Sun Enterprise Server Alternate Pathing Reference Manual
Contents

Intro(1M) 2
ap(1M) 3
apboot(1M) 4
apcheck(1M) 6
apconfig(1M) 7
ap_daemon(1M) 11
apdb(1M) 12
apdisk(1M) 14
apinst(1M) 17
apnet(1M) 18
ap_reboot_host(1M) 21
apssp(1M) 22
ap_ssp_daemon(1M) 23
Intro(7) 25
ap_dmd(7) 26
ap_nmd(7) 28
ap_nmd(7) 30
ap_nmd(7) 32
ap_nmd(7) 34
Maintenance Commands
### NAME
Intro – AP administration

### DESCRIPTION
This section describes commands, scripts, and programs executed in the Alternate Pathing environment.

### LIST OF COMMANDS
- **ap**(1M) alternate pathing
- **ap_daemon**(1M) alternate pathing daemon
- **ap_reboot_host**(1M) fast boot alternate path
- **ap_ssp_daemon**(1M) AP SSP daemon
- **apboot**(1M) set up system files for boot metadisk
- **apcheck**(1M) determine accessibility of a metadisk
- **apconfig**(1M) display and manage AP configuration
- **apdb**(1M) manage AP database
- **apdisk**(1M) manage disk pathgroups
- **apinst**(1M) identify disk host adapter instances, /dev/dsk targets
- **apnet**(1M) manage network pathgroups
- **apssp**(1M) client of AP SSP daemon
NAME      ap – alternate pathing

DESCRIPTION Alternate Pathing (AP) enables you to define and control alternate physical paths to peripheral devices. If a path to a device becomes unavailable, your Sun server can use an alternate path.

SEE ALSO Sun Enterprise Server Alternate Pathing User’s Guide
NAME
apboot – set up system files for boot metadisk

SYNOPSIS
apboot [-n] [-k system-name] [-v vfstab-name] device
apboot [-m metadevice-name]
apboot [-u metadevice-name]

DESCRIPTION
Use /usr/sbin/apboot to edit /etc/vfstab and /etc/system to make
the system bootable from either the boot disk file systems on an AP metadisk
or the boot disk file systems on a disk device that is not alternately pathed.
The apboot(1M) command enables AP to manage a mirrored boot device when
both that boot device and its mirror are under AP control.

In addition to editing /etc/vfstab/ and /etc/system, apboot checks the
current configuration of system swap and dump devices. If either is configured
as a partition of the boot disk, apboot calls swap(1M) or dump(1M), as
appropriate, to ensure that swap and dump devices are consistent with the
boot device.

OPTIONS
The following options are supported:

- n
  Print what would be done without actually doing it.

- k system-name
  Edit system-name instead of the default /etc/system file.

- v vfstab-name
  Edit vfstab-name instead of the default /etc/vfstab table of file system
defaults.

- m metadevice-name
  Enable boot mirror support for the specified AP metadevice.

- u metadevice-name
  Disable boot mirror support for the specified AP metadevice.

EXAMPLES
EXAMPLE 1 Using apboot with Metadisks
The following command edits /etc/system and /etc/vfstab to specify
that the boot-disk file systems are now on metadisk mc3t0d0.
EXAMPLE 2  Using apboot with Physical Devices

The following command edits /etc/system and /etc/vfstab to specify that the boot-disk file systems are now under the physical path /dev/dsk/c3t0d0.

    apboot c3t0d0

EXAMPLE 3  Using apboot with Mirrored Devices

The following command edit /etc/system and /etc/vfstab to specify that the boot disk file systems are now on metadisk mc3t0d0, with a mirror on mc1t0d1.

    apboot -m mc1t0d1

EXAMPLE 4  Using apboot to Disable Mirrored Devices

The following command disables AP support for the mirror device created in the previous example.

    apboot -u mc1t0d1

FILES

The following files are used by this utility:

/etc/system    Kernel patch file
/etc/vfstab    Table of file system defaults

SEE ALSO
dumpadm(1M), swap(1M), system(4), vfstab(4) in the man Pages(4): File Formats of the SunOS Reference Manual
### NAME

`apcheck` – determine accessibility of a metadisk

### SYNOPSIS

```
apcheck special
```

### DESCRIPTION

`/sbin/apcheck` ascertains whether a metadisk is usable. If it is able to locate dual paths, `apcheck` exits with a zero status; if not, it exits with a non-zero status.

### CAUTION

Do not execute `apcheck` on the command line; it is intended for use only by other commands or by authorized service providers.

### OPTIONS

The following options are supported:

- **special**
  
  This option represents the device node to be checked. This device node may reside under `/dev/ap/dsk` or `/dev/ap/rdsk`.

---

SunOS 5.7 Last modified May 1999
NAME        apconfig – display and manage AP configuration

SYNOPSIS    apconfig
            apconfig -D
            apconfig -F
            apconfig -N [-u]
            apconfig -P meta_network -a new_physical_path
            apconfig -P meta_disk -a new_physical_path
            apconfig -R
            apconfig -S [-u]

DESCRIPTION The /usr/sbin/apconfig command displays and helps you manage the
Alternate Pathing (AP) system configuration.

OPTIONS     The following options are supported:
            −D

            Display location and status information for all known copies of the host
database.

            −F

            Force the state (attached or detached) of every committed pathgroup
alternate to match the physical state of the system. Use this option if the two
states differ. It refreshes the Dynamic Reconfiguration (DR) flags for every
disk I/O port and physical network interface defined for all committed
pathgroups.

            −N [-u]

            Display network AP information only. For each pathgroup, apconfig -N
displays the metanetwork interface and the corresponding physical network
interfaces.

            If you specify the −u option, apconfig displays uncommitted pathgroup
information only. If you do not specify the −u option, apconfig displays
committed pathgroup information only. See "Character flags after meta
device names" and "Character flags after physical device paths", below.

            −P meta_network −a new_physical_path
Switch to the new physical path specified by -a for the metanetwork specified by -P.

`-P meta_disk_primary_path-a new_physical_path`

Switch to the new physical path specified by -a for the metadisk associated with the primary path specified by -P.

`-R`

Rebuild the metadisk device nodes in /dev/ap/dsk and /dev/ap/rdsk. The `apconfig` command creates links to /devices for all committed disk pathgroups in the database.

**Note** - You must execute `drvconfig -i ap_dmd` before you can execute `apconfig -R`. See `drvconfig(1M)` and `ap_dmd(7)`.

`-S [−u]`

Display alternate pathing information for disk pathgroups only. For each pathgroup, `apconfig` shows the names for the metadisk, its physical devices, and the disk I/O ports through which each physical device is accessed.

If you specify the -u option, `apconfig` displays only uncommitted pathgroup information. Otherwise, it displays only committed pathgroup information. See "Character flags after meta device names" and "Character flags after physical device paths", below.

### Character flags after meta device names

When you specify -N or -S, one or more of the following letters may be displayed after each meta-network or meta-disk name:

- **D** Marked for deletion. The metadisk or metanetwork remains in the database and continues to be used by AP until a commit is done. See `apdb(1M)`.
- **U** Uncommitted. Note that you cannot use a metadisk or metanetwork until a commit has been done.
- **R** Marked for use as a root device (−S only).
- **M** Marked as the mirror for a boot device (−S only).
The physical paths for this metadisk lead to different disks—that is, different SSAs (−S only).

When you specify −N or −S, one or more of the following letters may be displayed after each physical network path or physical disk I/O port path:

N  Automatic switching is not allowed for this physical device.

X  The physical paths for this metadisk lead to different disks (that is, different SSA’s). −S only.

A  The active alternate (to select another interface, use the −P and −a options)

DR  Marked as being drained by the DR daemon. A switch cannot be made to a device path in this state. See the Sun Enterprise Server Alternate Pathing User’s Guide.

DE  Marked as detached by the DR daemon.

P  The primary path (the primary path cannot be changed)

T  Path has been tried as active.

O  Marked as offline. See apdisk(1M) and apnet(1M).

EXAMPLES

EXAMPLE 1  Displaying Committed Disk Pathgroups

The following command displays all of the committed disk pathgroups in the AP database.

```
# apconfig −S
C6   pln0  A
C2   pln3  P
    metadiskname(s):
    mc2t5d0
    mc2t4d0  R
    mc2t3d0
    mc2t2d0
    mc2t1d0
    mc2t0d0
```

EXAMPLE 2  Displaying Uncommitted Network Pathgroups

The following example displays all of the uncommitted network pathgroups in the AP database:

```
# apconfig −N −u
metanetwork:  mqe0  U
```
physical devices:
qe1
qe0  P A

EXAMPLE 3  Switching the Active Pathgroup

The following example switches the active alternate of the disk pathgroup for
which the primary path is pln1. The new active alternate of that pathgroup is
pln0.

```bash
# apconfig -P pln1 -a pln0
```

EXAMPLE 4  Switching the Network Pathgroup

The following example switches the active alternate of the network pathgroup
identified by the metanetwork interface mqe0. The new active alternate of that
network pathgroup is qe1.

```bash
# apconfig -P mqe0 -a qe1
```

EXAMPLE 5  Displaying AP Database Information and Location

The following example displays the location and status information of all
known copies of the AP database.

```bash
# apconfig -D
path: /dev/rdsk/c3t3d0s1
  major: 32
  minor: 145
  timestamp: Wed Sep 28 18:45:58 1994
  checksum: 2636010350
  default: yes
  corrupt: no
  inaccessible: no

path: /dev/rdsk/c3t3d0s6
  major: 32
  minor: 150
  checksum: 2636010350
  default: no
  synced: yes
  corrupt: no
  inaccessible: no
```

SEE ALSO

Sun Enterprise Server Alternate Pathing User’s Guide

apdb(1M), apdisk(1M), apnet(1M), ap_dmd(7) in this reference manual
drvconfig(1M) in the man Pages(1M): System Administration Commands of
SunOS Reference Manual
NAME      ap_daemon – alternate pathing daemon
SYNOPSIS  ap_daemon
DESCRIPTION The /usr/sbin/ap_daemon is an RPC program that provides the interface to the Alternate Pathing (AP) driver.

Configuration Information
The ap_daemon RPC program name is AP_SVR; its RPC program number is 300473; and, its underlying protocol is TCP. It is invoked as an inetd server by using the TCP transport. The UID required for access to the daemon is ssp. This UID can be a non-login UID.

The entry for the daemon in the /etc/inetd.conf file is:
300473/1 tli rpc/tcp wait root /usr/sbin/ap_daemon ap_daemon

SEE ALSO Sun Enterprise Server Alternate Pathing User’s Guide
apconfig(1M), apdb(1M), apdisk(1M), apnet(1M)
NAME
apdb – manage AP database

SYNOPSIS
apdb -c raw_disk_slice [-k system_file] [-f]
apdb -d raw_disk_slice [-k system_file] [-f]
apdb -m major -n minor [-f]
apdb -C

DESCRIPTION
The /usr/sbin/apdb command helps you manage the AP database.

OPTIONS
The following options are supported:

- c raw_disk_slice  Create a database copy on the specified raw disk slice. You can create up to ten copies of the database. The minimum slice size is 300-KBytes.

- d raw_disk_slice  Delete a database copy from the specified raw disk slice.

- f                  Force the deletion of the specified database. This option is required for creating the first copy of the database and for deleting each of the last two copies of the database. If you try to delete a database copy without this option when fewer than two database copies exist, AP displays an error message.

- k system_file      Patch the database copy information to the kernel file system_file, rather than the default file, /etc/system.

- m major -n minor   Remove a database copy by specifying its location as a major-minor pair. Use -m to specify the major and -n for the minor. This option pair is useful when there is no path to the database because the device no longer exists.

- C                  Commit all uncommitted entries within the database.

- Z                  Copy the database in memory to all database copies. Note that all database copies are in sync with memory and are automatically updated at
system shutdown. The -Z option lets you update the database copies at your discretion.

EXAMPLES

**EXAMPLE 1** Creating an AP Database Copy

The following command creates a copy of the AP system database on /dev/rdsk/c2t0d0s1.

```
# apdb -c /dev/rdsk/c2t0d0s1
```

SEE ALSO

Sun Enterprise Server Alternate Pathing User’s Guide

apconfig(1M), apdisk(1M), apnet(1M)
NAME | apdisk – manage disk pathgroups

SYNOPSIS | apdisk -c -p primary_path -a alternate_path

| apdisk d primary_path
| apdisk -z primary_path
| apdisk -f io_controller_path
| apdisk -n io_controller_path
| apdisk -u -p primary_path -a alternate_path
| apdisk -w io_controller_path

DESCRIPTION | The /usr/sbin/apdisk command helps you manage disk pathgroups in the Alternate Pathing (AP) system.

OPTIONS | The following options are supported:

- -c -p primary_path -a alternate_path

Create database entries for disk arrays connected to two I/O ports. Give the I/O port names (for example, sf0 and sf1) as the primary_path and alternate_path.

- d primary_path

Delete AP information for the specified disk pathgroup. If the existing information is uncommitted, apdisk removes it immediately. If the existing information is already committed, it is only marked for deletion and existing metadevices continue to function until a commit is done, at which time the information is removed.

- z primary_path

Undelete AP information for the specified disk pathgroup. This option cancels a previous apdisk -d request that marked committed information for deletion.

- f io_controller_path

Mark the I/O controller path as offline. The corresponding metadisk interface can still be used if the other I/O controller path in the pathgroup is functioning properly. Note that you cannot mark an I/O controller path as offline if it is currently the active alternate.
Maintenance Commands

−n  *io_controller_path*

Mark the I/O controller path as online. Note that this operation does not automatically cause the I/O controller path to become the active alternate.

−u  −p  *primary_path* −a  *alternate_path*

Update existing database entries for the disk pathgroup identified by the primary path (for example, sf0). Disk targets that are no longer accessible through one or more paths are removed, and new disk targets are added. To update the metadisk device nodes execute the following two commands:

```
drvconfig -i ap_dmd
apconfig -R
```

See *apconfig(1M)* in this reference manual and *drvconfig(1M)* in the *SunOS Reference Manual*.

−w  *io_controller_path*

Clear the tried flag for the specified I/O controller path.

**EXAMPLES**

**EXAMPLE 1**  Creating Metadisk Nodes and AP Database Entries

The following commands create metadisk device nodes and AP database entries for disks that use the pln0 and pln1 interfaces, with pln0 specified as the primary path.

```
# apdisk -c -p pln0 -a pln1
# apdb -C
# drvconfig -i ap_dmd
# apconfig -R
```

**EXAMPLE 2**  Deleting Database Entries

The following commands delete the AP database entries for disks with sf1 specified as the primary path.

```
# apdisk -d sf1
# apdb -C
```

**EXAMPLE 3**  Clearing the Tried Flag

The following command clears the tried flag for sf1.

```
# apdisk -w sf1
```
SEE ALSO

`apdb(1M), apconfig(1M), apinst(1M), apnet(1M)` in this reference manual

devlinks(1M), drvconfig(1M) in the *man Pages(1M): System Administration Commands* in the *SunOS Reference Manual*
NAME
apinst – identify disk host adapter instances, /dev/dsk targets

SYNOPSIS
apinst

DESCRIPTION
The /usr/sbin/apinst program identifies all disk host bus adapters and provides the name, instance number, and /dev/dsk targets attached to each.

EXAMPLES
% apinst

isp0
/dev/dsk/c0t0d0
/dev/dsk/c0t1d0
/dev/dsk/c0t2d0

pln0
/dev/dsk/c1t0d0
/dev/dsk/c1t1d0
/dev/dsk/c1t2d0
/dev/dsk/c1t3d0
/dev/dsk/c1t4d0
/dev/dsk/c1t5d0

plnl
/dev/dsk/c2t0d0
/dev/dsk/c2t1d0
/dev/dsk/c2t2d0
/dev/dsk/c2t3d0
/dev/dsk/c2t4d0
/dev/dsk/c2t5d0

sf0
/dev/dsk/c3t0d0
/dev/dsk/c3t1d0
/dev/dsk/c3t2d0
/dev/dsk/c3t3d0
/dev/dsk/c3t4d0
/dev/dsk/c3t5d0

sf1
/dev/dsk/c4t0d0
/dev/dsk/c4t1d0
/dev/dsk/c4t2d0
/dev/dsk/c4t3d0
/dev/dsk/c4t4d0
/dev/dsk/c4t5d0
NAME  apnet – manage network pathgroups

SYNOPSIS  apnet 
           −c −p network_controller_path [−a network_controller_path]
           apnet
           −d meta_network
           apnet
           −z meta_network
           apnet
           −f network_controller_path
           apnet
           −n network_controller_path
           apnet
           −m meta_network −a network_controller_path
           apnet
           −m meta_network −r network_controller_path
           apnet
           −t meta_network
           apnet
           −w meta_network

DESCRIPTION  The /usr/sbin/apnet command helps you manage network pathgroups in the Alternate Pathing (AP) system.

OPTIONS  The following options are supported:
           −c −p network_controller_path [−a network_controller_path]

           Create a metanetwork and network pathgroup for the network connected to the specified network controller paths. If −a is given, apnet designates the specified network interface as the alternate for the metanetwork. (If you initially create a network pathgroup with only one path, you can later add an additional path using apnet −m.)

           Note: An example of a meta_network is mle0. An example of a network_controller_path is le0.

           −d meta_network

           Delete the specified metanetwork and corresponding network pathgroup. If the pathgroup is currently uncommitted, apnet removes the metanetwork and the pathgroup immediately. If the pathgroup is committed, the metanetwork and pathgroup are only marked for deletion, and the metanetwork interface continues to function until a commit is performed.

           −z meta_network

           SunOS 5.7  Last modified May 1999
Undelete the specified metanetwork and pathgroup. This option cancels a previous `apnet -d` request that marked a committed pathgroup for deletion.

`-f network_controller_path`

Mark the specified network controller path as offline, making it inaccessible through its metanetwork interface.

`-n network_controller_path`

Mark the specified network controller path as online, making it accessible through its metanetwork interface.

Note: A network controller path cannot be marked as offline if it is active.

`-m meta_network -a network_controller_path`

Add the network controller path as an alternate path for the specified metanetwork. You can use this option only if there is currently exactly one path associated with the metanetwork.

`-m meta_network -r network_controller_path`

Remove the network controller path from the specified metanetwork.

Note: When an alternate is added (-a) or removed (-r) from a committed pathgroup, a commit operation must be performed before the change takes effect. In practice, the existing metanetwork interface is marked for deletion, and a new metanetwork interface is created without affecting usage of the existing interface.

`-t meta_network`

Make the next alternate path (the path listed after the primary path) the temporary active path. This option is intended for scripts that are trying alternate paths in sequence until a working path is found. The command returns an error if the sequencing wraps back to the original primary.

`-w meta_network`

Make the current temporary active path the actual active path.
EXAMPLES

EXAMPLE 1  Creating a Network Pathgroup and Metanetwork Interface

The following command creates a network pathgroup and a metanetwork interface, mle0, which has le0 as its primary physical network interface and le1 as its alternate physical network interface.

```bash
# apnet -c -p le0 -a le1
# apdb -C
```

CODE EXAMPLE 1  Deleting a Network Pathgroup and Metanetwork Interface

The following example deletes the network pathgroup and metanetwork interface mle0:

```bash
# apnet -d mle0
# apdb -C
```

SEE ALSO

apconfig(1M), apdb(1M), apdisk(1M)
<table>
<thead>
<tr>
<th>NAME</th>
<th>ap_reboot_host – fast boot alternate path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>ap_reboot_host</td>
</tr>
<tr>
<td>AVAILABILITY</td>
<td>Sun Enterprise 10000 servers only</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>The $SSPOPT/bin/ap_reboot_host command is executed when a boot failure is detected. It determines the boot path of the previous boot and attempts to restart the host from an alternate path if one is available.</td>
</tr>
<tr>
<td>CAUTION</td>
<td>Do not execute ap_reboot_host on the command line; it is intended for use only by other commands.</td>
</tr>
<tr>
<td>NAME</td>
<td>apssp – client of AP SSP daemon</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>SYNOPSIS</td>
<td>apssp</td>
</tr>
<tr>
<td>AVAILABILITY</td>
<td>Sun Enterprise 10000 servers only</td>
</tr>
</tbody>
</table>
| DESCRIPTION | apssp is a client of the Alternate Pathing SSP daemon, `ap_ssp_daemon(1M)`. It takes information from `ap_ssp_daemon` and passes it to `ap_reboot_host(1M)`.
| CAUTION | Do not execute `apssp` on the command line; it is intended for use only by other commands. |
| SEE ALSO | `ap_ssp_daemon(1M), ap_reboot_host(1M)` |
NAME       ap_ssp_daemon – AP SSP daemon

SYNOPSIS   ap_ssp_daemon

AVAILABILITY Sun Enterprise 10000 servers only (this command is executed in the SSP environment)

DESCRIPTION The $SSPOPT/bin/ap_ssp_daemon command is an RPC program that maintains an SSP-based file that contains Alternate Pathing (AP) information for the boot disks. This file is updated automatically by ap_daemon(1M).

ap_ssp_daemon provides its information to apssp(1M), which then passes it to ap_reboot_host(1M). The SSP program apssp(1M) provides the interface to the ap_ssp_daemon.

The daemon’s only clients are apssp(1M) and ap_daemon(1M). The apssp(1M) client provides a way to access the information the daemon keeps. The ap_daemon(1M) updates the information.

SEE ALSO ap_daemon(1M), ap_reboot_host(1M), apssp(1M)
Device and Network Interfaces
This section describes AP files for your Sun Enterprise server.

### LIST OF FUNCTIONS

- **ap(7)**: alternate pathing librarian driver, `/dev/ap`
- **ap_dmd(7)**: AP disk meta-driver
- **ap_nmd(7)**: AP network meta-driver group
- **mge(7)**: GigabitEthernet special character device (see **ap_nmd(7)**)
- **mhme(7)**: SunFastEthernet 2.0 (see **ap_nmd(7)**)
- **mle(7)**: SCSI-2/Buffered Ethernet FSBE/S and DSBE/S (Lance Ethernet) special character device (see **ap_nmd(7)**)
- **mnf(7)**: SunFDDI 3.0.x and 4.x special character device (see **ap_nmd(7)**)
- **mqe(7)**: Quad Ethernet special character device (see **ap_nmd(7)**)
- **mqfe(7)**: Sun Quad FastEthernet special character device (see **ap_nmd(7)**)
NAME
ap_dmd – AP disk meta-driver

SYNOPSIS
ap_dmd @ target,lun:partition

DESCRIPTION
The ap_dmd driver works with the AP software to support Alternate Pathing for physical devices handled by the ssd SCSI disk driver. See ssd(7) in the SunOS Reference Manual.

The AP feature lets you configure alternate SCSI paths to a physical device. These paths are associated with a metadisk device, which is one of the file system special nodes associated with a particular metadriver.

The ap_dmd driver enables the AP Librarian, ap(7), to configure or unconfigure physical paths to a SCSI device by using an interface that allows APSET, APUNSET, and APSWITCH commands. These commands are issued by ap(7) at the request of the user-invoked AP commands and AP daemon. To change the SCSI path information associated with a particular ap_dmd device, use apconfig(1M), apdb(1M) and apdisk(1M). For more information, see the Sun Enterprise Server Alternate Pathing User’s Guide.

All device operations supported by the ssd driver are also valid on ap_dmd devices that have been created by using AP commands. See the other AP commands for details regarding other components of the AP software, and the ssd(7) man page for information about block/character file accesses, I/O requests, disk partitioning schemes, CD-ROM support, and ioctls.

ERRORS
The ENXIO function sets errno as listed for the following conditions:

ENXIO No physical SCSI path to the target device exists.

Other For information on other errors, see sd(7).

FILES
The following files are used by this utility:

apdmd.conf driver configuration file

/dev/ap/dsk/mn cn tn dn block files

/dev/ap/rdsk/mn cn tn dn raw files

where mn identifies the device as a metadevice and:

    cn Controller number
    tn Target number
    dn Logical unit number
    sn Slice (partition) number
DIAGNOSTICS

SEE ALSO
apconfig(1M), apdb(1M), ap_disk(1M), apnet(1M), ap_daemon(1M), ap(7), and ap_nmd(7) in this reference manual.
NAME
ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group

SYNOPSIS
/devices/pseudo/clone@0:mhme
/devices/pseudo/clone@0:mle
/devices/pseudo/clone@0:mnf
/devices/pseudo/clone@0:mqe
/devices/pseudo/clone@0:mqfe
/devices/pseudo/clone@0:mge

DESCRIPTION
The ap_nmd group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, dlpi(7), for hme(7) (SunFastEthernet 2.0), le(7) (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), nf(7) (SunFDDI 5.x), qe(7) (Quad Ethernet), qfe (Sun Quad FastEthernet), and ge (GigabitEthernet 2.0).

Note - SunOS man pages that describe drivers for optional packages, such as SunFDDI and SunFastEthernet, are available only on systems that have those packages installed.

The ap_nmd driver works with the AP software to support Alternate Pathing for physical network devices.

Device operations of ap_nmd are an extension of the operations of the underlying network drivers. The ap_nmd driver normally operates as a transparent pass-through module; it neither interprets nor modifies any of the STREAMS DLPI type messages. However, it does intercept and modify the DL_ATTACH_REQ and DL_INFO_ACK messages.

DL_ATTACH_REQ messages are captured and used to drive the initial connection between logical and physical devices. DL_INFO_ACK messages are captured and responded to with a prebuilt response to eliminate the possibility of the message response timing out due to induced message delays.

The cloning, character-special device /dev/mxx is used to access all device-specific instances of the ap_nmd driver within the system.

ap_nmd and AP
The ap_nmd driver provides an interface to support Alternate Pathing. The APSET interface enables a user to provide a mapping between physical path and logical path. APUNSET provides an interface to remove a physical-to-logical path mapping, and APSWITCH provides a mechanism to switch a logical path from its existing physical path to a new physical path.
For a more complete description of this AP capability, see the
Sun Enterprise Server Alternate Pathing User’s Guide.

ap_nmd and DLPI

The ap_nmd driver is a "style 2" Data Link Service provider. All DLPI
processing is handled by the underlying physical device driver. See the man
page that corresponds to each underlying driver.

ERRORS

The ap_nmd() function sets errno as listed for the following conditions:

EBUSY An attempt was made to unload a busy device or to
       APUNSET an active device.

EEXIST An attempt was made to APSET an existing
        logical-to-physical mapping and a logical path when the
        system was out of memory.

EIO An attempt to switch between physical devices failed.

ENODEV No physical mapping exists.

ENOMEM System memory was exhausted during an attempt to create a
        mapping between a physical path and a logical path.

FILES

The following files are used by this utility:

 mhme.conf driver configuration file
 mle.conf driver configuration file
 mnf.conf driver configuration file
 mqe.conf driver configuration file
 mqfe.conf driver configuration file
 mge.conf driver configuration file
 /dev/mhme hme special character device
 /dev/mle le special character device
 /dev/mnf nf special character device
 /dev/mqe qe special character device
 /dev/mqfe qfe special character device
 /dev/mge ge special character device

DIAGNOSTICS

See le(7) and qe(7) in the SunOS Reference Manual.

SEE ALSO

ap_daemon(1M), apconfig(1M), apdb(1M), apnet(1M), ap(7), and
ap_dmd(7) in this reference manual

driver.conf(4) in the SunOS Reference Manual

The SunOS Reference Manual and other optional reference manuals (for
example, the SunFDDI Reference Manual), as appropriate.
NAME
ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group

SYNOPSIS
```
/devices/pseudo/clone@0:mhme
/devices/pseudo/clone@0:mle
/devices/pseudo/clone@0:mnf
/devices/pseudo/clone@0:mqe
/devices/pseudo/clone@0:mqfe
/devices/pseudo/clone@0:mge
```

DESCRIPTION
The ap_nmd group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, dlpi(7), for hme(7) (SunFastEthernet 2.0), le(7) (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), nf(7) (SunFDDI 5.x), qe(7) (Quad Ethernet), qfe (Sun Quad FastEthernet), and ge (GigabitEthernet 2.0).

Note - SunOS man pages that describe drivers for optional packages, such as SunFDDI and SunFastEthernet, are available only on systems that have those packages installed.

The ap_nmd driver works with the AP software to support Alternate Pathing for physical network devices.

Device operations of ap_nmd are an extension of the operations of the underlying network drivers. The ap_nmd driver normally operates as a transparent pass-through module; it neither interprets nor modifies any of the STREAMS DLPI type messages. However, it does intercept and modify the DL_ATTACH_REQ and DL_INFO_ACK messages.

DL_ATTACH_REQ messages are captured and used to drive the initial connection between logical and physical devices. DL_INFO_ACK messages are captured and responded to with a prebuilt response to eliminate the possibility of the message response timing out due to induced message delays.

The cloning, character-special device /dev/max is used to access all device-specific instances of the ap_nmd driver within the system.

ap_nmd and AP
The ap_nmd driver provides an interface to support Alternate Pathing. The APSET interface enables a user to provide a mapping between physical path and logical path. APUNSET provides an interface to remove a physical-to-logical path mapping, and APSWITCH provides a mechanism to switch a logical path from its existing physical path to a new physical path.
For a more complete description of this AP capability, see the
Sun Enterprise Server Alternate Pathing User’s Guide.

ap_nmd and DLPI

The ap_nmd driver is a "style 2" Data Link Service provider. All DLPI
processing is handled by the underlying physical device driver. See the man
page that corresponds to each underlying driver.

ERRORS

The ap_nmd() function sets errno as listed for the following conditions:

EBUSY   An attempt was made to unload a busy device or to
        APUNSET an active device.

EEXIST  An attempt was made to APSET an existing
        logical-to-physical mapping and a logical path when the
        system was out of memory.

EIO      An attempt to switch between physical devices failed.

ENODEV  No physical mapping exists.

ENOMEM   System memory was exhausted during an attempt to create a
         mapping between a physical path and a logical path.

FILES

The following files are used by this utility:

mhme.conf driver configuration file
mle.conf driver configuration file
mnf.conf driver configuration file
mqe.conf driver configuration file
mqfe.conf driver configuration file
mge.conf driver configuration file
/dev/mhme hme special character device
/dev/mle le special character device
/dev/mnf nf special character device
/dev/mqe qe special character device
/dev/mqfe qfe special character device
/dev/mge ge special character device

DIAGNOSTICS

See le(7) and qe(7) in the SunOS Reference Manual.

SEE ALSO

Sun Enterprise Server Alternate Pathing User’s Guide

ap_daemon(1M), apconfig(1M), apdb(1M), apnet(1M), ap(7), and
ap_dmd(7) in this reference manual

driver.conf(4) in the SunOS Reference Manual

The SunOS Reference Manual and other optional reference manuals (for
example, the SunFDDI Reference Manual), as appropriate.
NAME | ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group

SYNOPSIS

```
/devices/pseudo/clone@0:mhme
/devices/pseudo/clone@0:mle
/devices/pseudo/clone@0:mnf
/devices/pseudo/clone@0:mqe
/devices/pseudo/clone@0:mqfe
/devices/pseudo/clone@0:mge
```

DESCRIPTION

The ap_nmd group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, dlpi(7), for hme(7) (SunFastEthernet 2.0), le(7) (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), nf(7) (SunFDDI 5.x), qe(7) (Quad Ethernet), qfe (Sun Quad FastEthernet), and ge (GigabitEthernet 2.0).

**Note** - SunOS man pages that describe drivers for optional packages, such as SunFDDI and SunFastEthernet, are available only on systems that have those packages installed.

The ap_nmd driver works with the AP software to support Alternate Pathing for physical network devices.

Device operations of ap_nmd are an extension of the operations of the underlying network drivers. The ap_nmd driver normally operates as a transparent pass-through module; it neither interprets nor modifies any of the STREAMS DLPI type messages. However, it does intercept and modify the DL_ATTACH_REQ and DL_INFO_ACK messages.

DL_ATTACH_REQ messages are captured and used to drive the initial connection between logical and physical devices. DL_INFO_ACK messages are captured and responded to with a prebuilt response to eliminate the possibility of the message response timing out due to induced message delays.

The cloning, character-special device /dev/max is used to access all device-specific instances of the ap_nmd driver within the system.

**ap_nmd and AP**

The ap_nmd driver provides an interface to support Alternate Pathing. The APSET interface enables a user to provide a mapping between physical path and logical path. APUNSET provides an interface to remove a physical-to-logical path mapping, and APSWITCH provides a mechanism to switch a logical path from its existing physical path to a new physical path.
For a more complete description of this AP capability, see the
Sun Enterprise Server Alternate Pathing User’s Guide.

**ap_nmd and DLPI**
The `ap_nmd` driver is a "style 2" Data Link Service provider. All DLPI
processing is handled by the underlying physical device driver. See the man
page that corresponds to each underlying driver.

**ERRORS**
The `ap_nmd()` function sets `errno` as listed for the following conditions:

- **EBUSY**
  An attempt was made to unload a busy device or to
  `APUNSET` an active device.

- **EEXIST**
  An attempt was made to `APSET` an existing
  logical-to-physical mapping and a logical path when the
  system was out of memory.

- **EIO**
  An attempt to switch between physical devices failed.

- **ENODEV**
  No physical mapping exists.

- **ENOMEM**
  System memory was exhausted during an attempt to create a
  mapping between a physical path and a logical path.

**FILES**
The following files are used by this utility:

- `mhme.conf` driver configuration file
- `mle.conf` driver configuration file
- `mnf.conf` driver configuration file
- `mqe.conf` driver configuration file
- `mqfe.conf` driver configuration file
- `mge.conf` driver configuration file
- `/dev/mhme` hme special character device
- `/dev/mle` le special character device
- `/dev/mnf` nf special character device
- `/dev/mqe` qe special character device
- `/dev/mqfe` qfe special character device
- `/dev/mge` ge special character device

**DIAGNOSTICS**
See `le(7)` and `qe(7)` in the *SunOS Reference Manual*.

**SEE ALSO**
*Sun Enterprise Server Alternate Pathing User’s Guide*

- `ap_daemon(1M)`, `apconfig(1M)`, `apdb(1M)`, `apnet(1M)`, `ap(7)`, and
- `ap_dmd(7)` in this reference manual

- `driver.conf(4)` in the *SunOS Reference Manual*

The *SunOS Reference Manual* and other optional reference manuals (for
example, the *SunFDDI Reference Manual*), as appropriate.
**NAME**

ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group

**SYNOPSIS**

/devices/pseudo/clone@0:mhme

/devices/pseudo/clone@0:mle

/devices/pseudo/clone@0:mnf

/devices/pseudo/clone@0:mqe

/devices/pseudo/clone@0:mqfe

/devices/pseudo/clone@0:mge

**DESCRIPTION**

The ap_nmd group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, dlpi(7), for hme(7) (SunFastEthernet 2.0), le(7) (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), nf(7) (SunFDDI 5.x), qe(7) (Quad Ethernet), qfe (Sun Quad FastEthernet), and ge (GigabitEthernet 2.0).

**Note** - SunOS man pages that describe drivers for optional packages, such as SunFDDI and SunFastEthernet, are available only on systems that have those packages installed.

The ap_nmd driver works with the AP software to support Alternate Pathing for physical network devices.

Device operations of ap_nmd are an extension of the operations of the underlying network drivers. The ap_nmd driver normally operates as a transparent pass-through module; it neither interprets nor modifies any of the STREAMS DLPI type messages. However, it does intercept and modify the DL_ATTACH_REQ and DL_INFO_ACK messages.

DL_ATTACH_REQ messages are captured and used to drive the initial connection between logical and physical devices. DL_INFO_ACK messages are captured and responded to with a prebuilt response to eliminate the possibility of the message response timing out due to induced message delays.

The cloning, character-special device /dev/mxx is used to access all device-specific instances of the ap_nmd driver within the system.

**ap_nmd and AP**

The ap_nmd driver provides an interface to support Alternate Pathing. The APSET interface enables a user to provide a mapping between physical path and logical path. APUNSET provides an interface to remove a physical-to-logical path mapping, and APSWITCH provides a mechanism to switch a logical path from its existing physical path to a new physical path.

34 SunOS 5.7 Last modified May 1999
For a more complete description of this AP capability, see the
Sun Enterprise Server Alternate Pathing User’s Guide.

**ap_nmd and DLPI**

The `ap_nmd` driver is a "style 2" Data Link Service provider. All DLPI
processing is handled by the underlying physical device driver. See the man
page that corresponds to each underlying driver.

**ERRORS**

The `ap_nmd()` function sets `errno` as listed for the following conditions:

- **EBUSY**
  An attempt was made to unload a busy device or to
  APUNSET an active device.

- **EEXIST**
  An attempt was made to APSET an existing
  logical-to-physical mapping and a logical path when the
  system was out of memory.

- **EIO**
  An attempt to switch between physical devices failed.

- **ENOMEM**
  System memory was exhausted during an attempt to create a
  mapping between a physical path and a logical path.

**FILES**

The following files are used by this utility:

- `mhme.conf` driver configuration file
- `mle.conf` driver configuration file
- `mnf.conf` driver configuration file
- `mqe.conf` driver configuration file
- `mqfe.conf` driver configuration file
- `/dev/mhme hme special character device`
- `/dev/mle le special character device`
- `/dev/mnf nf special character device`
- `/dev/mqe qe special character device`
- `/dev/mqfe qfe special character device`
- `/dev/mge ge special character device`

**DIAGNOSTICS**

See `le(7)` and `qe(7)` in the SunOS Reference Manual.

**SEE ALSO**

Sun Enterprise Server Alternate Pathing User’s Guide

- `ap_daemon(1M)`, `apconfig(1M)`, `apdb(1M)`, `apnet(1M)`, `ap(7)`, and
  `ap_dmd(7)` in this reference manual

- `driver.conf(4)` in the SunOS Reference Manual

The SunOS Reference Manual and other optional reference manuals (for
example, the SunFDDI Reference Manual), as appropriate.
NAME  
ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group

SYNOPSIS  
/devices/pseudo/clone@0:mhme
/devices/pseudo/clone@0:mle
/devices/pseudo/clone@0:mnf
/devices/pseudo/clone@0:mqe
/devices/pseudo/clone@0:mqfe
/devices/pseudo/clone@0:mge

DESCRIPTION  
The ap_nmd group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, dlpi(7), for hme(7) (SunFastEthernet 2.0), le(7) (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), nf(7) (SunFDDI 5.x), qe(7) (Quad Ethernet), qfe (Sun Quad FastEthernet), and ge (GigabitEthernet 2.0).

Note - SunOS man pages that describe drivers for optional packages, such as SunFDDI and SunFastEthernet, are available only on systems that have those packages installed.

The ap_nmd driver works with the AP software to support Alternate Pathing for physical network devices.

Device operations of ap_nmd are an extension of the operations of the underlying network drivers. The ap_nmd driver normally operates as a transparent pass-through module; it neither interprets nor modifies any of the STREAMS DLPI type messages. However, it does intercept and modify the DL_ATTACH_REQ and DL_INFO_ACK messages.

DL_ATTACH_REQ messages are captured and used to drive the initial connection between logical and physical devices. DL_INFO_ACK messages are captured and responded to with a prebuilt response to eliminate the possibility of the message response timing out due to induced message delays.

The cloning, character-special device /dev/mxx is used to access all device-specific instances of the ap_nmd driver within the system.

ap_nmd and AP  
The ap_nmd driver provides an interface to support Alternate Pathing. The APSET interface enables a user to provide a mapping between physical path and logical path. APUNSET provides an interface to remove a physical-to-logical path mapping, and APSWITCH provides a mechanism to switch a logical path from its existing physical path to a new physical path.
For a more complete description of this AP capability, see the 
Sun Enterprise Server Alternate Pathing User’s Guide.

**ap_nmd and DLPI**
The `ap_nmd` driver is a "style 2" Data Link Service provider. All DLPI 
processing is handled by the underlying physical device driver. See the man 
page that corresponds to each underlying driver.

**ERRORS**
The `ap_nmd()` function sets `errno` as listed for the following conditions:

- **EBUSY**  
  An attempt was made to unload a busy device or to 
  APUNSET an active device.

- **EEXIST**  
  An attempt was made to APSET an existing 
  logical-to-physical mapping and a logical path when the 
  system was out of memory.

- **EIO**  
  An attempt to switch between physical devices failed.

- **ENODEV**  
  No physical mapping exists.

- **ENOMEM**  
  System memory was exhausted during an attempt to create a 
  mapping between a physical path and a logical path.

**FILES**
The following files are used by this utility:

- `mhme.conf` driver configuration file
- `mle.conf` driver configuration file
- `mnf.conf` driver configuration file
- `mqe.conf` driver configuration file
- `mqfe.conf` driver configuration file
- `/dev/mhme` hme special character device
- `/dev/mle` le special character device
- `/dev/mnf` nf special character device
- `/dev/mqe` qe special character device
- `/dev/mqfe` qfe special character device
- `/dev/mge` ge special character device

**DIAGNOSTICS**
See le(7) and qe(7) in the SunOS Reference Manual.

**SEE ALSO**
Sun Enterprise Server Alternate Pathing User’s Guide

`ap_daemon`(1M), `apconfig`(1M), `apdb`(1M), `apnet`(1M), `ap`(7), and 
`ap_dmd`(7) in this reference manual

`driver.conf`(4) in the SunOS Reference Manual

The SunOS Reference Manual and other optional reference manuals (for 
example, the SunFDDI Reference Manual), as appropriate.
NAME
ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group

SYNOPSIS
/devices/pseudo/clone@0:mhme
/devices/pseudo/clone@0:mle
/devices/pseudo/clone@0:mnf
/devices/pseudo/clone@0:mqe
/devices/pseudo/clone@0:mqfe
/devices/pseudo/clone@0:mge

DESCRIPTION
The ap_nmd group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, dlpi(7), for hme(7) (SunFastEthernet 2.0), le(7) (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), nf(7) (SunFDDI 5.x), qe(7) (Quad Ethernet), qfe (Sun Quad FastEthernet), and ge (GigabitEthernet 2.0).

Note - SunOS man pages that describe drivers for optional packages, such as SunFDDI and SunFastEthernet, are available only on systems that have those packages installed.

The ap_nmd driver works with the AP software to support Alternate Pathing for physical network devices.

Device operations of ap_nmd are an extension of the operations of the underlying network drivers. The ap_nmd driver normally operates as a transparent pass-through module; it neither interprets nor modifies any of the STREAMS DLPI type messages. However, it does intercept and modify the DL_ATTACH_REQ and DL_INFO_ACK messages.

DL_ATTACH_REQ messages are captured and used to drive the initial connection between logical and physical devices. DL_INFO_ACK messages are captured and responded to with a prebuilt response to eliminate the possibility of the message response timing out due to induced message delays.

The cloning, character-special device /dev/mxx is used to access all device-specific instances of the ap_nmd driver within the system.

ap_nmd and AP
The ap_nmd driver provides an interface to support Alternate Pathing. The APSET interface enables a user to provide a mapping between physical path and logical path. APUNSET provides an interface to remove a physical-to-logical path mapping, and APSWITCH provides a mechanism to switch a logical path from its existing physical path to a new physical path.
For a more complete description of this AP capability, see the
*Sun Enterprise Server Alternate Pathing User’s Guide*.

**ap_nmd and DLPI**

The `ap_nmd` driver is a "style 2" Data Link Service provider. All DLPI processing is handled by the underlying physical device driver. See the man page that corresponds to each underlying driver.

**ERRORS**

The `ap_nmd()` function sets `errno` as listed for the following conditions:

- **EBUSY** An attempt was made to unload a busy device or to APUNSET an active device.
- **EEXIST** An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory.
- **EIO** An attempt to switch between physical devices failed.
- **ENODEV** No physical mapping exists.
- **ENOMEM** System memory was exhausted during an attempt to create a mapping between a physical path and a logical path.

**FILES**

The following files are used by this utility:

- `mhme.conf` driver configuration file
- `mle.conf` driver configuration file
- `mnf.conf` driver configuration file
- `mqe.conf` driver configuration file
- `mqfe.conf` driver configuration file
- `mge.conf` driver configuration file
- `/dev/mhme hme special character device`
- `/dev/mle le special character device`
- `/dev/mnf nf special character device`
- `/dev/mqe qe special character device`
- `/dev/mqfe qfe special character device`
- `/dev/mge ge special character device`

**DIAGNOSTICS**

See `le(7)` and `qe(7)` in the *SunOS Reference Manual*.

**SEE ALSO**

*Sun Enterprise Server Alternate Pathing User’s Guide*

- `ap_daemon(1M)`, `apconfig(1M)`, `apdb(1M)`, `apnet(1M)`, `ap(7)`, and `ap_dmd(7)` in this reference manual

driver.conf(4) in the *SunOS Reference Manual*

The *SunOS Reference Manual* and other optional reference manuals (for example, the *SunFDDI Reference Manual*), as appropriate.

Last modified May 1999
NAME  
ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group

SYNOPSIS

/devices/pseudo/clone@0:mhme
/devices/pseudo/clone@0:mle
/devices/pseudo/clone@0:mnf
/devices/pseudo/clone@0:mqe
/devices/pseudo/clone@0:mqfe
/devices/pseudo/clone@0:mge

DESCRIPTION

The ap_nmd group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, dlpi(7), for hme(7) (SunFastEthernet 2.0), le(7) (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), nf(7) (SunFDDI 5.x), qe(7) (Quad Ethernet), qfe (Sun Quad FastEthernet), and ge (GigabitEthernet 2.0).

Note - SunOS man pages that describe drivers for optional packages, such as SunFDDI and SunFastEthernet, are available only on systems that have those packages installed.

The ap_nmd driver works with the AP software to support Alternate Pathing for physical network devices.

Device operations of ap_nmd are an extension of the operations of the underlying network drivers. The ap_nmd driver normally operates as a transparent pass-through module; it neither interprets nor modifies any of the STREAMS DLPI type messages. However, it does intercept and modify the DL_ATTACH_REQ and DL_INFO_ACK messages.

DL_ATTACH_REQ messages are captured and used to drive the initial connection between logical and physical devices. DL_INFO_ACK messages are captured and responded to with a prebuilt response to eliminate the possibility of the message response timing out due to induced message delays.

The cloning, character-special device /dev/mxx is used to access all device-specific instances of the ap_nmd driver within the system.

ap_nmd and AP

The ap_nmd driver provides an interface to support Alternate Pathing. The APSET interface enables a user to provide a mapping between physical path and logical path. APUNSET provides an interface to remove a physical-to-logical path mapping, and APSWITCH provides a mechanism to switch a logical path from its existing physical path to a new physical path.
For a more complete description of this AP capability, see the
_Sun Enterprise Server Alternate Pathing User’s Guide._

**ap_nmd and DLPI**

The _ap_nmd_ driver is a “style 2” Data Link Service provider. All DLPI
processing is handled by the underlying physical device driver. See the man
page that corresponds to each underlying driver.

**ERRORS**

The _ap_nmd()_ function sets _errno_ as listed for the following conditions:

- **EBUSY** — An attempt was made to unload a busy device or to
  APUNSET an active device.

- **EEXIST** — An attempt was made to APSET an existing
  logical-to-physical mapping and a logical path when the
  system was out of memory.

- **EIO** — An attempt to switch between physical devices failed.

- **ENOMEM** — System memory was exhausted during an attempt to create a
  mapping between a physical path and a logical path.

**FILES**

The following files are used by this utility:

- mhme.conf  driver configuration file
- mle.conf   driver configuration file
- mnf.conf   driver configuration file
- mqe.conf   driver configuration file
- mqfe.conf  driver configuration file
- mge.conf   driver configuration file
- /dev/mhme  hme special character device
- /dev/mle   le special character device
- /dev/mnf   nf special character device
- /dev/mqe   qe special character device
- /dev/mqfe  qfe special character device
- /dev/mge   ge special character device

**DIAGNOSTICS**

See le(7) and qe(7) in the _SunOS Reference Manual._

**SEE ALSO**

_Sun Enterprise Server Alternate Pathing User’s Guide_

- _ap_daemon(1M), apconfig(1M), apdb(1M), apnet(1M), ap(7), and
  ap_dmd(7)_ in this reference manual

- _driver.conf(4)_ in the _SunOS Reference Manual_

The _SunOS Reference Manual_ and other optional reference manuals (for
example, the _SunFDDI Reference Manual_), as appropriate.

Last modified May 1999