### ▼ To Modify a Specific Parameter

- Use the `/usr/local/bin/nxge_config <if_name> set param_name>` command.

```
# /usr/local/bin/nxge_config eth4 set class_opt_ipv4_tcp 0xfe0
```

---

## 3.3 Tuning Performance on a Linux Platform

The following procedure improves the performance of the Netra CP32x0 ARTM-10G ports on a system running the Linux OS.

### ▼ To Tune Ethernet Port Performance

1. Using any ASCII text editor, create a new `.conf` file that will be passed to the `sysctl` utility.

For example, `sysctl_e1000.conf`

**CODE EXAMPLE 3-7** Sample `.conf` File for Tuning Ethernet Ports on a Linux Platform

```
### IPV4 specific settings
# turns TCP timestamp support off, default 1, reduces CPU use
net.ipv4.tcp_timestamps = 0
# turn SACK support off, default on systems with a VERY fast bus ->
# memory interface this is the big gainer
net.ipv4.tcp_sack = 0
# sets min/default/max TCP read buffer, default 4096 87380 174760
net.ipv4.tcp_rmem = 10000000 10000000 10000000
# sets min/pressure/max TCP write buffer, default 4096 16384 131072
net.ipv4.tcp_wmem = 10000000 10000000 10000000
# sets min/pressure/max TCP buffer space, default 31744 32256 32768
net.ipv4.tcp_mem = 10000000 10000000 10000000
### CORE settings (mostly for socket and UDP effect)
# maximum receive socket buffer size, default 131071
```

**CODE EXAMPLE 3-7** Sample .conf File for Tuning Ethernet Ports on a Linux Platform

```
net.core.rmem_max = 524287
# maximum send socket buffer size, default 131071
net.core.wmem_max = 524287
# default receive socket buffer size, default 65535
net.core.rmem_default = 524287
# default send socket buffer size, default 65535
net.core.wmem_default = 524287
# maximum amount of option memory buffers, default 10240
net.core.optmem_max = 524287
# number of unprocessed input packets before kernel starts dropping
# them, default 300
net.core.netdev_max_backlog = 300000
```

2. To apply your settings, invoke the `sysctl` utility, including the configuration file as follows:

```
# sysctl -p /etc/sysctl_nxge.conf
```

---

## 3.4 Enabling Host Board PCI-Express Communication

Presently, only the Netra CP3260 board can be configured to use PCI-express signals to communicate with the Netra CP32x0 ARTM-10G. By default, the host board is usually shipped with these signals disabled. The following procedure enables PCI-express signals.

### ▼ To Enable PCI-Express Signals

1. Do one of the following:
  - If using a SPARC-based host board, skip to [Step 2](#).
  - If using an x86-based host board, perform the following:
    - a. At boot, invoke the BIOS menu (ESC 2, F2, Ctrl-C, or Ctrl-E).
    - b. Under the Chipset menu/Southbridge menu, disable Spread Spectrum clocking for PCI-E.
    - c. Save and Exit the BIOS.
2. Establish a console connection with the shelf manager.

3. Type the shelf command `clia` to determine the slot location of the host board that will connect to the Netra CP32x0 ARTM-10G.

The shelf manager will show a Device ID String indicating the host board.

4. Enter the `clia sendcmd xx 2e 8b 00 00 2a 9 1c` command.

Where `xx` is the board address.

5. At the shelf manager, type `clia board` to find the correct number for your host board installation.

6. Type the shelf `clia boardreset 2` command.

---

## 3.5 Removing the nxge Driver

If you want to remove the `nxge` driver from your system, perform the following procedure, based on your OS.

### ▼ To Remove the Driver From a Solaris Platform

1. Determine the driver packages:

```
# pkginfo | grep nxge
SUNWnxge Sun x8 10G/1G Ethernet Adapter Driver
```

2. Remove the driver packages:

```
# pkgrm SUNWnxge
```

### ▼ To Remove the Driver From a Linux Platform

- Type the following:

```
# rpm -e nxge
```

# Network Configuration

---

This chapter describes how to manually configure the network interface and attributes, including optional methods to automate network configuration.

This chapter contains the following topics:

- [Section 4.1, “Configuring Network Host Files” on page 4-2](#)
- [Section 4.2, “Configuring Jumbo Frames” on page 4-5](#)
- [Section 4.3, “Configuring Link Aggregation” on page 4-10](#)
- [Section 4.4, “Configuring VLANs” on page 4-13](#)

---

## 4.1 Configuring Network Host Files

After installing the driver software as described in [Chapter 3](#), you can create a `hostname.nxge` file to configure each of the Netra CP32x0 ARTM-10G interfaces.

You must designate both an IP address and a host name for each Netra CP32x0 ARTM-10G Ethernet interface in the `/etc/hosts` file.

Adhere to the following guidelines when assigning host names:

- The `/etc/hostname.nxge` file must contain the host name for the appropriate `nxge` interface.
- The host name must have an IP address listed in the `/etc/hosts` file.
- The host name must be different from any other host name of any other interface, for example: `/etc/hostname.nxge0` and `/etc/hostname.nxge1` cannot share the same host name.

The following example shows the `/etc/hostname.nxge` file required for a system called `zardoz` that has an `nxge` driver (`zardoz-11`).

```
# cat /etc/hostname.nxge0
zardoz
# cat /etc/hostname.nxge1
zardoz-11
```

---

**Note** – To preserve `nxge` instance numbers for interfaces between reboots, only one product that uses the `nxge` driver can be used in a given slot. For example, numbers can change if between a series of reboots, a Netra CP32x0 ARTM-10G in one slot is removed and replaced by a different `nxge` driver-based network interface product, and later a Netra CP32x0 ARTM-10G reinstalled into that same slot.

---

## ▼ To Manually Configure Network Host Files

This method enables you to manually manipulate a network interface via the command line interface. Note that any changes made using this method will be lost when the system reboots.

**1. At the command line, search the `/etc/path_to_inst` file for `nxge` interfaces.**

The ARTM reports two `nxge` interfaces, and for each, includes three descriptive parameters:

- Instance path name in quotes
- Instance Number (integer, no quotes)
- Instance driver name in quotes

```
# grep nxge /etc/path_to_inst
# "/pci@7c0/pci@0/pci@9/network@0" 0 "nxge"
# "/pci@7c0/pci@0/pci@9/network@0,1" 1 "nxge"
```

**2. Write down the device path and instance, which in the example is**

`/pci@7c0/pci@0/pci@9/network@0 0`.

Your Netra CP32x0 ARTM-10G device path and instance will be similar. You need this information later to make changes to the `nxge.conf` file.

**3. Set up the Netra CP32x0 ARTM-10G's `nxge` interface.**

**4. Use the `ifconfig` command to assign an IP address to the network interface.**

Type the following at the command line, replacing *ip-address* with the Netra CP32x0 ARTM-10Gs IP address:

```
# ifconfig nxge0 plumb ip-address up
```

Refer to the `ifconfig(1M)` man page and the Solaris documentation for more information.

**5. Create an appropriate entry in the `/etc/hosts` file for each active `nxge` interface.**

Follow the guidelines for assigning IP addresses and host names, described at the beginning of this [Section 4.1, “Configuring Network Host Files”](#) on page 4-2.

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1 localhost
129.144.10.57 zardoz loghost
129.144.11.83 zardoz-11
```

## ▼ To Automatically Configure Network Host Files

This method enables you to create a file that provides a persistent setup, which is automatically loaded at each system reboot.

### 1. Create an `/etc/hostname.nxgex` file.

Where *x* is the instance number of the `nxge` interface you plan to use. For example, to auto-configure a Netra CP32x0 ARTM-10Gs `nxge` interface with instance 0, create an `/etc/hostname.nxge0` file, where 0 is the number of the `nxge` interface. If the instance number were 1, the file name would be `/etc/hostname.nxge1`.

---

**Note** – Do not create an `/etc/hostname.nxge` file for any Netra CP32x0 ARTM-10Gs you plan to leave unused.

---

### 2. Create an appropriate entry in the `/etc/hosts` file for each active `nxge` interface.

Follow the guidelines for assigning IP addresses and host names, described at the beginning of [Section 4.1, “Configuring Network Host Files”](#) on page 4-2.

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1 localhost
129.144.10.57 zardoz loghost
129.144.11.83 zardoz-11
```

---

## 4.2 Configuring Jumbo Frames

Configuring Jumbo Frames enables Ethernet interfaces to send and receive packets larger than the standard 1500 bytes. However, the actual transfer size depends on the switch capability and the device driver capability.

---

**Note** – Refer to the documentation that came with your switch for commands to configure Jumbo Frames.

---

### 4.2.1 Displaying Layer Configuration in a Solaris Environment

The Jumbo Frames checking occurs at Layer 2 or Layer 3, depending on the configuration method.

#### ▼ To Display Layer 2 Configuration

- View the maximum transmission unit (MTU) configuration of a `nxge` instance at any time with the `kstat` command:

```
# kstat nxge:0 | grep mac_mtu
```

The `kstat mac_mtu` variable represents the complete size of the Ethernet frame, which includes the Ethernet header, maximum payload, and CRC. The value of this variable should be equal to or less than the MTU configured on the switch.

## ▼ To Display Layer 3 Configuration

- **View the Layer 3 configuration by using the `dladm` command with the `show-link` option.**

```
# dladm show-link
e1000g0 type: non-vlan mtu: 1500 device: e1000g0
e1000g1 type: non-vlan mtu: 1500 device: e1000g1
e1000g2 type: non-vlan mtu: 1500 device: e1000g2
e1000g3 type: non-vlan mtu: 1500 device: e1000g3
nxge0 type: non-vlan mtu: 9194 device: nxge0
nxge1 type: non-vlan mtu: 9194 device: nxge1
nxge38001 type: vlan 38 mtu: 9194 device: nxge1
```

## 4.2.2 Displaying Driver Statistics

This section describes how to display all the statistics that the driver tracks for an interface.

## ▼ To Show Driver Statistics in a Solaris Environment

1. **Use the `kstat` command to display driver statistics.**

The output can be further filtered to display specific data of interest. The following example displays the receive packet counts on all eight receive DMA channels on `nxge` interface 1.

```
# kstat nxge:1 |grep rdc_packets
rdc_packets 798982054
rdc_packets 792546171
rdc_packets 803941759
rdc_packets 805674872
rdc_packets 798714912
rdc_packets 799293256
rdc_packets 806470537
rdc_packets 805413540
```

## 2. Use the `kstat` command to display driver statistics of a VLAN interface.

```
# kstat nxge:38001
module: nxge instance: 38001
name: nxge38001 class: net
brdcstrcv 0
brdcstxmt 0
collisions 0
crttime 3842.493000352
ierrors 0
ifspeed 10000000000
ipackets 2116069805
ipackets64 6411037101
multircv 0
multixmt 0
norcvbuf 0
noxmtbuf 0
obytes 2757388874
obytes64 23380264381002
oerrors 0
opackets 37606022
opackets64 4332573318
rbytes 2937141290
rbytes64 47178857920554
```

## ▼ To Show Driver Statistics in a Linux Environment

### 1. Use the `ifconfig` utility to display driver statistics.

```
# ifconfig eth6
eth6 Link encap:Ethernet HWaddr 00:14:4F:83:9E:1A
inet addr:192.168.11.189 Bcast:192.168.11.255
Mask:255.255.255.0
inet6 addr: fe80::214:4fff:fe83:9e1a/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:3 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:256 (256.0 b) TX bytes:0 (0.0 b)
```

In the previous example output, MTU indicates the complete size of the Ethernet frame, which includes the Ethernet header, maximum payload, and CRC. The value of MTU should be equal to or less than the MTU configured on the switch.

2. If you want to see only MTU, use the following command:

```
# ifconfig eth6 | grep MTU
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

3. Use the `ethtool` utility with the `-S` option to get more detailed information.

This option displays a large amount of statistics maintained by the driver. Output can be further filtered to display specific data of interest using `grep`.

```
# ethtool -S eth9 | grep rx_pac
rx_packets: 748274
rx_packets: 828
rx_packets: 112
rx_packets: 189088
rx_packets: 134
rx_packets: 196085
rx_packets: 177884
rx_packets: 93
rx_packets: 184050
```

## 4.2.3 Enabling Jumbo Frames

Enable Jumbo Frames if you want to configure Ethernet interfaces to send and receive packets larger than the standard 1500 bytes.

### 4.2.3.1 Enabling Jumbo Frames in a Solaris Environment

This section describes how to enable Jumbo Frames in both a SPARC and an x86 environment.

#### ▼ To Enable Jumbo Frames in a Solaris Environment

1. Enable Jumbo Frames for a port using the `nxge.conf` file.

```
name = "pciex108e,abcd" parent = "/pci@780/pci@0/pci@8/network@0"
unit-address = "0"
accept-jumbo=1;
```

2. Reboot the system.

```
# reboot -- -r
```

3. Set the maximum transmission unit (MTU) for maximum performance.

```
# ifconfig nxge mtu 9194
```

#### 4.2.3.2 Enabling Jumbo Frames in a Linux Environment

This section describes how to enable Jumbo Frames in Linux environment.

##### ▼ To Enable Jumbo Frames in a Linux Environment

1. Ensure that the `nxge` software is installed.

```
# modprobe nxge
```

2. Plumb the Netra CP32x0 ARTM-10G 10GbE SFP+ interface.

The `xxx.xxx.xx.xxx` represents the IP address of the interface.

```
# ifconfig eth2 xxx.xxx.xx.xxx up
```

3. Set the maximum transmission unit (MTU) to specify jumbo frames size.

Where *x* designates the instance number of the interface.

```
# ifconfig ethnumber mtu 9194
```

---

## 4.3 Configuring Link Aggregation

Link aggregation enables one or more network links to be aggregated together to form a link-aggregation group. This link-aggregation group appears to MAC clients as a regular link. Link aggregation is defined by IEEE 802.3ad and provides the following benefits:

- Increased bandwidth
- Linearly incremental bandwidth
- Load sharing
- Automatic configuration
- Rapid configuration and reconfiguration
- Deterministic behavior
- Low risk of duplication or incorrect ordering
- Support of existing IEEE 802.3 MAC clients

### 4.3.1 Configuring Link Aggregation in a Solaris Environment

This section explains how to configure link aggregation.

#### ▼ To Configure Link Aggregation in a Solaris Environment

1. **Aggregate `nxge0` and `nxge1` to form an aggregation and use a random number as key.**
2. **Unplumb the interfaces to be aggregated:**

```
# ifconfig down unplumb nxge0
# ifconfig down unplumb nxge1
```

### 3. Create a link aggregation group with a random number as a key, without specifying mode.

This example uses 33.

```
# dladm create-aggr -d nxge0 -d nxge1 33
```

When the command returns, one line appears in the `/etc/aggregation.conf` file and indicates that the default mode is off. For example:

```
# tail -1 /etc/aggregation.conf
33 L4 2 nxge0/0,nxge1/0 auto off short
# dladm show-aggr
key: 33 (0x0021) policy: L4 address: 0:3:ba:d8:9d:e8 (auto)
device address speed duplex link state
nxge0 0:3:ba:d8:9d:e8 10000 Mbps full up standby
nxge1 0:3:ba:d8:9d:e9 10000 Mbps full up standby
# dladm show-link aggr33
aggr33 type: non-vlan mtu: 1500 aggregation: key 33
```

### 4. Plumb up the interface `aggrkey`, which is `aggr33` in the following example:

```
# ifconfig aggr33 plumb
# ifconfig aggr33
aggr33: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
index 8
inet 0.0.0.0 netmask 0
ether 0:3:ba:d8:9d:e8
# ifconfig aggr33 192.168.1.1/24 broadcast + up
# ifconfig aggr33
aggr33: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu
1500 index 8
inet 192.168.1.1 netmask ffffffff0 broadcast 192.168.1.255
ether 0:3:ba:d8:9d:e8
```

### 5. Show link aggregation status again.

The state now is attached:

```
# dladm show-aggr
key: 33 (0x0021) policy: L4 address: 0:3:ba:d8:9d:e8 (auto)
device address speed duplex link state
nxge0 0:3:ba:d8:9d:e8 10000 Mbps full up attached
nxge1 0:3:ba:d8:9d:e9 10000 Mbps full up attached
```

**6. Use the `dladm show-aggr -s` command to display statistics:**

```
# dladm show-aggr -s
key: 33 ipackets rbytes opackets obytes %ipkts %opkts
Total 0 0 16 1182
nxge0 0 0 16 1182 - 100.0
nxge1 0 0 0 0 - 0.0
```

**7. Use the `dladm show-aggr -L` command to display LACP specific information:**

```
# dladm show-aggr -L
key: 33 (0x0021) policy: L4 address: 0:3:ba:d8:9d:e8 (auto)
LACP mode: off LACP timer: short
device activity timeout aggregatable sync coll dist defaulted
expired
nxge0 passive short yes no no no no no
nxge1 passive short yes no no no no no
```

For more information, refer to the man pages for the `dladm` command.

## 4.3.2 Configuring Link Aggregation in a Linux Environment

Many administrators familiar with Linux refer to Link Aggregation as “NIC Bonding.” Both refer to the IEEE802.3ad standard that defines Dynamic link aggregation.

### ▼ To Configure Link Aggregation in a Linux Environment

**1. Modify the `/etc/modprobe.conf` file for the 2.6 kernels file to add the following lines:**

```
alias bond0 bonding
options bonding max_bonds=2 mode=4 miimon=1000
```

Where:

- `bond0` is the bonding device.
- `max_bonds` is the number of bond interfaces to be created.
- `mode` specifies the bonding policies.
- `miimon` is the frequency in milliseconds that MII link monitoring occurs.

Refer to Linux documentation for more information.

## 2. Load the bonding driver:

```
# modprobe bonding
```

## 3. Configure the bond0 interface.

In this example, bond0 is the master of the two interfaces eth4 and eth5.

```
# ip addr add 192.12.38.64/24 brd + dev bond0
# ip link set dev bond0 up
# ifenslave bond0 eth4 eth5
```

---

# 4.4 Configuring VLANs

VLANs enable you to split your physical LAN into logical subparts, providing an essential tool for increasing the efficiency and flexibility of your network.

VLANs are commonly used to separate groups of network users into manageable broadcast domains, to create logical segmentation of workgroups, and to enforce security policies among each logical segment.

Each defined VLAN behaves as its own separate network. The traffic and broadcasts of each VLAN are isolated from the others, increasing the bandwidth efficiency within each logical group.

Although VLANs are commonly used to create individual broadcast domains or separate IP subnets, a server can have a presence on more than one VLAN simultaneously.

The Netra CP32x0 ARTM-10G supports multiple VLANs on a per port or per interface basis, allowing very flexible network configurations. With multiple VLANs on an ARTM, a single physical port can have a logical presence on multiple IP subnets.

By default, 128 VLANs can be defined for each VLAN-aware ARTM port in the chassis. However, you can increase this number by changing the system parameters. If your network does not require multiple VLANs, you can use the default configuration, in which case no further configuration is necessary.

---

**Caution** – If you change any of the VLAN configuration parameters, you must reboot the system before the changes take effect. If you make changes and do not reboot, your system might experience configuration problems.

---

VLANs can be created according to various criteria, but each VLAN must be assigned a VLAN tag or VLAN ID (VID). The VID is a 12-bit identifier between 1 and 4094 that identifies a unique VLAN. For each network interface (nxge0 and nxge1), 4094 possible VLAN IDs can be selected per port for up to 4 ports.

Tagging an Ethernet frame requires adding a tag header to the frame. Insert the header immediately following the destination MAC address and the source MAC address. The tag header consists of two bytes of Ethernet Tag Protocol identifier (TPID, 0x8100) and two bytes of Tag Control Information (TCI). [TABLE 4-1](#) shows the Ethernet tag header format.

**TABLE 4-1** Ethernet Tag Header Format

TAG Header Description			Octet
TPID, high byte (value = 0x81)			1
TPID, low byte (value = 0x00)			2
Priority (3 bits)	CFI (1 bit)	VLAN ID (4bits)	3
VLAN ID (8 bits)			4

By default, a single VLAN is configured for every port. This configuration groups all ports into the same broadcast domain, just as if there were no VLANs at all. VLAN tagging for the switch port is turned off.

---

**Note** – If you configure a VLAN virtual device for an ARTM, all traffic sent or received by that ARTM must be in VLAN-tagged format.

---

# 4.4.1 Configuring VLANs in a Solaris Environment

This section describes how to configure VLANs in a Solaris environment.

## ▼ To Configure VLANs in a Solaris Environment

1. **Create one `/etc/hostname.nxgex` file for each VLAN that will be configured for the installed ARTM.**

Where *x* is the instance number of the nxge interface you plan to use. For example, to configure a Netra CP32x0 ARTM-10Gs nxge interface with instance 0, create a `hostname.nxge0` file, where 0 is the number of the nxge interface. If the instance number were 1, the file name would be `hostname.nxge1`.

**2. Enter the VLAN ID (VID) and the physical point of attachment (PPA), as in the following example.**

```
VLAN logical PPA = 1000 * VID + Device PPA
nxge123000 = 1000*123 + nxge
```

This format limits the maximum number of PPAs (instances) you can configure to 1000 in the `/etc/path_to_inst` file.

For example, on a server blade with an Netra CP32x0 ARTM-10G having an instance of 0, belonging to a member of two VLANs (with VID 123 and 224), you would use `nxge123000` and `nxge224000`, respectively, as the two VLAN PPAs.

For more information about VID formats, see the introduction to this [Section 4.4, “Configuring VLANs”](#) on page 4-13.

**3. Use `ifconfig` to configure a VLAN virtual device.**

```
# ifconfig nxge123000 plumb up
# ifconfig nxge224000 plumb up
```

The output of `ifconfig -a` on a system having VLAN devices `nxge123000` and `nxge224000` would be as follows:

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232
index 1
inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
index 2
inet 129.144.131.91 netmask ffffffff00 broadcast
129.144.131.255
ether 8:0:20:a4:4f:b8
nxge123000: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 3
inet 199.199.123.3 netmask ffffffff00 broadcast
199.199.123.255
ether 8:0:20:a4:4f:b8
nxge224000: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 4
inet 199.199.224.3 netmask ffffffff00 broadcast
199.199.224.225
ether 8:0:20:a4:4f:b8
```

4. On the switch, set VLAN tagging and set VLAN ports to coincide with the VLANs you have set up on the server.

For formats, see the introduction to [Section 4.4, “Configuring VLANs” on page 4-13](#).

Continuing the examples used in [Step 2](#), you would set up VLAN ports 123 and 224 on the switch.

Refer to the documentation that came with your switch for specific instructions for setting VLAN tagging and ports.

## 4.4.2 Configuring VLANs in a Linux Environment

This section describes how to configure VLANs in a Linux environment.

### ▼ To Configure VLANs in a Linux Environment

1. Ensure that the `nxge` driver is installed:

```
# modprobe nxge
```

2. Plumb the Sun Dual 10GbE SFP+ PCIe ExpressModule interface.

Where `xxx.xxx.xx.xxx` = the IP address of the interface.

```
# ifconfig eth2 xxx.xxx.xx.xxx up
```

3. Insert the VLAN module:

```
# /sbin/modprobe 8021q
```

4. Add the VLAN instance (VID):

```
# /sbin/vconfig add eth2 5
```

For more information about VID formats, see the introduction to [Section 4.4, “Configuring VLANs” on page 4-13](#).

5. Configure the `nxge` VLAN (`eth2` in the following example):

```
# ifconfig eth2.5 xxx.xxx.xx.xxx up
```

## MMC and IPMI Functions

---

This chapter describes the Module Management Controller (MMC) and Intelligent Platform Management Interface (IPMI).

This chapter contains the following topics:

- [Section A.1, “MMC and IPMI Functions” on page A-2](#)
- [Section A.2, “Sensor Data Records” on page A-3](#)
- [Section A.5, “IPMI Commands” on page A-6](#)

---

## A.1 MMC and IPMI Functions

The Netra CP32x0 ARTM-10G module includes a Module Management Controller (MMC) device for reporting status information to the ATCA blade. This MMC uses an Intelligent Platform Management Interface (IPMI), which communicates with the AdvancedTCA.

The MMC controls and monitors the following:

- Hotswap communication with the shelf manager
- Inlet air temperature
- Voltage monitoring
- Electronic keying (e-Keying), as described in the AMC.0 specification
- FRU information
- BLUE LED indicators for hotswap
- OOS (out of service) LED indicator
- GREEN LED indicators for “ok”

From the management perspective, the MMC implementation conforms to the ARTM port commands specified in the AdvancedTCA specification. An ATCA board communicates to the ARTM using site ID 15, and thus an IPMBL address of 0x8E.

Additional features of the MMC implementation include:

- An I2C connection provides communication path between the MMC and the temperature sensor. Also, the I2C serial bus routes to the SFP+ modules.
- Supports hot-swap operation as defined for AMC modules in PICMG AMC.0 specification.
- Provides “fail-safe flash update;” if interrupted at anytime, the MMC firmware is able to respond and reflash.
- Provides “I2C hang recovery;” the MMC is able to detect and recover from an I2C bus hang.

---

## A.2 Sensor Data Records

The MMC monitors the status of the ARTM and provides data so it can be read by the shelf manager. The sensor data records (SDRs) that the Netra CP32x0 ARTM-10G monitors are described in [TABLE A-1](#).

**TABLE A-1** Sensor Data Records

Sensor	UNR	UC	UNC	LNC	LC	LNR	ID String
1.0V	1.150	1.100	1.070	0.930	0.900	0.850	+1.0V
1.2V	1.380	1.320	1.284	1.116	1.080	1.020	+1.2V
1.5V	1.725	1.650	1.605	1.395	1.350	1.275	+1.5V
2.5/1.8V CTV	2.875	2.750	2.675	1.674	1.620	1.530	+2.5V/+1.8V CTV
2.5V	2.875	2.750	2.675	2.325	2.250	2.125	+2.5V
3.3V	3.795	3.696	3.600	3.000	2.904	2.805	+3.3V
12V	13.800	13.200	12.840	11.160	10.800	10.200	+12V
Board Temp (LM60)	81.15	76.44	76.44	N/A	N/A	N/A	Board Temp
Inlet Temp (LM75)	80	75	75	N/A	N/A	N/A	Air Inlet Temp

---

## A.3 FRU Information Records

The Netra CP32x0 ARTM-10G includes the standard FRU data records per the IPMI FRU information storage definition, board information area, as shown in [TABLE A-2](#). The Netra CP32x0 ARTM-10G includes additional FRU records as defined in the PICMG 2.9 specification.

**TABLE A-2** Standard FRU Product Data Records

Product Version	Revision 50
Language Code	0
Manufacturer Name	Sun Microsystems, Inc.
Product Name	Netra CP32x0 ARTM-10G
Product Part/Model Number	375-3544-xx rev xx

**TABLE A-2** Standard FRU Product Data Records (*Continued*)

Product Version	50
Product Serial Number	2009MAC-YYWWNYSSSS <ul style="list-style-type: none"> <li>• YY = year of final assembly</li> <li>• WW = calendar week of final assembly</li> <li>• NT = multiuse code</li> <li>• SSSS = sequence number (does not reset each week or year)</li> </ul>
Asset Tag	0000000000000001
FRU Programmer File ID	fru-info.inf
Custom Product Info	
Management Controller ID String	ARTM-10G
Manufacturer ID	0x00002A
Product ID	0x0CE6

**TABLE A-3** Standard FRU Board Data Records

Version	1
Language Code	en
Mfg Date/Time	10/10/2005 Manufacturing time is defined as “minutes since 1/1/96” in the IPMI FRU specification.
Manufacturer	Sun Microsystems, Inc.
Product Name	ARTM-10G
Product Version	50
Serial Number	2009MAC-YYWWNYSSSS <ul style="list-style-type: none"> <li>• YY = year of final assembly</li> <li>• WW = calendar week of final assembly</li> <li>• NT = multiuse code</li> <li>• SSSS = sequence number (does not reset each week or year)</li> </ul>
Part Number	375-3544-xx rev xx
FRU Programmer File ID	fru-info.inf

## A.4 ARTM e-Keying Port Assignments

The Netra CP32x0 ARTM-10G ports assign functions in a fashion similar to the AMC cards. [TABLE A-4](#) shows the assignments for the ARTM.

**TABLE A-4** ARTM e-Keying Port Assignments

Port	Port Name	Link Type	Link Type Extension	ARTM PIN Groups
0	unused			
1	unused			
2	PCIe Lane0	AMC.1 PCI Express type 8	0, Gen1	J33 A1,B1, C1, D1
3	PCIe Lane1	AMC.1 PCI Express type 8	0, Gen1	J33 A2,B2, C2, D2
4	PCIe Lane2	AMC.1 PCI Express type 8	0, Gen1	J33 A3,B3, C3, D3
5	PCIe Lane3	AMC.1 PCI Express type 8	0, Gen1	J33 A4,B4, C4, D4
6	PCIe Lane4	AMC.1 PCI Express type 8	0, Gen1	J33 A5,B5, C5, D5
7	PCIe Lane5	AMC.1 PCI Express type 8	0, Gen1	J33 A6,B6, C6, D6
8	PCIe Lane6	AMC.1 PCI Express type 8	0, Gen1	J33 A7,B7, C7, D7
9	PCIe Lane7	AMC.1 PCI Express type 8	0, Gen1	J33 A8,B8, C8, D8
10	unused			
11	unused			
12	unused			
13	unused			
14	AMC1_EO_12			J31C1,D1, E1, F1
15	AMC1_EO_13			J31A1,B1, E2, F2
16	AMC1_EO_14			J31 A2,B2, C2, D2
17	AMC1_EO_15			J31 C3,D3, E3, F3
18	AMC1_EO_16			J31 A3,B3, E4, F4
19	AMC1_EO_17			J31 A4, B4, C4, D4
20	AMC1_EO_18			J31 C5, D5, E5, F5
21	AMC1_EO_19			J31 A5, B5, E6, F6
22-31	unused			

## A.5 IPMI Commands

The MMC communicates with the carrier controller through the local IPMB-L bus of the carrier, and MMC responds to all mandatory commands for AMC Module Management Controllers (as defined in the AdvancedTCA specification), as well as some optional commands. [TABLE A-5](#) lists supported IPMI commands.

**TABLE A-5** IPMI Commands

Command	IPMI/PICMG/AMC Specification	NetFn	CMD	MMC Req
<b>IPM Device Global Commands</b>				
Get Device ID	17.1	App	01h	Mandatory
Broadcast "Get Device ID"	17.9	App	01h	Mandatory
<b>Messaging Commands</b>				
Send Message	18.7	App	34h	Optional
<b>Event Commands</b>				
Platform Event	23.3	S/E	02h	Mandatory
<b>Sensor Device Commands</b>				
Get Device SDR Info	29.2	S/E	20h	Mandatory
Get Device SDR	29.3	S/E	21h	Mandatory
Reserve Device SDR Repository	29.4	S/E	22h	Mandatory
Get Sensor Reading Factors	29.5	S/E	23h	Optional
Set Sensor Hysteresis	29.6	S/E	24h	Optional
Get Sensor Hysteresis	29.7	S/E	25h	Optional
Set Sensor Threshold	29.8	S/E	26h	Optional
Get Sensor Threshold	29.9	S/E	27h	Optional
Set Sensor Event Enable	29.10	S/E	28h	Optional
Get Sensor Event Enable	29.11	S/E	29h	Optional
Rearm Sensor Events	29.12	S/E	2Ah	Optional
Get Sensor Event Status	29.13	S/E	2Bh	Optional
Get Sensor Reading	29.14	S/E	2Dh	Mandatory
<b>FRU Device Commands</b>				
Get FRU Inventory Area Info	28.1	Storage	10h	Mandatory

**TABLE A-5** IPMI Commands (*Continued*)

Read FRU Data	28.2	Storage	11h	Mandatory
Write FRU Data	28.3	Storage	12h	Mandatory
<b>AdvancedTCA™ Commands</b>				
Get PICMG Properties	3-9	PICMG	00h	Mandatory
FRU Control	3-22	PICMG	04h	Mandatory
Get FRU LED Properties	3-24	PICMG	05h	Mandatory
Get LED Color Capabilities	3-25	PICMG	06h	Mandatory
Set FRU LED State	3-26	PICMG	07h	Mandatory
Get FRU LED State	3-27	PICMG	08h	Mandatory
Get Device Locator Record ID	3-29	PICMG	0Dh	Mandatory
<b>AMC® Commands</b>				
Set AMC Port State	3-27	PICMG	19h	Mandatory
Get AMC Port State	3-28	PICMG	1Ah	Mandatory



# Specifications and Connectors

---

This chapter provides the specifications and connector Pin-outs for the Netra CP32x0 ARTM-10G.

This chapter contains the following topics:

- [Section B.1, “Specifications” on page B-2](#)
- [Section B.2, “Connectors and PIN Assignments” on page B-6](#)

# B.1 Specifications

This section provides mechanical, electrical, environmental, and other relevant specifications.

## B.1.1 Physical Dimensions

The Netra CP32x0 ARTM-10G is a 6U (233.35 mm) height board with 80 mm in depth for standard applications. It complies with IEEE 1101.11 mechanical standards, as required by the PICMG 2.0 Revision 3.0 specification. The Netra CP32x0 ARTM-10G is keyed to conform to the PICMG 2.10, “Keying of ATCA Boards and Backplanes” specification.

## B.1.2 Power Requirements

The Netra CP32x0 ARTM-10G derives all power from the front ATCA node board. The front board must provide the following voltages: 3.3V/5V/+12V/12V.

The power consumption of the ARTM is limited to 25W maximum. The 3.3V power to the management port is limited to 200ma and the current limit-set point for the 12V payload power is 2.5a.

## B.1.3 Environmental Specifications and Compliance

The environmental specifications for the Netra CP32x0 ARTM-10G assembly are presented in [TABLE B-1](#).

**TABLE B-1** Environmental Specifications

Specification	Value
Operating Temperature (airflow 5.0 CFM)	0°C ~ 55 °
Operating Temperature (airflow 2.0 CFM)	0°C ~ 23 °
Storage Temperature	-40 °~ 85 °
Operating Temperature Gradient	11 °/H (max)
Storage Temperature Gradient	20 °/H (max)
Shipping Temperature Gradient	20 °/H (max) *1

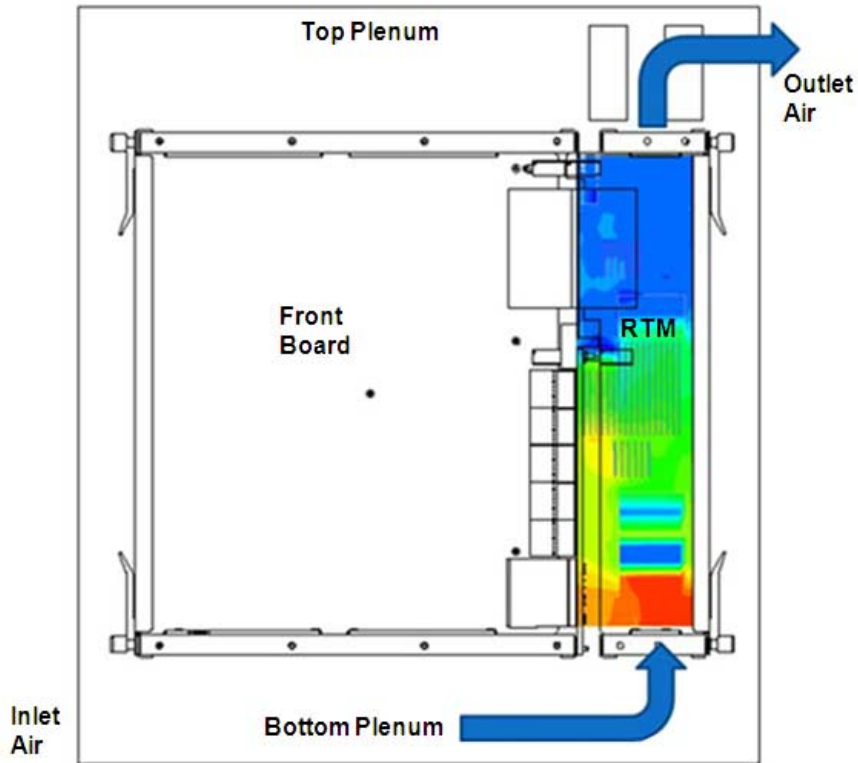
**TABLE B-1** Environmental Specifications (*Continued*)

Operating Humidity	8 % ~ 80 %
Storage Humidity	5 % ~ 95 %
Shipping Humidity	5 % ~ 95 % <sup>*1</sup>
Wet bulb Maximum Temperature	27 °
Condensation	No condensation
Atmospheric Pressure and Altitude	Operating: 0 ~ 3,000 m Shipping: 0 ~ 12,000 m
Operating Shock	See PICMG 3.0 specification, Regulatory guidelines.
Operating Vibration	See PICMG 3.0 specification, Regulatory guidelines.
RoHS	6 of 6 compliant

## B.1.4 Pressure Drop Versus Airflow

A thermal simulation model of the ARTM was constructed to analyze the pressure drop of the ARTM at incremental airflow conditions. System integrators may use these results as part of the selection criteria for the host chassis.

**FIGURE B-1** Pressure Drop Simulation Model



**TABLE B-2** Pressure Drop Versus Inlet Air Speed

Wind Speed	Inlet Pressure (N/M2)	Pressure Drop (N/M2/psf)
0 CFM (0m/s)	0	0/0
1 CFM (0.274m/s)	1.4	1.4/0.03
2 CFM (0.548m/s)	4	4/0.087
3 CFM (0.822m/s)	7.6	7.6/0.166

**TABLE B-2** Pressure Drop Versus Inlet Air Speed (*Continued*)

4 CFM (1.096m/s)	12.1	12.1/0.264
5 CFM (1.37m/s)	17.6	17.6/0.384
6 CFM (1.644m/s)	24	24/0.523
7 CFM (1.918m/s)	31.2	31.2/0.68
8 CFM (2.192m/s)	39.4	39.4/0.86
9 CFM (2.466m/s)	48.3	48.3/1.05
10 CFM (2.74m/s)	58.1	58.1/1.266

## B.1.5 NEBS Compliance

NEBS certifications are performed by the integrator at a system level (chassis, ATCA, AMC shelf managers, etc.). The ARTM module will not preclude the system from passing NEBS.

## B.1.6 Electromagnetic Compliance

The board is designed and implemented to minimize electromagnetic emissions, susceptibility, and electrostatic discharge effects. The ARTM carries the certifications shown in [TABLE B-3](#).

**TABLE B-3** EMC Emission Compliance

Certification	Compliance
US: FCC 47 CFR Part 15 Class A	Yes, Class A emissions requirements (USA)
Canada: ICES 003 Class A	Yes Class A Digital Apparatus emissions (Canada)
Japan: VCCI Class A	Yes Class A ITE emissions requirements (Japan)
Europe Commercial: EN 55022:1994 Class A	Yes, Class A ITE emissions requirements (EU)
Europe Commercial: EN 55024:1998 Class A	Immunity for ITE equipment
Europe Commercial: EN 61000-4-2,3,5,6,8,11: 2001	EMC Electrostatic discharge immunity
Europe Commercial: EN 61000-4-4: 2000 (Limits for harmonic current emissions)	Yes
Europe Commercial: EN 61000-3-2,3	Yes, Limits for harmonic current emissions

**TABLE B-3** EMC Emission Compliance (*Continued*)

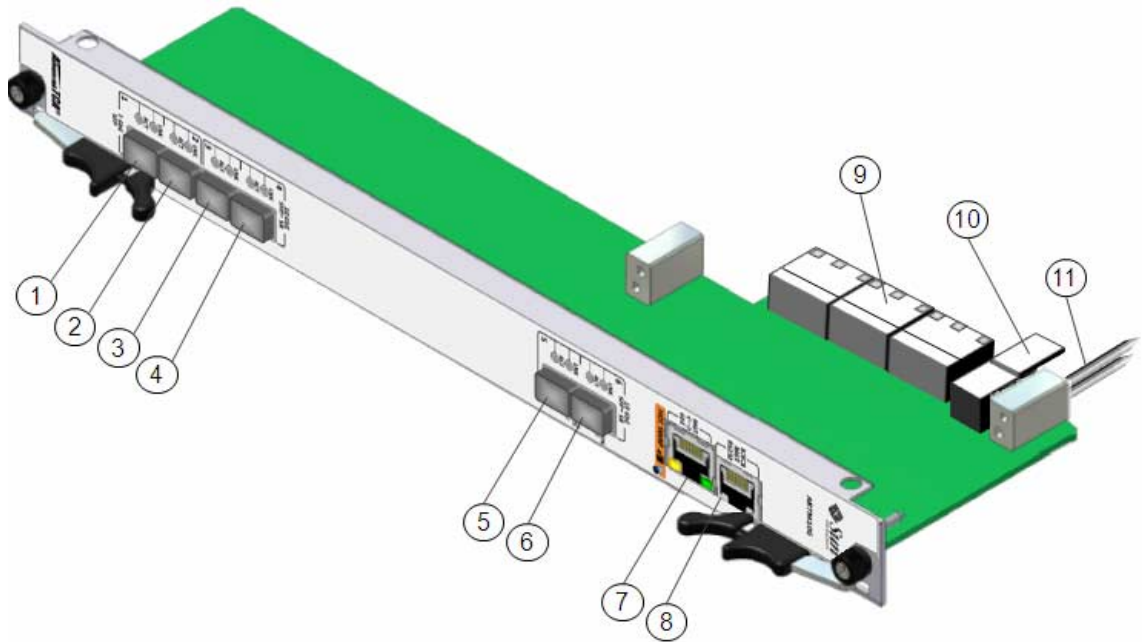
Europe Telecom Carrier: EN 300-386 v1.3.3 April 2005	Requirements for Telecom Network Equipment – Non-Telco Centers
Europe CE Mark	Yes
Australia: AS/NZS 3548 C-Tick	Yes, Class A ITE emissions requirements (Australia)
Korea: MIC	Yes
Taiwan: BSMI	Yes

---

## B.2 Connectors and PIN Assignments

This section provides position and PIN-out details of all connectors available on the Netra CP32x0 ARTM-10G.

**FIGURE B-2** Netra CP32x0 ARTM-10G Connectors



**TABLE B-4** Connector Port Identification and Location

1	1GbE SFP	7	10/100/1000 Mb management port
2	1GbE SFP	8	Serial console port
3	10GbE SFP+	9	Zone 3 Connectors
4	10GbE SFP+	10	Power connector
5	10GbE SFP+	11	RTM alignment PIN
6	10GbE SFP+		

# B.2.1 1GbE SFP Ethernet Ports

Ports 1 and 2 are shipped without SFP modules. The Netra CP32x0 ARTM-10G can be populated with either single mode or multimode variety, using LC connectors.

**TABLE B-5** 1GbE SFP Connector Characteristics

XSFP-SW-2GB-Z	(Short Reach) SFP Transceiver for the Express Module
SG-XSWCS-CU-SFPL-Z	1000BASE-T Copper SFP transceivers

# B.2.2 10GbE SFP+ Ethernet Ports

Ports 3, 4, 5 and 6 are shipped without SFP+ modules. The Netra CP32x0 ARTM-10G can be populated with either single mode or multimode variety, using LC connectors.

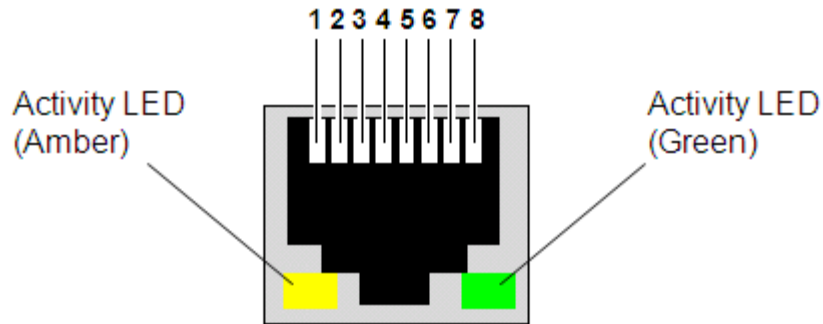
**TABLE B-6** 10GbE SFP+ Connector Characteristics

X5561A-Z -SR	(Short Reach) SFP+ Transceiver for the ExpressModule
	10GBASE-SR: Depending on fiber quality, up to 300 meters for high bandwidth multimode fiber

# B.2.3 10/100/1000 Mb Management Port

The Netra CP32x0 ARTM-10G provides a 1GbE management port on the faceplate via R-J45 jack (port 7). The port is wired as pass through to the front ATCA carrier via the J32/P32 Zone-3 connector.

**FIGURE B-3** 10/100/1000 Mb Management Port PIN Location



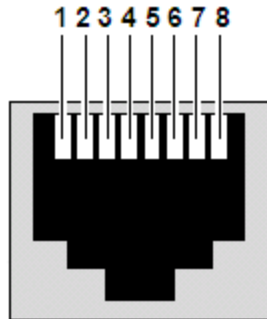
**TABLE B-7** 10/100/1000 Management Port PIN Locations

PIN	Signal Name	PIN	Signal Name
1	DA+	5	DC-
2	DA-	6	DB-
3	DB+	7	DD+
4	DC+	8	DD-

## B.2.4 Serial Console Management Port

Port 8 is a low-profile RJ-45 connector jack that provides RS-232 signaling. The port is wired as pass through to the front ATCA carrier via the J32/P32 Zone-3 connector.

**FIGURE B-4** Serial Console Management Port PIN Locations



**TABLE B-8** Serial Console Management Port PIN Locations

PIN	Signal Name	PIN	Signal Name
1	RTS	5	GND
2	DTR	6	RXD
3	TXD	7	DSR
4	GND	8	CTS

## B.2.5 Zone 3 Connectors

The Netra CP32x0 ARTM-10G routes all of the I/O signals to the front ATCA board through the Zone 3 connector complex consisting of connections P31, P32 and P33. These “P” connectors mate with “J” connectors located on the ATCA board. All of these connectors use a 3-PIN per column type, available from Tyco Electronics Corporation (<http://www.tycoelectronics.com>), part number 1469183-1.

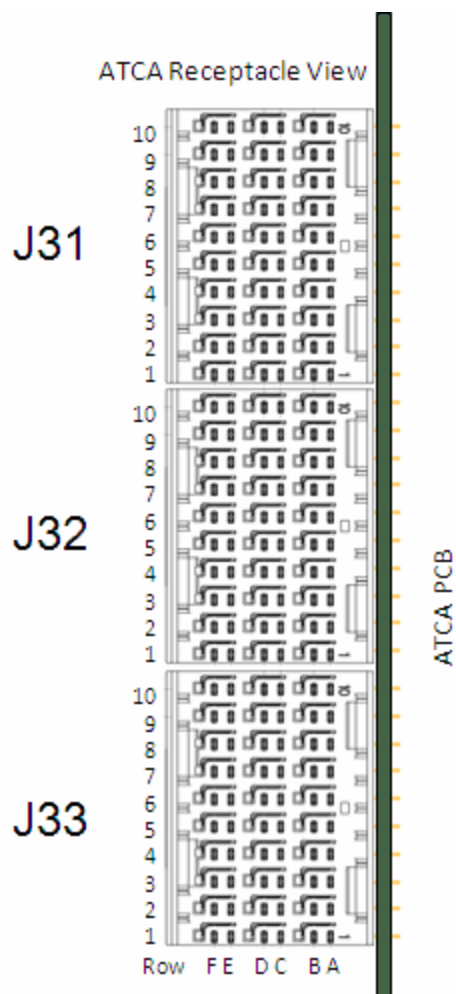
ATCA blades designed to connect with the Netra CP32x0 ARTM-10G should use a Tyco (or equivalent receptacle): 1469081-1, Receptacle 3-pair 10 column HM-Zd.

---

**Note** – The ARTM connector PIN-outs are presented from the point of view of the ATCA board, such that: “TX” refers to ATCA as the signal source, and ARTM as the signal receiver. And “RX” refers to the ATCA board as signal receiver, and ARTM as signal source.

---

**FIGURE B-5** Zone 3 Connector Port PIN Locations



## B.2.5.1 Connector P31, 10GbE XAUI PIN Assignments

The Netra CP32x0 ARTM-10G P31 connector information is in [TABLE B-9](#) and [TABLE B-10](#).

**TABLE B-9** Connector P31, 10GbE XAUI PIN Assignments

Row	Interface	A	B	C	D	E	F
1	XAUI	XAUI_0_TX1 +	XAUI_0_TX1-	XAUI_0_RX0 +	XAUI_0_RX0-	XAUI_0_TX0 +	XAUI_0_TX0-
2	XAUI	XAUI_0_RX2 +	XAUI_0_RX2-	XAUI_0_TX2 +	XAUI_0_TX2-	XAUI_0_RX1 +	XAUI_0_RX1-
3	XAUI	XAUI_1_TX0 +	XAUI_1_TX0-	XAUI_0_RX3 +	XAUI_0_RX3-	XAUI_0_TX3 +	XAUI_0_TX3-
4	XAUI	XAUI_1_RX1 +	XAUI_1_RX1-	XAUI_1_TX1 +	XAUI_1_TX1-	XAUI_1_RX0 +	XAUI_1_RX0-
5	XAUI	XAUI_1_TX3 +	XAUI_1_TX3-	XAUI_1_RX2 +	XAUI_1_RX2-	XAUI_1_TX2 +	XAUI_1_TX2-
6	XAUI					XAUI_1_RX3 +	XAUI_1_RX3-
7							
8							
9							
10							

**TABLE B-10** Connector P31, 10GbE XAUI Signal Descriptions

XAUI_0_TX	XAUI Port 0 TX signals – optional Zone 3 Extended Options overlay
XAUI_0_RX	XAUI Port 0 RX signals – optional Zone 3 Extended Options overlay
XAUI_1_TX	XAUI Port 1 TX signals – optional Zone 3 Extended Options overlay
XAUI_1_RX	XAUI Port 1 RX signals – optional Zone 3 Extended Options overlay

## B.2.5.2 Connector P32, Management Infrastructure PIN Assignments

The Netra CP32x0 ARTM-10G P32 connector information is in [TABLE B-11](#) and [TABLE B-12](#).

**TABLE B-11** Connector P32, Management Infrastructure PIN Assignments

Row	Interface	A	B	C	D	E	F
1	MDIO	XAUI_MDIO	XAUI_MDC				
2	RS-232					SR0_RTS	SR0_DTR
3	RS-232					SR0_TXD	SR0_RXD
4	RS-232					SR0_DSR	SR0_CTS
5							
6	LAN	LAN0_A+	LAN0_A-	LAN0_CTV	LAN0_CTV	LAN0_B+	LAN0_B-
7	LAN	LAN0_C+	LAN0_C-	ACT_LED#	LINK_LED#	LAN0_D+	LAN0_D-
8							
9							
10							

**TABLE B-12** Connector P32, Management Infrastructure Signal Descriptions

LAN0	10/100/1000BASE-T signals
LAN0_CTV	10/100/1000BASE-T transformer Center Tap signal, which could be used to terminate center tap of transformers, if they are placed on an ARTM. This signal is applicable if Ethernet PHY is located on ATCA blade, while GE transformers are located on an ARTM.
ACT_LED#	LAN 0 (Management) activity indicator signal for LED (active low)
LINK_LED#	LAN 0 (Management) LINK indicator signal for LED (active low)
SR0_...	RS-232 Serial Signals, Transmit, Receive, Clear To Send, Request To Send, Data Terminal Ready, Data Set Ready.
XAUI_MDIO	XAUI MDIO signal – optional Zone 3 Extended Option overlay. Together with MDC, this serial bus may be used to access registers within the Media Access Controller (MAC) for purposes of configuration or to retrieve PHY status.
XAUI_MDC	XAUI MDC signal – optional Zone 3 Extended Option overlay.

## B.2.5.3 Connector P33, Management Infrastructure PIN Assignments

The Netra CP32x0 ARTM-10G P33 connector information is in [TABLE B-13](#) and [TABLE B-14](#).

**TABLE B-13** Connector P33, PCIe and Miscellaneous ARTM PIN Assignments

Row	Interface	A	B	C	D	E	F
1	PCIe	PETx0+	PETx0-	PERx0+	PERx0-	FCLKA+	FCLK+
2	PCIe	PETx1+	PETx1-	PERx1+	PERx1-		
3	PCIe	PETx2+	PETx2-	PERx2+	PERx2-		
4	PCIe	PETx3+	PETx3-	PERx3+	PERx3-		
5	PCIe	PETx4+	PETx4-	PERx4+	PERx4-		
6	PCIe	PETx5+	PETx5-	PERx5+	PERx5-		
7	PCIe	PETx6+	PETx6-	PERx6+	PERx6-		
8	PCIe	PETx7+	PETx7-	PERx7+	PERx7-		PCI_RST#
9	Misc	NC	NC	NC	NC	ARTM#	PCI_CFG
10	Misc.	NC	NC	NC	NC	PS0#	Enable#

**TABLE B-14** Connector P33, PCIe and Miscellaneous ARTM Signal Descriptions

PETx...	PCI-Express transmit differential pair signals
PERx...	PCI-Express receive differential pair signals
FCLKA	Fabric clock, as defined in AMC.0 specification. Intended to be used for PCI-Express 100 MHz spread spectrum clock.
PCI_RST#	PCI-Express reset signal. Logic low shall reset PCI-Express switch and PCI-Express interfaces that are behind it. The Netra CP32x0 ARTM-10G asserts logic ground to this signal.
ARTM#	ARTM# signal is grounded by the Netra CP32x0 ARTM-10G to indicate a MMC controller is present (see AMC.0 specification.) The ATCA implementation shall include a 10Kohm pull-up resistor to management power on the ATCA. In this way, an ATCA is able to detect non-intelligent vs. intelligent ARTMs.

**TABLE B-14** Connector P33, PCIe and Miscellaneous ARTM Signal Descriptions (*Continued*)

PCI_CFG	PCI-Express bus configuration signal. It shall be grounded on ARTMs that use a single x8 PCI-Express bus and pulled up on the ATCA with 10Kohm resistor to management power. On ARTMs that expect two x4 PCI-Express buses this pin will float (not be connected). Two x4 PCI Express buses are intended for ARTM implementations that desire to avoid PCI-Express switch in order to reduce latency. The Netra CP32x0 ARTM-10G asserts logic ground to this signal.
PS0#	Active low ARTM present signal.PS0# shall be tied to logic GND on the ATCA blade. PS0# (Connector J33) and PS1# (Connector J30) shall be connected through a diode on the Netra CP32x0 ARTM-10G, exactly as defined in AMC.0 specification. PS1# is last mate on Power connector and PS0# is on the opposite end of the set of connectors. Logic low on PS1# indicates that ARTM is present and fully inserted.
Enable#	When low indicates to ARTM that it is fully inserted and that MMC can start execution, Logic high shall keep MMC in reset state. This signal shall have a pull-up resistor as indicated in AMC.0 specification.

## B.2.6 Connector P30, ARTM Power PIN Assignment

The Netra CP32x0 ARTM-10G derives 12V and the 3.3V stand-by voltages from the ATCA front board via the P30 connector. The P30 connector assignments also include the IPMI-L interface, PS1#. The ARTM converts the main 12V payload power into 1.2V, 1.5V, 2.5V and 3.3V by using the DC-DC converters.

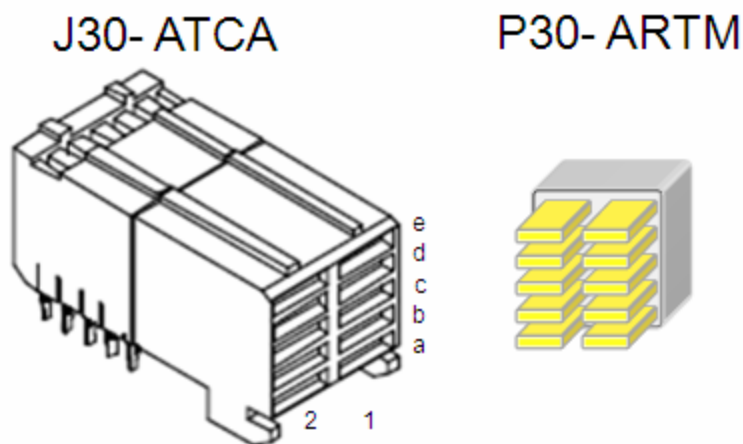
The Netra CP32x0 ARTM-10G adheres to the following maximum power ratings:

- Mean power: 25 watts
- Peak Power: 26 watts
- Standard Deviation of power from mean power: 0.25 watts

The Netra CP32x0 ARTM-10G connects to ATCA blades that use the Metral® 89096-xxx from FCI ([www.fciconnect.com](http://www.fciconnect.com)).

The Netra CP32x0 ARTM-10G P33 connector information is in [TABLE B-15](#) and [TABLE B-16](#).

**FIGURE B-6** J30/P30 Receptacle and Connector



The Netra CP32x0 ARTM-10G mates to the connector with a customized connector that features staggered pin lengths to enforce optimal mating sequence for hot plug:

- First mate: ground, power
- Second mate: all signals.
- Last mate: presence signals PS0# and PS1#

**Note** – The ARTM power connector J30 PIN-out is presented from the point of view of the ATCA board.

**TABLE B-15** P30, ARTM Power PIN Header Assignment

Row	Interface	1	2	Pin Length
e	Pwr	PS1#	NC	short
d	Pwr	+12V PP	+12V PP	long
c	IPMI	IPMI_SCL_L	IPMI_SDA_L	medium
b	Pwr	Logic_GND	+3.3V MP	long
a	Pwr	Logic_GND	Shelf_GND	long

**TABLE B-16** Connector P30, ARTM Power PIN Signal Descriptions

PS1#	Active low ARTM present signal. PS1# shall be pulled up to 3.3V Management Power on the ATCA blade. PS0# (Connector J33) and PS1# (Connector J30) shall be connected through a diode on the Netra CP32x0 ARTM-10G, exactly as defined in AMC.0 specification. PS1# is last mate on Power connector and PS0# is on the opposite end of the set of connectors. Logic low on PS1# indicates that ARTM is present and fully inserted.
IPMI_SCL_L	IPMI bus clock signal, as defined in AMC.0 specification. ARTM shall have a pull-up resistor for this signal as indicated in AMC.0 specification.
IPMI_SDA_L	IPMI bus data signal, as defined in AMC.0 specification. ARTM shall have a pull-up resistor for this signal as indicated in AMC.0 specification.
12VPP	12V Payload Power, enabled after successful e-Keying, following AMC.0 specification. ARTM shall meet requirements posted for payload power in AMC specification.
3.3V_MP	3.3V Management Power. ARTM shall meet requirements posted for management power in AMC.0 specification.
Shelf_GND	Frame/Chassis Safety Ground
Logic_GND	(Logic 0vdc). Logic Ground- Common return for Management Power Payload Power, reference potential for single ended logic signaling, and shielding for differential pair signals in the AMC Connector.



# Index

---

## Symbols

.properties command, 2-13  
/etc/aggregation.conf file, 4-11  
/etc/hosts file, 4-2  
/etc/modprobe.conf file, 4-12  
/kernel/drv/nxge.conf file, 3-10

## Numerics

1 GbE SFP Ethernet ports, B-8  
10 GbE SFP+ Ethernet ports, B-8  
10 GbE XAUI channels, 1-2  
10 GbE XAUI PIN assignments, B-12  
10/100/1000 port, PIN location, B-9  
10/100/1000BASE-T, 1-5  
10-Gigabit Ethernet ports, 1-5  
10-Gigabit XAUI interface, 1-2  
3-PIN per column, B-10

## A

ACT, LED, 1-8  
aggregation, 4-10  
aggrkey, 4-11  
airflow conditions, B-3  
AMC Module Management Controllers, A-6  
AMC modules  
    installation, 2-4  
    supporting hot-swap, A-2  
antistatic wrist strap, 2-2  
applications, terminating, 2-6  
ARTM, shut down, 2-6

ATCA shelf (chassis), 2-3  
auto negotiate, 1-5

## B

back-to-back slots, 2-3  
barcode labels, 1-9  
BLUE LED, 2-6, 2-10, A-2  
Board Info Area, A-3  
board removal, 2-6  
bond0 interface, 4-13  
bonding driver, 4-13  
boot process, 2-12

## C

certifications, B-5  
checksum, 1-5  
classification variables, 3-19  
clia command, 3-22  
commands, IPMI, A-6  
compatible node boards, 2-3  
compliance, EMI, 2-6  
configtool utility, 3-19  
connector mates, B-16  
connector PIN outs, B-1  
connectors and PIN assignments, B-6  
CPU  
    load balancing, 1-4  
    thread affinity, 1-4  
CRC32C support, 1-5

## D

- data corruption, 2-6
- depth, B-2
- device instance, specifying, 3-7
- digital components, 1-2
- direct pass-through connections, 3-2
- disabled ports, 3-2
- dladm command line utility, 3-2
- dladm show-aggr -L, displaying LACP, 4-12
- dladm show-aggr -s command, displaying statistics, 4-12
- dladm(1M) man page, 3-2
- DMA channels, 1-4, 3-2
- DMAs, 1-2
- documentation, 1-10
- driver
  - bonding, 4-13
  - nxge, 3-1
  - parameters, listing, 3-15
  - statistics, 4-6
- duplex LC fiber connector, 1-5

## E

- e-Keying port assignments, A-5
- electromagnetic emissions, B-5
- electromagnetic interference (EMI), 2-6
- electronic keying (e-Keying), 2-6, A-2
- electrostatic discharge (ESD), 2-2
- electrostatic discharge effects, B-5
- EMI, electromagnetic interference, 2-6
- empty chassis, installing boards, 2-5
- energy hazards, 2-2
- environmental specifications, B-2
- ESD precautions, 2-5
- Ethernet
  - ethtool utility, -S option, 4-8
  - interfaces, configuring, 4-2
  - interfaces, configuring Jumbo Frames, 4-8
  - management port, 1-5
  - packet traffic, 1-4
  - ports, tuning, 3-20
  - tag header format, 4-14

## F

- fail-safe flash update, A-2

- fcode firmware, 2-10
- file corruption, 2-6
- firmware, 2-10
- first mate, B-16
- flow classification, 1-4
- flow template, 3-19
- front- and rear-board installations, 2-3
- front ATCA node board, B-2
- FRU data records, A-3
- FRU information, A-2
- FRU Information Storage Definition, A-3
- full duplex ports, 1-5
- functional interconnect, 1-4

## G

- Gigabit Ethernet driver, 3-2
- GREEN LED, A-2

## H

- header parsing, 1-5
- height board, B-2
- host board
  - compatible, 2-4
  - PCI-express communication, 3-21
  - slot, 2-9
- host chassis, selection criteria, B-3
- host names, 4-2
  - guidelines, 4-2
- hostname.nxge file, 4-2
- hotswap communication, A-2
- hot-swap insertion, 2-6
- hot-swap software, 2-11
- HS, LED, 1-8

## I

- I/O interfaces, 1-3
- I2C
  - connection, A-2
  - hang recovery, A-2
  - serial bus, A-2
- ID 15, A-2
- IEEE 1101.11 mechanical standards, B-2
- IEEE 802.3 Ethernet standard, 3-2
- IEEE 802.3 MAC clients, 4-10

- IEEE 802.3ad, 4-10
- ifconfig utility, 4-7
- Inlet air temperature, A-2
- installing, 2-1, 2-7
- instance numbers, 3-10
- Intelligent Platform Management Interface (IPMI), A-1
- IP address, 4-2
- IP class, 3-19
- IPMB-L
  - address, A-2
  - bus, A-6
- IPMI commands, A-6

## J

- J32/P32 Zone-3 connector, B-9
- Jumbo Frames
  - commands, 4-5
  - configuring, 4-5
  - support, 1-5

## K

- kstat command, 4-6

## L

- L1 to L4 IPv4/v6 header parsing, 1-5
- last mate, B-16
- Layer 2 configuration. viewing, 4-5
- Layer 3 configuration, viewing, 4-6
- layer configuration, 4-5
- LC connectors, B-8
- LEDs
  - ACT, 1-8
  - AMBER, 1-8
  - back panel, 1-6
  - BLUE, 2-6, 2-10, A-2
  - Ethernet management port, 1-5
  - GREEN, 1-8, A-2
  - HS, 1-8
  - LNK, 1-8
  - OK, 1-8
  - OOS, 1-8, A-2
  - software control, 2-6
- link aggregation, 4-10
  - configuring in a Linux environment, 4-12
  - configuring in a Solaris environment, 4-10

- group, 4-10
  - status, 4-11
- LNK, LED, 1-8

## M

- MAC addresses, 1-5
- MAC clients, 4-10
- mates, B-16
- maximum link speed, 3-2
- maximum transmission unit (MTU),
  - configuration, 4-5
- midplane, 2-3
- modprobe nxge, 4-9
- Module Management Controller (MMC), 1-2, A-1

## N

- ndd utility, 3-7
- NEBS certifications, B-5
- network devices
  - listing on Linux host, 2-14
  - listing on SPARC host, 2-13
- network interface, configuring, 4-1
- networking I/O virtualization, 1-4
- node boards, compatible, 2-3
- nxge driver, 3-1
  - configuration file, 3-9
  - configuring, 3-6
  - installing on Linux OS systems, 3-4
  - installing on Solaris OS systems, 3-3
  - interfaces, 3-10
  - removing, 3-22
- nxge instance numbers, preserving, 4-2
- nxge interfaces, 4-6
- nxge.conf file, 3-9

## O

- OK, LED, 1-8
- OOS
  - LED, A-2
  - LEDs, 1-8
- OpenBoot (ok) prompt, 2-12
- operating systems, 1-10
- optical transceiver, 1-5

## P

- P30, power PIN assignment, B-15
- P32, management infrastructure PIN assignments, B-13
- P33, management infrastructure PIN assignments, B-14
- packet counts, 4-6
- packets
  - distributed, 3-19
  - sending and receiving large, 4-8
  - size, 4-5
- packing list, 2-3
- packing material, 2-3
- parallelism, extending, 1-4
- parameters
  - configuring driver on Linux, 3-15
  - configuring driver on Solaris OS, 3-6
  - listing for Linux, 3-15
  - listing for Solaris OS, 3-11
  - modifying in `nxge.conf` file, 3-10
  - modifying with `ndd` utility, 3-7
  - persistence, 3-11
  - RX coalesce, 3-18
- part number, 1-9
- pass-through, B-9
- PCI devices
  - listing on Solaris OS, 2-13
  - listing on x86 host, 2-14
- PCI Industrial Computer Manufacturers Group (PICMG) specifications, 1-3
- PCI-express signals, enabling, 3-21
- performance, tuning ports on Linux, 3-20
- persistence, 3-11
- physical dimensions, B-2
- physical point of attachment (PPA), 4-15
- PICMG AMC.0 specification, A-2
- PIN header assignment, B-16
- PINs
  - 10/100/1000 Mb management port, B-9
  - assignments, B-6
  - bent or damaged, 2-5
  - common, 2-3
  - details, B-6
- port commands, compliance, A-2
- ports, 1-3
  - 1 GbE SFP Ethernet, B-8

- 10 GbE Ethernet, B-8
- 10-Gigabit Ethernet, 1-5
- auto negotiate, 1-5
- disabled, 3-2
- e-Keying assignments, A-5
- Ethernet management, 1-5
- features, 1-4
- full duplex, 1-5
- MAC addresses, 1-5
- RS232 serial console, 1-6
- serial console management, B-9
- SFP, 3-1
  - tuning on Linux, 3-20
- power requirements, 2-4, B-2
- powering down, 2-6
- pressure drop, B-3

## R

- rails, alignment, 2-9
- RDC groups, 3-19
- reboot, 2-15
- redirect traffic, 3-2
- removing, 2-1, 2-11
- RJ-45 connector, 1-5
- RJ-45 connector jack, B-9
- RJ45 management ports, 1-2
- RS232 serial console port, 1-6
- RS232 signaling, B-9
- RX and TX packets, 1-2
- RX coalesce parameters, 3-18
- RX, signal receiver, B-10

## S

- `-S` option, `ethtool`, 4-8
- second mate, B-16
- sensor data records (SDRs), A-3
- serial console management port, B-9
- serial number, 1-9
- SFP
  - connections, 1-4
  - modules, A-2
  - ports, 3-1
- shelf manager, 2-6
- shut down, 2-6
- signals, 2-3

- receiver, B-10
- source, B-10
- single mode or multimode, B-8
- slot filler panel, 2-9
- slot usage, 2-6
- slot, compatible host, 2-3
- small form-factor pluggable (SFP), 1-5
- Solaris 10 (SPARC) Operating System, 1-10
- Solaris 10 (x86) Operating System, 1-10
- specifications, B-1
- statistics, driver, 4-6
- Sun MAC controller, 1-2

## **T**

- terminating applications, 2-6
- thermal simulation model, B-3
- TX, signal source, B-10

## **U**

- UL60950-1, compliance, 2-4
- unpacking, 2-1
- updates, firmware, 2-10
- upgrading instructions, 2-10
- utility
  - configtool, 3-19
  - ifconfig, 4-7
  - ndd, 3-7

## **V**

- verifying installation, 2-12
- virtualization, 1-4
- VLANs
  - configuration parameters, 4-13
  - configuring, 4-13
  - configuring in a Linux environment, 4-16
  - configuring in Solaris environment, 4-15
  - defining, 3-2
  - ID (VID), 4-14
  - tagged format, 4-14
  - virtual device, 4-14
- Voltage monitoring, A-2
- voltages, B-2

## **W**

- Wind-River Linux 4.0, 1-10

## **X**

- x8 PCIe, 1-2

## **Z**

- Zone 3 connector complex, 2-3
- Zone 3 connectors, B-10

