# Contents

1. Configuring the `ge` Device Driver  
   Configuring the Driver Parameters  1  
   Setting `ge` Driver Parameters  8  
   Network Configuration  14  
   ▼  To Configure the Host Files  14  
   Installing the Solaris Operating Environment Over a GigabitEthernet Network  16  
   ▼  To Install the Solaris Environment Over a GigabitEthernet Network  17
Preface

These Platform Notes provide instructions for configuring the software used by the Sun GigabitEthernet adapter. They also contain information for configuring the network. Unless otherwise noted, all instructions apply to both the Sun GigabitEthernet/P adapter and the Sun GigabitEthernet/S adapter.

Shell Prompts

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>C shell</td>
<td><code>machine_name%</code></td>
</tr>
<tr>
<td>C shell superuser</td>
<td><code>machine_name#</code></td>
</tr>
<tr>
<td>Bourne shell and Korn shell</td>
<td><code>$</code></td>
</tr>
<tr>
<td>Bourne shell and Korn shell superuser</td>
<td><code>#</code></td>
</tr>
</tbody>
</table>
Typographic Conventions

<table>
<thead>
<tr>
<th>Typeface or Symbol</th>
<th>Meaning</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>AaBbCc123</td>
<td>The names of commands, files, and directories; on-screen computer output</td>
<td>Edit your .login file. Use ls -a to list all files. % You have mail.</td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>What you type, when contrasted with on-screen computer output</td>
<td>% su Password:</td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>Book titles, new words or terms, words to be emphasized</td>
<td>Read Chapter 6 in the User’s Guide. These are called class options. You must be superuser to do this.</td>
</tr>
<tr>
<td></td>
<td>Command-line variable; replace with a real name or value</td>
<td>To delete a file, type rm filename.</td>
</tr>
</tbody>
</table>

Related Documentation

TABLE P-1 Related Documentation

<table>
<thead>
<tr>
<th>Title</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 8 System Administration Guide, Volume 3</td>
<td>806-0916-10</td>
</tr>
<tr>
<td>Solaris 8 Sun Hardware Platform Guide</td>
<td>806-2221-10</td>
</tr>
</tbody>
</table>

Ordering Sun Documentation

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.
For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at:

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

---

**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 (806-7745-10) of your document in the subject line of your email.
CHAPTER 1

Configuring the ge Device Driver

These Platform Notes provide instructions for configuring the software used by the Sun GigabitEthernet adapter. They also contain information for configuring the network. Unless otherwise noted, all instructions apply to both the Sun GigabitEthernet/P adapter and the Sun GigabitEthernet/S adapter.

The document contains the following sections:

- Configuring the Driver Parameters
- Setting ge Driver Parameters
- Network Configuration

Configuring the Driver Parameters

This section provides an overview of the capabilities of the GigabitEthernet ASIC used in the adapters, lists the available ge driver parameters, and describes how to configure these parameters.

The ge driver controls the Sun GigabitEthernet adapter devices. The Sun GigabitEthernet device is identified as network with the model property SUNW,sbus-gem or SUNW,pci-gem node. The ge driver is attached to the device with the compatible property pci108e,2bad for the Sun GigabitEthernet/P adapter or SUNW,sbus-gem for the Sun GigabitEthernet/S adapter. You can manually configure the parameters to customize each Sun GigabitEthernet adapter in your system.
GigabitEthernet MAC

The GigabitEthernet MAC (GEM) provides 1000BASE-SX networking interfaces. The driver automatically sets the link speed to 1000 Mbps and conforms to the IEEE 802.3z Ethernet standard. The GEM PCI ASIC provide the PCI interface, Media Access Control (MAC) functions, and Physical Code Sublayer (PCS) functions. The GEM SBus ASIC provides the SBus interface, MAC functions, and PCS functions. The External SERDES, which connects the 1000BASE-SX Compliant SC connector to the ASIC, provides the physical layer functions.

The GEM MAC and PCS are capable of all the operating speeds and modes listed in “Autonegotiation Mode” on page 11. The PCS performs autonegotiation with the remote end of the link (link partner) to select a common mode of operation.

The PCS also supports a forced mode of operation. You can select the speed and mode by creating a ge.conf file.

Driver Parameter Values and Definitions

TABLE 1-1 describes the parameters and settings for the ge driver.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_status</td>
<td>Read only</td>
<td>Defines the current status</td>
</tr>
<tr>
<td>link_speed</td>
<td>Read only</td>
<td>Defines the current status</td>
</tr>
<tr>
<td>link_mode</td>
<td>Read only</td>
<td>Defines the current status</td>
</tr>
<tr>
<td>ipg1</td>
<td>Read and write</td>
<td>Interpacket gap parameter</td>
</tr>
<tr>
<td>ipg2</td>
<td>Read and write</td>
<td>Interpacket gap parameter</td>
</tr>
<tr>
<td>instance</td>
<td>Read and write</td>
<td>Device instance</td>
</tr>
<tr>
<td>lance_mode</td>
<td>Read and write</td>
<td>Enable additional delay before transmitting a packet</td>
</tr>
<tr>
<td>ipg0</td>
<td>Read and write</td>
<td>Additional delay before transmitting a packet</td>
</tr>
<tr>
<td>adv_1000autoneg_cap</td>
<td>Read and write</td>
<td>Operational mode parameter</td>
</tr>
<tr>
<td>adv_1000fdx_cap</td>
<td>Read and write</td>
<td>Operational mode parameter</td>
</tr>
<tr>
<td>adv_1000hdx_cap</td>
<td>Read and write</td>
<td>Operational mode parameter</td>
</tr>
<tr>
<td>adv_pauseTX</td>
<td>Read and write</td>
<td>Operational mode parameter</td>
</tr>
<tr>
<td>adv_pauseRX</td>
<td>Read and write</td>
<td>Operational mode parameter</td>
</tr>
</tbody>
</table>
Defining the Current Status

The read-only parameters described in TABLE 1-2 explain the operational mode of the interface. Based on the value of these parameters, you can determine the current status of a link.

### TABLE 1-2  Read-Only Parameters Defining the Current Status

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_status</td>
<td>Current</td>
<td>Link status</td>
</tr>
<tr>
<td></td>
<td>link</td>
<td>0 = Link down</td>
</tr>
<tr>
<td></td>
<td>speed</td>
<td>1 = Link up</td>
</tr>
<tr>
<td>link_speed</td>
<td>Valid only</td>
<td>Link is up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Link is not up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000 = 1000 Mbps</td>
</tr>
<tr>
<td>link_mode</td>
<td>Valid only</td>
<td>Link is up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Half duplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Full duplex</td>
</tr>
</tbody>
</table>

TABLE 1-1  ge Driver Parameters, Status, and Descriptions (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000autoneg_cap</td>
<td>Read only</td>
<td>PCS autonegotiation capability</td>
</tr>
<tr>
<td>1000fdx_cap</td>
<td>Read only</td>
<td>PCS full duplex capability</td>
</tr>
<tr>
<td>1000hdx_cap</td>
<td>Read only</td>
<td>PCS half duplex capability</td>
</tr>
<tr>
<td>asm_dir_cap</td>
<td>Read only</td>
<td>PCS ASM_DIR capability</td>
</tr>
<tr>
<td>pause_cap</td>
<td>Read only</td>
<td>PCS Symmetric PAUSE capability</td>
</tr>
<tr>
<td>lp_1000autoneg_cap</td>
<td>Read only</td>
<td>Link partner autonegotiation capability</td>
</tr>
<tr>
<td>lp_1000fdx_cap</td>
<td>Read only</td>
<td>Link partner capability</td>
</tr>
<tr>
<td>lp_1000hdx_cap</td>
<td>Read only</td>
<td>Link partner capability</td>
</tr>
<tr>
<td>lp_asm_dir_cap</td>
<td>Read only</td>
<td>Link partner capability</td>
</tr>
<tr>
<td>lp_pause_cap</td>
<td>Read only</td>
<td>Link partner capability</td>
</tr>
</tbody>
</table>
Flow Control Parameters

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

For normal operations, GEM doesn’t need to source flow control frames. However, if GEM is operating on a slow bus (for instance, a 33-MHz PCI bus slot), and there is a lot of frame reception activity, there could be a performance degradation due to Receive FIFO overflow. If the link partner is capable of terminating PAUSE flow control frames, the performance of GEM could be improved by enabling adv_pauseTX and restarting autonegotiation.

Interpacket Gap Parameters

The GEM ASIC supports the programmable Interpacket Gap (IPG) parameters $ipg_1$ and $ipg_2$. The total IPG is the sum of $ipg_1$ and $ipg_2$: 0.096 microseconds for the link speed of 1000 Mbps.

TABLE 1-4 lists the default values and allowable values for the IPG parameters, $ipg_1$ and $ipg_2$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values (Byte-time)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ipg_1$</td>
<td>0, 255</td>
<td>$ipg_1 = 8$ (default at initialization)</td>
</tr>
<tr>
<td>$ipg_2$</td>
<td>0, 255</td>
<td>$ipg_2 = 4$ (default at initialization)</td>
</tr>
</tbody>
</table>
By default, the driver sets $ipg_1$ to 8-byte time and $ipg_2$ to 4-byte time, which are the standard values. (Byte time is the time it takes to transmit one byte on the link, with a link speed of 1000 Mbps.)

If your network has systems that use longer IPG (the sum of $ipg_1$ and $ipg_2$) and if those machines seem to be slow in accessing the network, increase the values of $ipg_1$ and $ipg_2$ to match the longer IPGs of other machines.

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

The GEM ASIC supports a programmable mode called `lance_mode`. The `ipg0` parameter is associated with `lance_mode`.

If `lance_mode` is enabled (the default), an additional delay is added by setting the `ipg0` parameter before transmitting the packet. This delay 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 transmission 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 media byte time delay.

**TABLE 1-5** defines the `lance_mode` and `ipg0` parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values and Descriptions</th>
</tr>
</thead>
</table>
| `lance_mode` | 0 = lance_mode disabled  
1 = lance_mode enabled (default) |
| `ipg0` | 0 to 30 = Additional IPG before transmitting a packet (after receiving a packet) |
Operational Mode Parameters

TABLE 1-6 describes the operational mode parameters and their default values.

**TABLE 1-6** Operational Mode Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adv_1000autoneg_cap</td>
<td>Local PCS capability advertised by the hardware</td>
</tr>
<tr>
<td></td>
<td>0 = Forced mode</td>
</tr>
<tr>
<td></td>
<td>1 = Autonegotiation (default)</td>
</tr>
<tr>
<td>adv_1000fdx_cap</td>
<td>Local PCS capability advertised by the hardware</td>
</tr>
<tr>
<td></td>
<td>0 = Not 1000 Mbits/sec full duplex capable</td>
</tr>
<tr>
<td></td>
<td>1 = 1000 Mbits/sec full duplex capable (default)</td>
</tr>
<tr>
<td>adv_1000hdx_cap</td>
<td>Local PCS capability advertised by the hardware</td>
</tr>
<tr>
<td></td>
<td>0 = Not 1000 Mbits/sec half duplex capable</td>
</tr>
<tr>
<td></td>
<td>1 = 1000 Mbits/sec half duplex capable (default)</td>
</tr>
<tr>
<td>adv_pauseTX</td>
<td>Local PCS capability advertised by the hardware</td>
</tr>
<tr>
<td></td>
<td>0 = Not Pause TX capable (default)</td>
</tr>
<tr>
<td></td>
<td>1 = Pause TX capable</td>
</tr>
<tr>
<td>adv_pauseRX</td>
<td>Local PCS capability advertised by the hardware</td>
</tr>
<tr>
<td></td>
<td>0 = Not Pause RX capable</td>
</tr>
<tr>
<td></td>
<td>1 = Pause RX capable (default)</td>
</tr>
</tbody>
</table>

Reporting Local PCS Capabilities

TABLE 1-7 describes the read-only PCS capabilities that GEM PCS supports. These parameters define the capabilities of the hardware.

**TABLE 1-7** Read-Only PCS Capabilities

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description (Local PCS Capabilities)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000autoneg_cap</td>
<td></td>
</tr>
<tr>
<td>0 = Not capable of autonegotiation</td>
<td></td>
</tr>
<tr>
<td>1 = Autonegotiation capable</td>
<td></td>
</tr>
<tr>
<td>1000fdx_cap</td>
<td></td>
</tr>
<tr>
<td>Local PCS Full Duplex capability</td>
<td></td>
</tr>
<tr>
<td>0 = Not 1000 Mbits/sec full-duplex capable</td>
<td></td>
</tr>
<tr>
<td>1 = 1000 Mbits/sec full-duplex capable</td>
<td></td>
</tr>
</tbody>
</table>
TABLE 1-8 describes the read-only link partner capabilities.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description (Local PCS Capabilities)</th>
</tr>
</thead>
<tbody>
<tr>
<td>lp_1000autoneg_cap</td>
<td>0 = No autonegotiation 1 = Autonegotiation</td>
</tr>
<tr>
<td>lp_1000fdx_cap</td>
<td>0 = No 1000 Mbits/sec full duplex transmission 1 = 1000 Mbits/sec full duplex</td>
</tr>
<tr>
<td>lp_1000hdx_cap</td>
<td>0 = No 1000 Mbits/sec half duplex transmission 1 = 1000 Mbits/sec half duplex</td>
</tr>
<tr>
<td>lp_asm_dir_cap</td>
<td>0 = Not Asymmetric Pause capable 1 = Asymmetric Pause (from Local Device) capable</td>
</tr>
<tr>
<td>lp_pause_cap</td>
<td>0 = Not Symmetric Pause capable 1 = Symmetric Pause capable</td>
</tr>
</tbody>
</table>

If the link partner is not capable of autonegotiation (when lp_1000autoneg_cap is 0), the remaining information described in TABLE 1-8 is not relevant and the parameter value equals 0.

If the link partner is capable of autonegotiation (when lp_autoneg_cap is 1), the speed and mode information is displayed when you use autonegotiation and get the link partner capabilities.
Setting ge Driver Parameters

You can set the ge driver parameters in two ways, depending on your needs:

- Using the ndd utility
- Using the ge.conf file

Use the ndd utility to set parameters that are valid until you reboot the system. It is also a good way to test parameter settings.

Use the ge.conf file to set parameters so they remain in effect after you reboot the system. Create a /kernel/drv/ge.conf file and add parameter values to this file when you need to set a particular parameter for a device in the system.

Setting Parameters Using the ndd Utility

Use the ndd utility to configure parameters that are valid until you reboot the system. The ndd utility supports any networking driver that implements the Data Link Provider Interface (DLPI).

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

▼ To Specify the Device Instance for the ndd Utility

Before you use the ndd utility to get or set a parameter for a ge device, you must specify the device instance for the utility.
1. Check the /etc/path_to_inst file to identify the instance associated with a particular device.

For Sun GigabitEthernet/P:

```
# grep ge /etc/path_to_inst
"/pci@4,4000/network@4" 2 "ge"
"/pci@6,2000/network@1" 1 "ge"
"/pci@4,2000/network@1" 0 "ge"
```

In this example, the three GigabitEthernet instances are from the adapters installed in perspective PCI slots.

For Sun GigabitEthernet/S:

```
# grep ge /etc/path_to_inst
"/sbus@b,0/network@2,100000" 0 "ge"
```

In this example, the GigabitEthernet instance is from an adapter installed in a perspective SBus slot.

2. Use the instance number to select the device.

```
# ndd -set /dev/ge instance instance_number
```

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 the non-interactive mode, you invoke the utility to execute a specific command. Once the command is executed, you exit the utility. In the 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 Non-Interactive Mode

This section describes how to modify and to display parameter values.
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/ge` driver instance, and assigns it to the parameter:
   
   ```bash
   # ndd -set /dev/ge 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:
   
   ```bash
   # ndd /dev/ge parameter
   ```

▼ **To Use the `ndd` Utility in Interactive Mode**

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

   The `ndd` utility then prompts you for the name of the parameter:
   
   ```bash
   # ndd /dev/ge
   name to get/set? (Enter the parameter name or ? to view all parameters)
   ```

▼ **To View the `ge` Driver Parameters**

After entering the parameter name, the `ndd` utility prompts you for the parameter value (see TABLE 1-1 through TABLE 1-8).
1. To list all the parameters supported by the ge driver, type `ndd /dev/ge \?`. (See TABLE 1-1 through TABLE 1-8 for parameter descriptions.)

```
# ndd /dev/ge \?

?                             (read only)
link_status                   (read only)
link_speed                    (read only)
link_mode                     (read only)
ipg1                          (read and write)
ipg2                          (read and write)
instance                      (read and write)
lance_mode                    (read and write)
ipg0                          (read and write)
adv_1000autoneg_cap           (read and write)
adv_1000fdx_cap               (read and write)
adv_1000hdx_cap               (read and write)
adv_pauseTX                   (read and write)
adv_pauseRX                   (read and write)
1000autoneg_cap               (read only)
1000fdx_cap                   (read only)
1000hdx_cap                   (read only)
asn_dir_cap                   (read only)
pause_cap                     (read only)
lp_1000autoneg_cap            (read only)
lp_1000fdx_cap                (read only)
lp_1000hdx_cap                (read only)
lp_asm_dir_cap                (read only)
lp_pause_cap                  (read only)
#
```

**Autonegotiation Mode**

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

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

▼ **To Set Autonegotiation to Off (Forced Mode)**

1. Select the following parameters: `adv_1000fdx_cap`, `adv_1000hdx_cap`, `adv_pauseTX` and `adv_pauseRX`, and set those values according to the user's manual that shipped with your link partner device (for example, switch). See TABLE 1-6 for parameter values.
2. Set `adv_1000autoneg_cap` to 0.

Setting Parameters Using the `ge.conf` File

You can also specify the properties described in this section on a per-device basis by creating a `ge.conf` file in the `/kernel/drv` directory. Use a `ge.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 2.

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

Understanding the `ge.conf` File

1. Obtain the hardware path names for the `ge` 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 ge /etc/path_to_inst
"/sbus@b,0/network@2,100000" 0 "ge"
```

- where:
  - “/sbus@b,0/network@2,100000” specifies the hardware node name in the device tree.
  - “0” is the instance number.
  - “ge” is the driver name.
- In the device path name, the last component after the last `/` character and before the `@` character (network) is the device name.
- The path name before the last component (sbus@b,0) is the parent name.

For Sun GigabitEthernet/S:

To identify an SBus device unambiguously in the `ge.conf` file, use the name and parent name of the device. Refer to the `sbus(4)` man page for more information about the SBus device specification.

In the previous example:
- Name = SUNW,sbus-gem
- Class = "sbus"
Note – The “name” property in the `ge.conf` file should be the same value as the “compatible” property. In this case, the value is SUNW,sbus-gem.

For Sun GigabitEthernet/P:

To identify a PCI device unambiguously in the `ge.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 = pci108e,2bad
- Parent = /pci@4,4000
- Unit-address = 4

In the second line in the previous example:
- Name = pci108e,2bad
- Parent = /pci@6,2000
- Unit-address = 1

In the third line in the previous example:
- Name = pci108e,2bad
- Parent = /pci@4,2000
- Unit-address = 1

▼ To Set Parameters Using the `ge.conf` File on an SBus adapter

1. Set the `ipg1` and `ipg2` parameters for the above devices in the `kernel/drv/ge.conf` file.

```plaintext
name = "SUNW,sbus-gem" class = "sbus"
reg=0x2,0x100000,0x14,0x2,0x200000,0x9060 ipg1=20 ipg2=10 ;
```

2. Save the `ge.conf` file.

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

4. If your system doesn’t support DR, reboot by typing the `init 6` command at the superuser prompt.
To Set Parameters Using the `ge.conf` File on a PCI adapter

1. **Set the `ipg1` and `ipg2` parameters for the above devices in the**
   `kernel/drv/ge.conf` file.

   ```
   name="pci108e,2bad" parent="/pci@4,4000" unit-address="4" ipg1=20 ipg2=10;
   name="pci108e,2bad" parent="/pci@6,2000" unit-address="1" ipg1=20 ipg2=10;
   name="pci108e,2bad" parent="/pci@4,2000" unit-address="1" ipg1=20 ipg2=10;
   ```

2. **Save the `ge.conf` file.**

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

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

---

**Network Configuration**

This section describes how to configure the driver after it has been installed on your system.

**To Configure the Host Files**

After installing the Sun GigabitEthernet adapter driver software, you must create a file for the adapter's Ethernet interface. You must also create both an IP address and a host name for the Ethernet interface in the `/etc/hosts` file.
1. At the command line, use the `grep` command to search the `/etc/path_to_inst` file for `ge` interfaces.

   *For Sun GigabitEthernet/P:*
   
The following example shows the device instance from an adapter installed in slot 1.

   ```
   # grep ge /etc/path_to_inst
   
   "pci@1f,4000/network@1" 0 "ge"
   ```

   *For Sun GigabitEthernet/S:*
   
The following example shows the device instance from an adapter installed in slot 0.

   ```
   # grep ge /etc/path_to_inst
   
   "sbus@1f,0/network@1" 0 "ge"
   ```

2. Create an `/etc/hostname.ge<num>` file, where `num` is the instance number of the `ge` interface you plan to use.

   If you wanted to use the adapter’s `ge` interface in the Step 1 example, you would need to create a `/etc/hostname.ge0` file, where 0 is the number of the `ge` interface. If the instance number were 1, the file name would be `/etc/hostname.ge1`.

   - Do not create an `/etc/hostname.genum` file for a Sun GigabitEthernet adapter interface you plan to leave unused.
   - The `/etc/hostname.genum` file must contain the host name for the appropriate `ge` interface.
   - The host name should have an IP address and should 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.ge0` and `/etc/hostname.ge1` cannot share the same host name.

   The following example shows the `/etc/hostname.genum` file required for a system called `zardoz` that has a Sun GigabitEthernet adapter (`zardoz-11`).

   ```
   # cat /etc/hostname.ge0
   zardoz
   
   # cat /etc/hostname.ge1
   zardoz-11
   ```
3. **Create an appropriate entry in the `/etc/hosts` file for each active ge interface.**

   For example:

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

   **Note** – The Internet Protocol, version 6 (IPv6), expands the capabilities of IPv4, which is the current version and the default. The GigabitEthernet 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 8 System Administration Guide, Volume 3.

4. **If your system does not support Dynamic Reconfiguration (DR), reboot.**

---

**Installing the Solaris Operating Environment Over a GigabitEthernet Network**

The *Solaris Advanced Installation Guide* describes the full procedure for installing the Solaris operating environment over the network. The procedure below assumes that you have an install server, which contains the image of the Solaris Operating Environment, and that you have set up the client system to be upgraded over the network.

Before you can install the Solaris operating environment on a client system with a GigabitEthernet adapter, you must first add the GigabitEthernet software packages to the install server. These software packages can be found on Solaris CD.

**Note** – Refer to the *Solaris Advanced Installation Guide* for more information about installing the Solaris operating environment over the network.
To Install the Solaris Environment Over a GigabitEthernet Network

1. Prepare the install server and client system to install the Solaris operating environment over the network.

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

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

2. Find the root directory of the client system.

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

   ```
   # grep client_name /etc/bootparams
   client_name  root=server_name:/netinstall/Solaris_8/Tools/Boot
   install=server_name:/netinstall boottype=:in rootopts=:rsize=32768
   ```

   In the example above, the root directory for the client is /netinstall.

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

3. If the client system is not already displaying the OpenBoot (ok) prompt, shut down and halt the client system.

   Use the `shutdown(1M)` command to display the OpenBoot (ok) prompt.

   ```
   # shutdown -i0 -g0 -y
   ...
   (shutdown command messages omitted)
   ...
   ok
   ```
4. **At the ok prompt, use the `show-nets` command to find the device path of the GigabitEthernet device.**

   The `show-nets` command lists the system devices. You should see the full path name of the network device, similar to the examples below.

   *For Sun GigabitEthernet/P:*

   ```
   ok show-nets
   a) /pci@1f,0/pci0l/network03
   b) /pci@1f,0/pci01,1/network01,1
   q) NO SELECTION
   Enter Selection, q to quit:
   ```

   *For Sun GigabitEthernet/S:*

   ```
   ok show-nets
   a) /sbus@1f,0/network01,100000
   b) /sbus@1f,0/SUNW,hme@e,8c00000
   q) NO SELECTION
   Enter Selection, q to quit:
   ```

5. **At the ok prompt, boot the client system using the full device path of the GigabitEthernet device.**

   Use the full device path name of the network device, similar to the examples below.

   *For Sun GigabitEthernet/P:*

   ```
   ok boot /pci@1f,0/pci@1/network@3
   ```

   *For Sun GigabitEthernet/S:*

   ```
   ok boot /sbus@1f,0/network@1,100000
   ```

6. **Proceed with the Solaris operating environment installation.**

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

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

   Although the Solaris software installation creates the client’s hosts files, you may need to edit these files to match your specific networking environment. See “To Configure the Host Files” on page 14 for more information about editing these files.
Note – The Internet Protocol, version 6 (IPv6), expands the capabilities of IPv4, which is the current version and the default. The GigabitEthernet 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 8 System Administration Guide, Volume 3.