Platform Notes: The dmfe Fast Ethernet Device Driver
Contents

Preface vii

1. The dmfe Device Driver 1
   Hardware Overview 1
   Operating Speeds and Duplex Modes 1
   Auto-Negotiation 2

2. Setting Parameters Temporarily or Permanently 3
   Introduction 3
   The Order of Precedence for the Parameter Settings 4
   Setting Parameters Temporarily Using ndd 4
      Identifying Devices On the Command Line 5
      Displaying the Available Parameters Using ndd 5
      Viewing the Current Setting for a Parameter 5
      Setting a New Value for a Parameter 6
      Specifying a Particular Speed and Duplex Setting 6
      Returning to Auto-Negotiation Mode 7
   Setting Parameters Permanently Using the dmfe.conf File 8
      Using dmfe.conf To Set Parameters for All dmfe Devices in a System 8
      Using dmfe.conf to Set Parameters for a Particular Device 9
3. Forcing the Network Speed and Duplex Settings for OBP If You Are Booting From the Network  
   Forcing the Speed and Duplex Mode If You Are Booting From the Network  12  
   Booting From the Network  13  

A. Parameter Definitions  15  
   Driver Parameter Values and Definitions  15  
      Inspecting the Current Status of the Device  16  
      Operational Mode Parameters  16  

B. Application Programming Interface  19
Tables

TABLE 2-1  Using nedd to List the Parameters for the dmfe Driver  5
TABLE A-1  dmfe Driver Parameters  15
TABLE A-2  Read-Only Parameters That Report the Link Status, Speed, and Mode  16
TABLE A-3  Operational Mode Parameters  16
Preface

This book describes how to configure the dmfe (Fast Ethernet) device driver for the Davicom (R) DM9102A devices in the Netra™ X1 and Sun Fire™ V100 Servers.

How This Book Is Organized

Chapter 1 describes the dmfe device driver for the Davicom (R) DM9102A device.

Chapter 2 describes how to set the dmfe device driver parameter values temporarily by using the ndd utility or permanently by editing the dmfe.conf file.

Chapter 3 describes how to set parameters for the dmfe device to enable you to boot from the network if you have connected your Netra X1 or Sun Fire V100 server to an Ethernet hub that does not support auto-negotiation.

Appendix A lists the parameters for the dmfe device driver.

Appendix B describes the application programming interface for the dmfe driver.

Using UNIX Commands

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

See one or more of the following for this information:
- AnswerBook2™ online documentation for the Solaris™ software environment
- Other software documentation that you received with your system
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>

Shell Prompts

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

Accessing Sun Documentation Online

The docs.sun.comSM 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://www.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 (816-2128-11) of this document in the subject line of your email.
The dmfe Device Driver

The dmfe (Davicom Fast Ethernet) device driver handles the Davicom(R) DM9102A devices on the Netra X1 and Sun Fire V100 servers.

This chapter has the following sections:
- “Hardware Overview” on page 11
- “Operating Speeds and Duplex Modes” on page 11
- “Auto-Negotiation” on page 12

Hardware Overview

The DM9102A device provides a 100BASE-TX or 10BASE-T Ethernet network interface, including the PCI interface and Media Access Control (MAC) functions. The internal transceiver, which connects to an RJ-45 connector, provides the physical layer functions. The device can automatically set the link speed to 10 or 100 Mbps and it conforms to the 100BASE-T IEEE 802.3u Ethernet Standard.

Operating Speeds and Duplex Modes

You can operate the link in any of the following combinations of speed and duplex mode:
- 100 Mbps, full-duplex
- 100 Mbps, half-duplex
- 10 Mbps, full-duplex
- 10 Mbps, half-duplex
These speeds and duplex settings are described in the *Ethernet Standard*.

**Auto-Negotiation**

A key feature of the Davicom DM9102A Fast Ethernet device is auto-negotiation. The auto-negotiation protocol, as specified by the *Ethernet Standard* allows the device to determine automatically the appropriate speed and duplex settings. When the system is powered on, the device uses the auto-negotiation protocol to:

- Advertise its own speed and duplex capabilities to the hub or switch it is connected to (the "link partner"),
- Accept notification of the link partner’s capabilities,
- And to select the mode of operation that permits the highest possible data-throughput given the link partner’s capabilities:
  - 100 Mbps, full-duplex
  - 100 Mbps, half-duplex
  - 10 Mbps, full-duplex
  - 10 Mbps, half-duplex

If the DM9102A device is connected to a remote system or interface that supports auto-negotiation, your system automatically selects the correct link speed and duplex mode.

If you connect the DM9102A device to a hub or switch with which the auto-negotiation protocol fails to operate successfully, you can configure the DM9102A device not to use auto-negotiation. If you do this you need to specify the speed and duplex setting for it to use (see Chapter 2 and Chapter 3).
This chapter describes how to configure the dmfe driver parameters.

This chapter contains the following sections:

- “Introduction” on page 23
- “The Order of Precedence for the Parameter Settings” on page 24
- “Setting Parameters Temporarily Using ndd” on page 24
- “Setting Parameters Permanently Using the dmfe.conf File” on page 28

Introduction

You can set the dmfe device driver parameters temporarily (by using ndd) or permanently (by editing a dmfe.conf file).

To set parameters for:

- A specific dmfe device in the system and to give it settings that remain valid only until the next system reboot, use the ndd utility (see “Setting Parameters Temporarily Using ndd” on page 24). Using ndd is a good way to test parameter settings.

- A specific dmfe device or for all dmfe devices in the system and to ensure that the settings remain valid after subsequent reboots, use the dmfe.conf file in /platform/SUNW,UltraAX-i2/kernel/drv (see “Setting Parameters Permanently Using the dmfe.conf File” on page 28.)
The Order of Precedence for the Parameter Settings

The default setting for all the dmfe driver’s read/write parameters is 1. When adv_autoneg_cap is set to 1, all of the other parameter settings are ignored by the driver.

However, if you set adv_autoneg_cap to 0 (in other words, if you disable auto-negotiation), the driver selects the combination of speed and duplex mode that gives the highest possible throughput permitted by the other parameter settings.

So, if you enable 100Mbps full-duplex operation (adv_100fdx_cap=1), the driver selects this combination of speed and duplex mode in preference to 100Mbps half-duplex operation. Note that, in this case, it selects 100Mbps full-duplex operation even if the parameter permitting 100Mbps half-duplex mode is also enabled (adv_100hdx_cap=1) or if parameters permitting other lower levels of throughput are also enabled.

Similarly, if you disable 100Mbps full-duplex operation (adv_100fdx_cap=0) but enable 100Mbps half-duplex operation (adv_100hdx_cap=1), the driver will select 100Mbps half-duplex operation in preference to 10Mbps full-duplex operation (adv_10fdx_cap=1) or 10Mbps half-duplex operation (adv_10hdx_cap=1).

So, to force the driver to select the combination of speed and duplex mode that gives the lowest possible throughput (10Mbps half-duplex), you must not only set adv_10hdx_cap to 1 (its default setting), you must also set to zero all the parameters that would otherwise enable higher levels of throughput. These are: adv_100fdx_cap, adv_100hdx_cap, and adv_10fdx_cap.

Setting Parameters Temporarily Using ndd

The ndd utility supports any networking driver that implements the Data Link Provider Interface (DLPI).

Use the ndd utility to configure parameters for a specific dmfe device in the system with settings that you intend to remain valid only until the next system reboot.

This section tells you how you use the ndd utility to modify (with the -set option) or display (without the -set option) the parameters for each dmfe device.
Identifying Devices On the Command Line

When you use the ndd utility to get or set a parameter for a dmfe device, you must specify the device instance for the utility. The device instance is always /dev/dmfenum, where num is the instance number, for example, 0 or 1.

Displaying the Available Parameters Using ndd

1. To view the parameters you can set for a dmfe device, type:

   ```
   # ndd /dev/dmfenum '?'
   ```

   Where num will be 0 or 1.

   **TABLE 2-1** Using ndd to List the Parameters for the dmfe Driver

   | # ndd /dev/dmfe0 '?'
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
</tr>
<tr>
<td>link_status</td>
</tr>
<tr>
<td>link_speed</td>
</tr>
<tr>
<td>link_mode</td>
</tr>
<tr>
<td>adv_autoneg_cap</td>
</tr>
<tr>
<td>adv_100fdx_cap</td>
</tr>
<tr>
<td>adv_100hdx_cap</td>
</tr>
<tr>
<td>adv_10fx_cap</td>
</tr>
<tr>
<td>adv_10hdx_cap</td>
</tr>
</tbody>
</table>

   (For parameter descriptions, see Appendix A.)

Viewing the Current Setting for a Parameter

1. To display the current setting of a parameter, specify the parameter name but do not specify a new value. The syntax is as follows:

   ```
   # ndd /dev/dmfenum parameter
   ```

   For example, to check whether auto-negotiation is currently enabled, type:

   ```
   # ndd /dev/dmfenum adv_autoneg_cap
   ```
The result displayed will be either 0 (which means that auto-negotiation is disabled) or 1 (which means that auto-negotiation is enabled).

Setting a New Value for a Parameter

1. To modify a parameter value, use the -set option.
   The syntax is as follows:

   ```
   % ndd -set /dev/dmfe num parameter value
   ```

   When you invoke the ndd utility with the -set option, the utility passes the parameter name and value down to the named dmfe driver instance. The driver then checks the new value and, if it is valid, assigns it to the parameter.

Specifying a Particular Speed and Duplex Setting

**Note** – Note that the values you set by using ndd remain valid only until the next system reboot.

To specify the speed and duplex mode (that is, to force the device to use a particular speed and duplex mode instead of negotiating these variables with the device it is connected to), do the following:

1. **Set one of the following parameters to 1**: adv_100fdx_cap, adv_100hdx_cap, adv_10fdx_cap, or adv_10hdx_cap.

2. **Set the other three parameters to zero**.

3. **Set adv_autoneg_cap to zero**. This disables the device's auto-negotiation facility and causes the other new parameters to take effect.
Note – When you use ndd to set the speed/duplex-mode parameters, the dmfe driver does not examine the value of these parameters until you subsequently set adv_autoneg_cap. This is why you must set adv_autoneg_cap last. In fact, for the new parameter values to take effect, you must set adv_autoneg_cap last even if this means setting it to the value it is already set to.

For example, to specify a network speed of 100Mbps with half-duplex operation for device dmfe0, you would type the following:

```
# ndd -set /dev/dmfe0 adv_100fdx_cap 0
# ndd -set /dev/dmfe0 adv_100hdx_cap 1
# ndd -set /dev/dmfe0 adv_10fdx_cap 0
# ndd -set /dev/dmfe0 adv_10hdx_cap 0
# ndd -set /dev/dmfe0 adv_autoneg_cap 0
```

If you enable more than one of the speed/duplex mode parameters (that is, if you set more than one of the parameters to a value of 1), the driver selects the combination of speed and duplex mode that allows the highest throughput.

**Returning to Auto-Negotiation Mode**

1. To return the device to its default operating mode (that is, to cause it to negotiate a transfer speed and duplex mode automatically with the link partner), return all the parameters to their default value of 1.

To return to auto-negotiation mode, type the following:

```
# ndd -set /dev/dmfe0 adv_100fdx_cap 1
# ndd -set /dev/dmfe0 adv_100hdx_cap 1
# ndd -set /dev/dmfe0 adv_10fdx_cap 1
# ndd -set /dev/dmfe0 adv_10hdx_cap 1
# ndd -set /dev/dmfe0 adv_autoneg_cap 1
```

For an explanation of why we recommend you to return all the parameters to their default values of 1, see “The Order of Precedence for the Parameter Settings” on page 24.
Setting Parameters Permanently Using the \texttt{dmfe.conf} File

You can specify parameters for all \texttt{dmfe} devices in a system or for specific devices by editing the \texttt{dmfe.conf} file in the /platform/SUNW,UltraAX-i2/kernel/drv directory. These parameters take effect after the next reboot and remain in effect across subsequent reboots.

Using \texttt{dmfe.conf} To Set Parameters for All \texttt{dmfe} Devices in a System

To configure driver parameters for all \texttt{dmfe} devices in a system by using the \texttt{dmfe.conf} file, follow the instructions in this section.

For more information, refer to the manual pages for \texttt{prtconf (1M)}, \texttt{system (4)} and \texttt{driver.conf (4)}.

\textbf{Note} – When you edit the \texttt{dmfe.conf} file, make sure you do not modify any settings other than those of the parameters specifically described in this documentation.

1. Use a text editor to open the \texttt{dmfe.conf} file in the following directory:
   /platform/SUNW,UltraAX-i2/kernel/drv

2. Add a line for each parameter you want to set globally.

   To force all \texttt{dmfe} devices in a system to use a speed of 100Mbps in half-duplex mode, you would make the following two entries:

   \begin{verbatim}
   adv_autoneg_cap=0;
   adv_100fdx_cap=0;
   \end{verbatim}

   The driver’s default values for the speed/duplex-mode settings are all 1. This means that, in the example above, the driver will read the settings of 0 for \texttt{adv_autoneg_cap} (which disables auto-negotiation) and \texttt{adv_100fdx_cap} (which disables 100Mbps full-duplex operation). Then, because the next parameter in the order of precedence is \texttt{adv_100hdx_cap} (100Mbps half-duplex operation) and this parameter still has its default setting of 1, the driver will select 100Mbps with half-duplex operation.
For more information about the order in which the driver selects the combination of speed and duplex mode to use, see “The Order of Precedence for the Parameter Settings” on page 24.

If you wanted to force the device to use 10Mbps full-duplex mode (this is next in the driver’s order of precedence, after 100Mbps half-duplex mode) you would also add the following entry:

```
adv_100hdx_cap=0;
```

### Using `dmfe.conf` to Set Parameters for a Particular Device

1. **Obtain the hardware path name for the device in the device tree.**
   
   Typically this path name and the associated instance number will be found in the `/etc/path_to_inst` file. Sample entries in `/etc/path_to_inst` are:
   
   ```
   
   In the entries above:
   
   - The first part within the double quotes specifies the hardware node name in the device tree.
   - The path name before the last `/` character is the parent name.
   - The number after the last `@` character is the unit-address.
   - The number in the middle (not enclosed by double quotes) is the instance number.
   - The last part in double quotes is the driver name.
   
   You need to use these elements in Step 3 when you make your entries in the `dmfe.conf` file.

2. **Use a text editor to open the `dmfe.conf` file in the following directory:**

   `/platform/SUNW,UltraAX-i2/kernel/drv`

3. **Type an entry into the `dmfe.conf` file you have created.**

   To identify a PCI device whose parameters you intend to set use the parent name, unit-address, and driver name for the device.

   Refer to the `pci` man page for more information about PCI device specification.

   In the first line of the sample `/etc/path_to_inst` in Step 1 above:
- Parent name = /pci@1f,0
- Unit-address = c
- Driver name = dmfe

Therefore an example of a `dmfe.conf` file in which the network speed and duplex mode for `dmfe0` only are set to 100Mbps and half duplex respectively is as follows:

```plaintext
parent="/pci@1f,0" unit-address="c" name="dmfe" adv_autoneg_cap=0 adv_100fdx_cap=0;
```

The driver’s default values for the speed/duplex-mode settings are all 1. This means that, in the example above, the driver will read the settings of 0 for `adv_autoneg_cap` (which disables auto-negotiation) and `adv_100fdx_cap` (which disables 100Mbps full-duplex operation). Then, because the next parameter in the order of precedence is `adv_100hdx_cap` (100Mbps half-duplex operation) and this parameter still has its default setting of 1, the driver will select 100Mbps half-duplex operation.

For more information about the order in which the driver selects the combination of speed and duplex mode to use, see “The Order of Precedence for the Parameter Settings” on page 24.

If you wanted to force `dmfe0` to operate at 10Mbps in full-duplex mode you would add `adv_100hdx_cap=0` to the previous example. The entry in `dmfe.conf` would then look as follows:

```plaintext
parent="/pci@1f,0" unit-address="c" name="dmfe" adv_autoneg_cap=0 adv_100fdx_cap=0 adv_100hdx_cap=0;
```
Forcing the Network Speed and Duplex Settings for OBP If You Are Booting From the Network

This chapter tells you how to force the network speed and duplex mode for the \texttt{dmfe} device if you are booting from the network and have attached the Netra X1 or Sun Fire V100 server to an Ethernet hub that does not support auto-negotiation.

The settings you specify by following the instructions in this chapter affect OpenBoot PROM only. This means that they remain valid only during booting and do not affect the settings used when the Solaris operating environment has loaded.

For information about setting \texttt{dmfe} parameters that remain valid both during and after system reboots, see Chapter 2.

The chapter contains the following sections:

- “Forcing the Speed and Duplex Mode If You Are Booting From the Network” on page 312
- “Booting From the Network” on page 313

\textbf{Note} – The current driver software does not support booting from \texttt{net1}. Until a fix is available for this, please use \texttt{net0} for booting the server.
Forcing the Speed and Duplex Mode If You Are Booting From the Network

If you are booting the Netra X1 or Sun Fire V100 server from the network and you have connected it to an Ethernet hub that does not support auto-negotiation, then follow the instructions below to force the DM9102A device to use the same network speed and/or duplex mode as the hub.

**Note** – The instructions in this section affect OBP (OpenBoot PROM) only. When the Solaris operating environment loads, it does not inherit any dmfe driver settings from OBP.

1. Type:

```
ok nvedit
```

2. If you have not used nvedit before on the system you are configuring, the first line you edit will be numbered ‘0:’.

   (If you have used nvedit before and existing entries are displayed, press Control-N until you reach an empty line.)

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

```
0: apply transfer-speed=speed net num
1: apply duplex-mode=mode net num
```

where speed is either 10 or 100 (depending on the speed you want to specify), mode is either half or full, and num is either 0 or 1 (depending on which network port you are using).

For example, to force net0 to operate at 10Mbps in half-duplex mode, type:

```
0: apply transfer-speed=10 net0
1: apply duplex-mode=half net0
```

1. If you need to set the network speed and/or duplex mode for net1, repeat Step 3. However, remember that you cannot currently use net1 for booting the server (see the Note at the start of this chapter).
2. Press Control-C.

3. At the `ok` prompt type:

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

4. Reset your system by typing:

```
ok reset-all
```

---

**Booting From the Network**

1. To boot from the network, at the `ok` prompt type:

```
ok boot netnum
```

where `num` is either 0 or 1 depending on the label (`net0` or `net1`) of the DM9102A device you are using to boot (the label is visible close to the port on the back panel of the server). However, see the Note below.

**Note** – The current driver software does not support booting from `net1`. Until a fix is available for this, please use `net0` for booting the server.
Parameter Definitions

This appendix lists the parameters for the dmfe driver. For information about using these parameters, see Chapter 2.

Driver Parameter Values and Definitions

The following sections describe the dmfe driver parameters. The parameters are listed in the following table.

<table>
<thead>
<tr>
<th>TABLE A-1</th>
<th>dmfe Driver Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Configurability</td>
</tr>
<tr>
<td>link_status</td>
<td>Read only</td>
</tr>
<tr>
<td>link_speed</td>
<td>Read only</td>
</tr>
<tr>
<td>link_mode</td>
<td>Read only</td>
</tr>
<tr>
<td>adv_autoneg_cap</td>
<td>Read and write</td>
</tr>
<tr>
<td>adv_100fdx_cap</td>
<td>Read and write</td>
</tr>
<tr>
<td>adv_100hdx_cap</td>
<td>Read and write</td>
</tr>
<tr>
<td>adv_10fdx_cap</td>
<td>Read and write</td>
</tr>
<tr>
<td>adv_10hdx_cap</td>
<td>Read and write</td>
</tr>
</tbody>
</table>
Inspecting the Current Status of the Device

The read-only parameters described in the following table report the status, network speed, and duplex mode used by the device.

### TABLE A-2  Read-Only Parameters That Report the Link Status, Speed, and Mode

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Notes</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_status</td>
<td>Current link status</td>
<td>1 = Link up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Link down</td>
</tr>
<tr>
<td>link_speed</td>
<td>Valid only if the link is up</td>
<td>100 = 100 Mbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 = 10 Mbps</td>
</tr>
<tr>
<td>link_mode</td>
<td>Valid only if the link is up</td>
<td>1 = Full duplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Half duplex</td>
</tr>
</tbody>
</table>

Operational Mode Parameters

The following table describes the operational mode parameters and their default values.

### TABLE A-3  Operational Mode Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>adv_autoneg_cap</td>
<td>Local transceiver capability advertised by the hardware</td>
<td>1 = Auto-negotiation enabled (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Auto-negotiation disabled</td>
</tr>
<tr>
<td>adv_100fdx_cap</td>
<td>Local transceiver capability advertised by the hardware</td>
<td>1 = 100 Mbps full-duplex (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Not 100 Mbps full-duplex capable</td>
</tr>
</tbody>
</table>
**Note** – The priority (in descending order) for the: adv\_100fdx\_cap, adv\_100hdx\_cap, adv\_10fdx\_cap, and adv\_10hdx\_cap parameters is: adv\_100fdx\_cap, adv\_100hdx\_cap, adv\_10fdx\_cap, adv\_10hdx\_cap.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
</table>
| adv\_100hdx\_cap | Local transceiver capability advertised by the hardware | 1 = 100 Mbps half-duplex (default)  
                  |                                                                  | 0 = Not 100 Mbps half-duplex capable |
| adv\_10fdx\_cap | Local transceiver capability advertised by the hardware | 1 = 10 Mbit/sec full-duplex (default)  
                  |                                                                  | 0 = Not 10 Mbps full-duplex capable |
| adv\_10hdx\_cap | Local transceiver capability advertised by the hardware | 1 = 10 Mbps half-duplex (default)  
                  |                                                                  | 0 = Not 10 Mbps half-duplex capable |
Application Programming Interface

The cloning character special device /dev/dmfe can be used to access all Davicom DM9102A devices installed in the system. Specific instances can be accessed via the device /dev/dmfePPA (where PPA stands for Physical Point of Attachment and is effectively the device instance number).

The dmfe driver depends on /kernel/misc/gld, a loadable kernel module that provides the dmfe driver with the DLPI and STREAMS functionality required of a LAN driver. See gld(7D) for more details on the primitives supported by the driver.

Where you open the clone device /dev/dmfe, you must use an explicit DL_ATTACH_REQ message to associate the opened stream with a particular device (PPA). The PPA ID is interpreted as an unsigned integer data type and indicates the corresponding device instance (unit) number. The driver returns an error (DL_ERROR_ACK) if the PPA field value does not correspond to a valid device instance number for this system. The device is initialized on first attach and de-initialized (stopped) at last detach.

The values returned by the driver in the DL_INFO_ACK primitive in response to DL_INFO_REQ are shown below.

- The maximum SDU is 1500 (ETHERMTU, defined in <sys/ethernet.h>).
- The minimum SDU is 0.
- The DLSAP address length is 8.
- The MAC type is DL_ETHER.
- The sap length value is -2, meaning the physical address component is followed immediately by a 2-byte sap component within the DLSAP address.
- The broadcast address value is the Ethernet/IEEE broadcast address (FFFFFFFF).

Once in the DL_ATTACHED state, send a DL_BIND_REQ to associate a particular Service Access Point (SAP) with the stream.