Platform Notes: The hme FastEthernet Device Driver

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara,CA 95054 U.S.A.
650-960-1300

Part No. 816-2348-10
May 2002, Revision A

Send comments about this document to: docfeedback@sun.com

Please
Recycle

Adobe PostScript

Contents

Preface

This book describes how to configure the `hme` driver for the SBus or PCI-bus based Sun™ Ultra™ workstations, Sun Enterprise™ servers, the SunSwift™ SBus Adapter, the SunFastEthernet™ Adapter 2.0, and the SunFastEthernet PCI Adapter.

Note that the 64-bit version of the Solaris operating environment uses the directory `/kernel/drv/sparcv9`. In this document and others, when `/kernel/drv` is mentioned, the `/kernel/drv/sparcv9` directory also applies.

How This Book Is Organized

Chapter 1 describes the hardware that uses the hme driver.

Chapter 2 describes how to configure the `SUNW,hme` device driver for your system.

Chapter 3 lists the values for each of the hme driver parameters.

Chapter 4 describes how to set the parameters for the hme driver.

Appendix A describes the auto-negotiation process.

Appendix B lists the `SUNW,hme` device driver parameters.

Appendix C lists commonly asked questions and explains troubleshooting techniques.

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	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

- The *Solaris on Sun Hardware Platform Guide* provides an overview of the `hme` driver and the Fast Ethernet Parallel Port SCSI (FEPS) ASIC.
- The *IEEE 802.3u Ethernet Standard* provides additional information about the Ethernet standard.

Accessing Sun Documentation Online

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:

docfeedback@sun.com

Please include the part number (816-2348-10) of your document in the subject line of your email.

CHAPTER 1

The hme Device Driver

This chapter gives a hardware overview of the SUNW,hme device, provides information on the operating speeds and modes for the SUNW,hme device, and discusses auto-negotiation. Note that the external transceiver is not present on the SunSwiftTM SBus Adapter or the SunSwiftTM PCI Adapter. The following sections describe features of the SUNW,hme device.

Supported Hardware

The hme device driver handles the SUNW,hme device on these hardware devices:

- SunSwift SBus Adapter
- SunSwift PCI Adapter
- SunFastEthernet Adapter 2.0
- SunFastEthernet PCI Adapter
- Sun Ultra systems and Sun Enterprise servers

TABLE 1-1 Transceivers Available In Sun Adapters

Adapters	Transceivers
SunSwiftTM SBus Adapter	Internal only
SunSwiftTM PCI Adapter	Internal only
SunTM FastEthernet Adapter 2.0	Both External and Internal
SunTM FastEthernet PCI Adapter	Both External and Internal

Hardware Overview

The SUNW,hme device provides 10BASE-TX or 100BASE-T networking interfaces using the Fast Ethernet Parallel Port SCSI (FEPS) ASIC and an internal 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 FEPS (SBus based) or PFEX (PCI-bus based) ASIC provides the SBus or PCI interface and Media Access Control (MAC) functions. The internal transceiver, which connects to an RJ-45 connector on all of the above hardware devices, provides the physical layer functions.

In addition to the RJ-45 connector, a Media Independent Interface (MII) connector, which is an Ultra DB 40 connector, is also provided on some adapters. The MII connects to an external transceiver that may use any physical media, such as copper or fiber, as specified in the 100BASE-TX standard. When an external transceiver is connected to the MII, the driver selects the external transceiver and disables the internal transceiver. The external transceiver may also support the 100BASE-T4 standard, which allows the link to operate in 100 Mbps speed using four pairs of category 3 or better cable.

Operating Speeds and Modes

You can operate the link in any of the following speeds and modes with the SUNW, hme device:

- 100 Mbps, full-duplex
- 100 BASE-T4 (with external transceiver only)
- 100 Mbps, half-duplex
- 10 Mbps, full-duplex
- 10 Mbps, half-duplex

The *100BASE-T IEEE 802.3u Ethernet Standard* describes these speeds and modes.

Auto-Negotiation

The auto-negotiation protocol, as specified by the *100BASE-T IEEE 802.3u Ethernet Standard*, selects the operation mode (half-duplex or full-duplex) at boot time or when the link state changes (the link fails or tries to connect). The auto-negotiation protocol also selects the speed and the full-duplex or half-duplex mode.

Details of the way the SUNW,hme device uses auto-negotiation are provided in Appendix A.

The auto-negotiation protocol does the following:

- Identifies all link partner-supported modes of operation
- Advertises its capabilities to the link partner
- Selects the highest common denominator mode of operation based on the following priorities (in decreasing order):
  - 100 BASE-T4
  - 100 Mbps, full-duplex
  - 100 Mbps, half-duplex
  - 10 Mbps, full-duplex
  - 10 Mbps, half-duplex

The link partner is the networking device (system, Ethernet hub, or Ethernet switch) at the other end of the link or cable.

If the SUNW,hme device is connected to a remote system or interface that is not capable of auto-negotiation, the system automatically selects the correct speed and half-duplex mode.

If adapters or systems are connected to a link partner and the auto-negotiation protocol fails to operate successfully, you can configure the device so it does not use this protocol. This forces the driver to set up the link in the mode and speed of your choice.

CHAPTER 2

Configuring the Driver Software

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

This chapter includes the following sections:

Installing the Driver Software

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

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

Before using the SUNW,hme device as your network interface, you will need to create and edit configuration files, as described in the next section.

Configuring the Hostname File

▼ To Configure the Hostname File

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

   For a Sun `hme` FastEthernet PCI adapter:

```
# grep hme /etc/path_to_inst
“/pci@1f,4000/network@1,1” 0 “hme”
“/pci@1f,4000/pci@4/SUNW,hme@0,1” 1 “hme”
```

   For a Sun `hme` FastEthernet SBus adapter:

```
# grep hme /etc/path_to_inst
“/sbus@1f,4000/network@1,1” 0 “hme”
“/sbus@1f,4000/pci@4/SUNW,hme@0,1” 1 “hme”
```

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

   If you want to use the network interface from the example in Step 1, you will need to create a file:

File Name	Instance Number
`/etc/hostname.hme0`	0

   - Do not create `/etc/hostname.hme`*num* files for Sun `hme` FastEthernet network interfaces you plan to leave unused.
   - The `/etc/hostname.hme`*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 the host name of any other interface, for example: `/etc/hostname.hme0` and `/etc/hostname.eri0` cannot share the same host name.

Using the instance examples in Step 1, the following example shows the two `/etc/hostname.hme`*num* files required for a system called `zardoz` that has a Sun `hme` FastEthernet (`zardoz, zardoz-11`).

```
# cat /etc/hostname.hme0
zardoz
# cat /etc/hostname.eri0
zardoz-11
```

3. **Create an appropriate entry in the `/etc/hosts` file for each active `hme` 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
```

Note – The Internet Protocol, version 6 (IPv6), expands the capabilities of IPv4, which is the current version and the default. The Sun `hme` FastEthernet device driver included in this release of the Solaris operating environment supports both IPv4 and IPv6. IPv4 uses the `/etc/hosts` configuration file, but IPv6 uses a different configuration file. To transition to, manage, and implement IPv6, refer to the Solaris System Administration documentation.

---

4. **Reboot your system.**

Booting From the Network

To use a Sun hme Ethernet interface as the boot device, perform the following tasks:

▼ To Boot From the Network

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

```
ok show-nets
```

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

For Sun hme FastEthernet PCI adapter*:*

```
/pci@1f,2000/pci@2/SUNW,hme@0,1
```

For Sun hme FastEthernet SBus adapter*:*

```
/sbus@1f,0/SUNW,hme@1,8c3000
```

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

---

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

```
ok boot full_path_name_of_the_hme_device
```

Optional Post-Installation Procedures

To customize the performance of the Sun hme FastEthernet device, perform the tasks in the following sections.

Setting Driver Parameters

The hme device driver, which is loaded from the Solaris CD-ROM, controls the SUNW,hme Ethernet devices. The device driver selects the link speed using the auto-negotiation protocol with the link partner. (See “Auto-Negotiation” on page 3.)

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

- Set a parameter on a per-device basis by creating the `hme.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 `hme` driver parameters generally for all `SUNW,hme` devices in the system by entering the parameter variables in the `/etc/system` file.

See Chapter 4 for more information.

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

▼ 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 `hme` devices, similar to the following examples:

   For Sun `hme` FastEthernet PCI adapter:

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

   For Sun `hme` FastEthernet SBus adapter:

```
/sbus@1f,0/SUNW,hme@1,8c30000
/sbus@1f,0/SUNW,hme@1,8c20000
/sbus@1f,0/SUNW,hme@1,8c10000
/sbus@1f,0/SUNW,hme@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_hme_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_*hme*_device***.**

   Perform Steps 2 to 4 to set the network speed for each hme 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 24 for more information on forcing network speed.

local-mac-address Property

Each of the network interfaces of the Sun hme FastEthernet 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 hme FastEthernet , set the NVRAM configuration variable local-mac-address? to true.

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

CHAPTER 3

Parameter Definitions

This chapter describes the parameters and settings for the hme device driver.

Driver Parameter Values and Definitions

The following sections describe the hme driver parameters, which are listed in TABLE 3-1.

- "Defining the Current Status" on page 13, Table 3-2.
- "Inter-Packet Gap Parameters" on page 13, Table 3-3.
- "Defining an Additional Delay Before Transmitting a Packet Using lance_mode and ipg0" on page 14, Table 3-4.
- "Operational Mode Parameters" on page 15, Table 3-5.
- "Operational Mode Priorities" on page 17, Table 3-6.
- "Defining the Number of Back-to-Back Packets to Transmit" on page 17, Table 3-7.
- "Reporting Transceiver Capabilities" on page 18, Table 3-8.
- "Reporting the Link Partner Capabilities" on page 19, Table 3-9.

TABLE 3-1 hme Driver Parameters, Status, and Descriptions

Parameter	Status	Description	Details
transciever_inuse	Read only	Defines the current status	
link_status	Read only	Defines the current status	"Defining the Current Status" on page 13
link_speed	Read only	Defines the current status	
link_mode	Read only	Defines the current status	
ipg1	Read and write	Inter-packet gap parameter	"Inter-Packet Gap Parameters" on page 13
ipg2	Read and write	Inter-packet gap parameter	

TABLE 3-1 hme Driver Parameters, Status, and Descriptions *(Continued)*

Parameter	Status	Description	Details
use_int_xcvr	Read and write	Operational mode parameter	
pace_size	Read and write	Operational mode parameter	"Defining the Number of Back-to-Back Packets to Transmit" on page 17
adv_autoneg_cap	Read and write	Operational mode parameter	"Reporting Transceiver Capabilities" on page 18
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	
autoneg_cap	Read only	Local transceiver auto negotiation capability	"Defining the Current Status" on page 13
100T4_cap	Read only	Local transceiver capability of the hardware	
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	"Reporting the Link Partner Capabilities" on page 19
lp_100T4_cap	Read only	Link partner capability	
lp_100fdx_cap	Read only	Link partner capability	
lp_100hdx_cap	Read only	Link partner capability	
lp_10fdx_cap	Read only	Link partner capability	
lp_10hdx_cap	Read only	Link partner capability	

TABLE 3-1 hme Driver Parameters, Status, and Descriptions *(Continued)*

Parameter	Status	Description	Details
instance	Read and write	Device instance	
lance_mode	Read and write	Additional delay before transmitting a packet	"Defining an Additional Delay Before Transmitting a Packet Using lance_mode and ipg0" on page 14
ipg0	Read and write	Additional delay before transmitting a packet	

Defining the Current Status

The read-only parameters described in TABLE 3-2 explain the operational mode of the interface. These parameters define the current status. A on this link to return to "Driver Parameter Values and Definitions" on page 11

TABLE 3-2 Read-Only Parameters Defining the Current Status

Parameter	Description	Values
link_status	Current link status	0 = Link down 1 = Link up
transceiver_inuse	Current transceiver status	0= Internal transceiver 1= External transceiver
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 Fast Ethernet Parallel Port SCSI (FEPS) ASIC supports programmable Inter-Packet Gap (IPG) parameters ipg1 and ipg2. The total IPG is the sum of ipg1 and ipg2. The total 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 IPG is 0.96 microseconds.

TABLE 3-3 lists the default values and allowable values for the IPG parameters, `ipg1` and `ipg2`. Click on this link to return to "Driver Parameter Values and Definitions" on page 11.

TABLE 3-3 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 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 following two ASICs support a programmable mode called `lance_mode`. The `ipg0` parameter is associated with `lance_mode`:

- Fast Ethernet Parallel Port SCSI (FEPS) for SBus)
- PCI Fast Ethernet and Expansion (PFEX for PCI)

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 set the additional delay with the `ipg0` parameter from 0 to 31, which is the nibble time delay. 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 3-4 defines the `lance_mode` and `ipg0 parameters.` Click on this link to return to “Driver Parameter Values and Definitions” on page 11 .

TABLE 3-4 Parameters Defining `lance_mode` and `ipg0`

Parameter	Values	Description
`lance_mode`	0	`lance_mode` disabled
	1	`lance_mode` enabled (default)
`ipg0`	0-31	Additional IPG before transmitting a packet (after receiving a packet) — Note 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 3-5 describes the operational mode parameters and their default values. Click on this link to return to “Driver Parameter Values and Definitions” on page 11.

TABLE 3-5 Operational Mode Parameters

Parameter	Description	Values
`adv_autoneg_cap`	Local transceiver capability advertised by the hardware	0 = Forced mode 1 = Auto-negotiation (default)
adv_100T4_cap	Local transceiver capability advertised by the hardware; read / write parameter	0 = Not 100BASE-T4 capable (default) 1 = 100BASE-T4 capable
`adv_100fdx_cap`	Local transceiver capability advertised by the hardware; read / write parameter	0 = Not 100 Mbit / sec full-duplex capable 1 = 100 Mbit / sec full-duplex capable (default)

TABLE 3-5 Operational Mode Parameters *(Continued)*

Parameter	Description	Values
`adv_100hdx_cap`	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`	Local transceiver capability advertised by the hardware; read / write parameter	0 = Not 10 Mbit / sec full-duplex capable 1 = 10 Mbit / sec full-duplex capable (default)
`adv_10hdx_cap`	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)
`use_int_xcvr`	Local transceiver capability selected for networking by the user	0 = External transceiver is used if connected 1 = Internal transceiver is used, even if the external transceiver is connected (default)

Selecting the Internal or External Transceiver

Use the `use_int_xcvr` parameter to change the default. For example, you can select the internal transceiver, even though the driver is capable of selecting the external transceiver. The driver, not the hardware, determines the internal and external transceiver.

If the external transceiver, which is connected to the MII interface, is present, the driver selects the external transceiver using the default `use_int_xcvr 0`. If the external transceiver is not present, the driver selects the internal transceiver. When this parameter is set to `use_int_xcvr 1,` the driver selects the internal transceiver, even if the external transceiver is connected.

Operational Mode Priorities

TABLE 3-6 lists the priority of the operational mode parameters. When the Sun `hme` FastEthernet Device Driver negotiates with a partner on the network, it implements the operational mode in the order shown in TABLE 3-6. Click on this link to return to "Driver Parameter Values and Definitions" on page 11.

TABLE 3-6 Operational Mode Priorities

Priority	Parameter
First	adv_100T4_cap
Second	adv_100fdx_cap
Third	adv_100hdx_cap
Fourth	adv_10fdx_cap
Fifth	adv_10hdx_cap

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

The `pace_size` parameter (see TABLE 3-7) 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. Click on this link to return to "Driver Parameter Values and Definitions" on page 11.

TABLE 3-7 Back-to-Back Packet Transmission Capability

Parameter	Values	Description
`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 3-8 describes the read-only transceiver capabilities (either the internal transceiver or the external transceiver), whichever is selected. Click on this link to return to “Driver Parameter Values and Definitions” on page 11.

TABLE 3-8 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
100T-4_cap	Local external transceiver capability of the hardware	0 = Not 100BASE-T4 capable 1 = 100BASE-T4 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

The parameters in TABLE 3-8 define the capabilities of the hardware. The internal transceiver can support all of these capabilities. The capabilities of the external transceiver are dependent on the device. If the external transceiver is not capable of auto-negotiation but has the capability of all speeds and modes (100 Mbps, 10 Mbps, half-duplex, and full-duplex), you must force the operational speed and mode of the external transceiver.

Reporting the Link Partner Capabilities

TABLE 3-9 describes the read-only link partner capabilities. Click on this link to return to “Driver Parameter Values and Definitions” on page 11.

TABLE 3-9 Read-Only Link Partner Capabilities

Parameter	Values
`lp_autoneg_cap`	0 = No auto-negotiation 1 = Auto-negotiation
lp_100T4_cap	0 = No 100BASE-T4 1 = 100BASE-T4
`lp_100fdx_cap`	0 = No 100 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 3-9 is not relevant and the parameter value = 0.

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

Click on this link to return to “Driver Parameter Values and Definitions” on page 11.

CHAPTER 4

Setting Parameters

This chapter describes three methods to configure the hme driver parameters. This chapter contains the following sections:

Parameter Setting Options

You can set the hme device driver parameters in three ways (ndd, /etc/system, and hme.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. If you want to test parameter settings, use the ndd utility described in "Setting Parameters Using ndd" on page 22. With ndd, the parameters are effective until you reboot the system.

To configure the hme driver parameters for all devices in the system so that the parameter values are always in effect (even after rebooting the system), enter the parameter values in the /etc/system file. When the system is rebooted, it reads the /etc/system file and sets the parameter values in that file.. "Setting Parameters in the /etc/system File" on page 26 describes this option.

To set the parameters for a particular device in the system, set the parameters in the hme.conf file in the /kernel/drv directory. The parameters set in the hme.conf file have precedence over the parameters set in the /etc/system file and override the parameters set in the /etc/system file. The parameters values set in hme.conf are always in effect (even after rebooting the system)."Setting Parameters Using the hme.conf File" on page 27 describes this option.

Setting Parameters Using ndd

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 hme driver and the ndd utility to modify (with the -set option) or display (without the -set option) the parameters for each SUNW,hme device.

Identifying Device Instances

Before you use the ndd utility to get or set a parameter for the hme device, you must specify the device instance for the utility if there is more than one SUNW,hme device.

Note – If there is only one SUNW,hme device, the device is automatically chosen by the ndd utility.

▼ 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.**
2. **Use that instance number to select the device as follows**:

```
% ndd -set /dev/hme instance instance#
```

The device remains selected until you change the selection.

Non-Interactive and Interactive Modes

You can use the ndd utility in two modes:

- Non-interactive
- Interactive

In non-interactive 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.)

Using the ndd Utility in Non-Interactive Mode

This section describes how to modify a parameter value and how to display a parameter value.

1. **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/hme` driver instance, and assigns it to the parameter:

   ```
   % ndd -set /dev/hme parameter value
   ```

2. **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/hme parameter
   ```

Using the ndd Utility in Interactive Mode

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

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

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

   After you enter the parameter name, the `ndd` utility prompts you for the parameter value (see TABLE 3-2 through TABLE 3-9).

2. **To list all the parameters supported by the `hme` driver, type `ndd /dev/hme \?`.**

   (See TABLE 3-2 through TABLE 3-9 for parameter descriptions.)

CODE EXAMPLE 4-1 Example of Listing All Parameters Supported by the hme Driver

```
example# ndd /dev/hme \?
?                               (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_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)
autoneg_cap                     (read only)
100T4_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_100T4_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)
example#
```

Setting Forced Mode

The following procedure describes how to set forced mode. When the SUNW,hme device is in forced mode it is not capable of auto-negotiation.

▼ To Select One Local Transceiver Capability and Setting Forced Mode

1. **Select one of the transceiver capabilities 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. The permitted values are as follows:

   - adv_100T4_cap
   - adv_100fdx_cap
   - adv_100hdx_cap
   - adv_10fdx_cap
   - adv_10hdx_cap

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 "Using the ndd Utility in Non-Interactive Mode" on page 23 in this chapter.

Setting Auto-Negotiation Mode

▼ To Set the Mode to Auto-Negotiation

1. **Select** *at least one* **of the five transceiver capabilities that you want to advertise to the remote system, and set its value to 1.**

   - adv_100T4_cap
   - adv_100fdx_cap
   - adv_100hdx_cap
   - adv_10fdx_cap
   - adv_10hdx_cap

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 "Using the ndd Utility in Interactive Mode" on page 23 in this chapter.

Setting Parameters in the /etc/system File

To configure the hme driver parameters for all SUNW,hme 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 hme module in the operating system kernel.

TABLE 4-1 lists the variables you need to set in the /etc/system file.

TABLE 4-1 Setting Variables in the /etc/system File

Parameter	Variable
ipg1	hme_ipg1
ipg2	hme_ipg2
use_int_xcvr	hme_use_int_xcvr
pace_size	hme_pace_size
adv_autoneg_cap	hme_adv_autoneg_cap
adv_100T4_cap	hme_adv_100T4_cap
adv_100fdx_cap	hme_adv_100fdx_cap
adv_100hdx_cap	hme_adv_100hdx_cap
adv_10fdx_cap	hme_adv_10fdx_cap
adv_10hdx_cap	hme_adv_10hdx_cap
lance_mode	hme_lance_mode
ipg0	hme_ipg0

These parameter values, described in Chapter 3, are applicable to all SUNW,hme devices on the system. See TABLE 3-2 through TABLE 3-9 for parameter descriptions. An example follows.

▼ To Set the ipg1 to 10 and ipg2 to 5 When Rebooting

1. **As superuser, add the following lines to the** `/etc/system` **file:**

```
set hme:hme_ipg1 = 10
set hme:hme_ipg2 = 5
```

2. **Save the** `/etc/system` **file.**
3. **Save all files and exit all programs. Exit the windowing system.**
4. **Reboot the system by typing** `init 6` **at the superuser prompt.**

   The system is halted and then rebooted.

Setting Parameters Using the hme.conf File

You can also specify the properties described in the section, “Setting Parameters in the /etc/system File” on page 26,” in this chapter on a per-device basis by creating the `hme.conf` file in the `/kernel/drv` directory. The properties set in the `hme.conf` file will override the parameters set in the `/etc/system` file. Use `hme.conf` 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 11.

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

▼ To Set ipg1 to 20 and ipg2 to 10 in SBus Slot 0xe

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 name (** `prtconf -v >` *filename*) **and print the redirected file.**

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

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

```
SUNW,hme, 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. **As superuser, create the** `hme.conf` **file in the** `/kernel/drv` **directory using a text editor and add lines similar to the following to the file**:

   a. **Specify** `name="hme"` **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.**

      These parameters are set to 20 and 10, respectively, in this example. The `ipg` parameters are defined in “Driver Parameter Values and Definitions” on page 11..

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

4. **Save the** `hme.conf` **file.**

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

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

Setting Driver Parameters for PCI-Bus hme Interfaces Using hme.conf

▼ To Configure Driver Parameters With PCI-Bus Based Systems

1. **Obtain the hardware path name for the device in the device tree.**

   Typically this path name and the associated instance number will be present in the /etc/path_to_inst file. For example, on a Sun Ultra 30 UPA/PCI system in which one SunSwift-PCI card is installed, the /etc/path_to_inst file will have the following two entries (in addition to entries for other devices):

```
"/pci@1f,4000/network@1,1" 0 "hme"
"/pci@1f,4000/pci@4/SUNW,hme@0,1" 1 "hme"
```

   - The first entry corresponds to the hme device on the motherboard. The second entry corresponds to the hme device on the SunSwift-PCI card.
   - In the previous 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 hme.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 = network
   - Parent = /pci@1f,4000
   - Unit-address = 1,1

   In the second line in the previous example:

   - Name = SUNW,hme
   - Parent = /pci@1f,4000/pci@4
   - Unit-address = 0,1

2. **Set the** `ipg1` **and** `ipg2` **parameters for the above two devices in the** `/kernel/drv/hme.conf` **file:**

```
name = “SUNW,hme” parent = “/pci@1f,4000” unit-address = “1,1” ipg1=10 ipg2=5;
name = “SUNW,hme” parent = “/pci@1f,4000/pci@4” unit-address = “0,1” ipg1=20
ipg2=10;
```

Note that for the motherboard device, the `SUNW,hme` device is used even though the path name uses the name `network`. The `SUNW,hme` device name is the value of the compatible property for this device.

APPENDIX **A**

Auto-Negotiation

Auto-negotiation matches the local device speed and mode with the link partner capability. This appendix offers detailed information on the auto-negotiation protocol.

The Auto-Negotiation Protocol

Auto-negotiation is a key feature of the Sun `hme` FastEthernet driver. The auto-negotiation protocol, as specified by the *100BASE-T IEEE 802.3u Ethernet 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 auto-negotiation protocol does the following:

- Identifies all link partner-supported modes of operation
- Advertises its capabilities to the link partner
- Selects the highest common denominator mode of operation based on the following priorities (in decreasing order):
  - 100 BASE-T4
  - 100 Mbps, full-duplex
  - 100 Mbps, half-duplex
  - 10 Mbps, full-duplex
  - 10 Mbps, half-duplex

The link partner is the networking device (system, Ethernet hub, or Ethernet switch) at the other end of the link or cable.

If adapters or systems are connected to a link partner and the auto-negotiation protocol fails to operate successfully, you can configure the device so it does not use this protocol. This forces the driver to set up the link in the mode and speed of your choice. For more information on this topice, see “Parameter Setting Options” on page 21

Boot Process on the Network

The auto-negotiation protocol does the following when the system is booted:

- Identifies all link partner-supported modes of operation
- Advertises its capabilities to the link partner
- Selects the highest common denominator mode of operation based on the following priorities (in decreasing order):

Priority	Line Speed and Mode
First	100 BASE T4
Second	100 Mbps, full-duplex
Third	100 Mbps, half-duplex
Fourth	10 Mbps, full-duplex
Fifth	10 Mbps, half-duplex

The link partner is the networking device (system, Ethernet hub, or Ethernet switch) at the other end of the link or cable.

Correcting Errors in Negotiating

If the SUNW,hme device is connected to a remote system or interface that is not capable of auto-negotiation, your system automatically selects the correct speed and half-duplex mode.

If the SUNW,hme device is connected to a link partner with which the auto-negotiation protocol fails to operate successfully, you can configure the device so it does not use this protocol. This forces the driver to set up the link in the mode and speed of your choice. For more information on this topice, see “Parameter Setting Options” on page 21

Internal (Local) Transceiver

The internal transceiver is supported by the driver and is capable of all the operating speeds and modes (except the 100BASE-T4 mode) listed in the section “Operating Speeds and Modes” on page 2,”. When the internal transceiver is used, the default is auto-negotation by the `hme` driver, which automatically selects the speed and mode of the link. The internal transceiver performs auto-negotiation with the remote end of the link (link partner) to select a common mode of operation.

The internal transceiver also supports a forced mode of operation. This is where the user selects the speed and mode using the `ndd` utility, the `/etc/system` file, or the `hme.conf` file. The `ndd` utility makes calls to the `hme` driver to choose the speed and mode.

External Transceiver

When an external transceiver (not present on the SunSwift SBus Adapter) is connected to the MII interface, the driver selects the external transceiver for networking operations.

- If the external transceiver supports auto-negotiation, the driver uses the auto-negotiation feature to select the link speed and mode.
- If the external transceiver does not support auto-negotiation, the driver selects the highest priority mode supported by the transceiver.

You can also manually select the speed and mode of the link. For example, two transceivers might not support the same mode and speed. Therefore, you must select the highest mode and speed that *both* transceivers support using the `ndd` utility. See the list of operating speeds and modes in the section “Operating Speeds and Modes” on page 2.

APPENDIX **B**

Driver Parameters

This appendix presents the complete list of SUNW,hme device driver parameters..

Driver Parameter Definitions

TABLE B-1 lists the hme driver parameters in the order they are encountered in the `/kernel/drv/hme.conf` file.

TABLE B-1 `hme` Driver Parameters, 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_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

TABLE B-1 hme Driver Parameters, Status, and Descriptions *(Continued)*

Parameter	Status	Description
adv_10hdx_cap	Read and write	Operational mode parameter
autoneg_cap	Read only	Local transceiver auto negotiation capability
100T4_cap	Read only	Local transceiver capability of the hardware
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_100T4_cap	Read only	Link partner capability
lp_100fdx_cap	Read only	Link partner capability
lp_100hdx_cap	Read only	Link partner capability
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

APPENDIX **C**

Troubleshooting

Common Problems

There are many steps involved in isolating and resolving network interface problems. Once you have configured your interface and network properly, these steps should be transparent to you. However, problems in your configuration may cause a failure at any number of points along the communication path. This section contains checks to determine the failure point, and resolution process. If you continue to experience problems, information gathered from these checks will help your service provider diagnose and resolve the problem.

How do I force the adapter to run at a particular speed?

You might need to force hme to 10 Mb half-duplex if using 10 MB switch or a hub. The same thing holds for forcing to 100 MB Full Duplex, and all of the other speeds. Replace the "0" with "1" for the required speed.

Reboot your system to apply the changes.

The following example uses a standard 10baseT using shared Hub or 10Base2 using Mii to Aui adapter. The following instructions will change the settings for all the hme adapters on the machine. Add the following lines to the /etc/system file using a text editor:

```
set hme:hme_adv_autoneg_cap=0

set hme:hme_adv_100fdx_cap=0

set hme:hme_adv_100hdx_cap=0

set hme:hme_adv_10hdx_cap=1

set hme:hme_adv_10hdx_cap=0
```

For individual adapters use ndd commands as follows:

In the following example, set the X value to the interface in question

```
# ndd -set /dev/hme use_int_xcvr 1
```

Use the following command to force to use internal Tx (100TX rj45 port)

```
# ndd -set /dev/hme use_int_xcvr 1
```

Use the following command to force off 100 Mb half-duplex

```
# ndd -set /dev/hme adv_100hdx_cap 0
```

Use the following command to force off 100Mb T-4

```
# ndd -set /dev/hme adv_100T4_cap 0
```

Use the following command to force on 100Mb full-duplex

```
# ndd -set /dev/hme adv_100fdx_cap 1
```

Use the following command to force off auto-negotiation

```
# ndd -set /dev/hme autoneg_cap 0
```

Example: force hme0 to 100 Mb Full Duplex

```
# ndd -set /dev/hme instance 0
# ndd -set /dev/hme adv_10hdx_cap 0
# ndd -set /dev/hme adv_100fdx_cap 1
# ndd -set /dev/hme adv_10fdx_cap 0
# ndd -set /dev/hme adv_10hdx_cap 0
# ndd -set /dev/hme adv_autoneg_cap 0
```

This will change the settings dynamically. There is no need to reboot your system. This set of commands affects only the selected interface. By default the instance number is "0". Place these commands in the rc scripts (Edit a file in /etc/rc2.d, for example, `S95nddsettings`) to make these changes permanent and effective after a reboot .

How do I change the IP address for my network interface?

Edit the `/etc/hosts` file. Change the IP address corresponding to this machine and then reboot.

Run the command `sys-unconfig`(1M) if you move your system to a different subnet or domain.

Caution – Read the `sys-unconfig`(1M) man pages before using the command. Many files are permanently altered and deleted.

How do I change the hostname for my network interface?

Edit the following files to change the host name associated with the IP address:

- `/etc/hosts`
- `/etc/hostname.<device_instance>`
- `/etc/net/ticlts/hosts`
- `/etc/net/ticots/hosts`
- `/etc/net/ticotsord/hosts`

How do I configure another IP address or subnet for my network interface?

Modify the files listed above from the previous question by adding the new additional hostname and IP address. In addition, add an entry needs to `/etc/netmasks`. The `netmasks(4)` man page provides the convention for this file.

Create a logical interface using the `ifconfig` command. For example, to configure the second interface on `hme0`

```
# ifconfig hme0:1 123.4.5.6 up
```

Use the following command for systems running Solaris 8 and later versions:

```
# ifconfig hme0 addif 123.4.5.6 up
```

This takes effect until the next reboot. To permanently configure an additional IP address for the `hme0` interface, add an additional line to `/etc/hostname.hme0`. Note that for Solaris 8 or newer systems, add the following line to the `/etc/hostname.hme<device_instance>:`

```
addif 123.4.5.6 up
```

How do I know if my network interface is up? I can not ping the device.

1. **Verify via the `ifconfig` command that the interface is configured for IP and that the adapter has the UP flag.**

```
# ifconfig -a
```

2. **If this is ok make sure the adapter has a link:**

   For example if the `ifconfig` command returns `hme0,` set the instance to query. This is needed on systems with multiple adapters.

```
# ndd -set /dev/hme instance 0
```

   Obtain the link status of the device. A return value of 1 indicates that the adapter is seeing a link, while a return value of 0 indicates there is no link.

```
# ndd -get /dev/hme link_status
```

   You cannot check `ndd` status until your adapter appears in the output to an `ifconfig -a` command. A stream must exist to the driver.

3. **Snoop the interface.**

```
# snoop -d hme0
Using device /dev/hme (promiscuous mode)
runnreff -> 10.10.192.255 UDP D=138 S=138 LEN=182
runnreff -> 10.10.192.255 UDP D=137 S=137 LEN=58
dhcp-192-238 -> 10.10.192.255 UDP D=137 S=137 LEN=58
```

   If you see packets then you know your interface is working. If your adapter is connected to a switch you will see only broadcast, multiast and unicast packets for this adapter. Verify that the switch port is configured correctly.

   If you see no packets, test the adapter hardware at the `ok` prompt via `watch-net-all` or `test <adapter path>`.

4. **If packets are seen with the snoop trace, then the network adapter is working correctly.**

   However, if `ping` still does not work, `traceroute(1M)` and `netstat(1M)` are helpful .

   The `traceroute` commands displays the route to your destination station:

```
# traceroute -i hme0 teriya
traceroute to teriya (129.150.143.68), 30 hops max, 40 byte packets
sw203-1 (192.168.203.1)  2.684 ms  2.444 ms  2.870 ms
nwk02rfsrb149 (129.150.149.241)  0.687 ms  0.704 ms  0.720 ms
teriya (129.150.143.68)  0.457 ms
```

   The `netstat -r` command displays the system routing tables:

```
# netstat -r

    Routing Table: IPv4
      Destination           Gateway           Flags  Ref   Use   Interface
      -------------------- -------------------- ----- ----- ------ ---------
      nwk02_1815_labnet1   atmsw-27             U        1    434  hme0
      199.99.234.0         cip234-27            U        1      1  ba0
      224.0.0.0            atmsw-27             U        1      0  hme0
      default              sw203-1              UG       1    252
      localhost            localhost            UH       2      6  lo0
```

How do I determine the current speed of my adapter?

Issue a dmesg command to display the last message that the driver sent to the console. It will state the speed of the link as well as the mode. However, often other messageshave caused the circular buffer to overflow erasing hme messages.

```
# dmesg | grep hme
...
Oct  1 17:42:03 atmsw-27 genunix: [ID 936769 kern.notice] hme0 is
/sbus@1f,0/SUNW,hme@e,8c00000
Oct  1 17:42:07 atmsw-27 hme: [ID 517527 kern.notice] SUNW,hme0 :
Internal Transceiver Selected.
Oct  1 17:42:07 atmsw-27 hme: [ID 517527 kern.notice] SUNW,hme0 :
Auto-Negotiated  100 Mbps Half-Duplex Link Up
    #
```

Use the ndd command to display and set selected configuration parameters in the hme driver. To see the parameters that are supported, type the following command:

```
# ndd /dev/hme \?
```

To display the link speed, issue a ndd command to tell the driver which instance of hme that you are intested querrying. Note that instance 0 is the default. Then issue an ndd command list the link speed.

```
# ndd -set /dev/hme instance 0
# ndd -get /dev/hme link_speed
1
# ndd -get /dev/hme link_mode
1
```

The return value of 1 for link_speed indicates that this link is running at 100Mbps. A value of 0 indicates a link speed of 10Mbps. A link_mode value of 1 indicates full duplex and a value of 0 indicates half-duplex.

Use the undocumented netstat -k command to list driver kstats. Below is an example from the output of a Solaris 8 system. Note that this option changes from release to release and is not guaranteed to work in the future.

CODE EXAMPLE C-1 The output from a `netstat -k` command

```
# netstat -k hme0
hme0:
ipackets 317522 ierrors 0 opackets 9790 oerrors 0 collisions 43
defer 0 framing 0 crc 0 sqe 0 code_violations 0 len_errors 0
ifspeed 100000000 buff 0 oflo 0 uflo 0 missed 0 tx_late_collisions 0
retry_error 0 first_collisions 0 nocarrier 0 nocanput 0
allocbfail 0 runt 0 jabber 0 babble 0 tmd_error 0 tx_late_error 0
rx_late_error 0 slv_parity_error 0 tx_parity_error 0 rx_parity_error 0
slv_error_ack 0 tx_error_ack 0 rx_error_ack 0 tx_tag_error 0
rx_tag_error 0 eop_error 0 no_tmds 0 no_tbufs 0 no_rbufs 0
rx_late_collisions 0 rbytes 141596475 obytes 1237354 multircv 0 multixmt 9
brdcstrcv 307824 brdcstxmt 59 norcvbuf 0 noxmtbuf 0   newfree 0
ipackets64 317522 opackets64 9790 rbytes64 141596475 obytes64 1237354
align_errors 0
fcs_errors 0   sqe_errors 0 defer_xmts 0 ex_collisions 0
macxmt_errors 0 carrier_errors 0 toolong_errors 0 macrcv_errors 0
link_duplex 0 inits 12 rxinits 0 txinits 0 dmarh_inits 0
dmaxh_inits 0 link_down_cnt 0 phy_failures 0 xcvr_vendor 524311
asic_rev 193
```

Notice the parameter `ifspeed`, indicats that this link is running at 100Mbps. The `link_duplex` parameter indicates that thislink is running at half-duplex.

I moved the board to another slot, but `ifconfig` does not work

Is the new card recognized by the system?

This can be detemined by inspecting the `/etc/path_to_inst` file for instances of `hme`.

Which instance did the network adapter move to?

The `path_to_inst` file records the mappings of physical device names to instance numbers. The system modifies this file when you move a board to a different slot and issue a boot -r command from the ok prompt. This information is persistent across reboots. The system remembers that your card was in another slot, so when

you place it in a new slot, it creates a new device_instance, leaving the old one intact. Move the /etc/hostname.hme<device_instance> to a file name with the new device_instance number. After a reconfiguration boot, boot -r, the system will use the new device_instance number to ifconfig and plumb your adapter.

1. **Inspect the file** /etc/path_to_inst **for** hme **instances.**

   Note the value of the highest hme instance. Also note the instance value of your adapter.

```
# grep hme /etc/path_to_inst
"/sbus@1f,0/SUNW,hme@e,8c00000" 0 "hme"
```

2. **Follow the instructions that came with your system to replace an I/O adapter. Move the card to a new slot and issue a** boot -r **command.**

```
ok boot -r
```

3. **Inspect the file** /etc/path_to_inst **for** hme **instances**

```
# grep hme /etc/path_to_inst
“/sbus@1f,0/SUNW,hme@e,8c00000" 0 "hme"
“/sbus@1f,0/SUNW,hme@e,8c00000" 1 "hme"
```

4. **Move the** hostname.hme<instance> **file to the new highest instance.**

   If you have moved the adapter into an slot that previously contained an hme adapter, then this is the instance that you should move the hostname.hme<instance> file to.

Error Messages

This section includes common error messages you might see while configuring and bringing up your hme interface. For each message, there is a brief explanation of the problem and a possible solution.

No response from Ethernet network : Link down -- cable problem?

The interface has detected Link Down while trying to transmit a packet.

- Check the cable if it is connected to the interface and another host/switch.
- Check the LEDs on the switch or host.
- Check the `ndd` parameters to confirm the speed selection or auto-negotiation mode on the switch or host,
- Force the interface to go into forced mode, that is, disable auto-negotiation.
- Try speeds from 100 to 10 and modes from full-duplex to half-duplex.

Driver is busy with upper layer

The device failed to detach because of open streams into it. The `cfgadm(DR)` and `modunload` commands can cause this message.

- Close all open streams to the driver (including any `snoop(1M)` to this device)
- Verify that the device is not in DR-Suspended state.
- Issue the command, `ifconfig hme<instance> down unplumb` before issuing the `modunload` command to remove the driver

Parallel detection fault

While attempting to auto-negotiate with a link partner, the interface encountered a parallel detection fault. The device will retry auto-negotion until the link is up.

- Check and replace the cable to the Switch or other host
- Try different port on the switch or a different switch or host.
- Set the device into force mode using `ndd`.
- Replace the adapter hardware.

Transceiver does not talk MII or "Transceiver isolate failed

The transceiver on the adapter fails to reset. The device will retry resetting the transceiver.

- Replace the adapter hardware.

No transceiver found

The device did not detect the external/internal transceiver. This often occurs when the external transceiver is not properly connected or is not functioning properly.

- Check if the External transceiver is present. Verify that it is inserted or connected properly.
- Replace the external transceiver with a known good(working) transciever.
- Replace the adapter hardware.

Dev not used - dev in slave only slot

The adapter is in a Slave-Only-Slot. The device failed to attach.

- Check the system I/O slots diagram.
- Identify Slave-only-slots and move the adapter to a non-Slave-Only-Slot.

Failed to initialize hardware/driver

The device failrf to intialise/setup/allocate resources needed to function properly.

- Check the system resources using `vmstat(1M)`.
- Reboot the system or DR the device to free up resources
- Replace the adapter hardware.

External Transceiver: anar not set with speed selection

Either the link partner's auto-negotiation capability cannot be read, or the switch or other host is not advertising its auto-negotiation capabilities. This often occurs when an external transceiver attached to the device.

Set the mode to half duplex and try the speed from 100 to 10 using the parallel detection scheme. This message indicates that the above configuration failed to set the speed.

- Verify that the external transceiver is attached.Verify the speed selection bits.
- Verify the switch or other host setup.
- Replace the external transceiver with a known working transciever.

Index

S

T

U