![Sun microsystems logo]

Sun™ Quad GbE UTP x8 PCIe ExpressModule User's Guide

Sun Microsystems, Inc.
www.sun.com

Part No. 820-1609-12
December 2009, Revision A

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

Adobe PostScript

Contents

Figures

Tables

Declaration of Conformity

Compliance Model Number: ATLS1QGC-EM
Product Family Name: Sun Quad GbE UTP x8 PCIe ExpressModule (X7287A-z)

EMC

USA—FCC Class A

This equipment complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This equipment may not cause harmful interference.
2. This equipment must accept any interference that may cause undesired operation.

European Union

This equipment complies with the following requirements of the EMC Directive 89/336/EEC:

As Information Technology Equipment (ITE) Class A per (as applicable):

EN 55022:1994 +A1:1995 +A2:1997	Class A
EN 61000-3-2:2000	Pass
EN 61000-3-3:1995 +A1:2000	Pass
EN 55024:1998 +A1:2001 +A2:2003	Required Limits:
IEC 61000-4-2	4 kV (Direct), 8kV (Air)
IEC 61000-4-3	3 V/m
IEC 61000-4-4	1 kV AC Power Lines, 0.5 kV Signal and DC Power Lines
IEC 61000-4-5	1 kV AC Line-Line and Outdoor Signal Lines, 2 kV AC Line-Gnd, 0.5 kV DC Power Lines
IEC 61000-4-6	3 V
IEC 61000-4-8	1 A/m
IEC 61000-4-11	Pass

Safety

This equipment complies with the following requirements of the Low Voltage Directive 73/23/EEC:

EC Type Examination Certificates:
    EN 60950-1:2001, 1st Edition
    IEC 60950-1:2001, 1st Edition
    Evaluated to all CB Countries
    UL and cUL/CSA 60950-1:2001, CSA C22.2 No. 60950-00 File:E138989-A92   Vol. 54

Supplementary Information

This equipment was tested and complies with all the requirements for the CE Mark.
This equipment complies with the Restriction of Hazardous Substances (RoHS) directive 2002/95/EC.

/S/ _____   DATE
Dennis P. Symanski
Worldwide Compliance Engineering
Sun Microsystems, Inc.
4150 Network Circle, MPK15-102
Santa Clara, CA 95054   U.S.A.
Tel: 650-786-3255
Fax: 650-786-3723

/S/ _____   DATE
Donald Cameron
Program Manager/Customer Quality
Sun Microsystems Scotland, Limited
Blackness Road, Phase I, Main Bldg.
Springfield, EH49 7LR
Scotland, United Kingdom
Tel: +44 1 506 672 539   Fax: +44 1 506 670 011

Preface

This guide provides instructions for installing both the hardware and software for the Sun Quad GbE UTP x8 PCIe ExpressModule. This guide also describes how to configure the `nxge` driver, which controls the ExpressModule.

These instructions are designed for enterprise system administrators with experience in installing network hardware and software.

How This Document Is Organized

Chapter 1 provides an overview of the Sun Quad GbE UTP x8 PCIe ExpressModule.

Chapter 2 explains how to install the `nxge` device driver software.

Chapter 3 describes how to install the ExpressModule in your system and verify that it has been installed correctly.

Chapter 4 describes how to configure the network host files after you install the ExpressModule on your system. This chapter also describes how to set up a Gigabit Ethernet network on a diskless client and install the Solaris Operating System over a Gigabit Ethernet network.

Chapter 5 explains how to set the `nxge` device driver parameters to customize each device in your system.

Chapter 6 describes how to enable the Jumbo Frames feature.

Chapter 7 describes how to configure link aggregation.

Chapter 8 explains Virtual Local Area Networks (VLANs) in detail and provides configuration instructions and examples.

Appendix A lists the specifications for the Sun Quad GbE UTP x8 PCIe ExpressModule.

Appendix B provides an overview of the SunVTS™ diagnostic application and instructions for updating the SunVTS software to recognize the ExpressModule.

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

Shell Prompts

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

Typographic Conventions

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

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

Related Documentation

The documents listed as online are available at:

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

Application	Title	Part Number	Format	Location
Release Notes	*Sun Quad GbE UTP x8 PCIe ExpressModule Release Notes*	820-1610	PDF HTML	Online
Getting Started	*Sun Quad GbE UTP x8 PCIe ExpressModule Getting Started Guide*	820-1611	Hardcopy	Ship kit
Safety and compliance	*Safety and Compliance Manual*	816-7190	PDF HTML	Online

Documentation, Support, and Training

Sun Function	URL
Documentation	http://www.sun.com/documentation/
Support	http://www.sun.com/support/
Training	http://www.sun.com/training/

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:

Sun Quad GbE UTP x8 PCIe ExpressModule User's Guide, part number 820-1609-12

Product Overview

This chapter provides an overview of the Sun Quad GbE UTP x8 PCIe ExpressModule, including:

Shipping Kit Contents

The carton in which your Sun Quad GbE UTP x8 PCIe ExpressModule was shipped should contain the following items:

- Sun Quad GbE UTP x8 PCIe ExpressModule

- *Sun Quad GbE UTP x8 PCIe ExpressModule Getting Started Guide*.

Product Description

The Sun Quad GbE UTP x8 PCIe ExpressModule provides high-performance packet processing capability optimized for throughput computing and throughput networking architecture. The ExpressModule provides four 10/100/1000BASE-T compliant Ethernet copper ports.

FIGURE 1-1 shows the ExpressModule.

FIGURE 1-1   Sun Quad GbE UTP x8 PCIe ExpressModule
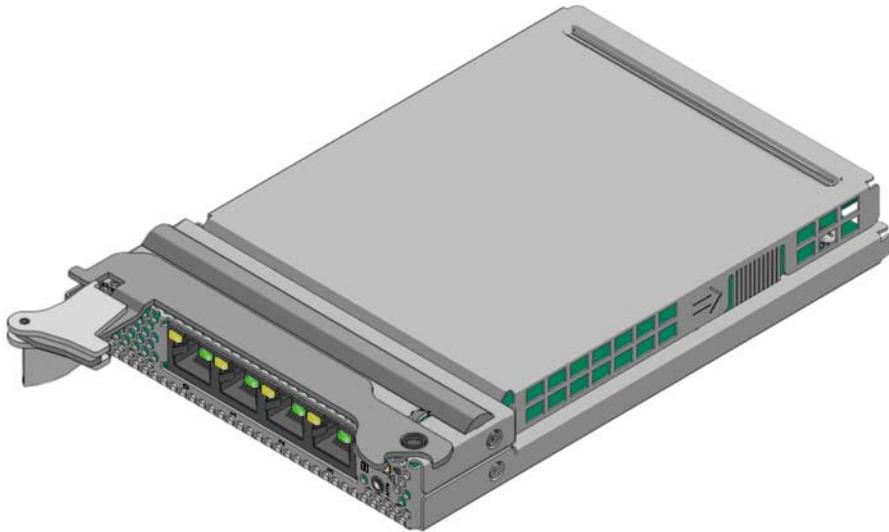
Hardware and Software Requirements

Before installing the ExpressModule, ensure that your system meets the hardware and software requirements. TABLE 1-1 lists the supported hardware and software.

TABLE 1-1   Hardware and Software Requirements

Requirements	Hardware or Software
Hardware	Sun Blade x6220 server module, Sun Blade T6300 server module, Sun Blade x8400 server module, Sun Blade x8420 server module
Operating System	Solaris 10 11/06 Operating System SuSE Linux Enterprise Server 10 RedHat Enterprise Linux 4.0_u4 RedHat Enterprise Linux 4.0_u3

FIGURE 1-2   Sun Quad GbE UTP x8 PCIe ExpressModule Ports
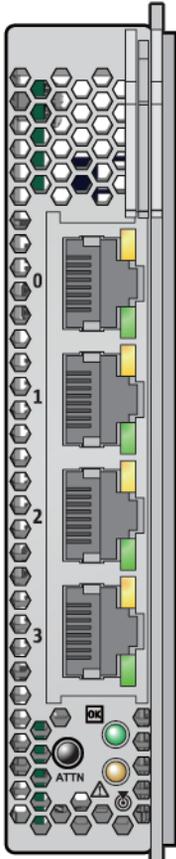
Front Panel LED Displays on the ExpressModule

Two indicators and four sets of LEDs (two for each port) are displayed on the front panel of the Sun Quad GbE UTP x8 PCIe ExpressModule. FIGURE 1-2 shows the location of the LEDs. TABLE 1-2 describes the meaning of each LED.

TABLE 1-2   Front Panel LEDs for the Sun Quad GbE UTP x8 PCIe ExpressModule

Label	Meaning if Lit	Color	Description
OK	Power is OK	Green	Hot plug power OK indicator
	Attention required	Amber	Hot plug ExpressModule attention indicator
ATTN	Attention Switch is a recessed pushbutton	N/A	Hot plug attention switch
LINK 0	Link is up on Port 0	Green	Indicator for GbE
ACT 0	Activity on Port 0	Amber	RX/TX activity indicator
LINK 1	Link is up on Port 1	Green	Indicator for GbE
ACT 1	Activity on Port 1	Amber	RX/TX activity indicator
LINK 2	Link is up on Port 2	Green	Indicator for 1GbE
ACT 2	Activity on Port 2	Amber	RX/TX activity indicator
LINK 3	Link is up on Port 3	Green	Indicator for GbE
ACT 3	Activity on Port 3	Amber	RX/TX activity indicator

Patch Requirements

The SunBlade T6300 was released on Solaris 10 11/06 Operating System with patches. For more information on the SunOS 5.10: SunBlade T6300 & Sun Fire (T1000, T2000) platform patch, refer to the *Sun Blade T6300 Server Module Product Notes* (820-0278) at the followingweb site:

```
http://www.sun.com/products-n-
solutions/hardware/docs/Servers/blade_servers/
```

If the patches are not available on SunSolve, contact your local sales or service representative.

Patches and Updates

Check the Sun Update Connection to ensure that you have the latest recommended patch clusters and security patches. You can download the latest recommended patch clusters and security patches at:

```
http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage
```

Features

The Sun Quad GbE UTP x8 PCIe ExpressModule addresses the following requirements and provides additional features and benefits:

- Four 10/100/1000BASE-T Ethernet interfaces (full-duplex only)
- IEEE 802.3ab compliant
- Uses Sun ASIC and software for innovative throughput networking design
- Networking I/O virtualization supporting Solaris LDOM1.0
- 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 and strand
- CPU/thread affinity and CPU load balancing at L1,L2,L3 and L4
- Jumbo frames support (up to 9KBytes)
- IPv4/IPv6 and IPMP support
- TCP/UDP/IP checksum and CRC32 support
- IEEE 802.1Q VLAN support

Installing and Setting Up the Driver

This chapter explains how to download and install the `nxge` driver. The `nxge` Gigabit Ethernet driver (`nxge(7D)`) is a multi-threaded, loadable, clonable, GLD-based STREAMS driver. The `nxge` driver is managed by the `dladm(1M)` 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.

This chapter contains the following sections:

- "Downloading and Installing the Driver on a Solaris SPARC or x86 Platform" on page 7

Downloading and Installing the Driver on a Solaris SPARC or x86 Platform

If your system uses the Solaris SPARC or x86 operating system you will need to download and install the `nxge` device driver for Solaris platforms.

▼ To Download the Driver on a Solaris Platform

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

   http://www.sun.com/products/networking/ethernet/index.html

2. **Uncompress the gzipped tar file:**

```
# gunzip nxge.tar.gz
```

3. **Unpack the tar file:**

```
# tar xvf nxge.tar
```

   a. **For SPARC systems, change to the following directory:**

```
# cd 10_GigabitEthernet/Solaris_10/sparc/Packages
```

b. **For x86 systems:**

```
# cd 10_GigabitEthernet/Solaris_10/i386/Packages
```

4. **For SPARC systems, determine which architecture your system is running:**

```
# uname -m
```

   a. **For sun4v systems, install the software packages by typing the following at the command line:**

```
# /usr/sbin/pkgadd -d SUNWnxge.v SUNWnxgem
```

   b. **For sun4u systems, install the software packages by typing the following at the command line:**

```
# /usr/sbin/pkgadd -d SUNWnxge.u SUNWnxgem
```

5. **For x86 systems, install the software packages by typing the following at the command line:**

```
# /usr/sbin/pkgadd -d
```

A menu similar to the following displays:

```
The following packages are available:

  1  SUNWnxge.u     Sun x8 10G/1G Ethernet Adapter Driver
                    (sparc.sun4u) 1.0,REV=2007.01.12.10.0
  2  SUNWnxge.v     Sun x8 10G/1G Ethernet Adapter Driver
                    (sparc.sun4v) 1.0,REV=2007.01.12.10.0
  3  SUNWnxgem      Sun x8 10G/1G Ethernet Adapter Driver Man Pages
                    (all) 1.0,REV=2007.01.12.10.0
Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]:
```

6. **Select the packages you want to install:**

■ Press Return or type **all** to accept the default and install all packages.

■ Type the specific numbers, separated by a space, if you prefer not to install any optional packages.

7. **Verify that the** nxge **driver is installed on the system:**

```
Sun x8 10G Ethernet Adapter Driver(sparc.sun4v) 1.0,REV=2007.01.06.10.0
Copyright 2007 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.
   8 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 <SUNWnxge> [y,n,?] y

Installing Sun x8 10G Ethernet Adapter Driver as <SUNWnxge>

## Installing part 1 of 1.
/platform/sun4v/kernel/drv/sparcv9/nxge
[ verifying class <none> ]
[ verifying class <preserve> ]
## Executing postinstall script.

Installation of <SUNWnxge> was successful.
```

▼ To Remove the Driver From a Solaris SPARC Platform

1. **Determine the driver packages:**

```
# pkginfo | grep SUNWnxge
system      SUNWnxge       Sun x8 10G Ethernet Adapter Driver
```

2. **Remove the driver packages:**

```
# pkgrm SUNWnxge  plus any other packages from the previous command
```

Downloading and Installing the Driver on a Linux Platform

1. **Login to your system.**

2. **Download the driver RPM for your operating system:**

   http://www.sun.com/download/products.xml?id=44eb1efd

   For example:

   ```
   nxge-1.0-1.x86_64.rpm
   ```

3. **Discover the network interfaces before adding the package by using the `ifconfig -a` command:**

   ```
   # ifconfig -a |grep eth
   eth0      Link encap:Ethernet  HWaddr 00:14:4F:20:F1:DC
   eth1      Link encap:Ethernet  HWaddr 00:14:4F:20:F1:DD
   eth2      Link encap:Ethernet  HWaddr 00:14:4F:20:F1:DE
   eth3      Link encap:Ethernet  HWaddr 00:14:4F:20:F1:DF
   ```

4. **Use the `rpm` tool to install the driver on SuSe and RedHat Linux.**

   ```
   # rpm -ivh /tmp/RHEL4U4-large/RPMS/x86_64/nxge-1.0-1.x86_64.rpm
   Preparing...          ###########################################
   1:nxge                ###########################################
   ```

Note – In RHEL5.0 and later releases, the driver is packaged in the kmod driver binary package format. This packaging allows forward and backward driver binary compatibility within the same flavors of RHEL5 releases. Driver packages nxge version 2.0.0 and later are not compatible with the earlier nxge-1.x-x releases. To upgrade from 1.x-x to 2.x.x rpm package, it is necessary to remove nxge-1.x-x package before installing the 2.x.x package.

The RHEL5 nxge 2.x.x kmod package contains two separate packages: One for the driver and another one for applications like nxge_config. You must install both of the following packages:

```
kmod-nxge-rhel-2.0-1.x86_64.rpm     (driver binary package)
nxge-apps-rhel-2.0-1.x86_64.rpm   (application package)
```

To install the complete package, enter the following:

```
# rpm -ivh nxge-apps-rhel-2.0-1.x86_64.rpm kmod-nxge-rhel-2.0-1.x86_64.rpm
```

To ensure that the driver is loaded after the rpm installation, enter the following:

```
# modprobe nxge
```

5. **Verify the new network interface instances corresponding to the Sun Quad GbE UTP x8 PCIe ExpressModule:**

```
# ifconfig -a |grep eth
eth0      Link encap:Ethernet   HWaddr 00:14:4F:20:F1:DC
eth1      Link encap:Ethernet   HWaddr 00:14:4F:20:F1:DD
eth2      Link encap:Ethernet   HWaddr 00:14:4F:20:F1:DE
eth3      Link encap:Ethernet   HWaddr 00:14:4F:20:F1:DF
eth4      Link encap:Ethernet   HWaddr 00:14:4F:6C:78:E8
eth5      Link encap:Ethernet   HWaddr 00:14:4F:6C:78:E9
```

The Sun Quad GbE UTP x8 PCIe ExpressModule instances, eth4 and eth5, are shown in **_bold italics_**.

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

```
alias eth4 nxge
alias eth5 nxge
```

7. **Use the** `ethtool` **command to check the parameter configurations that apply to the** `nxge` **driver.**

```
(For 10G)
# ethtool -i eth4
driver: nxge
version: 2.0.1
firmware-version:  2XGF PXE1.47 FCode 3.9 07/04/24
bus-info: 0000:84:00.0

(Equivalent for 1G)
# ethtool -i eth4
driver: nxge
version: 2.0.1
firmware-version: QGC PXE1.47 FCode 3.9 07/04/24
bus-info: 0000:02:00.2
```

▼ To Remove the Driver From a Linux Platform

● **To remove the driver packages from a Linux Platform use the** `rpm -e` **command:**

```
# rpm -e nxge-1.0-1
```

Installing the ExpressModule

This chapter describes how to install the Sun Quad GbE UTP x8 PCIe ExpressModule in your system and verify that it has been installed correctly.

This chapter contains the following section:

- "Installing the ExpressModule" on page 15
- "Verifying the Hardware Installation" on page 18

Installing the ExpressModule

If you are installing the Sun Quad GbE UTP x8 PCIe ExpressModule into a machine running Solaris 10, you *must* install the software *before* you install the hardware.

The following instructions describe the basic tasks required to install the ExpressModule. Refer to your system installation or service manual for detailed ExpressModule installation instructions.

Note – To maintain proper cooling for the ExpressModule in your chassis, all ExpressModule slots must be filled with either operating ExpressModules or filler panels.

▼ To Install the ExpressModule

1. **Halt and power off your system.**

2. **Power off all peripherals connected to your system.**

3. **Attach the adhesive copper strip of the antistatic wrist strap to the metal casing of the power supply. Wrap the other end twice around your wrist, with the adhesive side against your skin.**

4. **Remove the filler panel from the ExpressModule opening.**
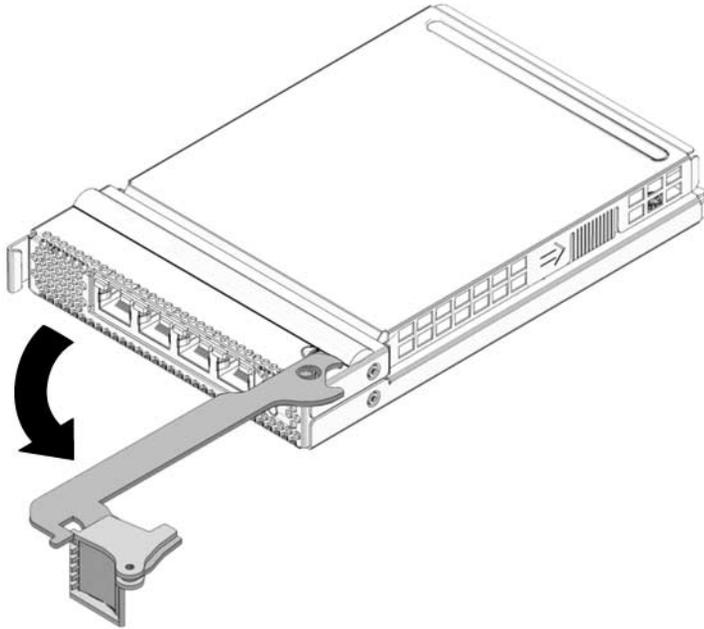
5. **Open the latch on the ExpressModule.**
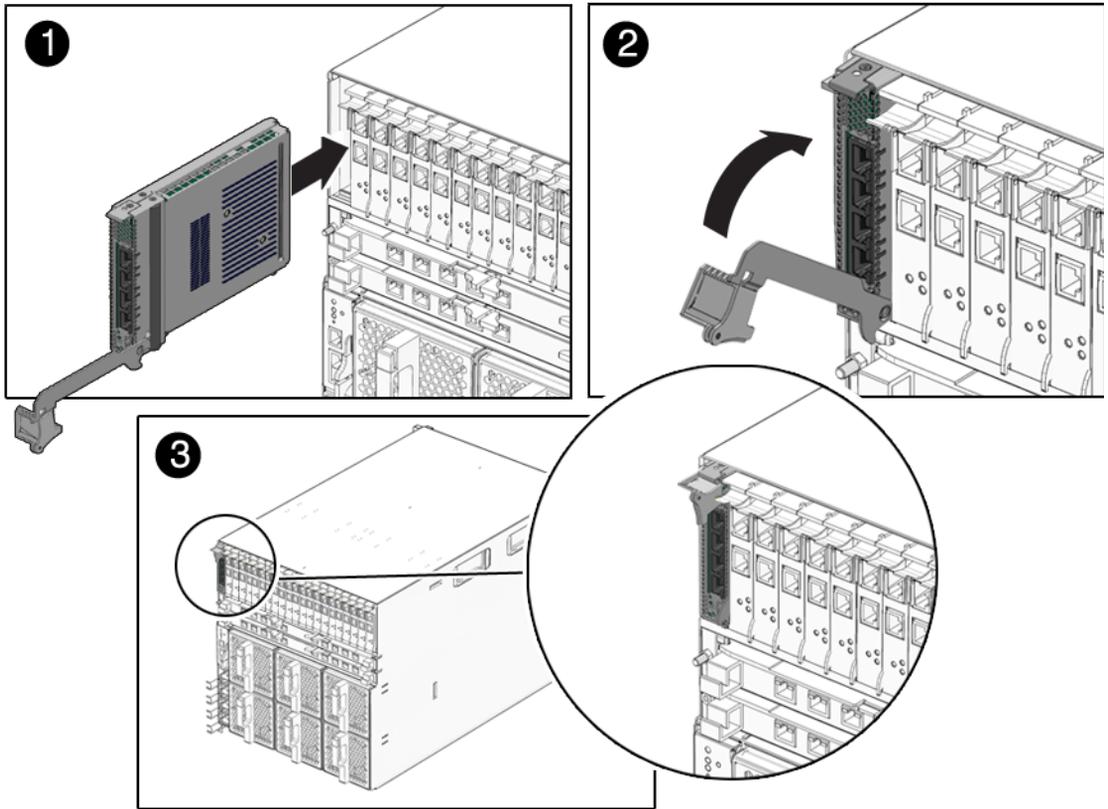
6. **Align the ExpressModule with the vacant ExpressModule slot (1).**

   Ensure that the ExpressModule's indicator lights on the front panel are facing toward you and that the ExpressModule ejector lever on the bottom is fully opened.

7. **Slide the ExpressModule into the vacant ExpressModule chassis slot until the ejector lever engages and starts to close (2).**

   Failure to align the ExpressModule correctly can result in damage with the ExpressModule's internal connection to the chassis midplane.

8. **Complete the installation by closing the ejector lever until the latch snaps into place (3).**

Caution – Do not use excessive force when installing the ExpressModule into the slot. You might damage the ExpressModule's connector. If the ExpressModule does not seat properly when you apply even pressure, remove the and carefully reinstall the ExpressModule.

9. **Detach the wrist strap.**
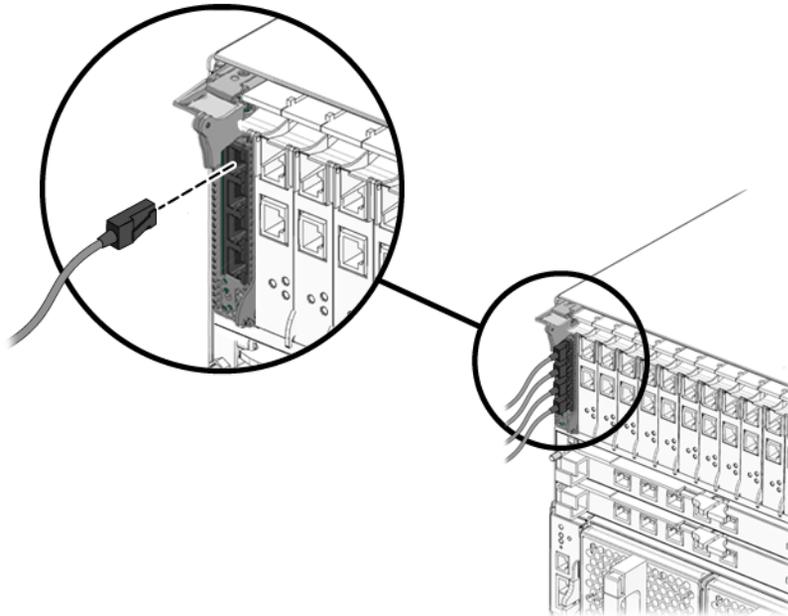
10. **Connect the Ethernet cables.**

Verifying the Hardware Installation

After you have installed the ExpressModule, but *before* you boot your system, perform the following tasks to verify the installation. Refer to the your Solaris documentation for the detailed instructions.

▼ To Verify the Hardware Installation

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

2. **List the network devices on your system.**

```
ok show-nets
a) /pci@7c0/pci@0/pci@9/network@0,3
b) /pci@7c0/pci@0/pci@9/network@0,2
c) /pci@7c0/pci@0/pci@9/network@0,1
d) /pci@7c0/pci@0/pci@9/network@0
e) /pci@7c0/pci@0/pci@8/network@0,1
f) /pci@7c0/pci@0/pci@8/network@0
g) /pci@7c0/pci@0/pci@2/network@0,1
h) /pci@7c0/pci@0/pci@2/network@0
i) /pci@780/pci@0/pci@1/network@0,1
j) /pci@780/pci@0/pci@1/network@0
m) MORE SELECTIONS
q) NO SELECTION
```

Checking the `.properties` output for each device is the surest way to identify the device. Since the Sun Quad GbE UTP x8 PCIe ExpressModule has four ports, the `show-nets` command displays four lines for the ExpressModule, usually `/pci@7c0/pci@0/pci@8` or `/pci@7c0/pci@0/pci@9` correspond to PCIe slots, so look at those devices first.

```
a) /pci@7c0/pci@0/pci@9/network@0,3
b) /pci@7c0/pci@0/pci@9/network@0,2
c) /pci@7c0/pci@0/pci@9/network@0,1
d) /pci@7c0/pci@0/pci@9/network@0
```

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

3. **View the device that you installed.**

Using the previous example, type:

```
ok cd /pci@7c0/pci@0/pci@9/network@0
```

4. **View the** `.properties` **file for a list of device properties.**

The `.properties` command displays the specific information about the installed ExpressModule. For this ExpressModule, your output will be similar to the following:

```
ok .properties
assigned-addresses     82770010 00000000 21000000 00000000
01000000
                       82770018 00000000 20300000 00000000 00008000
                       82770020 00000000 20308000 00000000 00008000
                       82770030 00000000 20400000 00000000 00100000
local-mac-address      00 14 4f 83 5b 9a
phy-type               mif
reg                    00770000 00000000 00000000 00000000 00000000
                       03770010 00000000 00000000 00000000 01000000
                       03770018 00000000 00000000 00000000 00008000
                       03770020 00000000 00000000 00000000 00008000
                       02770030 00000000 00000000 00000000 00100000
version                QGC-PEM 1G Ethernet Adapter FCode 3.9
07/04/24
board-model            501-7765-02
model                  SUNW,pcie-qgc-pem
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       00007a00
interrupts             00000001
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 finish looking at the** `.properties` **values:**

```
ok device-end
```

▼ To Reboot the System

After verifying the ExpressModule installation, use the `boot -r` command to perform a reconfiguration boot on your system.

● **Type the following**

```
ok boot -r
```

Network Configuration

This chapter describes how to configure the network host files after you install the ExpressModule on your system. This chapter also describes how to set up a Gigabit Ethernet network on a diskless client and install the Solaris Operating System over a Gigabit Ethernet network.

This chapter contains the following sections:

Configuring the Network Host Files

After installing the driver software, you must create a `hostname.nxge`*number* file for the ExpressModule's Ethernet interface. You must also create both an IP address and a host name for the ExpressModule's Ethernet interface in the `/etc/hosts` file.

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 Sun Quad GbE UTP x8 PCIe ExpressModule in one slot is removed and replaced by a different `nxge` driver-based network interface product, and later a Sun Quad GbE UTP x8 PCIe ExpressModule is reinstalled into that same slot.

---

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

```
# 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@7c0/pci@0/pci@9/network@0,2" 2 "nxge"
"/pci@7c0/pci@0/pci@9/network@0,3" 3 "nxge"
```

In this example, the device instance is from a Sun Quad GbE UTP x8 PCIe ExpressModule installed in Slot 1. The instance numbers are shown in **_bold italics_**.

Be sure to write down your device path and instance, which in the example is `"/pci@7c0/pci@0/pci@9/network@0"` 0. Your device path and instance will be similar. You need this information to make changes to the `nxge.conf` file. See "Setting Parameters Using the `nxge.conf` File" on page 34.

2. **Set up the ExpressModule's** `nxge` **interface.**

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 ExpressModule's IP address:

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

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

3. **(Optional) If you want a setup that remains the same after you reboot, create an** /etc/hostname.nxge*number* **file, where** *number* **is the instance number of the nxge interface you plan to use.**

To use the ExpressModule's `nxge` interface in the Step 1 example, create an `/etc/hostname.nxge0` file, where `0` is the number of the nxge interface. If the instance number were `1`, the filename would be `/etc/hostname.nxge1`. The filename would be `/etc/hostname.nxge2`, for instance 2, and so on.

Do not create an /etc/hostname.nxge*number* file for a Sun Quad GbE UTP x8 PCIe ExpressModule interface you plan to leave unused.

Follow these guidelines for the host name:

- The /etc/hostname.nxge*number* 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`*number* file required for a system called `zardoz` that has an Sun Quad GbE UTP x8 PCIe ExpressModule. Note the system name changes for each interface.

```
# cat /etc/hostname.nxge0
zardoz
# cat /etc/hostname.nxge1
zardoz-11
# cat /etc/hostname.nxge2
zardoz-12
# cat /etc/hostname.nxge3
zardoz-13
```

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

   For example:

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1      localhost
129.144.10.57 zardoz     loghost
129.144.11.83 zardoz-11
129.144.12.92 zardoz-12
129.144.13.45 zardoz-13
```

Setting Up a Gigabit Ethernet Port on a Diskless Client System

Before you can boot and operate a diskless client system across a Gigabit Ethernet port, you must first install the Gigabit Ethernet software packages into the root directory of the diskless client. You can find the Gigabit Ethernet software packages at the following web site:

http://www.sun.com/products/networking/ethernet/index.html

Refer to the *Solaris Advanced Installation Guide* and the *System Administration Guide* for more information about installing and administering diskless client systems.

▼ To Set Up a Gigabit Ethernet Port on a Diskless Client

1. **Locate the root directory of the diskless client on the host server.**

   The root directory of the diskless client system is commonly installed in the host server's /export/root/*client-name* directory, where *client-name* is the diskless client's host name. In this procedure, the root directory is:

   ```
   /export/root/client-name
   ```

2. **Download the software for the ExpressModule onto the server's drive.**

3. **Use the** pkgadd -R **command to install the software packages to the diskless client's root directory on the server.**

   Install the software packages to the client's root directory.

   ```
   ok pkgadd -R
   ```

4. **Create a** hostname.nxge*number* **file in the diskless client's root directory.**

   Create an /export/root/*client-name*/etc/hostname.nxge*number* file for the Gigabit Ethernet interface. See "Configuring the Network Host Files" on page 23 for instructions.

5. **Edit the** hosts **file in the diskless client's root directory.**

   Edit the /export/root/*client-name*/etc/hosts file to include the IP address of the Gigabit Ethernet interface. See "Configuring the Network Host Files" on page 23 for instructions.

6. **Set the MAC address on the server side and rebuild the device tree if you want to boot from the Gigabit Ethernet port.**

7. **Boot the diskless client from the Gigabit Ethernet port:**

   ```
   ok boot path-to-device:link-param
   ```

Installing the Solaris Operating System Over a Gigabit Ethernet Network

The *Solaris Advanced Installation Guide* describes the full procedure for installing the Solaris Operating System over the network. The following procedure assumes that you have created an install server, which contains the image of the Solaris CD, and that you have set up the client system to be installed over the network.

Before you can install the Solaris Operating System on a client system with a Gigabit Ethernet ExpressModule, you must first add the Gigabit Ethernet software packages to the install server. See Chapter 2 for the location of these software packages.

Note – Refer to the *Solaris Advanced Installation Guide* for more information about installing the Solaris Operating System over the network.

▼ To Install the Solaris Operating System Over a Gigabit Ethernet Network

1. **Prepare the install server and client system to install the Solaris Operating System over the network.**

   The *Solaris Advanced Installation Guide* describes how to create the install server and set up the client systems.

Note – If you want to install the client system over a network that is not part of the same subnet, you must also create a boot server. The *Solaris Advanced Installation Guide* describes how to create a boot server.

---

2. **Find the root directory of the client system.**

   The client system's root directory can be found in the install server's
   /etc/bootparams file. Use the grep command to search this file for the root
   directory.

   ```
   # grep client-name /etc/bootparams
   client_name root=server-name:/netinstall/Solaris_10/Tools/Boot
   install=server-name:/netinstall boottype=:in rootopts=:rsize=32768
   ```

   In this example, the root directory for the Solaris 10 client is /netinstall. In
   Step 4, you would replace *root-directory* with /netinstall.

Note – If the root directory is not found in the /etc/bootparams file, refer to the
Solaris Advanced Installation Guide for configuration instructions.

---

3. **Download the** nxge **driver onto the install server's hard drive.**

   The package is a folder SUNWnxge.v or SUNWnxge.u, which you can download
   from the following web site:

   http://www.sun.com/products/networking/ethernet/index.html

4. **On the install server, install the** nxge **software to the client's root directory, as
   determined in** Step 2**.**

   Replace *root-directory* with the location of the client's root directory.

   ```
   # cd location where you downloaded the packages
   # ls SUNWnxge*
   # pkgadd -R root-directory/Solaris_10/Tools/Boot -d . SUNWnxge.v
   ```

Note – Perform the following steps on the *client* system.

---

5. **Shut down and halt the client system.**

   Use the shutdown command to obtain the OpenBoot (ok) prompt.

   ```
   # shutdown -i0 -g0 -y
   . . .
   (shutdown command messages omitted)
   . . .
   ok
   ```

6. **At the** ok **prompt, use the** show-nets **command to find the device path of the Gigabit Ethernet device.**

   The show-nets command lists the system devices. You should see the full paths and names of the network devices, similar to the example below.

   ```
   ok show-nets
   a) /pci@7c0/pci@0/pci@9/network@0,3
   b) /pci@7c0/pci@0/pci@9/network@0,2
   c) /pci@7c0/pci@0/pci@9/network@0,1
   d) /pci@7c0/pci@0/pci@9/network@0
   e) /pci@7c0/pci@0/pci@8/network@0,1
   f) /pci@7c0/pci@0/pci@8/network@0
   g) /pci@7c0/pci@0/pci@2/network@0,1
   h) /pci@7c0/pci@0/pci@2/network@0
   i) /pci@780/pci@0/pci@1/network@0,1
   j) /pci@780/pci@0/pci@1/network@0
   m) MORE SELECTIONS
   q) NO SELECTION
   Enter Selection, q to quit:
   ```

7. **At the** ok **prompt, boot the client system using the full device path of the Gigabit Ethernet device. For example:**

   ```
   ok boot /pci@7c0/pci@0/pci@9/network@0
   ```

8. **Proceed with the Solaris Operating System installation.**

   Refer to the *Solaris Advanced Installation Guide* for more information about installing the Solaris Operating System over the network.

9. **After installing the Solaris Operating System, install the** nxge **driver software on the client system.**

   The software installed in Step 4 is required to boot the client system over the Gigabit Ethernet interface. You now need to install the software in order for the operating system to use the client's Gigabit Ethernet interfaces in normal operation.

   Before installing the nxge driver, ensure that the client system does not already have the driver installed. Use the pkginfo command to see if the nxge software packages are installed on the client system.

   ```
   # pkginfo | grep SUNWnxge
   ```

   ■ If the software is installed, the previous command will return the package name you typed in. In that case, skip to Step 10.

- If the software is not installed, install the software from the download center.

  See Chapter 2 for instructions on installing the required software packages.

10. **Confirm that the network host files have been configured correctly during the Solaris installation.**

    Although the Solaris software installation creates the client's network configuration files, you might need to edit these files to match your specific networking environment. See "Configuring the Network Host Files" on page 23 for more information about editing these files.

11. **Show configuration information for all data-links or the specified data-link.**

    By default, the system is configured to have one data-link for each known network device.

```
# dladm show-dev
e1000g0        link: up       speed: 1000   Mbps        duplex: full
e1000g1        link: down      speed: 0     Mbps        duplex: half
nxge2          link: up       speed: 1000   Mbps        duplex: full
nxge3          link: up       speed: 1000   Mbps        duplex: full
nxge4          link: up       speed: 1000   Mbps        duplex: full
nxge5          link: up       speed: 1000   Mbps        duplex: full
e1000g2        link: down      speed: 0     Mbps        duplex: half
e1000g3        link: down      speed: 0     Mbps        duplex: half
```

Configuring the `nxge` Device Driver Parameters

The `nxge` device driver controls the Sun Quad Gigabit Ethernet interfaces. You can manually set the `nxge` driver parameters to customize each device in your system.

This chapter lists the available device driver parameters and describes how you can set these parameters.

`nxge` Hardware and Software Overview

The Sun Quad GbE UTP x8 PCIe ExpressModule provides four Gigabit full-duplex networking interfaces. The device driver automatically sets the link speed to 1000 Mbit/sec and conforms to the IEEE 802.3ad Ethernet standard. Each interface has 4 receive DMA channels and 6 transmit DMA channels to allow for parallel processing of the packets.

The Sun Quad GbE UTP x8 PCIe ExpressModule extends CPU and OS parallelism to networking 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 ExpressModule enables a one-to-one correlation of Rx and Tx packets across the same TCP connection. This can help avoid cross-calls and context switching to deliver greater performance while reducing the need for CPU resources to support I/O processing. The Sun Quad GbE UTP x8 PCIe ExpressModule uses the Sun innovative MAC controller.

Setting `nxge` Driver Parameters on a Solaris Platform

You can set the `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.

▼ To Set the Driver Parameters

To set parameters so they remain in effect after you reboot the system, perform the following procedure:

1. **Create an nxge.conf file:**

```
/platform/sun4u/kernel/drv/nxge.conf
```

2. **Add parameter values to this file when you need to set a particular parameter for a device in the system.**

Setting Parameters Using the `ndd` Utility

Use the `ndd` utility to configure parameters that are valid until you reboot the system.

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

Noninteractive and Interactive Modes

You can use the `ndd` utility in two modes:

- Noninteractive

- Interactive

In Noninteractive mode, you invoke the utility to execute a specific command. Once the command is executed, you exit the utility. In Interactive mode, you can use the utility to get or set more than one parameter value. Refer to the ndd(1M) man page for more information.

▼ To Specify Device Instances for the ndd Utility

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

1. **Check the** /etc/path_to_inst **file to identify the instance associated with a particular 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@7c0/pci@0/pci@9/network@0,2" 2 "nxge"
"/pci@7c0/pci@0/pci@9/network@0,3" 3 "nxge"
```

▼ To Specify Parameter Values Using the ndd Utility

This section describes how to modify and display parameter values.

1. **Use the** -set **option to modify a parameter.**

   If you invoke the ndd utility with the -set option, the utility passes *value*, which must be specified, down to the named /dev/nxge*driver-instance*, and assigns the value to the parameter:

```
# ndd -set /dev/nxgenumber parameter-value
```

   where *number* is the driver instance, for example /dev/nxge0, /dev/nxge1.

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

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

```
# ndd /dev/nxgenumber parameter
```

▼ To Use the `ndd` Utility in Interactive Mode

1. Modify a parameter value in interactive mode by specifying `ndd` `/dev/nxge`*number***:**

```
# ndd /dev/nxge0
name to get/set? (Enter the parameter name or ? to view all
parameters)
```

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

2. List all the parameters supported by the `nxge` **driver, by typing** `?`**.**

```
# ndd /dev/nxge1 name to get/set ?
?                               (read only)
function_number                 (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)
```

Setting Parameters Using the `nxge.conf` File

Specify the driver parameter properties for each device by creating a `nxge.conf` file in the `/platform/sun4v/kernel/drv/sparcv9/` directory. Use a `nxge.conf` file when you need to set a particular parameter for a device in the system.

The man pages for prtconf(1M) and driver.conf(4) include additional details. The next procedure shows an example of setting parameters in a nxge.conf file.

▼ To Access a Man Page

● **To access any man page, type the** man **command plus the name of the man page.**

For example, to access man pages for prtconf(1M), type:

```
% man prtconf
```

▼ To Set Driver Parameters Using an nxge.conf File

1. **Obtain the hardware path names for the** nxge **devices in the device tree.**

   a. **Check the** /etc/driver_aliases **file to identify the name associated with a particular device:**

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

   b. **Locate the path names and the associated instance numbers in the** /etc/path_to_inst **file.**

```
# 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@7c0/pci@0/pci@9/network@0,2" 2 "nxge"
"/pci@7c0/pci@0/pci@9/network@0,3" 3 "nxge"
```

   In this example:

   - The first part within the double quotes specifies the hardware node name in the device tree.
   - The number not enclosed in quotes is the instance number (shown in **_bold italics_** for emphasis).
   - The last part in double quotes is the driver name.

To identify a PCIe device unambiguously in the `nxge.conf` file, use the name, parent name, and the unit-address for the device. Refer to the `pci(4)` man page for more information about the PCIe device specification.

In this example:

- `name = "pciex108e,abcd"`
- `parent = "/pci@7c0/pci@0/pci@9/"`
- `unit-address = "0"`

2. **Set the parameters for the** `nxge` **devices in the** `/platform/sun4u/kernel/drv/nxge.conf` **file.**

- The following parameters can be set using the `/platform/sun4u/kernel/drv/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
#               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).
#
# 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
/platform/sun4u/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
# 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;
```

■ In the following example, the ports of *all* the Sun Quad GbE UTP x8 PCIe ExpressModules are being set for load balancing Rx traffic based on 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;
```

The following example shows ports on two different cards being set. Only one node needs to be specified.

```
name = "pciex108e,abcd" parent = "/pci@780/pci@0/pci@8/" unit-
address = "0" class-opt-ipv4-tcp = 0x100;

name = "pciex108e,abcd" parent = "/pci@7c0/pci@0/pci@9/" unit-
address = "0" class-opt-ipv4-tcp = 0x40;
```

3. **Save the** nxge.conf **file.**

Configuring the Jumbo Frames Feature

This chapter describes how to enable the Jumbo Frames feature. It contains the following sections:

- "Jumbo Frames Overview" on page 41
- "Checking Jumbo Frames Configurations" on page 41
- "Enabling Jumbo Frames in a Solaris SPARC Environment" on page 43

Jumbo Frames Overview

Configuring Jumbo Frames enables the 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 Ethernet ExpressModule driver capability.

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

Checking Jumbo Frames Configurations

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

▼ To Show the Driver Statistics in a Solaris Environment

1. Use the `kstat` **command to display driver statistics, for example:**

```
# kstat nxge:1 |grep rdc_packets
        rdc_packets                        798982054
        rdc_packets                        792546171
        rdc_packets                        803941759
        rdc_packets                        805674872
```

The previous example diplays the receive packet counts on all of the eight receive DMA channels on interface 1. Using the `kstat nxge:1` shows all the statistics that the driver supports for that interface.

2. Use the `kstat` **command to display driver statistics of a VLAN interface, for example:.**

```
# kstat nxge:38001
module: nxge                            instance: 38001
name:   nxge38001                       class:    net
        brdcstrcv                        0
        brdcstxmt                        0
        collisions                       0
        crtime                           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
```

Enabling Jumbo Frames in a Solaris SPARC Environment

This section describes how to enable Jumbo Frames in a Solaris SPARC environment.

▼ To Enable Jumbo Frames in a Solaris Environment Using `nxge.conf`

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

   For example:

   ```
   name = "pciex108e,abcd" parent = "/pci@7c0/pci@0/pci@9/"
   unit-address = "0"
   accept-jumbo=1;
   ```

2. **Reboot the system:**

   ```
   % boot -r
   ```

▼ To Check Layer 2 Configuration

- **View the maximum transmission unit (MTU) configuration of an `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`. This value should be equal to or less than the MTU configured on the switch.

▼ To Check Layer 3 Configuration

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

For example:

```
# 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
nxge2          type: non-vlan mtu: 9194        device: nxge2
nxge3          type: non-vlan mtu: 9194        device: nxge3
nxge38001      type: vlan 38  mtu: 9194        device: nxge1
```

Configuring Link Aggregation

This chapter describes how to configure link aggregation. It contains the following sections:

Overview of 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 it provides the following benefits:

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

Configuring Link Aggregation in a Solaris Environment

This section explains how to configure link aggregation in a Solaris environment.

▼ To Configure Link Aggregation in a Solaris Environment

1. **Aggregate** nxge0, nxge1, nxge2, **and** nxge3 **to form an aggregation and a random number 33 as key.**

   a. **Unplumb the interfaces to be aggregated:**

   ```
   # ifconfig down unplumb nxge0
   # ifconfig down unplumb nxge1
   # ifconfig down unplumb nxge2
   # ifconfig down unplumb nxge3
   ```

   b. **Create a link-aggregation group with key 33 without specifying mode:**

   ```
   # dladm create-aggr -d nxge0 -d nxge1 -d nxge2 -d nxge3 33
   ```

   As the command returns, one line appears in the /etc/aggregation.conf file and indicates that the default mode is off, as shown in the following example:

```
# tail -1 /etc/aggregation.conf
# Use is subject to license terms.
#
# ident "@(#)aggregation.conf   1.1     05/09/01 SMI"
#
# DO NOT EDIT OR PARSE THIS FILE!
#
# Use the dladm(1m) command to change the contents of this file.

33      L4      2        nxge4/0,nxge5/0 auto    off      short
# dladm show-link aggr33
aggr33           type: non-vlan  mtu: 1500       aggregation: key 33
```

2. **Plumb up the interface** aggr*key*, **which is** aggr33 **is this case:**

```
# 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 ffffff00 broadcast 192.168.1.255
        ether 0:3:ba:d8:9d:e8
```

3. **Show link aggregation status again.**

Now the state should become attached:

```
# dladm show-aggr
key: 33 (0x0021)       policy: L4      address: 0:14:4f:6c:11:8 (auto)
          device         address          speed          duplex  link    state
          nxge0        0:14:4f:6c:11:8   1000  Mbps      full    up     attached
          nxge1        0:14:4f:6c:11:9   1000  Mbps      full    up     attached
          nxge2        0:14:4f:6c:11:a   1000  Mbps      full    up     attached
          nxge3        0:14:4f:6c:11:b   1000  Mbps      full    up     attached
```

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

```
# dladm show-aggr -s
key: 33               ipackets  rbytes      opackets  obytes    %ipkts    %opkts
          Total        380354    25872976     28        2648
          nxge0        95089     6468278      7         662       25.0      25.0
          nxge1        95089     6468278      7         662       25.0      25.0
          nxge2        95089     6468278      7         662       25.0      25.0
          nxge3        95087     6468142      7         662       25.0      25.0
```

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

```
# dladm show-aggr -L
key: 33 (0x0021)        policy: L4       address: 0:14:4f:6c:11:8 (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
    nxge2      passive  short   yes          no    no   no   no         no
    nxge3      passive  short   yes          no    no   no   no         no
```

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

Configuring VLANs

This chapter explains virtual local area networks (VLANs) in detail and provides configuration instructions and examples. It contains the following sections:

- "VLAN" on page 49
- "Configuring VLANs in a Solaris Environment" on page 51

VLAN

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, with its traffic and broadcasts isolated from the others, increasing the bandwidth efficiency within each logical group.

Although VLANs are commonly used to create individual broadcast domains and/or separate IP subnets, it can be useful for a server to have a presence on more than one VLAN simultaneously. Several Sun products support multiple VLANs on a per port or per interface basis, allowing very flexible network configurations.

With multiple VLANs on an ExpressModule, a server with a single ExpressModule can have a logical presence on multiple IP subnets. By default, 128 VLANs can be defined for each VLAN-aware ExpressModule on your server. 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.

FIGURE 8-1 shows an example network that uses VLANs.

FIGURE 8-1 Example of Servers Supporting Multiple VLANs With Tagging ExpressModules
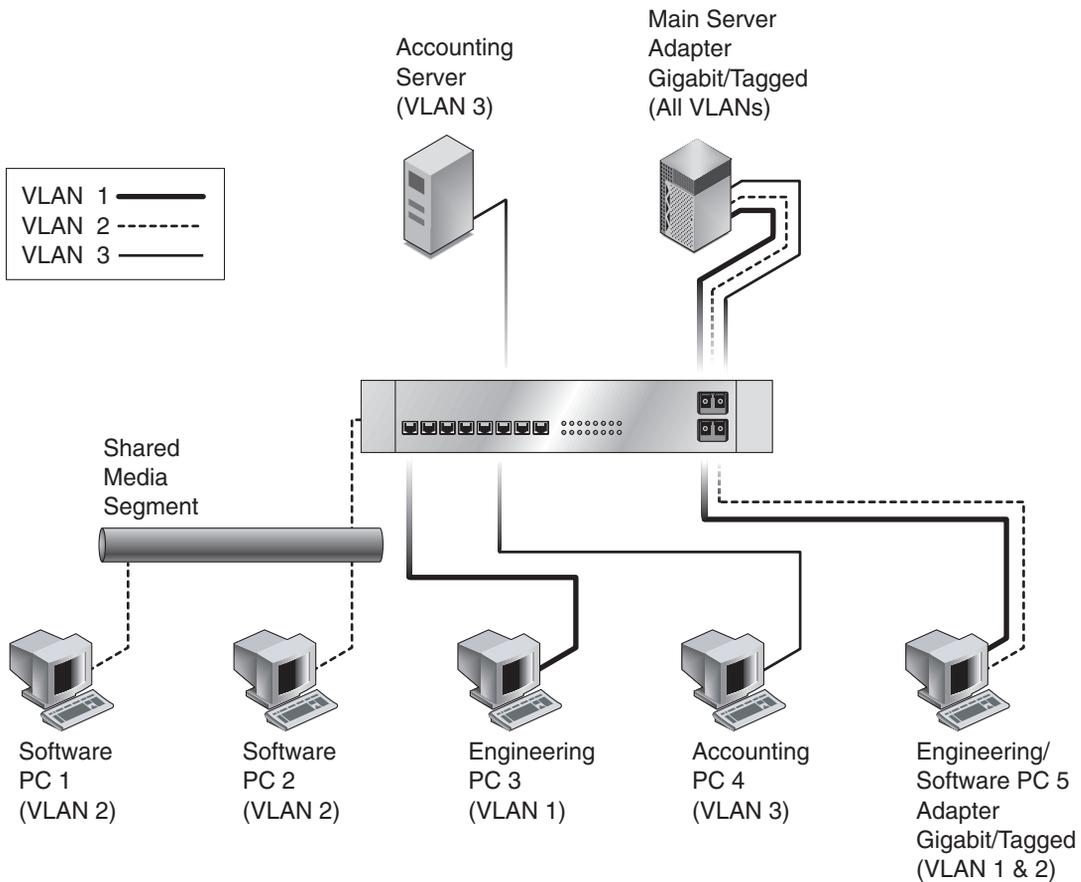
The example network has the following features:

The physical LAN network consists of a switch, two servers, and five clients. The LAN is logically organized into three different VLANs, each representing a different IP subnet.

- VLAN 1 is an IP subnet consisting of the Main Server, Client 3, and Client 5. This represents an engineering group.

- VLAN 2 includes the Main Server, Clients 1 and 2 by means of a shared media segment, and Client 5. This is a software development group.

- VLAN 3 includes the Main Server, the Accounting Server and Client 4. This is an accounting group.

The Main Server is a heavily used server that needs to be accessed from all VLANs and IP subnets. The server has a Sun Quad GbE UTP x8 PCIe ExpressModule installed. All three IP subnets are accessed by means of the single physical ExpressModule interface. The server is attached to one of the switch's Gigabit Ethernet ports, which is configured for VLANs 1, 2, and 3. Both the ExpressModule and the connected switch port have tagging turned on. The tagging VLAN capabilities of both devices enable the sever to communicate on all three IP subnets in this network, yet continue to maintain broadcast separation among the three subnets. The following list describes the components of this network:

■ The Accounting Server is available to VLAN 3 only. The Accounting Server is isolated from all traffic on VLANs 1 and 2. The switch port connected to the server has tagging turned off.

■ Clients 1 and 2 are attached to a shared media hub that is then connected to the switch. Clients 1 and 2 belong to VLAN 2 only, and are logically in the same IP subnet as the Main Server and Client 5. The switch port connected to this segment has tagging turned off.

■ Client 3 is a member of VLAN 1, and can communicate only with the Main Server and Client 5. Tagging is not enabled on Client 3's switch port.

■ Client 4 is a member of VLAN 3, and can communicate only with the servers. Tagging is not enabled on Client 4's switch port.

■ Client 5 is a member of both VLANs 1 and 2, and has a Sun Quad GbE UTP x8 PCIe ExpressModule installed. It is connected to switch port 10. Both the ExpressModule and the switch port are configured for VLANs 1 and 2 and have tagging enabled.

VLAN tagging is only required to be enabled on switch ports that create trunk links to other VLAN-aware Ethernet switches, or on ports connected to tag-capable end-stations, such as servers or workstations with VLAN-aware ExpressModules.

Configuring VLANs in a Solaris Environment

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`, `nxge1`, `nxge2`, and `nxge3`), 4094 possible VLAN IDs can be selected for each port.

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). FIGURE 8-2 shows the Ethernet tag header format.

FIGURE 8-2 Ethernet Tag Header Format
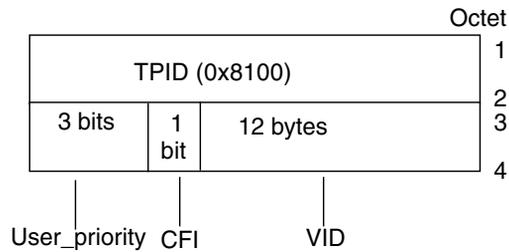
By default, a single VLAN is configured for every port. This groups all ports into the same broadcast domain, just as if there were no VLANs at all, VLAN tagging for the switch port is turned off.

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

▼ To Configure Static VLANs

1. **Create one** `hostname.nxge`*number* **file for each VLAN that will be configured for each ExpressModule on the server.**

   Use the following naming format, which includes both the VID and the physical point of attachment (PPA):

   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 with the Sun Quad GbE UTP x8 PCIe ExpressModule 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.

2. **Use the** `ifconfig(1M)` **to configure a VLAN virtual device.**

   For example:

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

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

   ```
   # 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 ffffff00 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 ffffff00 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 ffffff00 broadcast 199.199.224.225
           ether 8:0:20:a4:4f:b8
   ```

3. **On the switch, set VLAN tagging and set VLAN ports to coincide with the VLANs you've set up on the server.**

   Using the examples 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.

Specifications

This appendix lists the specifications for the Sun Quad GbE UTP x8 PCIe ExpressModule. It contains the following sections:

Connectors

FIGURE A-1 shows the connectors for the Sun Quad GbE UTP x8 PCIe ExpressModule.
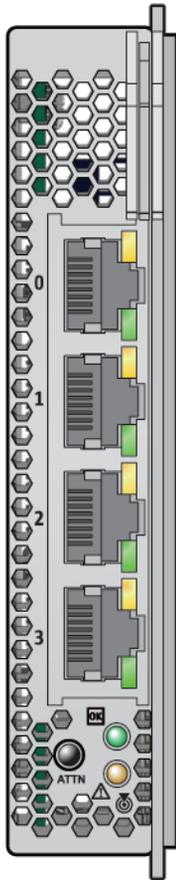
The green LED indicates the link status and the yellow LED indicates the link activity.

TABLE A-1 lists the characteristics of the connectors used by the Sun Quad GbE UTP x8 PCIe ExpressModule.

TABLE A-1 Connector Characteristics

Parameter	Description
Connector type	1x4 RJ45
Distance	100 meters

Performance Specifications

TABLE A-2 Performance Specifications

Feature	Specification
Bus type	x8 lane PCI Express 1.1
Bus width	x8 lane PCI Express 1.1
Bus speed (x8, encoded rate)	20 Gbit/sec unidirectional; 40 Gbit/sec bidirectional (theoretical)
Maximum Ethernet transfer rate	1 Gbps (full-duplex)

Physical Characteristics

TABLE A-3 Physical Characteristics

Dimension	Measurement
Length	170 mm (6.69 inches)
Width	21.5 mm (.85 inches)
Height	112 mm (4.41 inches)

Power Requirements

TABLE A-4   Power Requirements

Specification	Measurement
Power consumption	14.31W RMS typical
	17.56 W maximum
Voltage	12V @ 1.460A maximum (1.191A RMS typical)
	3.3V @ 0.012A maximum (0.005A RMS typical)

Diagnostic Software

This appendix provides an overview of the SunVTS diagnostic application and instructions for updating the SunVTS software to recognize the ExpressModule. This appendix contains the following sections:

- "SunVTS Diagnostic Software" on page 59
- "Updating SunVTS to Recognize the ExpressModule" on page 60
- "Using the SunVTS `netlbtest`" on page 61

SunVTS Diagnostic Software

The SunVTS software executes multiple diagnostic hardware tests from a single user interface and is used to verify the configuration and functionality of most hardware controllers and devices. The SunVTS software operates primarily from a graphical user interface, enabling test parameters to be set quickly and easily while a diagnostic test operation is being performed.

You can use the SunVTS `nettest` diagnostic to test all of the networking interfaces on the system, including the interfaces on the ExpressModule.

To use the `nettest` diagnostic, you must have the SunVTS software installed on your system. Refer to your Solaris documentation for installation instructions.

Refer to the SunVTS documentation (listed in TABLE B-1) for instructions on how to run and monitor the `nettest` diagnostic. These SunVTS documents are available online at the following URL:

http://www.sun.com/documentation

Search for title of the document you want to use.

Select the document for the Solaris release on your system.

TABLE B-1 SunVTS Documentation

Title	Description
SunVTS User's Guide	Describes the SunVTS diagnostic environment.
SunVTS Test Reference Manual	Describes each SunVTS test (including the `nettest` `netlbtest`) and describes the various test options and command-line arguments.
SunVTS Quick Reference	Provides an overview of the user interface.

Updating SunVTS to Recognize the ExpressModule

Use SunVTS 6.3 or later. You need to update the SunVTS configuration to recognize the ExpressModule.

▼ To Update SunVTS to Recognize the ExpressModule

1. **Plug in a loopback cable.**

2. **Ensure that the SunVTS software and the `nxge` driver are installed on your system.**

3. **Add the following lines to the** `/opt/SUNWvts/lib/conf/netlbtest.conf` **and** `/opt/SUNWvts/lib/conf/nettest.conf` **files:**

```
nxge     nxge     10gbaseT
```

Using the SunVTS `netlbtest`

You must have the Ethernet card and the device driver installed, a loopback connector in place, and Intervention mode enabled before running `netlbtest`. `netlbtest` cannot run if the network interface is connected to a live network, and requires that the Ethernet device be configured offline before running the test. Use the `ifconfig(1M)` command to halt the Ethernet device before running `netlbtest`.

▼ To Use the `netlbtest`

1. **Ensure that the SunVTS software and the** `nxge` **driver are installed on your system.**

2. **Plug in a loopback cable.**

3. **Unplumb the interface from the system, using the** `ifconfig` **command**:

```
# ifconfig nxgeinstance down
# ifconfig nxgeinstance unplumb
```

where *instance* is the instance number of the interface.

Refer to SunVTS documentation for instructions on how to run `netlbtest`.

Index