Platform Notes: SunFDDI™ Adapters

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A

Part No.: 805-6783-10
October 1998, Revision A

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

Please
Recycle

Adobe PostScript

CHAPTER 1

Configuring the Driver Software for SunFDDI Adapters

This chapter includes information and instructions for configuring the driver software used by the SunFDDI adapter. Unless otherwise noted, all instructions apply to both the SunFDDI PCI adapter (`pf`) and the SunFDDI SBus adapter (`nf`).

Installing the Driver Software

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

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

Before using the SunFDDI adapter, you will need to create and edit system host files, as described in the next section

Configuring the Host File

After installing the SunFDDI driver software, you must create a `hostname` file for the adapter's interface. You must also create both an IP address and a host name for the interface in the `/etc/hosts` file.

▼ To Configure the Host File

- **At the command line, use the `grep` command to search the `/etc/path_to_inst` file for `pf` or `nf` devices.**

  For a **SunFDDI PCI adapter**:

```
# grep pf /etc/path_to_inst
"/pci@1f,2000/pci@2/pf@0" 0 "pf"
```

  In the example above, the `pf@0` instance shows a SunFDDI PCI adapter installed in slot 1.

  For a **SunFDDI SBus adapter**:

```
# grep nf /etc/path_to_inst
"/sbus@1f,0/SUNW,nf@1,7880" 1 "nf"
```

  In the example above, the `nf@1` instance shows a SunFDDI SBus adapter installed in slot 2.

- **Create an `/etc/hostname.pf`*num* or `hostname.nf`*num* file, where *num* is the instance number of the interface you plan to use.**
  - Do not create `/etc/hostname.nf`*num* or `hostname.pf`*num* files for SunFDDI adapter network interfaces you plan to leave unused.
  - The `/etc/hostname.nf`*num* or `hostname.pf`*num* file must contain the host name for the appropriate network interface.
  - The host name should have an IP address that will need to be entered in the `/etc/hosts` file.
  - The host name should be different from any other host name of any other interface, for example: `/etc/hostname.hme0` and `/etc/hostname.nf1` cannot share the same host name.

  Using the instance example in SBus Step l, the following example shows the `/etc/hostname.nf`*num* files required for a system called `zardoz` that has a SunFDDI adapter (`zardoz-11`).

```
# cat /etc/hostname.hme0
zardoz
# cat /etc/hostname.nf0
zardoz-11
```

1. **Create an appropriate entry in the /etc/hosts file for each active `nf` or `pf` 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
```

2. **Reboot your system.**

Booting From the Network

To use a SunFDDI adapter interface as the boot device, perform the following tasks:

▼ To Boot From the Network

- **At the `ok` prompt type:**

```
ok show-devs
```

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

For **SunFDDI PCI adapter**:

```
/pci@1f,2000/pci@2/pf@0,1
```

For **SunFDDI SBus adapter**:

```
/sbus@1f,0/nf@1,7880
```

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

- **At the `ok` prompt type:**

```
ok boot full_path_name_of_the_fddi_device
```

`local-mac-address` Property

Each of the network interfaces of the SunFDDI adapter has been assigned a unique Media Access Control (MAC) address, which represents the 48-bit Ethernet address for that network interface. The OpenBoot firmware reports this MAC address via the `local-mac-address` property in the device nodes corresponding to the network interfaces.

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

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

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

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

CHAPTER 2

Using the SunFDDI Network Utilities

This chapter describes the network utilities of SunFDDI. While some examples show only `pf` or `nf`, unless otherwise noted, all instructions apply to both the SunFDDI PCI adapter (`pf`) and the SunFDDI SBus adapter (`nf`).

Throughout this chapter, it is assumed that you have installed the SunFDDI software under the default base directory <*basedir*> for your operating system:

The default base directory <*basedir*> is:

```
/opt/SUNWconn/bin
```

Changing the Default MAC Address

Each attachment to an FDDI network is identified by a unique 48-bit `MAC` address. By default, the first SunFDDI card takes the host-resident `MAC` address, which is stored in nonvolatile memory (NVRAM) on the motherboard of the machine. Each subsequent SunFDDI card adopts the card-resident `MAC` address stored in its own IDPROM.

In general, this convention is sufficient to ensure that each SunFDDI card installed in the machine has a unique `MAC` address. However, there may be a conflict with other `LAN` interfaces that also take the host-resident `MAC` address—for example, an Ethernet (`le`) interface or a SunFDDI 2.0 (`bf`) interface. In this event, change the default `MAC` address assigned to the first SunFDDI card installed in the system.

▼ To Change the Default MAC Address with `pf_macid` or `nf_macid`

Use the `pf_macid(1M)` or the `nf_macid(1M)` utility to recover the card-resident `MAC` address, and then modify the system files to override the default `MAC` address:

1. **Become superuser.**
2. **Use the** *pf_macid(1M)* **or** *nf_macid(1M)* **utility to recover the MAC address from the IDPROM on the SunFDDI interface identified by the instance number** <*inst*>**.**

```
# <basedir>/nf_macid nf<inst>
<mac_address>
```

3. **Modify the start-up file on your machine so that the MAC address is assigned correctly when the system is rebooted.**

   a. **Edit the** `/etc/rcS.d/S30rootusr.sh` **file to add the following** `if` **statement immediately** *after* **the** *ifconfig* **command that initializes the interface** `nf`<*inst*>**.**

   If you are changing the MAC address of more than one interface, add one `if` statement for each interface.

```
ifconfig $1 plumb
    if [ $1 = "nf<inst>" ]; then
        ifconfig nf<inst> ether <mac_address>
    fi
```

   On most systems, the `/etc/rcS.d/S30rootusr.sh` file is a hard link to the `/etc/rootusr` file.

4. **Reboot your machine to assign the new MAC address to the SunFDDI interface.**

When a SunFDDI card takes the host-resident `MAC` address, it can be swapped to another system without affecting the existing network. However, once a station starts sending packets on the network, the Address Resolution Protocol (ARP) updates the ARP tables on other stations to include the `MAC` address of its interface. The ES-IS protocol performs the same function for SunFDDI OSI running over FDDI. If you swap SunFDDI cards that use the card-resident `MAC` address, you must wait until the ARP entries time-out, or remove the ARP entries from every active station manually before packets can be routed correctly.

Displaying SunFDDI Statistics

The *pf_stat(1M)*or *nf_stat* utility interrogates a specified SunFDDI interface and displays the accumulated statistics. This command must be executed as `root` and has the general form:

```
# <basedir>/pf_stat [-m] pf<inst> [<interval>] [<count>]
```

`pf`<*inst*> specifies the SunFDDI interface

<*interval*> is the elapsed time (in seconds) between interrogations

<*count*> the total number of interrogations

The pf_stat utility displays information using column headings that conform to SMT revision 7.3, which differ from SMT revision 5.1 and 4.2 headings in the following cases:

- The ECM heading corresponds to the 5.1 MIM heading.
- The RMT heading does not have an analog in SMT revision 4.2. If you run SunFDDI at revision level 4.2, ignore any data displayed under the RMT heading of pf_stat.

Displaying Local Interface Statistics

When you enter the pf_stat command without the -m option, it displays statistics recovered from the local interface `pf`<*inst*>.

For example, to display the accumulated statistics for the interface `pf0`, type:

```
# <basedir>/pf_stat pf0
RingECMRMTPCMSRing_OPXmitPRecvP
UP  INRING_OPACTIVEc16fde1862d
```

You can also monitor the interface dynamically (active monitor), by specifying the *interval* (the elapsed time between interrogations) and *count* (the total number of interrogations). This displays the incremental difference between the current state and the previous state. The minimum interval is one second and the accumulated statistics are displayed after every tenth interrogation.

For example, to monitor the interface `pf0` once every 60 seconds for 3 minutes (a total of 3 interrogations), type:

```
# <basedir>/pf_stat pf0 60 3
RingECMRMTPCMSRing_OPXmitPRecvP
UP  INRING_OPACTIVEc131a0131aa
UP  INRING_OPACTIVE000
UP  INRING_OPACTIVE011
```

Interpreting Local Statistics

Running the `pf_stat` utility *without* the `–m` option displays information about the various SMT state machines and the network to which the local station is attached:

`Ring` *(Ring Status)*

The `Ring` status shows the current state of the physical connection to the FDDI network. The following states may be returned by `pf_stat` under the `Ring` heading

- `UP`–SunFDDI interface physically connected to the active network
- `DOWN`–SunFDDI interface disconnected, or connected to the inactive network

ECM (Entity Coordination Management)

ECM shows the current state of the Entity Coordination Management state machine, which controls the following features and facilities:

- Media availability
- Trace
- Path Test
- Optical Bypass (optional)
- Hold Policy (optional)

TABLE 2-1 lists the states that may be returned by `pf_stat` under the ECM heading.

TABLE 2-1 `pf_stat` States Under the ECM Heading

State	Meaning
`OUT`	ECM is inactive and is waiting for a connect request (initial state).
`IN`	ECM is active; normal state after successful connection request.
TRACE	ECM is propagating a trace request to the appropriate entity.

TABLE 2-1 `pf_stat` States Under the ECM Heading

LEAVE	ECM is closing all connections prior to the station leaving the ring.
PATH_TEST	ECM is entering a path test state following trace completion.
INSERT	ECM is sending a request to the optical bypass switch to indicate that the station is entering the ring. This disengages the switch.
CHECK	ECM is verifying that symbols are being received from the network.
DEINSERT	ECM is sending a request to the optical bypass switch, to indicate that the station is leaving the ring. This engages the switch.

RMT (Ring Management)

`RMT` shows the current state of the Ring Management state machine, which controls the following features and facilities:

- MAC availability
- Detection and resolution of duplicate addresses
- Identification of stuck beacon and initiation of trace

TABLE 2-2 lists the states that may be returned by `pf_stat` under the RMT heading.

TABLE 2-2 `pf_stat` States Under the RMT Heading

State	Meaning
`ISOLATED`	RMT is inactive (initial state).
`NON_OP`	RMT is waiting for an operational ring.
RING_OP	RMT is operating normally.
DETECT	RMT is checking for duplicate addresses (transient state during initialization).
NON_OP_DUP	RMT has detected that its address is duplicated and is initiating recovery. The ring is not operational in this state.
RING_OP_DUP	RMT has detected that the `MAC` address is duplicated and flagged the error. The ring is operational in this state.
DIRECTED	RMT has been beaconing for an extended period of time and is transmitting a stream of directed beacons prior to initiating recovery.
TRACE	RMT has initiated a trace to recover a stuck beacon.

PCM (Physical Connection Management)

PCM shows the current state of the Physical Connection Management state machine that controls the following features and facilities:

- Connection initialization
- Maintenance support

This heading is modified to indicate the type of port that is being managed:

- PCMS: single-attached station, S-port
- PCMA: dual-attached station, A-port
- PCMB: dual-attached station, B-port

TABLE 2-3 lists the states that may be returned by `pf_stat` under the PCM heading.

TABLE 2-3 `pf_stat` States Under the PCM Heading

State	Meaning
OFF	PCM is inactive (initial state).
BREAK	PCM is starting the connection synchronization phase.
CONNECT	PCM is synchronizing the connection end-points prior to the signaling sequence.
NEXT	PCM is transmitting PDUs prior to entering SIGNAL state.
SIGNAL	PCM is transmitting and receiving signal bits (information) following a NEXT state.
JOIN	First state in the sequence leading to a synchronized connection.
VERIFY	Second state in the sequence leading to a synchronized connection.
ACTIVE	Final state indicating that the port is successfully incorporated in the token path.
TRACE	PCM is localizing a stuck beacon condition.

The normal sequence of PCM states leading to a fully synchronized connection and incorporation of the port into the token path is shown in FIGURE 2-1. Note that the minimum interval between interrogations is one second and that this is not always fast enough to recover and display the complete sequence of PCM states.

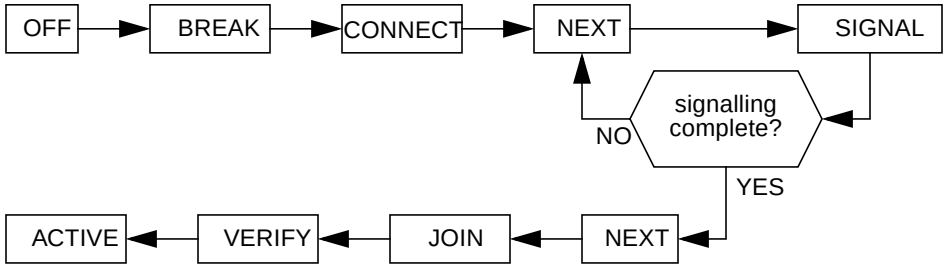

FIGURE 2-1 Normal Sequence of PCM States

Ring_OP *(Ring Operational)*

Ring_OP shows the number of Ring_OP (Ring Operational) signals received. This signal is generated when the station is incorporated into an operational network.

XmitP *(Transmit Packets)*

Running pf_stat *without* an interval and count, displays the total number of packets transmitted since the interface was activated. Running pf_stat *with* an interval and count, displays the number of packets transmitted since the last interrogation.

RecvP *(Receive Packets)*

Running pf_stat *without* an interval and count displays the total number of packets received since the interface was activated. Running pf_stat *with* an interval and count displays the number of packets received since the last interrogation.

Example Local Statistics

The following output was recovered from a single-attached station using the command shown. A temporary fault condition was simulated by disconnecting the FDDI cable from the SunFDDI card and then reconnecting it.

```
# <basedir>/pf_stat pf0 1 20
RingECMRMTPCMSRing_OPXmitPRecvP
UP  INRING_OPACTIVE2261d
UP  INRING_OPACTIVE000
UP  INRING_OPACTIVE000
UP  INRING_OPACTIVE000
DOWNINISOLATEDCONNECT011
DOWNINISOLATEDCONNECT000
DOWNINISOLATEDNEXT000
UP  INRING_OPACTIVE100
UP  INRING_OPACTIVE011

RingECMRMTPCMSRing_OPXmitPRecvP
UP  INRING_OPACTIVE32920
UP  INRING_OPACTIVE000
UP  INRING_OPACTIVE011
UP  INRING_OPACTIVE000
UP  INRING_OPACTIVE000
UP  INRING_OPACTIVE000
UP  INRING_OPACTIVE000
UP  INRING_OPACTIVE000
UP  INRING_OPACTIVE000
UP  INRING_OPACTIVE011
```

Note the following observations regarding this example:

- Accumulated statistics are displayed automatically after every tenth interrogation.
- The combination of `Ring=DOWN` and `RMT=ISOLATED` indicate that the station is disconnected from the network.
- The minimum interval of one second is not fast enough to recover and display the complete sequence of `PCM` states during the path re-establishment phase.
- A `Ring_OP` signal is received when the path is re-established indicating that the ring is operational.

The link status indicator mounted on the SunFDDI card displays the following sequence of events:

Green (connected) –> **Amber** (disconnected) —> **Green** (connected)

Displaying Statistics from Neighboring Stations

When you use the pf_stat or nf_stat command with the -m option, it displays information about the neighboring stations attached to the local interface `pf`<*inst*> and the frames received from the network.

For example, to display information about the neighboring stations attached to the interface `pf0`, type:

```
# <basedir>/pf_stat –m pf0
PhySFrameErrorLostSAUNADNA
M   b43eb203<mac_addr1><mac_addr2><mac_addr3>
```

You can also monitor the neighboring stations dynamically (active monitor), by specifying the *interval* (the elapsed time in seconds between interrogations) and *count* (the total number of interrogations). The minimum interval is one second and the accumulated statistics are displayed after every tenth interrogation.

For example, to monitor the stations attached to `pf0` once every 10 seconds for 1 minute (a total of 6 interrogations), type:

```
# <basedir>/pf_stat –m pf0 10 6
PhySFrameErrorLostSAUNADNA
M   c460a6d03<mac_addr1><mac_addr2><mac_addr3>
M   2722400 <mac_addr1><mac_addr2><mac_addr3>
M   2722700 <mac_addr1><mac_addr2><mac_addr3>
M   2722000 <mac_addr1><mac_addr2><mac_addr3>
M   2722e00 <mac_addr1><mac_addr2><mac_addr3>
M   2722300 <mac_addr1><mac_addr2><mac_addr3>
```

Interpreting Statistics from Neighboring Stations

Running the pf_stat utility with the -m option displays information about the neighboring stations attached to the local interface `pf`<*inst*>.

Phy (Physical Connection)

PHY shows the type of physical connection to the FDDI network. This heading is modified to indicate the type of port being managed:

- PhyS: single-attached station, port S
- PhyA: dual-attached station, port A
- PhyB: dual-attached station, port B

The following states may be returned by pf_stat under the Phy heading:

TABLE 2-4 pf_stat States Under the PHY Heading

State	Meaning
NONE	Port disconnected
M	Port connected to Port M on a concentrator
S	Port connected to Port S on a single-attached station
A	Port connected to Port A on a dual-attached station
B	Port connected to Port B on a dual-attached station

Frame (Frames Received)

Running pf_stat or nf_stat *without* an interval and count displays the total number of SMT frames received since the interface was activated. Running pf_stat or nf_stat *with* an interval and count displays the number of SMT frames received since the last interrogation.

More detailed information about the SMT frames can be recovered using the pf_smtmon*(1M)* or *nf_smtmon(1M)* utility described in "Monitoring SMT Frames" on page 18.

Error (Error Frames)

Running pf_stat or nf_stat *without* an interval and count displays the total number of error frames received since the interface was activated. Running pf_stat or nf_stat *with* an interval and count displays the number of error frames received since the last interrogation. An error frame is defined as an SMT frame whose E (error) bit is set, and whose E bit is first detected by the local station. It does not indicate the location of the cause of the error. Frequent error frames can indicate a noise problem on the network, either dirt (optical fiber) or electrical interference (UTP).

Lost (Lost Frames)

Running pf_stat or nf_stat *without* an interval and count displays the total number of lost frames since the interface was activated. Running pf_stat or nf_stat *with* an interval and count displays the number of lost frames since the last interrogation. A lost frame is defined as an SMT frame whose reception is

aborted by the `local` station. It does not indicate the location of the cause of the error. A large number of `lost` frames can indicate a noise problem on the network, either dirt (optical fiber) or electrical interference (UTP).

SA (Station Address)

Displays the `MAC` address for the local station.

UNA (Upstream Neighbor Address)

Displays the `MAC` address for the neighboring station, connected upstream on the ring from the local station.

DNA (Downstream Neighbor Address)

Displays the `MAC` address for the neighboring station, connected downstream on the ring from the local station.

Example Neighbor Statistics

The following output was recovered from a single-attached station using the command shown. A temporary fault condition was simulated by disconnecting the FDDI cable from the SunFDDI card and then reconnecting it.

```
# <basedir>/pf_stat –m pf0 1 20
PhySFrameErrorLostSAUNADNA
    Mc45d546311b<mac_addr1><mac_addr2><mac_addr3>
    M274370 0<mac_addr1><mac_addr2><mac_addr3>
    M274270 0<mac_addr1><mac_addr2><mac_addr3>
    M274350 0<mac_addr1><mac_addr2><mac_addr3>
    NONE182f100<mac_addr1><mac_addr2><mac_addr3>
    NONE00  0<mac_addr1><mac_addr2><mac_addr3>
    NONE00  0<mac_addr1><mac_addr2><mac_addr3>
    Md4320  7<mac_addr1><mac_addr2><mac_addr3>
    M2707e0 0<mac_addr1><mac_addr2><mac_addr3>

PhySFrameErrorLostSAUNADNA
    Mc46e5ce7122<mac_addr1><mac_addr2><mac_addr3>
    M272280 0<mac_addr1><mac_addr2><mac_addr3>
    M272300 0<mac_addr1><mac_addr2><mac_addr3>
    M272270 0<mac_addr1><mac_addr2><mac_addr3>
    M2722e0 0<mac_addr1><mac_addr2><mac_addr3>
    M2722c0 0<mac_addr1><mac_addr2><mac_addr3>
    M272280 0<mac_addr1><mac_addr2><mac_addr3>
    M272310 0<mac_addr1><mac_addr2><mac_addr3>
    M2722b0 0<mac_addr1><mac_addr2><mac_addr3>
    M272270 0<mac_addr1><mac_addr2><mac_addr3>
```

Note the following observations regarding this example:

- Accumulated statistics are displayed automatically after every tenth interrogation.
- The combination of `PhyS=NONE` and the loss of frame activity indicates that the station is disconnected from the network.

Monitoring SMT Frames

The *pf_smtmon(1M)* or *nf_smtmon(1M)* utility is an active monitor that displays the `SMT` frames received by the local station. It is particularly useful for diagnosing communication problems with the SunNet Manager proxy agent.

This command must be executed as root (or superuser) and has the general form:

```
# <basedir>/pf_smtmon [-i pf<inst>] [–x] [–h] [<frameclass>]
```

-i pf<*inst*> specifies the SunFDDI interface

-x displays the received SMT frames in hexadecimal

-h displays help information, including a list of valid frame classes

<*frameclass*> specifies one or more SMT frame classes (used to filter output)

If you do not specify an interface, *pf_smtmon(1M)* or *nf_smtmon(1M)* returns the SMT frames received by pf0. If you do not specify a frame type, pf_smtmon displays all the SMT frames that it receives. Use Ctrl-C to stop *pf_smtmon(1M)* or *nf_smtmon(1M)*.

To display the encoded SMT frames received by interface pf1, type:

```
# <basedir>/pf_smtmon -i pf1
pf1: nif_request  v=0x1 t=0xfc03e781 s=10-0-4-48-6f-a5 i=0x28
pf1: nif_response v=0x1 t=0xfc03e781 s=10-0-4-8-24-5c  i=0x28
pf1: nif_request  v=0x1 t=0xfc00dec6 s=10-0-4-b8-6e-ab i=0x28
pf1: nif_request  v=0x1 t=0xfc03e787 s=10-0-4-48-6f-a5 i=0x28
pf1: nif_response v=0x1 t=0xfc03e787 s=10-0-4-8-24-5c  i=0x28
```

TABLE 2-5 The elements of the SMT frames are defined as follows:

TABLE 2-6 Elements of the SMT Frames

Element	Definition
class_type	Identifies the SMT frame class and type (see page 21)
v	Version ID; identifies the structure of the SMT information field
t	Transaction ID; used to pair SMT response and request frames
s	Station ID; uniquely identifies the station transmitting the frame
i	InfoField Length; defines the length of the SMT information field

- **To display the SMT frames received by interface `pf1` in hexadecimal format, type:**

```
# <basedir>/pf_smtmon -i pf1 –x
pf1: nif_request v=0x1 t=0x170 s=10-0-4-8-24-5c i=0x28
004DC000 0000004F FFFFFFFF FFFF1000 0408245C 01020001 00000170 00001000
0408245C 00000028 00010008 00001000 04B86EAB 00020004 00010100 00030004
00002100 200B0008 00000001 00000001 76C467A0

pf1: nif_request v=0x1 t=0x5e0f s=10-0-d4-78-42-4d i=0x28
004D0000 0000004F FFFFFFFF FFFF1000 D478424D 01020001 00005E0F 00001000
D478424D 00000028 00010008 00001000 0408245C 00020004 01010208 00030004
00001200 200B0008 0000000B 00000002 A522BBA1

pf1: nif_response v=0x1 t=0xfc00d94a s=10-0-4-8-24-5c i=0x28
004D0000 00000041 100004B8 6EAB1000 0408245C 01030001 FC00D94A 00001000
0408245C 00000028 00010008 00001000 04B86EAB 00020004 00010100 00030004
00002100 200B0008 00000001 00000001 865549E2 0049C020 F0154E4F FFFFFFFF
FFFF1000 04B86EAB 01020001 FC00D94A 00001000 04B86EAB 00000028 00010008
00001000 D478424D 00020004 00010100 00030004 00002000 200B0008 00000001
00000001

pf1: nif_request v=0x1 t=0x5e13 s=10-0-d4-78-42-4d i=0x28
004D0000 0000004F FFFFFFFF FFFF1000 D478424D 01020001 00005E13 00001000
D478424D 00000028 00010008 00001000 0408245C 00020004 01010208 00030004
00001200 200B0008 0000000B 00000002 4AD75A79

pf1: nif_request v=0x1 t=0x5e17 s=10-0-d4-78-42-4d i=0x28
004D0000 0000004F FFFFFFFF FFFF1000 D478424D 01020001 00005E17 00001000
D478424D 00000028 00010008 00001000 0408245C 00020004 01010208 00030004
00001200 200B0008 0000000B 00000002 DCEBADA2

pf1: nif_request v=0x1 t=0x171 s=10-0-4-8-24-5c i=0x28
004DC000 0000004F FFFFFFFF FFFF1000 0408245C 01020001 00000171 00001000
0408245C 00000028 00010008 00001000 04B86EAB 00020004 00010100 00030004
00002100 200B0008 00000001 00000001 127B1D3B

pf1: nif_request v=0x1 t=0x5e1b s=10-0-d4-78-42-4d i=0x28
004D0000 0000004F FFFFFFFF FFFF1000 D478424D 01020001 00005E1B 00001000
D478424D 00000028 00010008 00001000 0408245C 00020004 01010208 00030004
00001200 200B0008 0000000B 00000002 626FA878
```

SMT Frame Classes and Types

SMT frames are used for peer-to-peer (station-to-station) management. They are divided into *classes*, which define the function of the frame. Each class is then divided into up to three *types*, which define whether the frame is an announcement (information only), a request for service, or a response to a request. Refer to the *ANSI/FDDI Station Management (SMT) X3.299 R7.3 Specification* for a detailed description of SMT frames and their functions.

The *pf_smtmon(1M)* or *nf_smtmon(1M)* utility is used to monitor the following SMT frame classes:

NIF (Neighbor Information Frames)

These are the most common frames displayed when you run *pf_smtmon(1M)* or *nf_smtmon(1M)*. As the name suggests, they carry information about a neighboring station (for example, address, description, state, MAC capabilities) and are used as keep-alive notifications that a station is still attached to the ring and functioning. An NIF frame can be an announcement, a request, or a response.

SIF (Status Information Frames)

These frames carry more detailed information about a station. SIF configuration frames describe the station configuration (for example, number of ports, number of MAC entities, connection policy). SIF operation frames describe the current status of the station. A SIF frame can be either a request or a response.

ECF (Echo Frames)

These frames are equivalent to ICMP ping packets and are used to test connectivity between stations. An ECF frame can be either a request or a response.

RDF (Request Denied Frame)

These frames are used to indicate that the request is rejected. If an SMT agent (such as the SunNet Manager proxy agent delivered with SunFDDI) receives an unsupported or unrecognized request, it issues an RDF frame to indicate that the request is rejected. An RDF frame is always a response.

ESF (Extended Service Frame)

These frames are implementation dependent. An ESF frame can be an announcement, a request, or a response.

PMF (Parameter Management Frame)

These frames are used to access remote station attributes. The Parameter Management Protocol supports both get (display) and set (modify) functions. However, the *pf_smtmon(1M)* or *nf_smtmon(1M)* utility can display only PMF_get frames. A PMF_get frame can be either a request or a response.

Filtering Output from pf_smtmon

By default, *pf_smtmon(1M)* or *nf_smtmon(1M)* displays all of the SMT frames received by the local station. You can filter the output generated by *pf_smtmon(1M)* or *nf_smtmon(1M)* by specifying one or more frame classes on the command-line: nif, sif_config, sif_operat, ecf, rdf, esf, pmf_get.

For example:

- To display only the RDF frames received by interface pf0, type:

```
# <basedir>/pf_smtmon rdf
```

- To display the SIF configuration and SIF operation frames received by interface pf1, type:

```
# <basedir>/pf_smtmon –i pf1 sif_config sif_operat
```

- To display the NIF frames, SIF configuration frames, and SIF operation frames received by interface pf1, type:

```
# <basedir>/pf_smtmon –i pf1 nif sif_config sif_operat
```