Sun™ Quad GigaSwift Ethernet UTP Adapter Installation and User's Guide

Sun Microsystems, Inc.
www.sun.com

Part No. 817-2974-10
December 2003, Revision A

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

Regulatory Compliance Statements

Your Sun product is marked to indicate its compliance class:

- Federal Communications Commission (FCC) — USA
- Industry Canada Equipment Standard for Digital Equipment (ICES-003) — Canada
- Voluntary Control Council for Interference (VCCI) — Japan
- Bureau of Standards Metrology and Inspection (BSMI) — Taiwan

Please read the appropriate section that corresponds to the marking on your Sun product before attempting to install the product.

FCC Class A Notice

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

1. This device may not cause harmful interference.
2. This device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy, and if it is not installed and used in accordance with the instruction manual, it may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Modifications: Any modifications made to this device that are not approved by Sun Microsystems, Inc. may void the authority granted to the user by the FCC to operate this equipment.

FCC Class B Notice

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

1. This device may not cause harmful interference.
2. This device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/television technician for help.

Modifications: Any modifications made to this device that are not approved by Sun Microsystems, Inc. may void the authority granted to the user by the FCC to operate this equipment.

ICES-003 Class A Notice - Avis NMB-003, Classe A

This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

ICES-003 Class B Notice - Avis NMB-003, Classe B

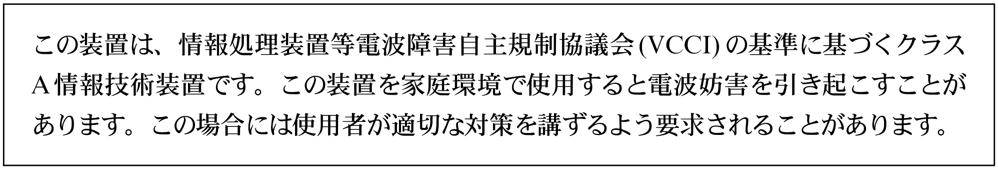
This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe B est conforme à la norme NMB-003 du Canada.

VCCI 基準について

クラス A VCCI 基準について

クラス A VCCIの表示があるワークステーションおよびオプション製品は、クラスA情報技術装置です。これらの製品には、下記の項目が該当します。

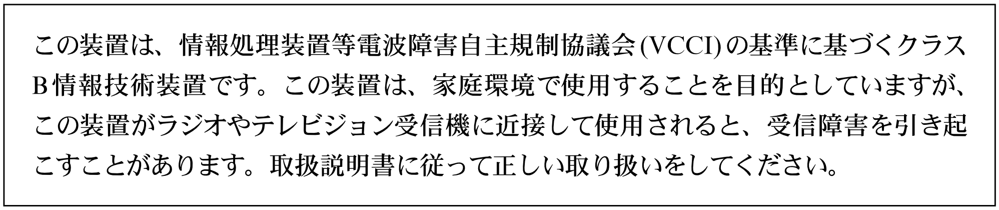


この装置は、情報処理装置等電波障害自主規制協議会(VCCI)の基準に基づくクラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

Graphic showing the Japanese VCCI-A regulatory statement.

クラス B VCCI 基準について

クラス B VCCI の表示 VCCI があるワークステーションおよびオプション製品は、クラスB情報技術装置です。これらの製品には、下記の項目が該当します。

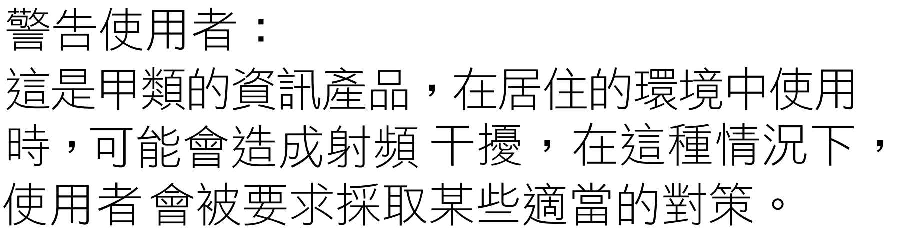


この装置は、情報処理装置等電波障害自主規制協議会(VCCI)の基準に基づくクラスB情報技術装置です。この装置は、家庭環境で使用することを目的としていますが、この装置がラジオやテレビジョン受信機に近接して使用されると、受信障害を引き起こすことがあります。取扱説明書に従って正しい取り扱いをしてください。

Graphic showing the Japanese VCCI-B regulatory statement.

BSMI Class A Notice

The following statement is applicable to products shipped to Taiwan and marked as Class A on the product compliance label.



警告使用者：
這是甲類的資訊產品，在居住的環境中使用
時，可能會造成射頻 干擾，在這種情況下，
使用者 會被要求採取某些適當的對策。

Graphic showing the BSMI Class A Notice for products shipped to Taiwan.

Declaration of Conformity

Compliance Model Number:	QGEPCI
Product Family Name:	Sun Quad Gigabit Ethernet QGE PCI Card (X4444A)

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 Telecommunication Network Equipment (TNE) in Both Telecom Centers and Other Than Telecom Centers per (as applicable):

EN300-386 V.1.3.1 (09-2001) Required Limits:

EN55022/CISPR22	Class A
EN61000-3-2	Pass
EN61000-3-3	Pass
EN61000-4-2	6 kV (Direct), 8 kV (Air)
EN61000-4-3	3 V/m 80-1000MHz, 10 V/m 800-960 MHz, and 1400-2000 MHz
EN61000-4-4	1 kV AC and DC Power Lines, 0.5 kV Signal Lines
EN61000-4-5	2 kV AC Line-Gnd, 1 kV AC Line-Line and Outdoor Signal Lines, 0.5 kV Indoor signal Lines > 10m
EN61000-4-6	3 V
EN61000-4-11	Pass

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

EN55022:1998/CISPR22:1997	Class A
EN55024:1998 Required Limits:	
EN61000-4-2	4 kV (Direct), 8 kV (Air)
EN61000-4-3	3 V/m
EN61000-4-4	1 kV AC Power Lines, 0.5 kV Signal and DC Power Lines
EN61000-4-5	1 kV AC Line-Line and Outdoor Signal Lines, 2 kV AC Line-Gnd, 0.5 kV DC Power Lines
EN61000-4-6	3 V
EN61000-4-8	1 A/m
EN61000-4-11	Pass
EN61000-3-2:1995 + A1, A2, A14	Pass
EN61000-3-3:1995	Pass

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

EC Type Examination Certificates:

EN60950:2000, 3rd Edition	TÜV Rheinland Certificate No. xxxxxxxxxxxx		
IEC 60950:2000, 3rd Edition	CB Scheme Certificate No. xxxxxxxxxxxx		
Evaluated to all CB Countries			
UL 60950, 3rd Edition, CSA C22.2 No. 60950-00	File:	Vol.	Sec.
UL 60950, 3rd Edition, CSA C22.2 No. 950-00	File:	Vol.	Sec.
FDA DHHS Accession Number (Monitors Only)			

Supplementary Information: This product was tested and complies with all the requirements for the CE Mark.

/S/

Dennis P. Symanski DATE
Manager, 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/

Pamela J. Dullaghan DATE
Quality Program Manager
Sun Microsystems Scotland, Limited
Springfield, Linlithgow
West Lothian, EH49 7LR
Scotland, United Kingdom
Tel: +44 1 506 672 395 Fax: +44 1 506 670 011

Contents

1. Product Overview 1

Product Description 1

Hardware and Software Requirements 2

Features 3

2. Installing the Adapter 5

Installing the Adapter Without Dynamic Reconfiguration 6

▼ To Install the Adapter 6

Connecting the Ethernet Cables 7

▼ To Connect the Ethernet Cables 7

Attaching the Adapter to the System 7

▼ To Verify the Installation 7

Setting the `local-mac-address?` Variable 10

▼ To Set the GigaSwift Ethernet Device Primary Boot Device 10

Rebooting the System 11

Installing the Adapter With Dynamic Reconfiguration 12

▼ To Install an Adapter in a Dynamic Reconfiguration System 13

Using the Installation Script 16

▼ To Install the Software Using the Installation Script 17

Example 1: All Requirements Met 18

Example 2: Updates Required 19

Example 3: Packages Missing 21

3. Configuring Software for the Sun Quad GigaSwift Ethernet Adapter 23

Attaching the Gigabit Ethernet Interfaces to the Network 24

▼ To Attach the Gigabit Ethernet Interfaces to the Network 24

Jumbo Frames Capability 27

▼ To Enable Jumbo Frames 27

Managing MTU Using `ifconfig` 28

Reboot Persistence 30

Autonegotiation Protocol 31

Setting the `local-mac-address?` Variable 31

▼ To Set the `local-mac-address?` Variable to the Network Interfaces 32

Booting the System Over a Gigabit Ethernet Interface 34

▼ To Boot the System Over the Network 34

4. Configuring the Sun Quad GigaSwift Ethernet Device Driver Parameters 37

Sun Quad GigaSwift Ethernet Device Driver Parameters 38

Hardware Overview 38

Device Driver Parameter Values and Definitions 39

Operational Mode Parameters 40

Flow Control Parameters 41

Gigabit Link Clock Mastership Controls 42

Interpacket Gap Parameters 42

Interrupt Parameters 44

Random Early Drop Parameters 44

PCI Bus Interface Parameters 45

Jumbo Frames 46

Setting `ce` Driver Parameters 46

Setting Parameters Using the ndd Utility 47

▼ To Specify Device Instances for the ndd Utility 47

Noninteractive and Interactive Modes 47

▼ To Use the ndd Utility in Noninteractive Mode 48

▼ To Use the ndd Utility in Interactive Mode 48

Setting the Autonegotiation Mode 49

▼ To Disable Auto-negotiation Mode 50

Setting Parameters Using the ce.conf File 50

▼ To Set Driver Parameters Using a ce.conf File 51

Usability Enhancements to the Driver 53

GigaSwift Ethernet Driver Operating Statistics 54

Reporting the Link Partner Capabilities 55

▼ To Check Link Partner Settings 58

▼ To Discover Link Partner Capabilities 61

▼ To Discover Link Settings 62

Configuration Checking 62

A. Specifications 63

Performance Specifications 63

Physical Characteristics 64

Power Requirements 64

Performance Specifications 65

Interface Signals 66

B. Installing the Software Manually 67

Installing the Driver Software Manually 67

Verifying Patches 71

▼ To Install Patches 71

▼ To Install a Sun VLAN Utility Patch 73

C. Testing the Adapter 75

Using the SunVTS Diagnostic Software 75

Using the SunVTS `netlbtest` 76

▼ To Use the `netlbtest` 76

Using the OpenBoot PROM FCode Self-Test 77

▼ To Run the FCode Self-Test Diagnostic 77

Figures

FIGURE 1-1 Sun Quad GigaSwift Ethernet Adapter 2

FIGURE 1-2 Sun Quad GigaSwift Ethernet Adapter Ethernet Channels 3

FIGURE A-1 32-Pin RJ-45 Connectors 66

Tables

TABLE 1-1 Hardware and Software Requirements 2

TABLE 4-1 ce Driver Parameter, Status, and Descriptions 39

TABLE 4-2 Operational Mode Parameters 40

TABLE 4-3 Read-Write Flow Control Keyword Descriptions 41

TABLE 4-4 Forced Mode Parameter 42

TABLE 4-5 Parameters Defining enable-ipg0 and ipg0 43

TABLE 4-6 Read-Write Interpacket Gap Parameter Values and Descriptions 43

TABLE 4-7 RX Blanking Register for Alias Read 44

TABLE 4-8 RX Random Early Detecting 8-Bit Vectors 44

TABLE 4-9 PCI Bus Interface Parameters 45

TABLE 4-10 accept-jumbo Parameters 46

TABLE 4-11 Read-Only ce Device Capabilities 54

TABLE 4-12 Read-Only Link Partner Capabilities 55

TABLE 4-13 Transmit and Receive Parameters 56

TABLE A-1 Performance Specifications 63

TABLE A-2 Physical Characteristics 64

TABLE A-3 Power Requirements 64

TABLE A-4 Performance Specifications 65

TABLE A-5 Pin Descriptions for the 32-Pin RJ-45 Connectors—Channels 0 - 3 66

TABLE 4-14 Files and Directories on the CD-ROM 68

TABLE B-1 Patch Versions 71

TABLE C-1 SunVTS Documentation 76

Preface

The *Sun Quad GigaSwift Ethernet UTP Adapter Installation and User's Guide* provides installation instructions for the Sun Quad GigaSwift Ethernet UTP adapter. This manual also describes how to configure the driver software.

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

How This Book Is Organized

Chapter 1 provides a description of the adapter, including hardware and software.

Chapter 2 describes how to install the hardware and use the automated installation script to install the software.

Chapter 3 describes how to configure the driver parameters used by the Sun Quad GigaSwift Ethernet adapter.

Chapter 4, lists the available `ce` device driver parameters and describes how you can set these parameters.

Appendix A lists the specifications for the Sun Quad GigaSwift Ethernet adapter.

Appendix B describes how to installthe driver software manually.

Appendix C describes how to disable the adapter's `ce` interfaces prior to removing the adapter from a system, and how to activate the interfaces after replacing the adapter.

Appendix C provides an overview of the SunVTS™ diagnostic application and instructions for testing the adapter using the onboard FCode selftest. An additional section outlines some common troubleshooting issues.

Using UNIX Commands

This document may not contain information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices.

See one or more of the following for this information:

- Online documentation for the Solaris™ operating environment
- Other software documentation that you received with your system

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	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type `rm` *filename*.

Shell Prompts

Shell	Prompt
C shell	*machine_name*%
C shell superuser	*machine_name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#

Related Documentation

Application	Title
PCI Adapter Installation	Your system installation or service manual
Storage Device Installation	Your storage device installation or service manual
Dynamic Reconfiguration Installation	*Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems Dynamic Reconfiguration User's Guide*
Diagnostic Software	*SunVTS User's Guide* *SunVTS Test Reference Manual*
OpenBoot™ Commands	*OpenBoot 3.x Command Reference Manual*

Accessing Sun Documentation Online

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

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

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can email your comments to Sun at:

`docfeedback@sun.com`

Please include the part number (817-2974-05) of your document in the subject line of your email.

CHAPTER 1

Product Overview

This chapter contains an overview of the Sun Quad GigabitEthernet adapter, including:

- "Product Description" on page 1
- "Hardware and Software Requirements" on page 2
- "Features" on page 3

Product Description

The Sun Quad GigaSwift Ethernet adapter is a full-length PCI card with implementation of four Ethernet ports for 10/100/1000 Mbps. It features half and full duplex ports for 10/100 Mbps, and a full duplex port for 1000 Mpbs. Each port has its own unique MAC address.

FIGURE 1-1 and FIGURE 1-2 show views of the adapter.

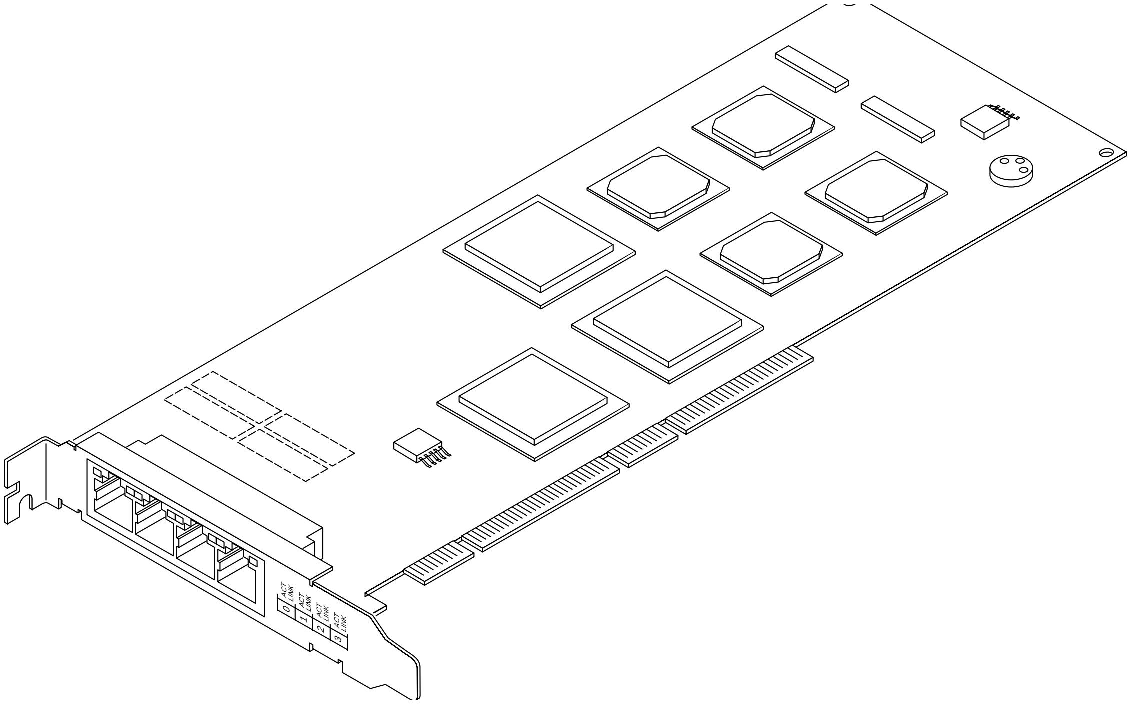


FIGURE 1-1 Sun Quad GigaSwift Ethernet Adapter

Hardware and Software Requirements

Before installing the adapter, make sure your system meets the following hardware and software requirements:

TABLE 1-1 Hardware and Software Requirements

Requirements	Hardware or Software
Hardware	Sun Enterprise™ E220, E420, E450, Netra™ 1150, Netra T20/Sun Blade 1000, Sun Fire™ V210, Sun Fire V240, Sun Fire V440, Sun Blade™ 150, Sun Blade 2000, Sun Fire V280, Sun Fire V480, Sun Fire V880, Sun Fire 12k, Sun Fire 15k, Sun Fire 2500, Sun Fire 4800, Sun Fire 6800
Software	Solaris™ 7, 8, and 9 operating environment

Note – The preceeding information is up-to-date as of the printing of this manual. Visit Sun on the World Wide Web at `http://www.sun.com/` for the latest information.

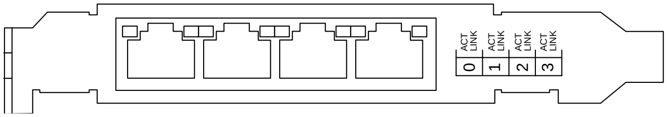


FIGURE 1-2 Sun Quad GigaSwift Ethernet Adapter Ethernet Channels

Features

This PCI adapter offers the following features:

- PCI Local Bus Revision 2.2 compliant
- 33/66-MHz, 32- or 64-bit bus master IEEE 802.3x compatible
- Universal PCI card (12.283-inch x 4.2-inch long card)
- IPv4 and IPv6 support
- Four ports of IEEE 10BASE-T, 100BASE-TX, 1000BASE-T compliant network interfaces
- Link integrity and activity LEDs for each channel on the RJ-45 connector
- Trunking 1.3 support
- FCode PROM support
- RAS support
- Dynamic reconfiguration (DR) and redundancy/failover support
- Full flow control support
- Jumbo frames support up to 9216 bytes, including VLAN Ethernet header and CRC

CHAPTER 2

Installing the Adapter

This chapter describes how to install the adapter in your system and verify that it has been installed correctly. It then describes how to install the driver software using the automated script.

This chapter contains the following sections:

- "Installing the Adapter Without Dynamic Reconfiguration" on page 6
- "Connecting the Ethernet Cables" on page 7
- "Attaching the Adapter to the System" on page 7
- "Installing the Adapter With Dynamic Reconfiguration" on page 12
- "Using the Installation Script" on page 16

Note – If you have a Sun Enterprise system that supports dynamic reconfiguration (DR), refer to the *Sun Enterprise Dynamic Reconfiguration User's Guide* and your system's documentation for further information about dynamic reconfiguration.

Installing the Adapter Without Dynamic Reconfiguration

▼ To Install the Adapter

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

---

1. **Halt and power off your system.**
2. **Power off all of the peripherals connected to your system.**
3. **Open the system unit.**
4. **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.**
5. **Holding the PCI adapter by the edges, unpack it and place it on an antistatic surface.**
6. **Using a No. 1 Phillips screwdriver, remove the PCI filler panel from the slot in which you want to insert the PCI adapter.**

   Save the filler panel screw for Step 9.
7. **Holding the PCI adapter by the edges, align the adapter edge connector with the PCI slot. Slide the adapter face plate into the small slot at the end of the PCI opening.**

8. **Applying even pressure at both corners of the adapter, push the PCI adapter until it is firmly seated in the slot.**

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

9. **Secure the adapter to the PCI slot using the screw you removed in Step 6.**
10. **Detach the wrist strap and close the system unit.**

Connecting the Ethernet Cables

After installing the adapter, attach the Ethernet cables to the four Ethernet ports.

▼ To Connect the Ethernet Cables

1. **Connect one end of the Ethernet cables to the ports on the card.**
2. **Connect the other end of the Ethernet cable to an active Ethernet network.**
3. **Repeat for each Ethernet cable you want to connect.**

Attaching the Adapter to the System

After installing the adapter, you must make the system recognize the new adapter and its Ethernet interfaces.

▼ To Verify the Installation

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

Note – Verification is not required if your system supports dynamic reconfiguration (DR).

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
```

Use the `show-nets` command to list the system devices. You should see the full path name of the `network` devices, similar to the example below.

```
ok show-nets
a) /pci@9,600000/pci@1/pci@4/network@3
b) /pci@9,600000/pci@1/pci@4/network@2
c) /pci@9,600000/pci@1/pci@0/network@1
d) /pci@9,600000/pci@1/pci@0/network@0
e) /pci@9,700000/network@1,1
f) /pci@8,600000/network@1
q) NO SELECTION
Enter Selection, q to quit: q
```

In the previous example, items a, b, c, and d, are ports on the Sun Quad GigaSwift Ethernet adapter.

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

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

Using the previous example, type:

```
ok cd /pci@9,600000/pci@1/pci@0/network@0
```

Be sure to write down your device path, which in the example is `/pci@9,600000/pci@1/pci@0/network@0`. While your device path might be different, it will be similar. You need this information to make changes to the `ce.conf` file. See "Setting `ce` Driver Parameters" on page 46.

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

   Your output might look different from the following example:

```
ok .properties
mac-address              00 03 ba 29 b8 3f
link-clock               auto
duplex                   auto
speed                    auto
assigned-addresses       82020010 00000000 02000000 00000000 00200000
                         82020030 00000000 02200000 00000000 00100000
d-fru-len                00 00 00 00
d-fru-off                00 00 e8 00
d-fru-dev                eeprom
s-fru-len                00 00 08 00
s-fru-off                00 00 e0 00
s-fru-dev                eeprom
compatible               70 63 69 31 30 30 62 2c 33 35 2e 31 00 70 63 69
reg                      00020000 00000000 00000000 00000000 00000000
                         02020010 00000000 00000000 00000000 00200000
                         02020030 00000000 00000000 00000000 00100000
address-bits             00 00 00 30
max-frame-size           00 00 40 00
network-interface-type   ethernet
device_type              network
name                     network
local-mac-address        08 00 20 e3 60 38
version                 Sun PCI Quad Gigaswift 10/100/1000Base-T FCode 2.12 03/
11/11
phy-type                 mif
board-model              501-6522
model                    SUNW,pci-qge
fcode-rom-offset         00000000
66mhz-capable
fast-back-to-back
devsel-speed             00000002
class-code               00020000
interrupts               00000001
                         00000002
                         00000003
                         00000004
latency-timer            00000040
cache-line-size          00000010
max-latency              00000040
min-grant                00000040
revision-id              00000030
device-id                00000035
vendor-id                0000100b
```

The assigned value of the `phy-type` is `mif`.

Note – If you are going to set the `local-mac-address?` variable, note the local MAC address of your device at this time. See "Setting the `local-mac-address?` Variable" on page 10 for more information.

Setting the `local-mac-address?` Variable

Note – Enabling the `local-mac-address?` variable is only required if you boot from the network.

The network interface of the Sun Quad GigaSwift Ethernet adapter has been assigned a unique media access control (MAC) address that represents the 48-bit Ethernet address for that interface. The OpenBoot firmware reports this MAC address via the `local-mac-address?` variable in the device nodes corresponding to the network interface.

A system is not obligated to use this assigned MAC address if it has a system-wide MAC address. In such cases, the system-wide MAC address applies to all network interfaces on the system.

The device driver, or any other adapter utility, can use the network device's MAC address (`local-mac-address?`) while configuring it. In the Solaris operating environment, you can use the MAC address when booting over the network.

The `mac-address` property of the network device specifies the network address (system-wide or `local-mac-address?`) used for booting the system. To start using the MAC address assigned to the network interface of the Sun Quad GigaSwift Ethernet adapter, set the NVRAM configuration variable `local-mac-address?` to `true`.

```
ok setenv local-mac-address? true
```

▼ To Set the GigaSwift Ethernet Device Primary Boot Device

Use this procedure *only* if you want the GigaSwift Ethernet device to be your primary boot device.

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

```
ok show-nets
a) /pci@9,600000/pci@1/pci@4/network@3
b) /pci@9,600000/pci@1/pci@4/network@2
c) /pci@9,600000/pci@1/pci@0/network@1
d) /pci@9,600000/pci@1/pci@0/network@0
e) /pci@9,700000/network@1,1
f) /pci@8,600000/network@1
q) NO SELECTION
Enter Selection, q to quit: q
```

2. **Set the Sun Quad GigaSwift Ethernet device to be your default boot device by typing:**

```
 ok setenv boot-device /pci@9,600000/pci@1/pci@0/network@0
```

Note that the correct command has a space rather than an equal sign (`=`) between `boot-device` and `/pci@1f,2000/pci@1/network@4`.

Rebooting the System

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

```
ok boot -r
```

You can enable verbosity of the boot process with the following command:

```
 ok setenv boot -file -v
```

Installing the Adapter With Dynamic Reconfiguration

If you have a Sun Enterprise system that supports dynamic reconfiguration (DR), you do not have to reboot your system after installing the adapter.

The process of adding and configuring an adapter with DR involves (1) connecting the attachment point and (2) configuring its occupant. In most cases, the `cfgadm`(1M) command can perform both steps at once.

▼ To Install an Adapter in a Dynamic Reconfiguration System

1. **Verify that the selected board slot is ready for the adapter.**

```
# cfgadm

Ap_Id                 Type         Receptacle   Occupant     Condition
IO2                   unknown      empty        unconfigured unknown
IO5                   HPCI         disconnected unconfigured unknown
IO7                   HPCI         disconnected unconfigured unknown
IO9                   HPCI         connected    configured   ok
IO13                  WPCI         connected    configured   ok
IO14                  HPCI         connected    configured   ok
SB2                   unknown      empty        unconfigured unknown
SB7                   CPU          disconnected unconfigured unknown
SB9                   CPU          connected    configured   ok
SB13                  CPU          disconnected unconfigured unknown
SB15                  CPU          connected    configured   ok
SB17                  CPU          disconnected unconfigured unknown
c0                    fc-private   connected    configured   unknown
c1                    scsi-bus     connected    configured   unknown
c2                    scsi-bus     connected    unconfigured unknown
c3                    fc-private   connected    configured   unknown
c4                    scsi-bus     connected    unconfigured unknown
c5                    scsi-bus     connected    unconfigured unknown
c6                    fc-private   connected    configured   unknown
c7                    fc-private   connected    configured   unknown
c8                    fc-private   connected    configured   unknown
c9                    scsi-bus     connected    unconfigured unknown
pcisch0:e09b1slot1    pci-pci/hp   connected    configured   ok
pcisch1:e09b1slot0    pci-pci/hp   connected    configured   ok
pcisch2:e09b1slot3    pci-pci/hp   connected    configured   ok
pcisch3:e09b1slot2    pci-pci/hp   connected    configured   ok
pcisch4:e14b1slot1    pci-pci/hp   connected    configured   ok
pcisch5:e14b1slot0    pci-pci/hp   connected    configured   ok
pcisch6:e14b1slot3    scsi/hp      connected    configured   ok
pcisch7:e14b1slot2    pci-pci/hp   connected    configured   ok
pcisch8:e13b1slot1    pci-pci/hp   connected    configured   ok
pcisch9:e13b1slot0    pci-pci/hp   connected    configured   ok
```

The states and conditions should be:

```
Receptacle state—Empty
Occupant state—Unconfigured
Condition—Unknown
```

Or:

```
Receptacle state—Disconnected
Occupant state—Unconfigured
Condition—Unknown
```

2. **If the status of the slot is not `empty` or `disconnected`, type:**

```
# cfgadm -c disconnect sysctrl-number:IO-boardslot-number
```

Where:

sysctrl-number = `pcisch`*X* and is obtained from Step 1 output

IO-boardslot-number = available slot # on the IO board for the adapter

For example, if you were installing the adapter into slot number 0 of IO board 9 on a Sun Fire 15000, you would obtain something like `"pcisch1:e09b1slot0"` from the Step 1 output. So you would type:

```
# cfgadm -c disconnect pcisch1:e09b1slot0
```

3. **Insert the adapter into the slot and look for an acknowledgement on the console, such as, "***name* `board inserted into slot3`**."**

After the adapter is inserted, the states and conditions should become:

```
Receptacle state—Disconnected
Occupant state—Unconfigured
Condition—Unknown
```

Any other states or conditions indicate an error.

4. **Connect any peripheral cables and interface modules to the adapter.**

5. **Connect the cables to the card and to an Ethernet network.**

6. **Configure the adapter with the following command:**

```
# cfgadm -v -c configure sysctrl-number:slot-number
```

This command should both connect and configure the receptacle. Verify the condition with the `cfgadm` command.

The states and conditions for a connected and configured attachment point should be:

```
Receptacle state—Connected
Occupant state—Configured
Condition—OK
```

The system is aware of the usable devices that reside on the adapter. All devices can be mounted or configured for use.

If the command fails to connect and configure the adapter and slot (the status should be `configured` and `ok`), do the connection and configuration as separate steps:

a. **Connect the adapter and slot:**

```
# cfgadm -v -c connect sysctrl-number:slot-number
```

The states and conditions for a connected attachment point should be:

```
Receptacle state—Connected
Occupant state—Unconfigured
Condition—OK
```

The system is aware of the adapter, but not the usable devices that reside on the adapter. The temperature is monitored; power and cooling affect the attachment point condition.

b. Configure the adapter and slot:

```
# cfgadm -v -c configure sysctrl#:slot#
```

The states and conditions for a configured attachment point should be:

```
Receptacle state—Connected
Occupant state—Configured
Condition—OK
```

The system is aware of the usable devices that reside on the adapter. All devices can be mounted or configured for use.

7. **Reconfigure the devices on the adapter:**

```
# drvconfig; devlinks; disks; ports; tapes;
```

The console should now display a list of devices and their addresses.

Using the Installation Script

The *Sun Quad GigaSwift Ethernet Driver* CD provides an automated installation for the Quad GigaSwift Ethernet driver software. In most cases, you can use the following procedure and execute the installation script. If you have any problems, see Appendix B for manual installation procedures.

When you have executed the script, select the interface card that is installed on your system. For this product, select the “Quad GigaSwift Ethernet Adapter.”

The script first checks to ensure that the software driver package has already been installed. If the package is installed, the script verifies the patch level and updates the driver with the updated patch if necessary. If the packages are not installed on Solaris 7, 8, or 9 software, the installation script installs the latest pre-patched packages. If the system does not require updating because the patch level is at or beyond the current install version, the script exits with an appropriate message.

▼ To Install the Software Using the Installation Script

1. **At the system console, become superuser (root).**
2. **Insert the** *Sun GigaSwift Ethernet Driver CD* **into a CD-ROM drive that is connected to your system.**

   If your system is running *Sun Enterprise Volume Manager*, it should automatically mount the CD-ROM to the `/cdrom/cdrom0` directory.

   If your system is not running *Sun Enterprise Volume Manager*, mount the CD-ROM as follows:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom
```

Note – If you are installing the software from the download site, follow the online download instructions.

---

3. **Change to the directory were the installation script resides:**

```
# cd /media_path/
```

Where:

- *media_path* = `/usr/cdrom/cdrom0`, if you are installing from the CD-ROM

Or

- unzipped *GigaSwift_path*/`Sun_GigaSwift_Ethernet_Driver`

4. **Execute the install script:**

```
# ./install
```

The script displays a menu of products:

```
    Product Installation

    1 - Dual Fast Ethernet Dual SCSI/P Adapter
    2 - GigaSwift Ethernet Adapter (copper/fiber)
    3 - GigaSwift Ethernet Compact PCI Adapter
    4 - Quad GigaSwift Ethernet Adapter

         Please select the product to install:
```

5. **Select 4, the Quad GigaSwift Ethernet adapter, and answer the questions, if required, in the installation script.**

Note – After the installation script runs, you will see a summary of the software package or patch that was installed. You can also find a log of the installation in `/var/tmp`. The log file information is displayed when the installation completes unless the software meets or exceeds the required package or patch level.

Example 1: All Requirements Met

Following is an example of an installation script output if the driver and patch requirements are met:

CODE EXAMPLE 2-1 Script Output When Requirements are Met

```
            Product Installation

    1 - Dual Fast Ethernet Dual SCSI/P Adapter
    2 - GigaSwift Ethernet Adapter (copper/fiber)
    3 - GigaSwift Ethernet Compact PCI Adapter
    4 - Quad GigaSwift Ethernet Adapter

          Please select the product to install: 4

Quad GigaSwift Ethernet Adapter Driver Installation.
Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
```

CODE EXAMPLE 2-1 Script Output When Requirements are Met

```
Use is subject to license terms.

*** Checking for correct version of GigaSwift Ethernet Driver
Packages...
*** Checking for correct version of GigaSwift Ethernet Driver
Patches...

*** Checking for correct version of VLAN Utility Driver Packages..
*** Checking for correct version of VLAN Utility Driver Patches...

The CE/VLAN driver(s) installed on this system for:

        Quad GigaSwift Ethernet Adapter Driver

meets or exceeds the required package or patch version. No changes
to the system are required - exiting.
```

Example 2: Updates Required

The following is an example of an installation script output, if the packages are found and need updating by patching:

CODE EXAMPLE 2-2 Script Output When Packages Require Patches

```
            Product Installation

   1 - Dual Fast Ethernet Dual SCSI/P Adapter
   2 - GigaSwift Ethernet Adapter (copper/fiber)
   3 - GigaSwift Ethernet Compact PCI Adapter
   4 - Quad GigaSwift Ethernet Adapter

         Please select the product to install: 4

Quad GigaSwift Ethernet Adapter Driver Installation.

Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.

*** Checking for correct version of GigaSwift Ethernet Driver
Packages...
*** Checking for correct version of GigaSwift Ethernet Driver
Patches...

*** Checking for correct version of VLAN Utility Driver Packages..
*** Checking for correct version of VLAN Utility Driver Patches...
```

CODE EXAMPLE 2-2 Script Output When Packages Require Patches *(Continued)*

```
This script is about to take the following actions:
- Install GigaSwift Ethernet Patch: 112817-16.
- Install VLAN Ethernet Utility Patch: 114600-02.

Press return to continue, or 'Q' followed by a return to quit:

*** Installing patch 112817-16 for Solaris 9...

Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...

Patch number 112817-16 has been successfully installed.
See /var/sadm/patch/112817-16/log for details

Patch packages installed:
  SUNWcea
  SUNWceax
  SUNWced
  SUNWcedu
  SUNWcedu

*** Installing patch 114600-02 for Solaris 9...

Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...

Patch number 114600-02 has been successfully installed.
See /var/sadm/patch/114600-02/log for details

Patch packages installed:
  SUNWvld
  SUNWvldu
  SUNWvldx

Installation completed. Summary:

 Quad GigaSwift Ethernet Adapter Driver has been updated using
patch
 112817-16. It will be necessary to reboot the system to reload
 the updated driver.

 Optional VLAN Utility has been updated using  114600-02. It
 may be necessary to reboot the system to reload the updated
 driver.
```

CODE EXAMPLE 2-2 Script Output When Packages Require Patches *(Continued)*

```
A log of this Install can be found at:
    /var/tmp/GigaSwift.install.2003.11.17.1213
```

Example 3: Packages Missing

Following is an example of an installation script output if the package(s) are not found:

CODE EXAMPLE 2-3 Script Output When Packages Are Not Found

```
           Product Installation

   1 - Dual Fast Ethernet Dual SCSI/P Adapter
   2 - GigaSwift Ethernet Adapter (copper/fiber)
   3 - GigaSwift Ethernet Compact PCI Adapter
   4 - Quad GigaSwift Ethernet Adapter

         Please select the product to install: 4

Quad GigaSwift Ethernet Adapter Driver Installation.

Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.

*** Checking for correct version of GigaSwift Ethernet Driver
Packages...

*** Checking for correct version of VLAN Utility Driver Packages..

The following supporting driver(s) can be used with Quad GigaSwift
Ethernet Adapter
Driver:

    VLAN Utility Driver software
Install the Optional VLAN Utility Driver Software [y,n,?,q] y

This script is about to take the following actions:
- Install Quad GigaSwift Ethernet Adapter Driver packages.
- Optionally Install VLAN Utility Driver Software.

Press return to continue, or 'Q' followed by a return to quit:

*** Installing Sun GigaSwift Ethernet  packages...
Copyright 2002 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
```

CODE EXAMPLE 2-3 Script Output When Packages Are Not Found *(Continued)*

```
System configuration files modified but ce driver not loaded or
attached.

Installation of <SUNWced> was successful.
Copyright 2002 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.

Installation of <SUNWcedx> was successful.
Copyright 2002 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.

Installation of <SUNWcedu> was successful.
Copyright 2002 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.

Installation of <SUNWcea> was successful.
Copyright 2002 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.

Installation of <SUNWceax> was successful.

*** Installing Sun VLAN Utility  packages...
Copyright 2001 Sun Microsystems, Inc. All rights reserved.

Installation of <SUNWvld> was successful.
Copyright 2001 Sun Microsystems, Inc. All rights reserved.

Installation of <SUNWvldx> was successful.
Copyright 2001 Sun Microsystems, Inc. All rights reserved.

Installation of <SUNWvldu> was successful.

Installation completed. Summary:

 GigaSwift Ethernet Driver packages have been installed.

 Optional VLAN Utility packages have been installed.

A log of this Install can be found at:
    /var/tmp/GigaSwift.install.2003.11.17.1205
```

CHAPTER 3

Configuring Software for the Sun Quad GigaSwift Ethernet Adapter

This chapter describes how to configure files associated with the Sun Quad GigaSwift Ethernet adapter. This chapter contains the following sections:

- "Attaching the Gigabit Ethernet Interfaces to the Network" on page 24
- "Autonegotiation Protocol" on page 31
- "Setting the `local-mac-address?` Variable" on page 31
- "Booting the System Over a Gigabit Ethernet Interface" on page 34

Note – The configuration procedures in this chapter are correct for the Solaris 8 6/00 operating environment. If your system is running a subsequent compatible operating environment, refer to the *Platform Notes: The Sun Quad GigaSwift Ethernet Device Driver* for the latest configuration procedures. You can view this document on the `http://docs.sun.com` website.

Attaching the Gigabit Ethernet Interfaces to the Network

The adapter's Gigabit Ethernet network interfaces are controlled by the Sun GigaSwift Ethernet `ce`(1M) device driver. To use the adapter's `ce` interfaces, you must use the Solaris `ifconfig`(1M) command to activate the interfaces and to assign them Internet protocol (IP) addresses.

To make the interface changes persist over system reboots, you must create a `hostname.ce`*instance* file for each Ethernet interface on the adapter that you plan to use. For example, if you plan to use only two of the four Ethernet interfaces on the adapter, you will need to create two `hostname.ce`*ceinstance* files. You must also add both an IP address and a host name for the new Ethernet interfaces to the `/etc/inet/hosts` file.

Note – For more information about configuring Solaris networks, refer to the *Solaris System Administration Guide,* volumes 1 through 3. You can view these documents on the `http://docs.sun.com` website.

Now check the configuration

```
ifconfig ce0
ce0: flags=1000843<.....> mtu 1500
inet ip_address .......
ether 8:0:20:c4:51:df
ifconfig ce123000
ce123000: flags=1000843<.....> mtu 9194
inet ip_address-vlan-1 ......
ether 8:0:20:c4:51:df
```

▼ To Attach the Gigabit Ethernet Interfaces to the Network

1. **Log into the server.**

   Refer to your server's documentation for the correct procedure.

2. **At the terminal prompt, use the `grep` command to search the `/etc/path_to_inst` file for `ce` devices.**

```
% grep ce /etc/path_to_inst
"/pci@9,600000/pci@1/pci@0/network@0" 0 "ce"
"/pci@9,600000/pci@1/pci@0/network@1" 1 "ce"
"/pci@9,600000/pci@1/pci@4/network@2" 2 "ce"
"/pci@9,600000/pci@1/pci@4/network@3" 3 "ce"
```

In the preceding example, the four `network` instances are from a Sun Quad GigaSwift Ethernet adapter. The `path_to_inst` file contains the full device path names to the four `network` instances, the instance numbers (shown in bold type), and the driver names (`ce`). Refer to the `path_to_inst`(4) man page for more information about this file.

You will need to know the instance numbers when creating the `/etc/hostname.ce`*instance* files (where *instance* = `ce` instance number) in Step 3.

Note – Refer to your system's service manual for more information on how the adapter's device path names correspond to the PCI slot numbers in the system.

---

3. **Create an `/etc/hostname.ce`*instance* file for each network interface you plan to use.**

For each file, replace *instance* with the instance number you found in Step 2. For example, if you want to use all of the network interfaces from the example in Step 2, you must create four files.

File Name	Instance Number	Adapter Ethernet Network Interface
`/etc/hostname.ce0`	0	0
`/etc/hostname.ce1`	1	1
`/etc/hostname.ce2`	2	2
`/etc/hostname.ce3`	3	3

Follow these rules when creating `/etc/hostname.ce`*instance* files:

- Do not create `/etc/hostname.ce`*instance* files for adapter network interfaces you do not plan to use.
- The `/etc/hostname.ce`*instance* file must contain the host name for the appropriate network interface.
- The host name must be different from any other host name of any other interface. For example: `/etc/hostname.hme0` and `/etc/hostname.ce2` cannot have the same host name.

- The `/etc/hosts` file must contain an IP address for the hostname (see Step 4).

Using the instance examples in Step 2, the following example shows the four `/etc/hostname.ce`*instance* files required for a system called `zardoz`, which has a Sun Quad GigaSwift Ethernet adapter (`zardoz-11`, `zardoz-12`, `zardoz-13`, and `zardoz-14`).

```
# cat /etc/hostname.ce0
zardoz-11
# cat /etc/hostname.ce1
zardoz-12
# cat /etc/hostname.ce2
zardoz-13
# cat /etc/hostname.ce3
zardoz-14
```

4. **Create an entry in the `/etc/inet/hosts` file for each active `ce` network interface.**

You must create an entry for each `/etc/hostname.ce`*instance* file you created in Step 3.

Using the examples from Step 3, you would have:

```
# cat /etc/inet/hosts
#
# Internet host table
#
127.0.0.1localhost
129.144.10.57 zardozloghost
129.144.11.57 zardoz-11
129.144.12.57 zardoz-12
129.144.13.57 zardoz-13
129.144.14.57 zardoz-14
```

Refer to the `hosts`(4) man page and the Solaris documentation for more information about this file.

Note – The Internet protocol, version 6 (IPv6), expands the capabilities of IPv4, which is the current and default version. The device driver included in the Solaris operating environment supports both IPv4 and IPv6. IPv4 uses the `/etc/inet/hosts` configuration file, but IPv6 uses different configuration files. To transition to, manage, and implement IPv6, refer to the *Solaris System Administration Guide, Volume 3*.

---

5. **Activate the new** `ce` **interfaces using the** `ifconfig` **command.**

   For each adapter interface, type the following at the command line, replacing *instance* with the interface's instance number and *ip_address* with the interface's IP address:

```
# ifconfig ceinstance plumb ip_address  up
```

   For example, if you wanted to activate all of the sample `ce` interfaces shown in Step 4, you would type:

```
# ifconfig ce0 plumb 129.144.11.57 up
# ifconfig ce1 plumb 129.144.12.57 up
# ifconfig ce2 plumb 129.144.13.57 up
# ifconfig ce3 plumb 129.144.14.57 up
```

Jumbo Frames Capability

The `ce` driver is capable of accepting jumbo frames. You can enable this capability using the `accept-jumbo` driver configuration variable. See "Jumbo Frames" on page 46.

With jumbo frames enabled the interface can operate with an ethernet frame size as large as 9216 bytes, which includes the maximum size ethernet header for VLAN of 18 bytes and the CRC. Control of the actual MTU the link uses is provided through the `ifconfig` utility.

Note – Jumbo frames is not supported for 10/100 Mbps environments.

▼ To Enable Jumbo Frames

You can enable jumbo frames in three ways:

- **To set jumbo frames globally on all ce devices, add a line to** `ce.conf` **file (**`accept-jumbo=1`**). Note: this enables jumbo frames on *all*** `ce` **devices.**

- **To set jumbo frames for a specific instance of** `ce`**, add a line into** `ce.conf` **file for that particular instance to be set to** `accept-jumbo`**:**

```
# grep ce /etc/path_to_inst
  "/pci@1f,0/network@a" 0 "ce"
  "/pci@1f,0/network@a" 0 "ce" accept-jumbo=1;
```

- **Use the** `ndd` **command to enable jumbo frames that will not persist when you reboot the system:**

```
# ndd -set /dev/ce instance 0
# ndd -set /dev/ce accept-jumbo 1
```

Note – If you want to use the jumbo frames capability, be sure that jumbo frames is enabled on the link partner. If you are using Sun Trunking 1.3, make sure all members of the trunk have jumbo frames enabled and are using the same MTU.

Managing MTU Using `ifconfig`

The `ifconfig` utility allows you to manage the MTU by lowering it from the maximum provided by DLPI driver.

The default setting of the MTU as seen by `ifconfig` is defined as the MTU advertised by the driver. For this example, the default setting for the MTU is 9194, and is limited by this value based on `ce` hardware limitation.

Once the `accept-jumbo` property is set to 1 any `ifconfig` on a stream will automatically be set for jumbo frames.

```
ifconfig ce0 plumb ip_address up
ifconfig ce0
ce0: flags=1000843<.....> mtu 9194
inet ip_address netmask ffffff00 broadcast 10.4.127.255
ether 8:0:20:c4:51:df
```

You might want to individually configure the MTU down from the default setting to the regular MTU of Ethernet.

Use the `ifconfig mtu` property to lower the setting:

```
ifconfig ce0 mtu 1500
ifconfig ce0
ce0: flags=1000843<.....> mtu 1500
inet ip_address netmask ffffff00 broadcast 10.4.127.255
ether 8:0:20:c4:51:df
```

The DLPI driver remains in jumbo frames mode, but the upper layer rejects packets greater than 1500.

The following examples illustrate this further.

Example 1

The primary interface is the regular Ethernet MTU, but a network contained in a VLAN with VID 123 would work best with jumbo frames enabled. Set `accept-jumbo` globally.

```
ifconfig ce0 plumb ip_address mtu 1500 up
ifconfig ce123000 plumb ip_address-vlan-1 up
```

Check the configuration.

```
ifconfig ce0
ce0: flags=1000843<.....> mtu 1500
inet ip_address .......
ether 8:0:20:c4:51:df
ifconfig ce123000
ce123000: flags=1000843<.....> mtu 9194
inet ip_address-vlan-1 .......
ether 8:0:20:c4:51:df
```

Example 2

This example is similar to Example 1 with one exception. A new VLAN with VID 124 does not support jumbo frames as part of the configuration. `accept-jumbo` is set globally.

```
ifconfig ce123000 plumb ip_address-vlan-2 mtu 1500 up
```

Check the configuration.

```
ifconfig ce0
ce0: flags=1000843<.....> mtu 1500
inet ip_address .......
ether 8:0:20:c4:51:df
ifconfig ce123000
ce123000: flags=1000843<.....> mtu 9194
inet ip_address-vlan-1 .......
ether 8:0:20:c4:51:df
ifconfig ce124000
ce0: flags=1000843<.....> mtu 1500
inet ip_address-vlan-2 .......
ether 8:0:20:c4:51:df
```

This example can be maintained through a reboot using the `hostname.ce`*instance* file. To achieve that the following hostname files need to be created for the previous examples.

Reboot Persistence

The following examples build on the preceding dynamic configuration to enable the final established configuration to persist through a reboot.

Example 1

```
hostname.ce0
ip_address mtu 1500
hostname.ce123000
ip_address-vlan-1
```

Example 2

Since this builds on the system configured in Example 1, only another hostname file is necessary. In this case, specify the MTU.

```
hostname.ce124000
ip_address-vlan-2 mtu 1500
```

Autonegotiation Protocol

A key feature of the Sun Quad GigaSwift Ethernet adapter adapter is auto-negotiation. The *auto-negotiation* protocol, as specified by the IEEE 802.3 standard, selects the operation mode (half-duplex or full-duplex), and the auto-sensing protocol selects the speed (10 Mbps, 100 Mbps or 1000Mbps) for the adapter.

The link speed and modes supported by the Sun Quad GigaSwift Ethernet adapter are listed in decreasing order of priority:

- 1000Mbps full-duplex
- 1000Mbps half-duplex (not supported)
- 100 Mbps, full-duplex
- 100 Mbps, half-duplex
- 10 Mbps, full-duplex
- 10 Mbps, half-duplex

When the system is booted, the Sun Quad GigaSwift Ethernet adapter advertises these capabilities to the link partner at the other end of the link (a hub, switch, or another network interface card in a host system). If the link partner also supports auto-negotiation, it advertises its capabilities over the link. The highest common priority mode supported by both sides is selected for the link operation.

If the Sun Quad GigaSwift Ethernet adapter is connected to a remote system or interface that is not capable of auto-negotiation, your system automatically selects the speed and half-duplex mode.

If the adapter is connected to a link partner with which the auto-negotiation protocol fails to operate successfully, you can configure the device not to use this protocol and force the driver to set up the link in the mode and speed of your choice. See “Setting Parameters Using the `ndd` Utility” on page 47 for more information.

Setting the `local-mac-address?` Variable

Each of the four network interfaces of the adapter has been assigned a unique media access control (MAC) address, which represents the 48-bit Ethernet address for that network interface. The OpenBoot firmware reports this MAC address by means of the `local-mac-address?` variable in the device nodes corresponding to the network interfaces.

A system is not obligated to use this assigned MAC address if it has a system-wide MAC address. In such cases, the system-wide MAC address applies to all network interfaces on the system.

The device driver, or any other adapter utility, can use the network device's MAC address (`local-mac-address?`) while configuring the device. A network interface's MAC address can be used when booting over the network.

The `mac-address` property of the network device specifies the network address (system-wide or `local-mac-address?`) used for booting the system. To start using the MAC addresses assigned to the network interfaces of the adapter, set the NVRAM configuration variable `local-mac-address?` to `true`, as shown in the procedure below.

To Set the `local-mac-address?` Variable to the Network Interfaces

Before starting this procedure, make sure that all network activity over the adapter's network interfaces has stopped.

1. **Log into the system as superuser.**

   Refer to the system's documentation for the correct procedure.

2. **At a terminal prompt, type the following command to set the property to** `true`:

```
% eeprom local-mac-address?=true
```

3. **Use the** `ifconfig` **command to stop all network activity traveling over the adapter's Ethernet interfaces.**

   For example, if you were using the adapter's four interfaces, and they were named `ce0`, `ce1`, `ce2`, and `ce3`, you would type:

```
# ifconfig ce0 down unplumb
# ifconfig ce1 down unplumb
# ifconfig ce2 down unplumb
# ifconfig ce3 down unplumb
```

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

4. **Disconnect and unconfigure the adapter from the system using the appropriate basic DR command.**

   The command and usage may be unique to different systems, so refer to your system's service manual for the appropriate procedure.

   The following example is for a server with a Sun Quad GigaSwift Ethernet adapter installed in the second PCI slot (with the `pci_pci0:PCI_slot2` attachment point ID.

```
# cfgadm -c unconfigure pci_pci0:PCI_slot2
# cfgadm -c disconnect pci_pci0:PCI_slot2
```

5. **To make the adapter's network interfaces use the local MAC addresses, connect and configure the adapter to the system using the appropriate basic hot plug command.**

   The command and usage might be unique to different systems, so refer to your system's service manual for the appropriate procedure.

   The following example is for a server with a Sun Quad GigaSwift Ethernet adapter installed in the second PCI slot (with the `pci_pci0:PCI_slot2` attachment point ID.

```
# cfgadm -c connect pci_pci0:PCI_slot2
# cfgadm -c configure pci_pci0:PCI_slot2
```

6. **Use the `ifconfig` command to activate the adapter's Ethernet interfaces.**

   For each adapter interface, type the following at the command line, replacing instance with the interface's instance number and ip_address with the interface's IP address:

```
# ifconfig ceinstance plumb ip_address up
```

   For example, if you were using the adapter's four interfaces, and they were named `ce0`, `ce1`, `ce2`, and `ce3`, you would type:

```
# ifconfig ce0 plumb ip_address up
# ifconfig ce1 plumb ip_address up
# ifconfig ce2 plumb ip_address up
# ifconfig ce3 plumb ip_address up
```

Booting the System Over a Gigabit Ethernet Interface

Perform the following procedure to use a Sun Quad GigaSwift Ethernet adapter Ethernet interface as the boot device.

▼ To Boot the System Over the Network

1. **Before shutting down the operating environment and halting the system, make sure that all significant application activity on the server has stopped.**

2. **Follow the appropriate procedures, as documented in the system's service manual, to shut down and halt the system.**

   Refer to the system's documentation for the complete power off procedure.

   After shutting down the system, you should see the OpenBoot PROM `ok` prompt on the system console.

3. **At the** `ok` **prompt, type** `show-nets` **to list the network devices on your system.**

   You should see the full path name of the `ce` devices, similar to the following example:

```
ok show-nets
a) /pci@9,600000/pci@1/pci@4/network@3
b) /pci@9,600000/pci@1/pci@4/network@2
c) /pci@9,600000/pci@1/pci@0/network@1
d) /pci@9,600000/pci@1/pci@0/network@0
e) /pci@9,700000/network@1,1
f) /pci@8,600000/network@1
q) NO SELECTION
Enter Selection, q to quit:
```

4. Select the `ce` device that you want to use as a boot device.

Using the show-nets command, select the letter of the device you want to use. For example:

```
Enter Selection, q to quit: c
/pci@9,600000/pci@1/pci@0/network@1 has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev ^Y
       for creating devalias mydev for /pci@9,600000/pci@1/pci@0/
network@1
```

Note – Select only one of these `ce` devices for booting.

---

5. At the `ok` prompt the type the `boot` command and press the Control-Y key sequence to paste the full path name of the selected `ce` device

You can

- Type `boot` and the full path name to the `ce` device.

Or

- Type `boot` and press the Control-Y key sequence.

In this example, the Control-Y key sequence would be replaced by the full path name of the `ce` device:

```
ok boot /pci@9,600000/pci@1/pci@0/network@1
```

6. Press the Return key to boot the system over the `ce` device.

CHAPTER 4

Configuring the Sun Quad GigaSwift Ethernet Device Driver Parameters

The ce(1M) device driver controls the four Sun Quad GigaSwift Ethernet interfaces. You can manually set the ce device driver parameters to customize each device in your system. This appendix lists the available ce device driver parameters and describes how you can set these parameters.

This appendix contains the following sections:

- "Usability Enhancements to the Driver" on page 53
- "GigaSwift Ethernet Driver Operating Statistics" on page 54

Note – The configuration procedures in this chapter are correct for the Solaris 8 6/00 operating environment. If your system is running a future compatible operating environment, refer to the *Platform Notes: The Sun GigaSwift Ethernet Device Driver* document for the latest configuration procedures. You can view this manual on the `http://docs.sun.com` website.

Sun Quad GigaSwift Ethernet Device Driver Parameters

This section describes the hardware overview of the local transceiver used in the adapter and defines the `ce` device driver parameters.

Hardware Overview

Each of the four `ce` channels provides 10BASE-T, 100BASE-TX, or 1000BASE-T networking interfaces. The device driver automatically sets the link speed to 10, 100, or 1000 Mbps and conforms to the IEEE 802.3 Ethernet standard. The single MAC/PHY chip provides the PCI interface and medium access control (MAC) functions. The local transceiver, which connects to an RJ-45 connector, provides the physical-layer functions.

The local transceiver is capable of all the operating speeds and modes listed below:

- 1000 Mbps, full-duplex
- 1000 Mbps, half-duple (not supported)
- 100 Mbps, full-duplex
- 100 Mbps, half-duplex
- 10 Mbps, full-duplex
- 10 Mbps, half-duplex

The Sun Quad GigaSwift Ethernet adapter performs autonegotiation with the remote end of the link (the link partner) to select a common mode of operation.

Sun Quad GigaSwift Ethernet adapter also supports a non-autonegotiated mode of operation. You can set the speed and mode using the `ndd` utility, or, to make sure the changes persist after a system reboot, you can create a `ce.conf` file. See “To Set Driver Parameters Using a `ce.conf` File” on page 51 for more information about setting driver parameters.

Device Driver Parameter Values and Definitions

TABLE 4-1 lists the ce device driver parameters.

TABLE 4-1 ce Driver Parameter, Status, and Descriptions

Parameter	Status	Description
instance	Read and write	Device instance
adv-autoneg-cap	Read and write	Operational mode parameter
adv-1000fdx-cap	Read and write	Operational mode parameter
adv-1000hdx-cap	Read and write	Operational mode parameter
adv-100T4-cap	Read and write	Operational mode parameter
adv-100fdx-cap	Read and write	Operational mode parameter
adv-100hdx-cap	Read and write	Operational mode parameter
adv-10fdx-cap	Read and write	Operational mode parameter
adv-10hdx-cap	Read and write	Operational mode parameter
adv-asmpause-cap	Read and write	Flow control parameter
adv-pause-cap	Read and write	Flow control parameter
master-cfg-enable	Read and write	Gigabit link clock mastership controls
master-cfg-value	Read and write	Gigabit link clock mastership controls
use-int-xcvr	Read and write	
enable-ipg0	Read and write	Enable additional delay before transmitting a packet
ipg0	Read and write	Additional delay before transmitting a packet
ipg1	Read and write	Interpacket gap parameter
ipg2	Read and write	Interpacket gap parameter
rx-intr-pkts	Read and write	Receive interrupt blanking values
rx-intr-time	Read and write	Receive interrupt blanking values
red-dv4to6k	Read and write	Random early detection and packet drop vectors
red-dv6to8k	Read and write	Random early detection and packet drop vectors
red-dv8to10k	Read and write	Random early detection and packet drop vectors
red-dv10to12k	Read and write	Random early detection and packet drop vectors
tx-dma-weight	Read and write	PCI interface parameter
rx-dma-weight	Read and write	PCI interface parameter

TABLE 4-1 `ce` Driver Parameter, Status, and Descriptions *(Continued)*

Parameter	Status	Description
`infinite-burst`	Read and write	PCI interface parameter
`disable-64bit`	Read and write	PCI interface parameter
`accept-jumbo`	Read and write	

Operational Mode Parameters

The following parameters determine the transmit and receive speed and duplex. TABLE 4-2 describes the operational mode parameters and their default values.

TABLE 4-2 Operational Mode Parameters

Parameter	Description
`adv-autoneg-cap`	Local interface capability advertised by the hardware 0 = Forced mode 1 = Autonegotiation (default)
`adv-1000fdx-cap`	Local interface capability advertised by the hardware 0 = Not 1000 Mbit/sec full-duplex capable 1 = 1000 Mbit/sec full-duplex capable (default)
`adv-1000hdx-cap`	Local interface capability advertised by the hardware 0 = Not 1000 Mbit/sec half-duplex capable 1 = 1000 Mbit/sec half-duplex capable (default)
`adv-100T4-cap`	Local interface capability advertised by the hardware 0 = Not 100T4 capable (default) 1 = 100T4 capable
`adv-100fdx-cap`	Local interface capability advertised by the hardware 0 = Not 100 Mbit/sec full-duplex capable 1 = 100 Mbit/sec full-duplex capable (default)
`adv-100hdx-cap`	Local interface capability advertised by the hardware 0 = Not 100 Mbit/sec half-duplex capable 1 = 100 Mbit/sec half-duplex capable (default)
`adv-10fdx-cap`	Local interface capability advertised by the hardware 0 = Not 10 Mbit/sec full-duplex capable 1 = 10 Mbit/sec full-duplex capable (default)
`adv-10hdx-cap`	Local interface capability advertised by the hardware 0 = Not 10 Mbit/sec half-duplex capable 1 = 10 Mbit/sec half-duplex capable (default)

Note – If a parameter's initial setting is 0, it cannot be changed. If you try to change it, it will revert back to 0.

If all these parameters are set to 1, autonegotiation uses the highest speed possible. If all these parameters are set to 0, you receive the following error message:

```
NOTICE: Last setting will leave ce1 with no link capabilities.
WARNING: ce1: Restoring previous setting.
```

Flow Control Parameters

The `ce` device is capable of sourcing (transmitting) and terminating (receiving) pause frames conforming to the IEEE 802.3x Frame Based Link Level Flow Control Protocol. In response to received flow control frames, the `ce` device can slow down its transmit rate. On the other hand, the `ce` device is capable of sourcing flow control frames, requesting the link partner to slow down, provided that the link partner supports this feature. By default, the driver advertises both transmit and receive pause capability during autonegotiation.

TABLE 4-3 provides flow control keywords and describes their function.

TABLE 4-3 Read-Write Flow Control Keyword Descriptions

Keyword	Description
`adv-asmpause-cap`	The adapter supports asymmetric pause, which means it can pause only in one direction. 0=Off (default) 1=On
`adv-pause-cap`	This parameter has two meanings depending on the value of `adv-asmpause-cap`. (Default=0) If `adv-asmpause-cap` = 1 while `adv-pause-cap` = 1 pauses are received. If `adv-asmpause-cap` = 1 while `adv-pause-cap` = 0 pauses are transmitted. If `adv-asmpause-cap` = 0 while `adv-pause-cap` = 1 pauses are sent and received. If `adv-asmpause-cap` = 0 then `adv-pause-cap` determines whether Pause capability is on or off.

Gigabit Link Clock Mastership Controls

The concept of link clock mastership was introduced with gigabit twisted-pair technology. One side of the link is the master that provides the link clock and the other is the slave that uses the link clock. Once this relationship is established, the link is up, and data can be communicated. Two physical layer parameters control whether your side is the master or the slave or whether mastership is negotiated with the link partner.

TABLE 4-4 Forced Mode Parameter

Parameter	Description
`master-cfg-enable`	Determines whether or not during the auto-negotiation process the link clock mastership is setup automatically.
`master-cfg-value`	• If the `master-cfg-enable` parameter is set then the mastership is not set up automatically but depends on the value of `master-cfg-value`. If the `master-cfg-value` is set then the physical layer expects the local device to be the link master. If it is not set then it expects the link partner to be the master. • If Auto-negotiation is not enabled, then the value of `master-cfg-enable` is ignored and the value of `master-cfg-value` is key to the link clock mastership. If the `master-cfg-value` is set, then the physical layer expects the local device to be the link master. If the value is not set then it expects the link partner to the master.

Caution – Do not adjust the link clock mastership parameters unless you clearly understand the settings of the link partner. Incorrect link clock mastership configuration results in link up failure.

Interpacket Gap Parameters

The `ce` device supports a programmable mode called `enable-ipg0`.

When a driver receives a packet with `enable-ipg0` set (the default), it adds an additional time delay before transmitting the packet. This delay, set by the `ipg0` parameter, is in addition to the delay set by the `ipg1` and `ipg2` parameters. The additional `ipg0` delay helps to reduce collisions.

If `enable-ipg0` is disabled, the value of `ipg0` is ignored and no additional delay is set. Only the delays set by `ipg1` and `ipg2` will be used. Disable `enable-ipg0` if other systems keep sending a large number of back-to-back packets. Systems that have `enable-ipg0` set might not have enough time on the network.

You can add the additional delay by setting the ipg0 parameter from 0 to 255, which is the media byte time delay.

TABLE 4-5 defines the enable-ipg0 and ipg0 parameters.

TABLE 4-5 Parameters Defining enable-ipg0 and ipg0

Parameter	Values	Description
enable-ipg0	0 1	enable-ipg0 reset enable-ipg0 set (Default=8)
ipg0	0 to 255	The additional time delay (or gap) before transmitting a packet (after receiving the packet) (Default=8)

The ce device supports the programmable interpacket gap (IPG) parameters ipg1 and ipg2. The total IPG is the sum of ipg1 and ipg2. The total IPG is 0.096 microseconds for the link speed of 1000 Mbps.

TABLE 4-6 lists the default values and allowable values for the IPG parameters.

TABLE 4-6 Read-Write Interpacket Gap Parameter Values and Descriptions

Parameter	Values (Byte-time)	Description
ipg1	0 to 255	Interpacket gap 1 (Default = 8)
ipg2	0 to 255	Interpacket gap 2 (Default = 4)

By default, the driver sets ipg1 to 8-byte time and ipg2 to 4-byte time, which are the standard values. (Byte time is the time it takes to transmit one byte on the link, with a link speed of 1000 Mbps.)

If your network has systems that use longer IPG (the sum of ipg1 and ipg2), and if those machines seem to be slow in accessing the network, increase the values of ipg1 and ipg2 to match the longer IPGs of other machines.

Interrupt Parameters

TABLE 4-7 describes the receive interrupt blanking values.

TABLE 4-7 RX Blanking Register for Alias Read

Field Name	Values	Description
`rx-intr-pkts`	0 to 511	Interrupt after this number of packets have arrived since the last packet was serviced. A value of zero indicates no packet blanking. (Default=8)
`rx-intr-time`	0 to 524287	Interrupt after 4.5 microseconds ticks have elapsed since the last packet was serviced. A value of zero indicates no time blanking. (Default=3)

Random Early Drop Parameters

TABLE 4-8 describes the RX random early detection 8-bit vectors, which allows you to enable random early drop (RED) thresholds. When received packets reach the RED range packets are dropped according to the preset probability. The probability should increase when the FIFO level increases. Control packets are never dropped and are not counted in the statistics.

TABLE 4-8 RX Random Early Detecting 8-Bit Vectors

Field Name	Values	Description
`red-dv4to6k`	0 to 255	Random early detection and packet drop vectors for when FIFO threshold is greater than 4096 bytes and less than 6,144 bytes. Probability of drop can be programmed on a 12.5 percent granularity. For example, if bit 0 is set, the first packet out of every eight will be dropped in this region. (Default=0)

TABLE 4-8 RX Random Early Detecting 8-Bit Vectors *(Continued)*

Field Name	Values	Description
`red-dv6to8k`	0 to 255	Random early detection and packet drop vectors for when FIFO threshold is greater than 6,144 bytes and less than 8,192 bytes. Probability of drop can be programmed on a 12.5 percent granularity. For example, if bit 0 is set, the first packet out of every eight will be dropped in this region. (Default=0)
`red-dv8to10k`	0 to 255	Random early detection and packet drop vectors for when FIFO threshold is greater than 8,192 bytes and less than 10,240 bytes. Probability of drop can be programmed on a 12.5 percent granularity. For example, if bit 1 and 6 are set, the second and seventh packets out of every eight will be dropped in this region. (Default=0)
`red-dv10to12k`	0 to 255	Random early detection and packet drop vectors for when FIFO threshold is greater than 10,240 bytes and less than 12,288 bytes. Probability of drop can be programmed on a 12.5 percent granularity. If bit 2, 4 and 6 are set, then the third, fifth and seventh packets out of every eight will be dropped in this region. (Default=0)

PCI Bus Interface Parameters

These parameters enable you to modify PCI interface features to gain better PCI performance for a given application.

TABLE 4-9 PCI Bus Interface Parameters

Parameter	Description
`tx-dma-weight`	Determines the multiplication factor for granting credit to the TX side during a weighted round robin arbitration. Values are 0 to 3. (Default=0) Zero means no extra weighting. The other values are power of 2 extra weighting, on that traffic. For example of `tx-dma-weight` = 0 and `rx-dma-weight` = 3 then as long as RX traffic is continuously arriving its priority will be 8 times greater than TX to access the PCI.

TABLE 4-9 PCI Bus Interface Parameters

Parameter	Description
`rx-dma-weight`	Determines the multiplication factor for granting credit to the RX side during a weighted round robin arbitration. Values are 0 to 3. (Default=0)
`infinite-burst`	Enables the infinite burst capability to be utilized. When this is in effect and the system supports infinite burst, the adapter does not free the bus until complete packets are transferred across the bus. Values are 0 or 1. (Default=0)
`disable-64bit`	Switches off 64-bit capability of the adapter. In some cases, it is useful to switch off this feature. Values are 0 or 1. (Default=0, which enables 64 bit capability)

Jumbo Frames

The variable `accept-jumbo` allows the driver to instruct the upper layers that the driver is operating in jumbo frames mode.

TABLE 4-10 `accept-jumbo` Parameters

Parameter	Values	Description
`accept-jumbo`	1	Enables jumbo frames mode and sets MTU to greater than 1500.
	0	Jumbo frames is not enabled. MTU is at 1500.

Setting `ce` Driver Parameters

You can set the `ce` device driver parameters in two ways:

- Using the `ndd` utility
- Using the `ce.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 parameters so they remain in effect after you reboot the system, create a `/platform/sun4u/kernel/drv/ce.conf` file and 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 `ndd` utility supports any networking driver, which implements the Data Link Provider Interface (DLPI).

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

▼ To Specify Device Instances for the `ndd` Utility

Before you use the `ndd` utility to get or set a parameter for a `ce` 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 ce /etc/path_to_inst
"/pci@9,600000/pci@1/pci@0/network@0" 0 "ce"
"/pci@9,600000/pci@1/pci@0/network@1" 1 "ce"
"/pci@9,600000/pci@1/pci@4/network@2" 2 "ce"
"/pci@9,600000/pci@1/pci@4/network@3" 3 "ce"
```

In the previous example, the four GigaSwift Ethernet instances are from the installed adapter. The instance numbers are in bold italics for clarity.

2. **Use the instance number to select the device.**

```
# ndd -set /dev/ce instance instance
```

The device remains selected until you change the selection.

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 Use the `ndd` Utility in Noninteractive Mode

This section describes how to modify and display parameter values.

- **To modify a parameter value, use the** `-set` **option.**

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

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

When you change any adv parameter, a message similar to the following appears:

```
ce1: xcvr addr:0x01 - link up 1000 Mbps full duplex
```

- **To display the value of a parameter, specify the parameter name and omit the value.**

When you omit the `-set` option, a query operation is assumed and the utility queries the named driver instance, retrieves the value associated with the specified parameter, and prints it:

```
# ndd /dev/ce parameter
```

▼ To Use the `ndd` Utility in Interactive Mode

- **To modify a parameter value in interactive mode, specify** `ndd /dev/ce`**, as shown below.**

The `ndd` utility then prompts you for the name of the parameter:

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

After typing the parameter name, the `ndd` utility prompts you for the parameter value (see TABLE 4-1 through TABLE 4-12).

- **To list all the parameters supported by the `ce` driver, type `ndd /dev/ce`.**

  (See TABLE 4-1 through TABLE 4-12 for parameter descriptions.)

```
# ndd /dev/ce
name to get/set ? ?
?                             (read only)
instance                      (read and write)
adv_autoneg_cap               (read and write)
adv_1000fdx_cap               (read and write)
adv_1000hdx_cap               (read and write)
adv_100T4_cap                 (read and write)
adv_100fdx_cap                (read and write)
adv_100hdx_cap                (read and write)
adv_10fdx_cap                 (read and write)
adv_10hdx_cap                 (read and write)
adv_asmpause_cap              (read and write)
adv_pause_cap                 (read and write)
master_cfg_enable             (read and write)
master_cfg_value              (read and write)
use_int_xcvr                  (read and write)
enable_ipg0                   (read and write)
ipg0                          (read and write)
ipg1                          (read and write)
ipg2                          (read and write)
rx_intr_pkts                  (read and write)
rx_intr_time                  (read and write)
red_dv4to6k                   (read and write)
red_dv6to8k                   (read and write)
red_dv8to10k                  (read and write)
red_dv10to12k                 (read and write)
tx_dma_weight                 (read and write)
rx_dma_weight                 (read and write)
infinite_burst                (read and write)
disable_64bit                 (read and write)
accept_jumbo                  (read and write)
name to get/set ?
#
```

Setting the Autonegotiation Mode

By default, autonegotiation is set to `on`. This means that the adapter communicates with its link partner to determine a compatible network speed, duplex mode, and flow control capability.

To Disable Auto-negotiation Mode

If your network equipment does not support autonegotiation, or if you want to specify your network speed, you can set autonegotiation to `off` on the `ce` device.

Note – Disabling auto-negotiation mode can cause collisions.

---

1. **Set the following driver parameters to the values that are described in the documentation that shipped with your link partner (for example, a switch):**

   - `adv-1000fdx-cap`
   - `adv-1000hdx-cap`
   - `adv-100fdx-cap`
   - `adv-100hdx-cap`
   - `adv-10fdx-cap`
   - `adv-10hdx-cap`
   - `adv-asmpause-cap`
   - `adv-pause-cap`

   See TABLE 4-2 for the descriptions and possible values of these parameters.

Note – According to IEEE 802.3 specifications, when using 1000 mbps speed over copper media, auto-negotiation is not optional. In the absense of auto-negotiation, (for example, while using forced mode), link syncing between link partners may not happen and the link may not come up.

---

2. **Set the** `adv-autoneg-cap` **parameter to** `0`**.**

```
# ndd -set /dev/ce adv-autoneg-cap 0
```

When you change any `ndd` link parameter, a message similar to the following appears:

```
xcvr addr:0x00 - link up 100 Mbps full duplex
```

Setting Parameters Using the ce.conf File

Specify the driver parameter properties on a per-device basis by creating a ce.conf file in the /platform/sun4u/kernel/drv directory. Use a ce.conf file when you need to set a particular parameter for a device in the system. The parameters you set are read and write parameters that are listed in “Device Driver Parameter Values and Definitions” on page 39.

Note – Configuring the parameters by putting ndd commands in rcX.d scripts is not supported.

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

- **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 a ce.conf File

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

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

```
# grep ce /etc/driver_aliases
ce "pci108e,abba"
ce "pci100b,35"
```

Note – ce "pci100b,35" is used by the Sun Quad GigaSwift Ethernet device.

---

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

```
# grep ce /etc/path_to_inst
"/pci@9,600000/pci@1/pci@0/network@0" 0 "ce"
"/pci@9,600000/pci@1/pci@0/network@1" 1 "ce"
"/pci@9,600000/pci@1/pci@4/network@2" 2 "ce"
"/pci@9,600000/pci@1/pci@4/network@3" 3 "ce"
```

- In the previous 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).
  - The last part in double quotes is the driver name.
- In the hardware node name, the last component after the last / character and before the @ character is the device name.
- The path name before the last component is the parent name.
- The number after the final @ character within quotes is referred to as unit-address.

To identify a PCI device unambiguously in the `ce.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 PCI device specification.

In the first line in the previous example:

- `parent` = “`pci@9,600000`″
- `unit-address` = “`0`″

In the second line in the previous example:

- `parent` = “`pci@9,600000`″
- `unit-address` = “`1`″

In the third line in the previous example:

- `parent` = “`pci@9,600000`”
- `unit-address` = “`2`″

In the fourth line in the previous example:

- `parent` = “`pci@9,600000`”
- `unit-address` = “`3`″

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

In the following example, the `adv_autoneg_cap` and `adv_1000fdx_cap` parameters are set for all Sun Quad GigaSwift Ethernet devices. See the `driver.conf(4)` man page for more information.

```
adv-autoneg-cap=0 adv-1000fdx-cap=0;
```

In the following example, the `adv-autoneg-cap` and `adv-1000fdx-cap` parameters are set for a single instance of the Sun Quad GigaSwift Ethernet device.

```
name="pci100b,35" parent="pci@9,600000" unit-address="2"
adv-autoneg-cap=0 adv-100hdx-cap=0 adv-100fdx-cap=1 adv-1000fdx-cap=0 adv-
10hdx-cap=0 adv-10fdx-cap=0 adv-1000hdx-cap=0 adv-100T4-cap=0;
```

Note – The difference between setting parameters for all Sun Quad GigaSwift Ethernet devices and setting parameters for a single instance of the device depends on whether you include the `name=`, `parent=`, and `unit-address=`. If you omit these definitions, the settings become global to all Quad Sun GigaSwift Ethernet instances.

3. **Save the** `ce.conf` **file.**

Usability Enhancements to the Driver

In older Ethernet device drivers, determining the link status for a driver required a two-step process:

First, you had to set the instance:

```
# ndd -set /dev/ce instance 2
```

Then you had to get the link status:

```
# ndd -get /dev/ce adv_autoneg_cap
1
```

Starting with the Sun Quad GigaSwift Ethernet adapter in the Solarlis 9 operating environment this method of determining the link status is discouraged, and in some cases, it is removed completely as a driver feature.

The new improved approach moves all read-only parameters from the ndd options into kstat. This simplifies getting link status information by allowing you to do it with one simple command.

- **To get the link status of a driver, type the following command:**

```
# kstat ce:n | grep link_up
link_up                          0
```

Where *n* = instance.

For more information about the kstat command, refer to the kstat man page.

GigaSwift Ethernet Driver Operating Statistics

These statistics are part of the statistics presented by the netstat -k command.

TABLE 4-11 describes the read-only Media Independent Interface (MII) capabilities. These parameters define the capabilities of the hardware. The Gigabit Media Independent Interface (GMII) supports all of the following capabilities.

TABLE 4-11 Read-Only ce Device Capabilities

Parameter	Description (Local interface Capabilities)
cap_autoneg	0 = Not capable of autonegotiation 1 = Autonegotiation capable
cap_1000fdx	Local interface full-duplex capability 0 = Not 1000 Mbit/sec full-duplex capable 1 = 1000 Mbit/sec full-duplex capable
cap_1000hdx	Local interface half-duplex capability 0 = Not 1000 Mbit/sec half-duplex capable 1 = 1000 Mbit/sec half-duplex capable
cap_100fdx	Local interface full-duplex capability 0 = Not 100 Mbit/sec full-duplex capable 1 = 100 Mbit/sec full-duplex capable

TABLE 4-11 Read-Only `ce` Device Capabilities *(Continued)*

Parameter	Description (Local interface Capabilities)
`cap_100hdx`	Local interface half-duplex capability 0 = Not 100 Mbit/sec half-duplex capable 1 = 100 Mbit/sec half-duplex capable
`cap_10fdx`	Local interface full-duplex capability 0 = Not 10 Mbit/sec full-duplex capable 1 = 10 Mbit/sec full-duplex capable
`cap_10hdx`	Local interface half-duplex capability 0 = Not 10 Mbit/sec half-duplex capable 1 = 10 Mbit/sec half-duplex capable
`cap_asm_pause`	Local interface flow control capability 0 = Not asymmetric pause capable 1 = Asymmetric pause (from the local device) capable
`cap_pause`	Local interface flow control capability 0 = Not Symmetric pause capable 1 = Symmetric pause capable

Reporting the Link Partner Capabilities

TABLE 4-12 describes the read-only link partner capabilities.

TABLE 4-12 Read-Only Link Partner Capabilities

Parameter	Description
`lp_cap_autoneg`	0 = No autonegotiation 1 = Autonegotiation
`lp_cap_1000fdx`	0 = No 1000 Mbit/sec full-duplex transmission 1 = 1000 Mbit/sec full-duplex
`lp_cap_1000hdx`	0 = No 1000 Mbit/sec half-duplex transmission 1 = 1000 Mbit/sec half-duplex
`lp_cap_100fdx`	0 = No 100 Mbit/sec full-duplex transmission 1 = 100 Mbit/sec full-duplex
`lp_cap_100hdx`	0 = No 100 Mbit/sec half-duplex transmission 1 = 1000 Mbit/sec half-duplex
`lp_cap_10fdx`	0 = No 10 Mbit/sec full-duplex transmission 1 = 10 Mbit/sec full-duplex

TABLE 4-12 Read-Only Link Partner Capabilities *(Continued)*

Parameter	Description
`lp_cap_10hdx`	0 = No 10 Mbit/sec half-duplex transmission 1 = 10 Mbit/sec half-duplex
`lp_cap_asm_pause`	0 = Not asymmetric pause capable 1 = Asymmetric pause toward link partner capability
`lp_cap_pause`	0 = Not symmetric pause capable 1 = Symmetric pause capable

If the link partner is *not* capable of autonegotiation (when `lp_cap_autoneg` is 0), the remaining information described in TABLE 4-12 is not relevant and the parameter value = 0.

If the link partner *is* capable of autonegotiation (when `lp_cap_autoneg` is 1), then the speed and mode information is displayed when you use autonegotiation and the link partner capabilities.

TABLE 4-13 describes the `netstat -k` transmit and receive parameters:

TABLE 4-13 Transmit and Receive Parameters

Parameter	Description
`xcvr_inits`	Number of physical layer re-initializations every time you change link parameters using NDD this increments.
`rev_id`	Revision ID of the GigaSwift Ethernet device. Useful for recognizing the device being used in the field.
`xcvr_addr`	GMII/MII physical layer device address for management interface.
`xcvr_id`	GMII/MII physical layer device Identification Decimal copy of MII registers 2 and 3.
`lb_mode`	Copy of the Loopback mode the device is in, if any.
`qos_mode`	When zero, the TX queues operate in a simple round robin queueing scheme, based on TCP/UDP destination port number. If set, the TX queues operate in a scheme designed to provide VLAN priorities.
`tx_starts`	Number of times that the driver attempted to transmit a packet.
`tx_dma_bind_fail`	Number of times a page table entry was not available to allow the driver to map the kernel memory to device accessible memory for transmission.
`tx_queue0`	Number of packets queued for transmission on the first hardware transmit queue.

TABLE 4-13 Transmit and Receive Parameters

Parameter	Description
tx_queue1	Number of packets queued for transmission on the second hardware transmit queue.
tx_queue2	Number of packets queued for transmission on the third hardware transmit queue.
tx_queue3	Number of packets queued for transmission on the fourth hardware transmit queue.
tx_max_pend	Maximum number of transmits pending on any of the four queues.
tx_jumbo_pkts	Number of packets transmitted that were greater than 1522 bytes.
rx_jumbo_pkts	Number of packets received that were greater than 1522 bytes.
rx_hdr_pkts	Number of packets received that were less than 256 bytes.
rx_mtu_pkts	Number of packets received that were greater than 256 bytes and less than 1514 bytes.
rx_split_pkts	Number of packets that were split across two pages.
rx_no_comp_wb	Number of times the hardware cannot post completion entries for received data.
rx_no_buf	Number of times the hardware cannot receive data because there is no more receive buffer space.
rx_new_pages	Number of pages that were replaced during reception.
rx_new_hdr_pgs	Number of pages filled with packets less than 256 bytes that were replaced during reception.
rx_new_mtu_pgs	Number of pages filled with packets greater than 256 bytes and less than 1514 that were replaced during reception.
rx_new_nxt_pgs	Number of pages containing packets split across pages that qwew replaced during reception.
rx_hdr_drops	Number of times an entire page of packets less than 256 bytes was dropped because the driver was unable to map a replacement page.
rx_mtu_drops	Number of times an entire page of packets greater than 256 bytes and less than 1514 was dropped because the driver was unable to map a replacement page.
rx_nxt_drops	Number of times a page with a split packet was dropped because the driver was unable to map a replacement page.
rx_rel_flow	Number of times the driver released a flow.

To Check Link Partner Settings

- **If you are running the Solaris 7 or 8 operating evironment, type the** `netstat -k` **command as superuser:**

```
# netstat -k ce:0
ce0:
ipackets 0 ipackets64 0 ierrors 0 opackets 0 opackets64 0
oerrors 0 collisions 0 rbytes 0 rbytes64 0 obytes 0 obytes64 0
multircv 0 multixmt 0 brdcstrcv 0 brdcstxmt 0 norcvbuf 0
noxmtbuf 0 first_collision 0 excessive_collisions 0 late_collisions 0
peak_attempts 0 length_err 0 alignment_err 0 crc_err 0 code_violations 0
ifspeed 1000000000 promisc off rev_id 48 xcvr_inits 1 xcvr_inuse 1
xcvr_addr 1 xcvr_id 536894584 cap_autoneg 1 cap_1000fdx 1 cap_1000hdx 1
cap_100T4 0 cap_100fdx 1 cap_100hdx 1 cap_10fdx 1 cap_10hdx 1
cap_asmpause 0 cap_pause 0 lp_cap_autoneg 1 lp_cap_1000fdx 1
lp_cap_1000hdx 1 lp_cap_100T4 0 lp_cap_100fdx 1 lp_cap_100hdx 1
lp_cap_10fdx 1 lp_cap_10hdx 1 lp_cap_asmpause 0 lp_cap_pause 1
link_T4 0 link_speed 1000 link_duplex 2 link_asmpause 0 link_pause 0
link_up 1 mac_mtu 0 lb_mode 0 qos_mode 0 trunk_mode 0 tx_inits 0
tx_starts 0 tx_nocanput 0 tx_msgdup_fail 0 tx_allocb_fail 0
tx_no_desc 0 tx_dma_bind_fail 0 tx_uflo 0 tx_queue0 0 tx_queue1 0
tx_queue2 0 tx_queue3 0 tx_hdr_pkts 0 tx_ddi_pkts 0 tx_dvma_pkts 0
mdt_reqs 0 mdt_hdr_bufs 0 mdt_pld_bufs 0 mdt_pkts 0 mdt_hdrs 0
mdt_plds 0 mdt_hdr_bind_fail 0 mdt_pld_bind_fail 0 tx_jumbo_pkts 0
tx_max_pend 0 rx_inits 0 rx_jumbo_pkts 0 rx_hdr_pkts 0 rx_mtu_pkts 0
rx_split_pkts 0 rx_no_buf 0 rx_no_comp_wb 0 rx_ov_flow 0
rx_len_mm 0 rx_tag_err 0 rx_nocanput 0 rx_msgdup_fail 0
rx_allocb_fail 0 rx_new_pages 0 rx_new_hdr_pgs 0 rx_new_mtu_pgs 0
rx_new_nxt_pgs 0 rx_hdr_drops 0 rx_mtu_drops 0 rx_nxt_drops 0
rx_rel_flow 0 rx_rel_bit 0 rx_pkts_dropped 0 pci_bus_speed 66
pci_err 0 pci_rta_err 0 pci_rma_err 0 pci_parity_err 0 pci_bad_ack_err 0
pci_drto_err 0 pci_dmawz_err 0 pci_dmarz_err 0 rx_taskq_waits 0
ipackets_cpu00 0 ipackets_cpu01 0 ipackets_cpu02 0 ipackets_cpu03 0
```

- **If you are running the Solaris 9 operating environment, type the** `kstat` **command as superuser:**

CODE EXAMPLE 4-1 Output from the `kstat` Command

```
# kstat ce:0
module: ce                                  instance: 0
name:   ce0                         class:   net
        alignment_err                       0
        brdcstrcv                           0
        brdcstxmt                           0
        cap_1000fdx                         1
```

CODE EXAMPLE 4-1 Output from the `kstat` Command *(Continued)*

```
# kstat ce:0
        cap_1000hdx                     1
        cap_100T4                       0
        cap_100fdx                      1
        cap_100hdx                      1
        cap_10fdx                       1
        cap_10hdx                       1
        cap_asmpause                    0
        cap_autoneg                     1
        cap_pause                       0
        code_violations                 0
        collisions                      0
        crc_err                         0
        crtime                          2345.407585961
        excessive_collisions            0
        first_collision                 0
        ierrors                         0
        ifspeed                         0
        ipackets                        0
        ipackets64                      0
        ipackets_cpu00                  0
        ipackets_cpu01                  0
        ipackets_cpu02                  0
        ipackets_cpu03                  0
        late_collisions                 0
        lb_mode                         0
        length_err                      0
        link_T4                         0
        link_asmpause                   0
        link_duplex                     0
        link_pause                      0
        link_speed                      0
        link_up                         0
        lp_cap_1000fdx                  0
        lp_cap_1000hdx                  0
        lp_cap_100T4                    0
        lp_cap_100fdx                   0
        lp_cap_100hdx                   0
        lp_cap_10fdx                    0
        lp_cap_10hdx                    0
        lp_cap_asmpause                 0
        lp_cap_autoneg                  0
        lp_cap_pause                    0
        multircv                        0
        multixmt                        0
        norcvbuf                        0
        noxmtbuf                        0
```

CODE EXAMPLE 4-1 Output from the `kstat` Command *(Continued)*

```
# kstat ce:0
        obytes                          0
        obytes64                        0
        oerrors                         0
        opackets                        0
        opackets64                      0
        pci_bad_ack_err                 0
        pci_dmarz_err                   0
        pci_dmawz_err                   0
        pci_drto_err                    0
        pci_err                         0
        pci_parity_err                  0
        pci_rma_err                     0
        pci_rta_err                     0
        peak_attempts                   0
        promisc                         off
        qos_mode                        0
        rbytes                          0
        rbytes64                        0
        rev_id                          1
        rx_allocb_fail                  0
        rx_hdr_drops                    0
        rx_hdr_pkts                     0
        rx_inits                        0
        rx_len_mm                       0
        rx_msgdup_fail                  0
        rx_mtu_drops                    0
        rx_mtu_pkts                     0
        rx_new_hdr_pgs                  0
        rx_new_mtu_pgs                  0
        rx_new_nxt_pgs                  0
        rx_new_pages                    0
        rx_no_buf                       0
        rx_no_comp_wb                   0
        rx_nocanput                     0
        rx_nxt_drops                    0
        rx_ov_flow                      0
        rx_pkts_dropped                 0
        rx_rel_bit                      0
        rx_rel_flow                     0
        rx_split_pkts                   0
        rx_tag_err                      0
        rx_taskq_waits                  0
        snaptime                        2352.506837979
        tx_allocb_fail                  0
        tx_ddi_pkts                     0
        tx_dma_bind_fail                0
```

CODE EXAMPLE 4-1 Output from the `kstat` Command *(Continued)*

```
# kstat ce:0
        tx_dma_hdr_bind_fail            0
        tx_dma_pld_bind_fail            0
        tx_dvma_pkts                    0
        tx_hdr_pkts                     0
        tx_inits                        0
        tx_max_desc                     0
        tx_max_pend                     0
        tx_msgdup_fail                  0
        tx_no_desc                      0
        tx_nocanput                     0
        tx_queue0                       3
        tx_queue1                       0
        tx_queue2                       0
        tx_queue3                       0
        tx_starts                       0
        tx_uflo                         0
        xcvr_addr                       1
        xcvr_id                         2121809
        xcvr_inits                      1
        xcvr_inuse                      1
```

▼ To Discover Link Partner Capabilities

- **Use the** `kstat` **command to discover link partner capabilities.**

```
# kstat ce:0 | grep lp_
         lp_cap_1000fdx                  1
         lp_cap_1000hdx                  1
        lp_cap_100T4                    0
        lp_cap_100fdx                   0
        lp_cap_100hdx                   0
        lp_cap_10fdx                    0
        lp_cap_10hdx                    0
        lp_cap_asmpause                 0
        lp_cap_autoneg                  1
        lp_cap_pause                    0
```

To Discover Link Settings

- **Use the** `kstat` **command to discover link settings.**

```
# kstat ce:0 | grep link
            link_T4                         0
            link_asmpause                   0
            link_duplex                     2
            link_pause                      0
            link_speed                      1000
            link_up                         1
```

Configuration Checking

The jumbo frame configuration checking occurs at Layer 2 and Layer 3, depending on the configuration methods.

Layer 2 Configuration Checking

You can view the MTU configuration of a `ce` instance at any time using the `kstat` command. The `kstat` `mac_mtu` variable represents the complete size of the Ethernet frame, which includes the Ethernet header, maximum payload, and `crc`. This value is consistant with the definition of MTU presented by switch vendors.

Use the following command to get the information:.

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

Layer 3 Configuration Checking

You can do this by invoking the `ifconfig` command on a preconfigured interface as shown in the above examples:.

```
ifconfig ce123000
ce123000: flags=1000843<.....> mtu 9194
inet ip_address-vlan-1 .......
ether 8:0:20:c4:51:df
```

APPENDIX **A**

Specifications

This appendix provides various specifications for the Sun Quad GigaSwift Ethernet adapter.

- "Performance Specifications" on page 63
- "Physical Characteristics" on page 64
- "Power Requirements" on page 64
- "Performance Specifications" on page 65
- "Interface Signals" on page 66

Performance Specifications

TABLE A-1 Performance Specifications

Feature	**Specification**
PCI clock	33/66 MHz max
PCI data burst transfer rate	Up to 64-byte bursts
PCI data/address width	32/64 -bit
PCI modes	Master/slave
Ethernet TX rate	10/100/1000 Mbps (full-duplex)

Physical Characteristics

TABLE A-2 Physical Characteristics

Dimension	Measurement
Length	12.283 inches (31.2 centimeters))
Width	4.2 inches (10.668 centimeters)

Power Requirements

TABLE A-3 Power Requirements

Specification	Measurement
Maximum power consumption	21.32 Watts
Voltage	3.3V and 5V

Performance Specifications

TABLE A-4 Performance Specifications

Specification	Performance
Maximum Ethernet transfer rate	10/100/1000 Mbps per channel.
Host interface	304-pin PBGA PCI to PCI bridge with a 64-bit primary and 64-bit secondary interface. Operates transparently in either 64-bit or 32-bit slots by means of a PCI-specific Universal 64-bit Card Edge. Supports both 3.3v and 5v signaling environments.
Network interface	1000BASE-T using Category 5E cable. 100BASE-TX using Category (data-grade) cable. 10BASE-T using Category 3 (voice-grade) cable or better. Single UTP RJ-45 for both 10 and 100 Mbps per channel.
Ethernet version	Conforms to IEEE 802.3x.
PCI clock	66 MHz max.
PCI data/address Lines	AD63-0.
PCI modes	Master/slave.

Interface Signals

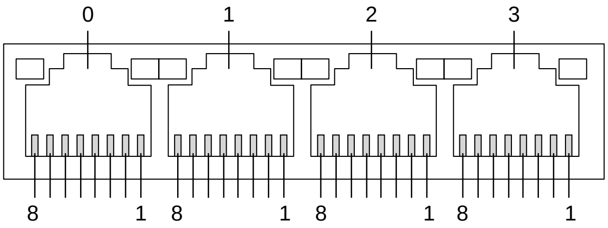


FIGURE A-1 32-Pin RJ-45 Connectors

TABLE A-5 Pin Descriptions for the 32-Pin RJ-45 Connectors—Channels 0 - 3

Description	CH0	CH1	CH2	CH3
TRD0+	1	1	1	1
TRD0-	2	2	2	2
TRD1+	3	3	3	3
TRD2+	4	4	4	4
TRD2-	5	5	5	5
TRD1-	6	6	6	6
TRD3+	7	7	7	7
TRD3-	8	8	8	8

APPENDIX **B**

Installing the Software Manually

This appendix describes how to install the driver software manually. It contains the following sections:

- "Installing the Driver Software Manually" on page 67
- "Verifying Patches" on page 71

Installing the Driver Software Manually

The *Sun Quad GigaSwift Ethernet Driver* CD contains the driver software required to operate the adapter. Refer to the documentation that shipped with the *Solaris Supplement* CD-ROM for a listing of the available network drivers. Install the Sun Quad GigaSwift Ethernet driver for your version of the Solaris operating environment.

1. **Become superuser.**
2. **Insert the** *Sun Quad GigaSwift Ethernet Driver* **CD into a CD-ROM drive that is connected to your system.**
   - If your system is running Sun Enterprise Volume Manager™, it should automatically mount the CD-ROM to the `/cdrom/cdrom0` directory.

- If your system is not running Volume Manager, mount the CD-ROM as follows:

```
# mkdir /cdrom
# mkdir /cdrom/cdrom0
# mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom/cdrom0
```

You will see the following files and directories in the `/cdrom/sun_gigaswift_ethernet_driver` directory or the `/cdrom/cdrom0` directory, depending on how you mounted the CD-ROM.

TABLE 4-14 Files and Directories on the CD-ROM

File or Directory	Contents
`Copyright`	U.S. copyright file
`FR_Copyright`	French copyright file
`QuadGigaSwiftEthernet/Solaris_`*OS-VER/* Packages/	Contains the Sun Quad GigaSwift Ethernet software packages applicable to your version (*OS-VER)* of the Solaris software: • SUNWcea—32-bit adb macros • SUNWceax—64-bit adb macros • SUNWced.u—32-bit adapter driver • SUNWcedx.u—64-bit adapter driver • SUNWcedu—adapter driver headers • SUNWcem—man pages (optional, Solaris 8 only) The optional VLAN packages for Solaris 8 only: • SUNWvld—VLAN utility routines • SUNWvldx—VLAN utility routines (64-bit) • SUNWvldu—VLAN utility headers
QuadGigaSwiftEthernet/ Docs/en	Contains PDF copy of the user manual

Note – If you intend to use VLAN, you must install VLAN packages when you install the software packages.

3. . Determine if the `ce` and `vlan` drivers are installed on the system:

```
#/usr/bin/pkginfo SUNWced SUNWcedx SUNWvld SUNWvldx
system      SUNWced        Sun GigaSwift Ethernet Adapter (32-bit Driver)
system      SUNWcedx       Sun GigaSwift Ethernet Adapter (64-bit Driver)
system      SUNWvld        Sun Ethernet Vlan Utility Routines
system      SUNWvldx       Sun Ethernet Vlan Utility Routines (64-bit)
```

- If the output of the command is similar to the example in Step 3, skip to Step 6.
- If nothing is reported back on SUNWced or SUNWcedx, then follow Step 4 and select SUNWcea, SUNWceax, SUNWced.u, SUNWcedx.u, SUNWcedu, and SUNWcem (if present).
- If nothing is reported back for SUNWvld or SUNWvldx, then follow Step 4 and select SUNWvld, SUNWvldx, SUNWvldu packages.
- If nothing is reported back, the follow Step 4 and select "all" packages.

4. **Install the software packages by typing the following at the command line, replacing the *OS_VER* with your version (7, 8, or 9) of the Solaris Operating Environment:**

```
# /usr/sbin/pkgadd -d /cdrom/cdrom0/QuadGigaSwiftEthernet/Solaris_OS_VER/
Packages/
```

For example, if your system is running the Solaris 8 Operating Environment, you would type the following:

```
# /usr/sbin/pkgadd -d /cdrom/cdrom0/QuadGigaSwiftEthernet/Solaris_8/Packages/
```

A menu similar to the following displays:

```
The following packages are available:
  1  SUNWcea       Sun Quad GigaSwift Ethernet Adapter Driver 32 bit adb Macros
                    (sparc) 1.0,REV=2001.05.04
  2  SUNWceax      Sun Quad GigaSwift Ethernet Adapter Driver 64 bit adb Macros
                    (sparc) 1.0,REV=2001.05.04
  3  SUNWced.u      Sun Quad GigaSwift Ethernet Adapter (32-bit Driver)
                    (sparc.sun4u) 1.0,REV=2001.05.04
  4  SUNWcedu       Sun Quad GigaSwift Ethernet Adapter Driver Headers
                    (sparc) 1.0,REV=2001.05.04
  5  SUNWcedx.u     Sun Quad GigaSwift Ethernet Adapter (64-bit Driver)
                    (sparc.sun4u) 1.0,REV=2001.05.04
  6  SUNWcem        Sun Quad GigaSwift Ethernet Adapter Driver Man Pages
                    (sparc) 1.0,REV=2001.05.04
  7  SUNWvld        Sun Ethernet Vlan Utility Routines
                    (sparc) 1.0,REV=2001.05.04
  8  SUNWvldu       Sun Ethernet Vlan Utility Headers
                    (sparc) 1.0,REV=2001.05.04
  9  SUNWvldx       Sun Ethernet Vlan Utility Routines (64-bit)
                    (sparc) 1.0,REV=2001.05.04

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

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

   ■ Press Return or type **all** to accept the default and install all packages (if neither the QuadGigaSwfit Ethernet Driver nor VLAN Utility packages are installed).

   ■ Type the numbers of all but the optional packages if you prefer not to install them.

   See TABLE 4-14 to identify the optional packages.

6. **If the Packages are already installed, verify the patch version.**

   See the following section.

Verifying Patches

Verify whether your system has the needed patches for your version of the Solaris Operating Environment to ensure that you do not need to add patches.

Note – If patches meet the requirements, nothing needs to be installed on the install system.

TABLE B-1 lists and describes the patch versions:

TABLE B-1 Patch Versions

Solaris	Patch Version	Description
7	112327-17	Sun GigaSwift Ethernet 1.0: driver patch
8	111883-23	SunOS 5.8: Sun GigaSwift Ethernet 1.0 driver patch
	112119-04	SunOS 5.8: vlan driver patch*
9	112817-16	SunOS 5.9: Sun GigaSwift Ethernet 1.0 driver patch
	114600-02	SunOS 5.9: vlan driver patch*

* Only required if packages SUNWvld or SUNWvldx is installed.

▼ To Install Patches

1. **Determine the version of the Solaris operating environment your system is running:**

```
% cat /etc/release
Solaris 9 8/03 s9s_u4wos_08a SPARC
Copyright 2003 Sun Microsystems, Inc.  All Rights Reserved.
                        Use is subject to license terms.
                             Assembled 13 June 2003
```

2. Verify patch versions for the GigaSwift Etnernet driver:

- To verify patches for Solaris 7 Operating Environments, type the following:

```
# showrev -p | grep 112327
```

The patch version should be -17 or greater.

- To verify patches for Solaris 8 Operating Environments, type the following:

```
# showrev -p | grep 111883
```

The patch version should be -23 or greater.

- To verify patches for Solaris 9 Operating Environments, type the following:

```
# showrev -p | grep 112817
```

The patch version should be -16 or greater.

3. **If the patch version is not correct, install the correct patch:**

```
# cd /cdrom/cdrom0
# patchadd QuadGigaSwiftEthernet/OS/Patches/patch_id
```

Where:

OS = Solaris_7, Solaris_8, or Solaris_9

patch_id = 112327-17 for Solaris 7, 111883-23 for Solaris 8, or 112817-16 for Solaris 9.

For example, using the information shown in Step 1, you would type the following:

```
# /usr/sbin/patchadd QuadGigaSwiftEthernet/Solaris_9/Patches/112817-16
Checking installed patches...
Executing prepatch script...


Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...

Patch number 112817-16 has been successfully installed.
See /var/sadm/patch/112817-16/log for details

Patch packages installed:
SUNWcea
SUNWceax
SUNWced
SUNWcedu
SUNWcedx
```

▼ To Install a Sun VLAN Utility Patch

Note – The VLAN Utility is only supported in Solaris 8 and Solaris 9 software.

---

1. **Determine the version of the Solaris operating environment your system is running:**

```
% cat /etc/release
Solaris 9 8/03 s9s_u4wos_08a SPARC
Copyright 2003 Sun Microsystems, Inc.  All Rights Reserved.
                        Use is subject to license terms.
                             Assembled 13 June 2003
```

2. **Verify patch versions for the VLAN utility software:**
   - To verify patches for Solaris 8 Operating Environments, type the following:

     ```
     # showrev -p | grep 112119
     ```

     The patch version should be -04 or greater.
   - To verify patches for Solaris 9 Operating Environments, type the following:

     ```
     # showrev -p | grep 114600
     ```

   Verify that the patch version is -02 or higher.

3. **If the patch versions are not correct, install the correct patch:**

   ```
   # cd /cdrom/cdrom0
   # patchadd QuadGigaSwiftEthernet/OS/Patches/patch_id
   ```

   Where:

   OS = Solaris_8 or Solaris_9

   patch_id = 112119-04 for Solaris 8, or 114600-02 for Solaris 9.

   For example, using the information shown in Step 1, you would type the following:

```
# /usr/sbin/patchadd QuadGigaSwiftEthernet/Solaris_9/Patches/114600-02
Checking installed patches...
Executing prepatch script...


Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...

Patch number 114600-02 has been successfully installed.
See /var/sadm/patch/114600-02/log for details

Patch packages installed:
SUNWvld
SUNWvldu
SUNWvldx
```

APPENDIX **C**

Testing the Adapter

This appendix describes how to test the adapter using either the SunVTS diagnostic application or the onboard FCode self-test.

This appendix contains the following sections:

- "Using the SunVTS Diagnostic Software" on page 75
- "Using the OpenBoot PROM FCode Self-Test" on page 77

Using the SunVTS Diagnostic Software

The SunVTS software executes multiple diagnostic hardware tests from a single user interface. It is used to verify the configuration and functionality of most hardware controllers and devices. The SunVTS software operates primarily from a user interface that enables you to set test parameters while diagnostic operations are being performed.

The SunVTS `nettest` diagnostic can be used to test all of the networking interfaces on the system, including the interfaces on the adapter.

To use the `nettest` diagnostic, you must have the SunVTS software installed on your system. Refer to the *Solaris on Sun Hardware Platform Guide* for SunVTS installation instructions.

Refer to the SunVTS documentation (listed in TABLE C-1) for instructions on how to run and monitor the `nettest` diagnostic. These SunVTS documents are available on the *Solaris on Sun Hardware Collection*, which is provided on the Solaris Supplement CD for the Solaris release on your system.

TABLE C-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`) and describes the various test options and command-line arguments
SunVTS Quick Reference	Provides an overview of the user interface

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 bring the Ethernet device down before running `netlbtest`.

▼ To Use the `netlbtest`

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

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

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

```
# ifconfig ceinstance down
# ifconfig ceinstance unplumb
```

Where *instance* is the instance numbr of the interface.

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

Using the OpenBoot PROM FCode Self-Test

The Sun Quad GigabitEthernet adapter’s self-test verifies the correct operation of the adapter and identifies any hardware problems with the adapter. The self-test is a suite of tests that reside in the adapter’s FCode PROM.

You can only invoke the FCode self-test diagnostics using the OpenBoot PROM `test` or `test-all` commands. In order to access the OpenBoot PROM, you must shut down and halt your system to display the `ok` prompt.

You can use the adapter’s self-test any time you want to determine the status of the hardware. If you encounter an error while running the diagnostic test, the self-test displays appropriate error messages.

Note – Although the self-test does not require the adapter’s interface to be connected to an active network, you will see failing error messages if it is not connected.

▼ To Run the FCode Self-Test Diagnostic

For more information about the OpenBoot commands (including `test`) in this section, refer to the *OpenBoot Command Reference* manual.

1. **Before shutting down the operating environment and halting the system, ensure that all significant application and network activity on the server has stopped.**

2. **Follow the appropriate procedures, as documented in the system’s service manual, to shut down and halt the system.**

   Refer to the system’s documentation for the complete shutdown procedure.

   After shutting down the system, you should see the OpenBoot PROMs `ok` prompt on the system console.

3. **At the `ok` prompt, set the `auto-boot?` configuration variable to `false`.**

```
ok setenv auto-boot? false
```

4. **Set the** `diag-switch?` **parameter to** `true` **in order to see the diagnostic power-on self-test (POST) messages.**

```
ok setenv diag-switch? true
```

5. **Reset the system to activate these parameter changes.**

```
ok reset-all
```

6. **Use the** `show-nets` **command to display the list of network devices on the system.**

   Among the list of devices, you should see the four interfaces specific to the Sun Quad GigabitEthernet adapter you want to test, as shown in the following example.

```
ok show-nets
a) /pci@9,600000/pci@1/pci@4/network@3
b) /pci@9,600000/pci@1/pci@4/network@2
c) /pci@9,600000/pci@1/pci@0/network@1
d) /pci@9,600000/pci@1/pci@0/network@0
e) /pci@9,700000/network@1,1
f) /pci@8,600000/network@1
q) NO SELECTION
Enter Selection, q to quit: q
```

Note – The preceding device paths are examples. The actual device paths on your system may be different from the examples shown here.

---

7. **Use the** `test` **command, and the device paths found in Step 6, to test the adapter when you use the FCode self-test.**

```
ok test device_path
```

   Refer to the appropriate *OpenBoot Command Reference Manual* for more information on the `test` command.

   The following tests are run when you execute the `test` command:

- MAC internal loopback
- 1000 Mbps PHY loopback
- 100 Mbps PHY loopback
- 10 Mbps PHY loopback

Note – To test all of the interfaces of the adapter, you need to execute the `test` command separately for each `SUNW,qge` interface.

If all of the tests pass, you see these messages:

```
ok test /pci@9,600000/pci@1/pci@0/network@0
Testing /pci@9,600000/pci@1/pci@0/network@0
Register tests: passed
Internal loopback test: passed
/pci@9,600000/pci@1/pci@0/network@0: 1000 Mbps full duplex link up
ok
```

If the adapter's interface is not connected to a network, you see the following error messages:

```
ok test /pci@9,600000/pci@1/pci@0/network@0
Testing /pci@9,600000/pci@1/pci@0/network@0
Register tests: passed
Internal loopback test: passed
/pci@9,600000/pci@1/pci@0/network@0: Timed out waiting for Auto-
Negotation to complete
/pci@9,600000/pci@1/pci@0/network@0: Cannot establish link via
Auto-Negotation
Please check cable and/or connection
/pci@9,600000/pci@1/pci@0/network@0: link down
ok
```

Note – If the `test` command fails, verify that the adapter hardware is installed correctly and the Ethernet device is connected to an active network. If necessary, replace the adapter or contact your service provider for support.

8. **After testing the adapter, type the following commands to return the OpenBoot PROM to the standard operating mode.**

```
ok setenv diag-switch? false
ok test /pci@9,600000/pci@1/pci@0/network@0
Testing /pci@9,600000/pci@1/pci@0/network@0
Register tests: passed
Internal loopback test: passed
/pci@9,600000/pci@1/pci@0/network@0: link down
```

9. **Reset and reboot the system.**

```
ok reset-all
```

Refer to the system’s documentation for the correct procedures for bringing up the system.

Index

SYMBOLS

.properties file, 9

A

auto-negotiation, 31

autonegotiation
- setting
  - disabling, 49

C

ce driver parameter, status, and descriptions, 39

D

device properties, 9

diagnostics, 75 to 80

documentation
- SunVTS, 76

driver parameters, 39

Dynamic Reconfiguration, 6

dynamic reconfiguration, 5

Dynamic Reconfiguration (DR), 12

E

editing
- `hostname.qfe#` files, 24
- `hosts` file, 24

Ethernet channels, 25, 66

F

files and directories on the CD-ROM, 68

flow control keywords, 41

full-duplex operation mode, 31

H

half-duplex operation mode, 31

hardware
- features, 3
- local transceiver, 38
- product overview, 1
- requirements, 2

`hostname.qfe#` files
- creating, 25
- editing, 24
- examples, 26

`hosts` file
- editing, 24
- example, 26

I

`ifconfig` command, 24, 31, 32

installing the adapter with DR, 12

installing the adapter without DR, 6

instance numbers
  with hostname.qfe# files, 25

interface signals, 66

internal transceiver, 38

IPv6 protocol, 26

L

LEDs, 3

link partner settings
  checking, 58

local transceiver, 38

local-mac-address property, 10

local-mac-address? variable, 32

M

MAC address, 10

MAC address, using the adapter’s address, 31

mac-address property, 10

Media Access Control (MAC), 10

N

ndd utility, 47

NVRAM configuration, 10

O

OpenBoot PROM
  FCode self-test, 77
  local-mac-address? variable, setting, 31
  ok prompt, 34, 77
  test command, 78
  viewing device path names, 25

operation mode
  full-duplex, 31
  half-duplex, 31

operational mode parameters, 40

P

parameter values
  how to modify and display, 48

PCI connector, 2

performance specifications, 65

physical dimensions, 64

pin descriptions, 66

product overview, 1

Q

qfe driver
  attaching qfe interfaces, 24

R

random early detection register, 44

receive interrupt blanking values, 39, 44

requirements
  hardware, 2
  software, 2

RJ-45 connectors, 38, 66

S

self-test diagnostics, 77

setting ce driver parameters
  using ndd
    using ce.conf, 46

setting qfe driver parameters, 37

show-nets command output, 78

software
  attaching qfe devices, 24
  auto-negotiation, 31
  plumbing interfaces, 24
  qfe driver parameters, 39
  requirements, 2
  unplumbing interfaces, 32

specifications
  performance, 65

SunVTS
  diagnostics, 75
  documentation, 76

SUNW,qfe devices, 37, 78

T

`test` diagnostic command, 77

testing, 75 to 80

to set, 10

U

unplumbing interfaces, 32

V

verify the installation, 7