Platform Notes: The Sun™ Quad FastEthernet™ Device Driver

Solaris™ 2.6 Hardware: 5/98

Send comments about this document to: smcc-docs@sun.com

Please
Recycle

Adobe PostScript™

Contents

Preface

This book describes how to configure the `qfe` driver and driver parameters for the Sun™ Quad FastEthernet™ PCI adapter and the Sun™ Quad FastEthernet™ SBus adapter.

How This Book Is Organized

Chapter 1 describes configuring the driver for the Sun Quad FastEthernet PCI adapter and the Sun Quad FastEthernet SBus adapter.

Chapter 2 describes setting the driver parameters for the Sun Quad FastEthernet PCI adapter and the Sun Quad FastEthernet SBus adapter.

Typographic Conventions

TABLE P-1 Typographic Conventions

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

Shell Prompts

TABLE P-2 Shell Prompts

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

Ordering Sun Documents

SunDocs[SM] is a distribution program for Sun Microsystems technical documentation. Contact SunExpress[TM] for easy ordering and quick delivery. You can find a listing of

available Sun documentation on the Web.

TABLE P-3   SunExpress Contact Information

Country	Telephone	Fax
Belgium	02-720-09-09	02-725-88-50
Canada	1-800-873-7869	1-800-944-0661
France	0800-90-61-57	0800-90-61-58
Germany	01-30-81-61-91	01-30-81-61-92
Holland	06-022-34-45	06-022-34-46
Japan	0120-33-9096	0120-33-9097
Luxembourg	32-2-720-09-09	32-2-725-88-50
Sweden	020-79-57-26	020-79-57-27
Switzerland	0800-55-19-26	0800-55-19-27
United Kingdom	0800-89-88-88	0800-89-88-87
United States	1-800-873-7869	1-800-944-0661
World Wide Web: `http://www.sun.com/sunexpress/`		

Sun Documentation on the Web

The `docs.sun.com` web site enables you to access Sun technical documentation on the Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at:

```
http://docs.sun.com
```

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

```
smcc-docs@sun.com
```

Please include the part number of your document in the subject line of your email.

Configuring the Driver Software for Sun Quad FastEthernet Adapters

This chapter includes information and instructions for configuring the driver software used by the Sun Quad FastEthernet adapter. Unless otherwise noted, all instructions apply to both the Sun Quad FastEthernet PCI adapter and the Sun Quad FastEthernet SBus adapter.

Installing the Driver Software

The Solaris CD-ROM contains the software that must be installed in order to use the Sun Quad FastEthernet adapter.

Note – Do not use the installation CD-ROM that shipped with your Quad FastEthernet adapter. The software on the Solaris CD-ROM is more current and replaces previous versions of the driver.

Before using the four network interfaces of the adapter, you will need to create and edit system host files, as described in the next section.

Configuring the Host File

After installing the Sun Quad FastEthernet driver software, you must create a `hostname.qfe`*num* file for the Ethernet interfaces on the adapter. You must also create both an IP address and a host name for the Ethernet interfaces in the `/etc/hosts` file.

▼ To Configure the Host File

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

For a Sun Quad FastEthernet PCI adapter:

```
# grep qfe /etc/path_to_inst
"/pci@1f,2000/pci@2/SUNW,qfe@0,1" 0 "qfe"
"/pci@1f,2000/pci@2/SUNW,qfe@1,1" 1 "qfe"
"/pci@1f,2000/pci@2/SUNW,qfe@2,1" 2 "qfe"
"/pci@1f,2000/pci@2/SUNW,qfe@3,1" 3 "qfe"
```

In the example above, the four SUNW,qfe@*x*,1 instances are from a Sun Quad FastEthernet PCI adapter installed in slot 2. For clarity, the instance numbers are bold.

For a Sun Quad FastEthernet SBus adapter:

```
# grep qfe /etc/path_to_inst
"/sbus@1f,0/SUNW,qfe@1,8c10000" 1 "qfe"
"/sbus@1f,0/SUNW,qfe@1,8c00000" 0 "qfe"
"/sbus@1f,0/SUNW,qfe@1,8c30000" 3 "qfe"
"/sbus@1f,0/SUNW,qfe@1,8c20000" 2 "qfe"
```

In the example above, the four SUNW,qfe@1 instances are from a Sun Quad FastEthernet SBus adapter installed in slot 1. For clarity, the instance numbers are bold.

2. Create an `/etc/hostname.qfe`*num* **file, where** *num* **is the instance number of each interface you plan to use.**

If you want to use all of the network interfaces from the example in Step 1, you will need to create four files:

File Name	Instance Number	Adapter Ethernet Network Interface
/etc/hostname.qfe0	0	0

File Name	Instance Number	Adapter Ethernet Network Interface
/etc/hostname.qfe1	1	1
/etc/hostname.qfe2	2	2
/etc/hostname.qfe3	3	3

- Do not create /etc/hostname.qfe*num* files for Sun Quad FastEthernet adapter network interfaces you plan to leave unused.
- The /etc/hostname.qfe*num* file must contain the host name for the appropriate network interface.
- The host name should have an IP address that will need to be entered in the /etc/hosts file.
- The host name should be different from any other host name of any other interface, for example: /etc/hostname.hme0 and /etc/hostname.qfe2 cannot share the same host name.

Using the instance examples in Step 1, the following example shows the four /etc/hostname.qfe*num* files required for a system called zardoz that has a Sun Quad FastEthernet adapter (zardoz-11, zardoz-12, zardoz-13, and zardoz-14).

```
# cat /etc/hostname.hme0
zardoz
# cat /etc/hostname.qfe0
zardoz-11
# cat /etc/hostname.qfe1
zardoz-12
# cat /etc/hostname.qfe2
zardoz-13
# cat /etc/hostname.qfe3
zardoz-14
```

3. **Create an appropriate entry in the** /etc/hosts **file for each active** qfe **network interface.**

Using the previous example, you will have:

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1      localhost
129.144.10.57 zardoz     loghost
129.144.11.83 zardoz-11
129.144.12.41 zardoz-12
129.144.13.67 zardoz-13
129.144.14.30 zardoz-14
```

4. **Reboot your system.**

Booting From the Network

To use a Sun Quad FastEthernet adapter Ethernet interface as the boot device, perform the following tasks:

▼ To Boot From the Network

1. **At the** `ok` **prompt type:**

```
ok show-devs
```

The `show-devs` command lists the system devices. You should see the full path name of the `qfe` devices, similar to the following examples:

For Sun Quad FastEthernet PCI adapter:

```
/pci@1f,2000/pci@2/SUNW,qfe@0,1
/pci@1f,2000/pci@2/SUNW,qfe@1,1
/pci@1f,2000/pci@2/SUNW,qfe@2,1
/pci@1f,2000/pci@2/SUNW,qfe@3,1
```

For Sun Quad FastEthernet SBus adapter:

```
/sbus@1f,0/SUNW,qfe@1,8c30000
/sbus@1f,0/SUNW,qfe@1,8c20000
/sbus@1f,0/SUNW,qfe@1,8c10000
/sbus@1f,0/SUNW,qfe@1,8c00000
```

Note – You need to select only one of these `qfe` devices for booting.

2. **At the** `ok` **prompt type:**

```
ok boot full_path_name_of_the_qfe_device
```

Optional Post-Installation Procedures

To customize the performance of the Sun Quad FastEthernet adapter, perform the tasks in the following sections.

Setting Driver Parameters

The `qfe` device driver, which is loaded from the Solaris CD-ROM, controls the `SUNW,qfe` Ethernet devices. The device driver selects the link speed using the auto-negotiation protocol with the link partner. (See "Auto-Negotiation" on page 9.)

You can manually set the `qfe` device driver parameters to customize each `SUNW,qfe` device in your system in one of three ways:

- Set a parameter on a per-device basis by creating the `qfe.conf` file in the `/kernel/drv` directory.
- Use the `ndd` utility to *temporarily* change a parameter. This change is lost when you reboot the system.
- Set the `qfe` driver parameters generally for all `SUNW,qfe` devices in the system by entering the parameter variables in the `/etc/system` file.

See "Setting the qfe Driver Parameters" on page 17," for more information.

Note – In the future, the `/etc/system` file will not be available. It is not dynamic reconfiguration compatible.

▼ To Increase Performance on Solaris 2.6

If you are running your Solaris 2.6 system with a large TCP window and you experience slowdowns during high network traffic, you should decrease the TCP default maximum retransmission timeout interval variable (`tcp_rexmit_interval_max`) to **60000**.

- **As superuser type:**

```
# ndd -set /dev/tcp tcp_rexmit_interval_max 60000
```

The changes are immediate and affect all the networking interfaces in the system. However, the changes are lost when you reboot your machine.

▼ To Avoid Losing TCP/IP Parameter Changes

Add the parameter change to a run control script in the `/etc/rc2.d` directory, similar to the following example:

```
#!/sbin/sh
# Local kernel modifications
#
case "$1" in
'start')
    echo "Setting local kernel parameters...\c"
    ndd -set /dev/tcp tcp_rexmit_interval_max 60000
    echo ""
    ;;
'stop')
    echo "No kernel parameters changed."
    ;;
*)
    echo "Usage: $0 {start|stop}"
    ;;
esac
exit 0
```

▼ To Force Network Speed Between 10 Mbps and 100 Mbps

1. **At the** `ok` **prompt, use the** `show-devs` **command to list the system devices.**

   You should see the full path names of the `qfe` devices, similar to the following examples:

For Sun Quad FastEthernet PCI adapter:

```
/pci@1f,2000/pci@2/SUNW,qfe@0,1
/pci@1f,2000/pci@2/SUNW,qfe@1,1
/pci@1f,2000/pci@2/SUNW,qfe@2,1
/pci@1f,2000/pci@2/SUNW,qfe@3,1
```

For Sun Quad FastEthernet SBus adapter:

```
/sbus@1f,0/SUNW,qfe@1,8c30000
/sbus@1f,0/SUNW,qfe@1,8c20000
/sbus@1f,0/SUNW,qfe@1,8c10000
/sbus@1f,0/SUNW,qfe@1,8c00000
```

2. **Type:**

```
ok nvedit
```

3. **Type the following, pressing the Return key at the end of line** `0`**:**

```
0: probe-all install-console banner
1: apply transfer-speed=10 full_ path_name_of_a_qfe_device
```

Note – If you already have commands in NVRAM, append these lines to the end of the file.

4. **Press Control-C after typing** *full_ path_name_of_a_*`qfe` **device.**

   Perform Steps 2 to 4 to set the network speed for each `qfe` network interface.

Note – In the preceding example, the speed is forced to 10 Mbps. To force the speed to 100 Mbps, replace 10 with 100.

5. **At the** `ok` **prompt type:**

```
ok nvstore
ok setenv use-nvramrc? true
```

6. **Reboot your system.**

   See "Setting Forced Mode" on page 21 for more information on forcing network speed.

Auto-Negotiation

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

The link speed and modes supported by the Sun Quad FastEthernet adapter are listed as follows in decreasing order of priority:

- 100 Mbps, full-duplex
- 100 Mbps, half-duplex
- 10 Mbps, full-duplex
- 10 Mbps, half-duplex

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

If the Sun Quad FastEthernet 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 Sun Quad FastEthernet adapter is connected to a link partner with which the auto-negotiation protocol fails to operate successfully, you can configure the device to not use this protocol and force the driver to set up the link in the mode and speed of your choice.

`local-mac-address` Property

Each of the network interfaces of the Sun Quad FastEthernet 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 via the `local-mac-address` property in the device nodes corresponding to the network interfaces.

A system is not obligated to use this assigned MAC address if it has a systemwide MAC address. In such cases, the systemwide 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. 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 (systemwide or `local-mac-address`) used for booting the system. To start using the MAC addresses assigned to the network interfaces of the Sun Quad FastEthernet adapter, set the NVRAM configuration variable `local-mac-address?` to `true`.

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

Setting Parameters for the Sun Quad FastEthernet Device Driver

The `qfe` device driver controls the Sun Quad FastEthernet adapter's `SUNW,qfe` Ethernet devices. You can manually set the `qfe` device driver parameters to customize each `SUNW,qfe` device in your system. This chapter provides an overview of the local transceiver used in the adapter, lists the available `qfe` device driver parameters, and describes how you can set these parameters.

The Local Transceiver (Internal)

The four `SUNW,qfe` channels each provide 10BASE-T or 100BASE-TX networking interfaces, using the PCI FastEthernet and Expansion (PFEX) ASIC or the SBus FastEthernet Parallel SCSI (FEPS) ASIC and a local transceiver. The driver automatically sets the link speed to 10 or 100 Mbps and conforms to the 100BASE-T IEEE 802.3u Ethernet standard. The PFEX ASIC provides the PCI interface and Media Access Control (MAC) functions. The FEPS ASIC provides the SBus interface and 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 in "Auto-Negotiation" on page 9. The local transceiver performs auto-negotiation with the remote end of the link (link partner) to select a common mode of operation.

The local transceiver also supports a forced mode of operation. You can select the speed and mode using the `ndd` utility, by editing the `/etc/system` file, or creating a `qfe.conf` file in the `/kernel/drv` directory.

Driver Parameter Values and Definitions

This section describes the parameters and settings for the `qfe` device driver. TABLE 2-1 lists these parameters.

TABLE 2-1   `qfe` Driver Parameter, Status, and Descriptions

Parameter	Status	Description
transceiver_inuse	Read only	Defines the current status
link_status	Read only	Defines the current status
link_speed	Read only	Defines the current status
link_mode	Read only	Defines the current status
ipg1	Read and write	Inter-packet gap parameter
ipg2	Read and write	Inter-packet gap parameter
use_int_xcvr	Read and write	Operational mode parameter
pace_size	Read and write	Operational mode parameter
adv_autoneg_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
autoneg_cap	Read only	Local transceiver auto-negotiation capability
100fdx_cap	Read only	Local transceiver capability of the hardware
100hdx_cap	Read only	Local transceiver capability of the hardware
10fdx_cap	Read only	Local transceiver capability of the hardware
10hdx_cap	Read only	Local transceiver capability of the hardware
lp_autoneg_cap	Read only	Link partner auto-negotiation capability
lp_100fdx_cap	Read only	Link partner capability
lp_100hdx_cap	Read only	Link partner capability

TABLE 2-1 qfe Driver Parameter, Status, and Descriptions *(Continued)*

Parameter	Status	Description
lp_10fdx_cap	Read only	Link partner capability
lp_10hdx_cap	Read only	Link partner capability
instance	Read and write	Device instance
lance_mode	Read and write	Additional delay before transmitting a packet
ipg0	Read and write	Additional delay before transmitting a packet

Defining the Current Status

The read-only parameters described in TABLE 2-2 explain the operational mode of the interface. These parameters define the current status.

TABLE 2-2 Read-Only Parameters for Defining the Current Status

Parameter	Description	Values
link_status	Current link status	0 = Link down 1 = Link up
link_speed	Valid only if the link is up	0 = 10 Mbps 1 = 100 Mbps
link_mode	Valid only if the link is up	0 = Half duplex 1 = Full duplex

Inter-Packet Gap Parameters

The PCI FastEthernet and Expansion (PFEX) ASIC and the SBus FastEthernet Parallel SCSI (FEPS) ASIC support the programmable inter-packet gap (IPG) parameters ipg1 and ipg2. The total IPG is the sum of ipg1 and ipg2. The total default IPG is 9.6 microseconds when the link speed set, by the auto-negotiation protocol, is 10 Mbps. When the link speed is 100 Mbps, the total default IPG is 0.96 microseconds.

TABLE 2-3 lists the default values and allowable values for the IPG parameters, `ipg1` and `ipg2`.

TABLE 2-3 Read-Write Inter-Packet Gap Parameter Values and Descriptions

Parameter	Values (Byte Time)	Description
ipg1	0, 255	ipg1 = 8 (default at initialization)
ipg2	0, 255	ipg2 = 4 (default at initialization)

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 either 100 Mbps or 10 Mbps.)

If your network has systems that use a 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.

Defining an Additional Delay Before Transmitting a Packet Using `lance_mode` and `ipg0`

The PCI FastEthernet and Expansion (PFEX) ASIC and the SBus FastEthernet Parallel Port SCSI (FEPS) ASIC support a programmable mode called `lance_mode`. The `ipg0` parameter is associated with `lance_mode`.

After a packet is received with `lance_mode` enabled (default) an additional delay is added by setting the `ipg0` parameter 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 delay set by `ipg0` helps to reduce collisions. Systems that have `lance_mode` enabled might not have enough time on the network.

If `lance_mode` is disabled, the value of `ipg0` is ignored and no additional delay is set. Only the delays set by `ipg1` and `ipg2` are used. Disable `lance_mode` if other systems keep sending a large number of back-to-back packets.

You can add the additional delay by setting the `ipg0` parameter from 0 to 31, which is the nibble time delay. Note that nibble time is the time it takes to transfer four bits on the link. If the link speed is 10 Mbps, nibble time is equal to 400 ns. If the link speed is 100 Mbps, nibble time is equal to 40 ns.

For example, if the link speed is 10 Mbps, and you set `ipg0` to 20 nibble times, multiply 20 by 400 ns to get 8000 ns. If the link speed is 100 Mbps, and you set `ipg0` to 30 nibble times, multiply 30 by 40 ns to get 1200 ns.

TABLE 2-4 defines the `lance_mode` and `ipg0` parameters.

TABLE 2-4   Parameters Defining `lance_mode` and `ipg0`

Parameter	Values	Description
`lance_mode`	0 1	`lance_mode` disabled `lance_mode` enabled (default)
`ipg0`	0-31[1]	Additional IPG before transmitting a packet (after receiving a packet)

1. The default value is 16 nibble-times, which is 6.4 microseconds for 10 Mbps and 0.64 microseconds for 100 Mbps.

Operational Mode Parameters

TABLE 2-5 describes the operational mode parameters and their default values.

TABLE 2-5   Operational Mode Parameters

Parameter	Description	Values
`adv_autoneg_cap`	Local transceiver capability advertised by the hardware	0 = Forced mode 1 = Auto-negotiation (default)
`adv_100fdx_cap`[1]	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 100 Mbit/sec full-duplex capable (default in the Solaris 2.5 and 2.5.1 software environments) 1 = 100 Mbit/sec full-duplex capable (default in the Solaris 2.6 software environment)
`adv_100hdx_cap`[1]	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 100 Mbit/sec half-duplex capable 1 = 100 Mbit/sec half-duplex capable (default)
`adv_10fdx_cap`[1]	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 10 Mbit/sec full-duplex capable (default) 1 = 10 Mbit/sec full-duplex capable
`adv_10hdx_cap`[1]	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 10 Mbit/sec half-duplex capable 1 = 10 Mbit/sec half-duplex capable (default)

1. The priority (in descending order) for these parameters is: `adv_100fdx_cap`, `adv_100hdx_cap`, `adv_10fdx_cap`, `adv_10hdx_cap`.

Defining the Number of Back-to-Back Packets to Transmit

The `pace_size` parameter (see TABLE 2-6) defines the maximum number of back-to-back packets you can transmit at one time. If the value is zero, there is no limit to the number of back-to-back packets that can be transmitted.

TABLE 2-6   Back-to-back Packet Transmission Capability

Parameter	Values
`pace_size`	1 to 255 = Number of back-to-back packets transmitted at one time
	0 = No limit to the number of back-to-back packets that can be transmitted (default)

Reporting Transceiver Capabilities

TABLE 2-7 describes the read-only transceiver capabilities. These parameters define the capabilities of the hardware. The local transceiver can support all of these capabilities.

TABLE 2-7   Read-Only Transceiver Capabilities

Parameter	Description	Values
`autoneg_cap`	Local transceiver capability of the hardware	0 = Not capable of auto-negotiation 1 = Auto negotiation capable
`100fdx_cap`	Local transceiver capability of the hardware; initialized at startup	0 = Not 100 Mbit/sec full-duplex capable 1 = 100 Mbit/sec full-duplex capable
`100hdx_cap`	Local transceiver capability of the hardware; initialized at startup	0 = Not 100 Mbit/sec half-duplex capable 1 = 100 Mbit/sec half-duplex capable
`10fdx_cap`	Local transceiver capability of the hardware; initialized at startup	0 = Not 10 Mbit/sec full-duplex capable 1 = 10 Mbit/sec full-duplex capable
`10hdx_cap`	Local transceiver capability of the hardware; initialized at startup	0 = Not 10 Mbit/sec half-duplex capable 1 = 10 Mbit/sec half-duplex capable

Reporting the Link Partner Capabilities

TABLE 2-8 describes the read-only link partner capabilities.

TABLE 2-8 Read-Only Link Partner Capabilities

Parameter	Values
lp_autoneg_cap	0 = No auto-negotiation 1 = Auto-negotiation
lp_100fdx_cap	0 = No100 Mbit/sec full-duplex transmission 1 = 100Mbit/sec full-duplex
lp_100hdx_cap	0 = No 100 Mbit/sec half-duplex transmission 1 = 100 Mbit/sec half-duplex
lp_10fdx_cap	0 = No 10 Mbit/sec full-duplex transmission 1 = 10 Mbit/sec full-duplex
lp_10hdx_cap	0 = No 10 Mbit/sec half-duplex transmission 1 = 10 Mbit/sec half-duplex

If the link partner is not capable of auto-negotiation (when lp_autoneg_cap is 0) the information described in TABLE 2-8 is not relevant and the parameter value = 0.

If the link partner is capable of auto-negotiation (when lp_autoneg_cap is 1), the speed and mode information are displayed when you use auto-negotiation and get the link partner capabilities.

Setting the qfe Driver Parameters

You can set the qfe device driver parameters in three ways (ndd, /etc/system, and qfe.conf), depending on your needs. To set parameters that are valid until you reboot the system, use the ndd utility. Using ndd is a good way to test parameter settings.

▼ To set parameters so they remain in effect after you reboot the system

- Add the parameter values to the /etc/system file when you want to set parameters for all devices in the system.

- Create a `/kernel/drv/qfe.conf` file and add parameter values to the 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 set parameters that are valid until you reboot the system. The `ndd` utility supports any networking driver, that implements the Data Link Provider Interface (DLPI).

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

Identifying Device Instances

Before you use the `ndd` utility to get or set a parameter for a `qfe` device, you must specify the device instance for the utility since there will be at least four `SUNW,qfe` devices.

▼ To Specify the Device Instance for the `ndd` Utility

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

   For Sun Quad FastEthernet PCI adapter:

   ```
   # grep qfe /etc/path_to_inst
   "/pci@1f,2000/pci@2/SUNW,qfe@0,1" 0 "qfe"
   "/pci@1f,2000/pci@2/SUNW,qfe@1,1" 1 "qfe"
   "/pci@1f,2000/pci@2/SUNW,qfe@2,1" 2 "qfe"
   "/pci@1f,2000/pci@2/SUNW,qfe@3,1" 3 "qfe"
   ```

   In the preceding example, the four `SUNW,qfe@x,1` instances are from a Sun Quad FastEthernet PCI adapter installed in slot 2. For clarity, the instance numbers are bold.

For Sun Quad FastEthernet SBus adapter:

```
# grep qfe /etc/path_to_inst
"/sbus@1f,0/SUNW,qfe@1,8c10000" 1 "qfe"
"/sbus@1f,0/SUNW,qfe@1,8c00000" 0 "qfe"
"/sbus@1f,0/SUNW,qfe@1,8c30000" 3 "qfe"
"/sbus@1f,0/SUNW,qfe@1,8c20000" 2 "qfe"
```

In the preceding example, the four SUNW,qfe@1 instances are from a Sun Quad FastEthernet SBus adapter installed in slot 1. For clarity, the instance numbers are bold.

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

```
# ndd -set /dev/qfe 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/qfe driver instance, and assigns it to the parameter:

```
# ndd -set /dev/qfe parameter_value
```

- **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/qfe parameter
  ```

▼ To Use the ndd Utility in Interactive Mode

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

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

  ```
  # ndd /dev/qfe
  name to get/set? parameter_name_or_?
  ```

  After entering the parameter name, or ? to view all parameters, the ndd utility prompts you for the parameter value (see TABLE 2-1 through TABLE 2-8).

- **To list all the parameters supported by the** `qfe` **driver, type** `ndd /dev/qfe \?`.
  (See TABLE 2-1 through TABLE 2-8 for parameter descriptions.)

```
# ndd /dev/qfe \?
?                          (read only)
transceiver_inuse          (read only)
link_status                (read only)
link_speed                 (read only)
link_mode                  (read only)
ipg1                       (read and write)
ipg2                       (read and write)
use_int_xcvr               (read and write)
pace_size                  (read and write)
adv_autoneg_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)
autoneg_cap                (read only)
100fdx_cap                 (read only)
100hdx_cap                 (read only)
10fdx_cap                  (read only)
10hdx_cap                  (read only)
lp_autoneg_cap             (read only)
lp_100fdx_cap              (read only)
lp_100hdx_cap              (read only)
lp_10fdx_cap               (read only)
lp_10hdx_cap               (read only)
instance                   (read and write)
lance_mode                 (read and write)
ipg0                       (read and write)
#
```

FIGURE 2-1   Example of Listing All Parameters Supported by the `qfe` Driver

Setting Forced Mode

This section describes how to set forced mode (not capable of auto-negotiation).

▼ To Select One Local Transceiver Capability and Set Forced Mode

1. **Select one of the four capabilities (**`adv_100fdx_cap`, `adv_100hdx_cap`, `adv_10fdx_cap`, **or** `adv_10hdx_cap`**), and set its value to 1.**

   If you select more than one of the local transceiver capabilities, the driver selects the one that is highest in the priority order (see the footnote from TABLE 2-5).

2. **Set the local transceiver capabilities advertised by the hardware to forced mode = 0, which is not capable of auto-negotiation:** `adv_autoneg_cap  0`

   Use the `ndd` utility as described in "To Use the ndd Utility in Interactive Mode" on page 20.

Auto-Negotiation Mode

This section describes how to select at least one of the four local transceiver capabilities and set the mode to auto-negotiation.

▼ To Set the Mode to Auto-Negotiation

1. **Select** *at least one* **of the five capabilities (**`adv_100fdx_cap`, `adv_100hdx_cap`, `adv_10fdx_cap`, `adv_10hdx_cap`**) that you want to advertise to the remote system, and set its value to 1.**

2. **Set the local transceiver capabilities advertised by the hardware to 1**, **the auto-negotiation setting:** `adv_autoneg_cap  1`

   Use the `ndd` utility as described in "To Use the ndd Utility in Interactive Mode" on page 20.

Setting Parameters in the `/etc/system` File

To set the `qfe` driver parameters for all `SUNW,qfe` devices in the system so that the parameter variables are always effective (even after rebooting the system), enter the parameter variables in the `/etc/system` file. When you reboot the system, the system reads the `/etc/system` file and sets these parameter variables in the `qfe` module in the operating system kernel.

TABLE 2-9 lists the variables you can set in the /etc/system file.

TABLE 2-9   Setting Variables in the /etc/system File

Parameter	Variable
ipg1	qfe_ipg1
ipg2	qfe_ipg2
use_int_xcvr	qfe_use_int_xcvr
pace_size	qfe_pace_size
adv_autoneg_cap	qfe_adv_autoneg_cap
adv_100fdx_cap	qfe_adv_100fdx_cap
adv_100hdx_cap	qfe_adv_100hdx_cap
adv_10fdx_cap	qfe_adv_10fdx_cap
adv_10hdx_cap	qfe_adv_10hdx_cap
lance_mode	qfe_lance_mode
ipg0	qfe_ipg0

These parameter values, described in "Driver Parameter Values and Definitions" on page 12, are applicable to all SUNW,qfe devices on the system. See TABLE 2-1 through TABLE 2-8 for the descriptions of these parameters.

Note – In the future, the /etc/system file will not be available. It is not dynamic reconfiguration compatible.

▼ To Set Parameters in the /etc/system File

1. **Become superuser.**

2. **Add the appropriate lines to the** /etc/system **file. For example, to set the** ipg1 **and** ipg2 **parameters, make the following changes to the** /etc/system **file:**

```
set qfe:qfe_ipg1 = 10
set qfe:qfe_ipg2 = 5
```

3. **Save the** /etc/system **file.**

4. **Save all files and exit all programs, exit the windowing system.**

5. **Reboot the system by typing** `init 6` **at the superuser prompt.**

Setting Parameters Using the `/kernel/drv/qfe.conf` File

You can also specify the properties described in the section, "Setting Parameters in the /etc/system File" on page 22, on a per-device basis by creating a `qfe.conf` file in the `/kernel/drv` directory. The properties set in the `qfe.conf` file override the parameters set in the `/etc/system` file. Use a `qfe.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 "Driver Parameter Values and Definitions" on page 12.

The man pages for `prtconf(1M)`, `system(4)` and `driver.conf(4)` include additional details.

▼ To Set Driver Parameters on a Sun Quad FastEthernet PCI adapter

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

Typically the path names and the associated instance numbers will be present in the `/etc/path_to_inst` file.

```
# grep qfe /etc/path_to_inst
"/pci@1f,2000/pci@2/SUNW,qfe@0,1" 4 "qfe"
"/pci@1f,2000/pci@2/SUNW,qfe@1,1" 5 "qfe"
"/pci@1f,2000/pci@2/SUNW,qfe@2,1" 6 "qfe"
"/pci@1f,2000/pci@2/SUNW,qfe@3,1" 7 "qfe"
```

- In the above lines:
  - The first part within the double quotes specifies the hardware node name in the device tree.
  - The second number is the instance number.
  - The last part in double quotes is the driver name.
- In the device path 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 comma-separated numbers after the @ character at the end represent the device and function numbers, which are together referred to as unit-address.

To identify a PCI device unambiguously in the `qfe.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 of the previous example:

- Name = `SUNW,qfe`
- Parent = `/pci@1f,2000/pci@2`
- Unit-address = `0,1`

In the second line in the previous example:

- Name = `SUNW,qfe`
- Parent = `/pci@1f,2000/pci@2`
- Unit-address = `1,1`

In the third line in the previous example:

- Name = `SUNW,qfe`
- Parent = `/pci@1f,2000/pci@2`
- Unit-address = `2,1`

In the fourth line in the previous example:

- Name = `SUNW,qfe`
- Parent = `/pci@1f,2000/pci@2`
- Unit-address = `3,1`

2. **For example, set the `ipg1` and `ipg2` parameters for the above four devices in the `/kernel/drv/qfe.conf` file.**

```
name = "SUNW,qfe" parent = "/pci@1f,2000/pci@2" unit-address = "0,1" ipg1=20
ipg2=10;
name = "SUNW,qfe" parent = "/pci@1f,2000/pci@2" unit-address = "1,1" ipg1=20
ipg2=10;
name = "SUNW,qfe" parent = "/pci@1f,2000/pci@2" unit-address = "2,1" ipg1=20
ipg2=10;
name = "SUNW,qfe" parent = "/pci@1f,2000/pci@2" unit-address = "3,1" ipg1=20
ipg2=10;
```

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

4. **Save and close all files and programs, and exit the windowing system.**

5. **Halt and reboot the system by typing `init 6` at the superuser prompt.**

▼ To Set Driver Parameters on a Sun Quad FastEthernet SBus adapter

1. **Invoke the** `prtconf -v` **command and pipe the output to the** `more` **command (** `prtconf -v | more` **) or redirect the output of the command to a file (** `prtconf -v >` *filename***) and print the redirected file.**

2. **Find the section in the** `prtconf -v` **output for** `SUNW,qfe,instance #0`, **or** `SUNW,qfe,instance #1`**, and so on.**

   The output for `SUNW,qfe,instance #0` for a Sun Ultra 1 Creator Series system follows:

   ```
   SUNW,qfe, instance #0
               Driver software properties:
                   name <pm_norm_pwr> length <4>
                       value <0x00000001>.
                   name <pm_timestamp> length <4>
                       value <0x30743b26>.
               Register Specifications:
                   Bus Type=0xe, Address=0x8c00000, Size=108
                   Bus Type=0xe, Address=0x8c02000, Size=2000
                   Bus Type=0xe, Address=0x8c04000, Size=2000
                   Bus Type=0xe, Address=0x8c06000, Size=2000
                   Bus Type=0xe, Address=0x8c07000, Size=20
   ```

3. **Become superuser.**

4. **Create the** `qfe.conf` **file in the** `/kernel/drv` **directory using a text editor and add lines appropriate for the parameter you are setting. Use the following example for setting** `ipg1` **and** `ipg2` **parameters:**

   a. **Specify** `name="qfe"` **and** `class="sbus"`.

   b. **Use the** `reg` **property to specify the device,** `0xe` **in this case. Use the value following** `Bus Type` **in the** `prtconf -v` **output.**

   c. **Type the addresses followed by the specified sizes. Precede each size with** `0x` **and leading zeros, as indicated in the following screen.**

   d. **Set** `ipg1` **and** `ipg2`. **Type a semicolon (;) after the last value.**

In this example, these parameters are set to 20 and 10, respectively.

```
name="qfe" class="sbus"
reg=0xe,0x8c00000,0x00000108,0xe,0x8c02000,0x00002000,0xe,
0x8c04000,0x00002000,0xe,0x8c06000,0x00002000,0xe,0x8c07000,
0x00000020
ipg1=20 ipg2=10;
```

5. **Save the** `qfe.conf` **file.**

6. **Save and close all files and exit all programs; exit the windowing system.**

7. **Halt and reboot the system by typing** `init 6` **at the # prompt.**

Index