



---

# Sun Enterprise Server Alternate Pathing 2.0.1 Reference Manual

---

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

Part No: 806-1165-10  
May 1999

Copyright 1999 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, SunDocs, Java, the Java Coffee Cup logo, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

**RESTRICTED RIGHTS:** Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

Copyright 1999 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, SunDocs, Java, le logo Java Coffee Cup, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



# Contents

---

Intro(1M)	2
ap(1M)	3
ap_daemon(1M)	4
ap_reboot_host(1M)	5
ap_ssp_daemon(1M)	6
apboot(1M)	7
apcheck(1M)	9
apconfig(1M)	10
apdb(1M)	14
apdisk(1M)	16
apinst(1M)	18
apnet(1M)	19
apssp(1M)	22
Intro(7)	24
ap(7)	25
ap_dmd(7)	26
ap_nmd(7)	28
ap_nmd(7)	31
ap_nmd(7)	34

ap_nmd(7)	37
ap_nmd(7)	40
ap_nmd(7)	43
ap_nmd(7)	46

# Maintenance Commands

<b>NAME</b>	Intro – AP administration
<b>DESCRIPTION</b>	This section describes commands, scripts, and programs executed in the Alternate Pathing environment.
<b>LIST OF COMMANDS</b>	<p>The following commands are supported:</p> <p><b>ap(1M)</b> alternate pathing</p> <p><b>ap_daemon(1M)</b> alternate pathing daemon</p> <p><b>ap_reboot_host(1M)</b> fast boot alternate path</p> <p><b>ap_ssp_daemon(1M)</b> AP SSP daemon</p> <p><b>apboot(1M)</b> set up system files for boot metadisk</p> <p><b>apcheck(1M)</b> determine accessibility of a metadisk</p> <p><b>apconfig(1M)</b> display and manage AP configuration</p> <p><b>apdb(1M)</b> manage AP database</p> <p><b>apdisk(1M)</b> manage disk pathgroups</p> <p><b>apinst(1M)</b> identify disk host adapter instances, /dev/dsk targets</p> <p><b>apnet(1M)</b> manage network pathgroups</p> <p><b>apssp(1M)</b> client of AP SSP daemon</p>

<b>NAME</b>	ap – alternate pathing
<b>DESCRIPTION</b>	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.
<b>SEE ALSO</b>	<i>Sun Enterprise Server Alternate Pathing User's Guide</i>

<b>NAME</b>	ap_daemon – alternate pathing daemon
<b>SYNOPSIS</b>	<b>ap_daemon</b>
<b>DESCRIPTION</b>	The <code>/usr/sbin/ap_daemon</code> is an RPC program that provides the interface to the Alternate Pathing (AP) driver.
<b>Configuration Information</b>	<p>The <code>ap_daemon</code> RPC program name is <code>AP_SVR</code>; its RPC program number is 300473; and, its underlying protocol is TCP. It is invoked as an <code>inetd</code> server by using the TCP transport. The UID required for access to the daemon is <code>ssp</code>. This UID can be a non-login UID.</p> <p>The entry for the daemon in the <code>/etc/inetd.conf</code> file is:</p> <pre>300473/1 tli rpc/tcp wait root /usr/sbin/ap_daemon ap_daemon</pre>
<b>SEE ALSO</b>	<p><i>Sun Enterprise Server Alternate Pathing User's Guide</i></p> <p><b>apconfig(1M)</b>, <b>apdb(1M)</b>, <b>apdisk(1M)</b>, <b>apnet(1M)</b></p>



<b>NAME</b>	ap_reboot_host - fast boot alternate path
<b>SYNOPSIS</b>	<b>ap_reboot_host</b>
<b>AVAILABILITY</b>	Sun Enterprise 10000 servers only
<b>DESCRIPTION</b>	The <code>\$\$SSPOPT/bin/ap_reboot_host</code> 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.
<b>CAUTION</b>	Do not execute <code>ap_reboot_host</code> on the command line; it is intended for use only by other commands.

<b>NAME</b>	ap_ssp_daemon – AP SSP daemon
<b>SYNOPSIS</b>	<b>ap_ssp_daemon</b>
<b>AVAILABILITY</b>	Sun Enterprise 10000 servers only (this command is executed in the SSP environment)
<b>DESCRIPTION</b>	<p>The <code>\$SSPOPT/bin/ap_ssp_daemon</code> 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 <code>ap_daemon(1M)</code>.</p> <p><code>ap_ssp_daemon</code> provides its information to <code>apssp(1M)</code>, which then passes it to <code>ap_reboot_host(1M)</code>. The SSP program <code>apssp(1M)</code> provides the interface to the <code>ap_ssp_daemon</code>.</p> <p>The daemon's only clients are <code>apssp(1M)</code> and <code>ap_daemon(1M)</code>. The <code>apssp(1M)</code> client provides a way to access the information the daemon keeps. The <code>ap_daemon(1M)</code> updates the information.</p>
<b>SEE ALSO</b>	<code>ap_daemon(1M)</code> , <code>ap_reboot_host(1M)</code> , <code>apssp(1M)</code>

<b>NAME</b>	apboot – set up system files for boot metadisk
<b>SYNOPSIS</b>	<p><b>apboot</b> [-n] [-k <i>system-name</i>] [-v <i>vfstab-name</i>] <i>device</i></p> <p><b>apboot</b> [-m <i>metadevice-name</i>]</p> <p><b>apboot</b> [-u <i>metadevice-name</i>]</p>
<b>DESCRIPTION</b>	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; or to allow AP to coherently manage a mirrored boot device when both that host device and its mirror are under AP control.
<b>OPTIONS</b>	<p>The following options are supported:</p> <p>-n</p> <p>Print what would be done without actually doing it.</p> <p>-k <b><i>system-name</i></b></p> <p>Edit <i>system-name</i> instead of the default /etc/system file.</p> <p>-v <b><i>vfstab-name</i></b></p> <p>Edit <i>vfstab-name</i> instead of the default /etc/vfstab table of file system defaults.</p> <p>-m <b><i>metadevice-name</i></b></p> <p>Enable boot mirror support for the specified AP metadevice.</p> <p>-u <b><i>metadevice-name</i></b></p> <p>Disable boot mirror support for the specified AP metadevice.</p>
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> Using apboot with Metadisks</p> <p>The following command edits /etc/system and /etc/vfstab to specify that the boot-disk file systems are now on metadisk mc3t0d0.</p> <pre>apboot mc3t0d0</pre>

**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**

`system(4)`, `vfstab(4)` in the *man Pages(4): File Formats* of the *SunOS Reference Manual*

<b>NAME</b>	apcheck – determine accessibility of a metadisk
<b>SYNOPSIS</b>	<b>apcheck</b> <i>special</i>
<b>DESCRIPTION</b>	<code>/sbin/apcheck</code> ascertains whether a metadisk is usable. If it is able to locate dual paths, <code>apcheck</code> exits with a zero status; if not, it exits with a non-zero status.
<b>CAUTION</b>	Do not execute <code>apcheck</code> on the command line; it is intended for use only by other commands or by authorized service providers.
<b>OPTIONS</b>	The following options are supported: <b><i>special</i></b> This option represents the device node to be checked. This device node may reside under <code>/dev/ap/dsk</code> or <code>/dev/ap/rdsk</code> .

<b>NAME</b>	apconfig – display and manage AP configuration
<b>SYNOPSIS</b>	<p><b>apconfig</b> -D</p> <p><b>apconfig</b> -F</p> <p><b>apconfig</b> -N [-u]</p> <p><b>apconfig</b> -P <i>meta_netwk_intrfc</i> -a <i>new_phys_path</i></p> <p><b>apconfig</b> -P <i>primary_path</i> -a <i>new_phys_path</i></p> <p><b>apconfig</b> -R</p> <p><b>apconfig</b> -S [-u]</p>
<b>DESCRIPTION</b>	The /usr/sbin/apconfig command displays and helps you manage the Alternate Pathing (AP) system configuration.
<b>OPTIONS</b>	<p>The following options are supported:</p> <p>-D</p> <p>Display location and status information for all known copies of the host database.</p> <p>-F</p> <p>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.</p> <p>-N [-u]</p> <p>Display network AP information only. For each pathgroup, apconfig -N displays the metanetwork interface and the corresponding physical network interfaces.</p> <p>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> <p>-P <i>meta_netwk_intrfc</i> -a <i>new_phys_path</i></p>

Switch to the new physical path specified by `-a` for the metanetwork specified by `-P`.

`-P primary_path-a new_phys_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/rdisk`. 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 "Letters after names and paths", below.

#### Letters after names and paths

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 <code>apdb(1M)</code> .
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. <code>-S</code> only.
M	Marked as the mirror for a boot device. <code>-S</code> only.

X	The physical paths for this metadisk lead to different disks (that is, different SSAs). <code>-S</code> only.
	When you specify <code>-N</code> or <code>-S</code> , one or more of the following letters may be displayed after each physical network path or physical disk I/O port name:
N	Automatic switching is not allowed for this physical device.
X	The physical paths for this meta-disk lead to different disks (that is, different SSA's). <code>-S</code> only.
A	The active alternate. To select another interface, use the <code>-P</code> and <code>-a</code> options.
DR	Marked as being drained by Dynamic Reconfiguration. A switch cannot be made to a device path in this state. See the <i>Sun Enterprise Server Alternate Pathing User's Guide</i> .
DE	Marked as detached by Dynamic Reconfiguration.
P	The primary path. The primary path cannot be changed.
T	Path has been tried as active.
O	Marked as offline. See <code>apdisk(1M)</code> and <code>apnet(1M)</code> .

**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`.

```
# 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`.

```
# 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.

```
# apconfig -D

path: /dev/rdisk/c3t3d0s1
major: 32
minor: 145
timestamp: Wed Sep 28 18:45:58 1994
checksum: 2636010350
default: yes
corrupt: no
inaccessible: no

path: /dev/rdisk/c3t3d0s6
major: 32
minor: 150
timestamp: Wed Sep 28 18:50:43 1994
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*

<b>NAME</b>	apdb – manage AP database
<b>SYNOPSIS</b>	<p><b>apdb</b> -c <i>raw_disk_slice</i> [-k <i>system_file</i>] [-f]</p> <p><b>apdb</b> -d <i>raw_disk_slice</i> [-k <i>system_file</i>] [-f]</p> <p><b>apdb</b> -m <i>major</i> -n <i>minor</i> [-f]</p> <p><b>apdb</b> -C</p> <p><b>apdb</b> -Z</p>
<b>DESCRIPTION</b>	The /usr/sbin/apdb command helps you manage the AP database.
<b>OPTIONS</b>	<p>The following options are supported:</p> <p>-c <b><i>raw_disk_slice</i></b> 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.</p> <p>-d <b><i>raw_disk_slice</i></b> Delete a database copy from the specified raw disk slice.</p> <p>-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.</p> <p>-k <b><i>system_file</i></b> Patch the the database copy information to the kernel file <i>system_file</i>, rather than the default file, /etc/system.</p> <p>-m <b><i>major</i></b> -n <b><i>minor</i></b> 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.</p> <p>-C Commit all uncommitted entries within the database.</p> <p>-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</p>

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/rdisk/c2t0d0s1`.

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

**SEE ALSO**

*Sun Enterprise Server Alternate Pathing User's Guide*

`apconfig(1M)`, `apdisk(1M)`, `apnet(1M)`

<b>NAME</b>	apdisk – manage disk pathgroups
<b>SYNOPSIS</b>	<p><b>apdisk</b> -c -p <i>primary_path</i> -a <i>alternate_path</i></p> <p><b>apdisk</b> d <i>primary_path</i></p> <p><b>apdisk</b> -z <i>primary_path</i></p> <p><b>apdisk</b> -f <i>io_port</i></p> <p><b>apdisk</b> -n <i>io_port</i></p> <p><b>apdisk</b> -u -p <i>primary_path</i> -a <i>alternate_path</i></p> <p><b>apdisk</b> -w <i>io_port</i></p>
<b>DESCRIPTION</b>	The /usr/sbin/apdisk command helps you manage disk pathgroups in the Alternate Pathing (AP) system.
<b>OPTIONS</b>	<p>The following options are supported:</p> <p>-c -p <i>primary_path</i> -a <i>alternate_path</i></p> <p>Create database entries for disks (i.e., a SPARCStorage Arrays) connected to two I/O ports. Give the I/O port names (e.g., p1n0 and p1n1) as the <i>primary_path</i> and <i>alternate_path</i>.</p> <p>-d <i>primary_path</i></p> <p>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.</p> <p>-z <i>primary_path</i></p> <p>Undelete AP information for the specified disk pathgroup. This option cancels a previous apdisk -d request that marked committed information for deletion.</p> <p>-f <i>io_port</i></p> <p>Mark the I/O port 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.</p>

**-n *io\_port***

Mark the I/O port 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 (e.g., `pln0`). 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 `drvconfig -i ap_dmd` and `apconfig -R`.

See `apconfig(1M)` in this reference manual and `drvconfig(1M)` in *man Pages(1M): System Administration Commands* of the *SunOS Reference Manual*.

**-w *io\_port***

Clear the `tried` flag for the specified I/O port.

## 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
```

## SEE ALSO

*Sun Enterprise Server Alternate Pathing User's Guide*

`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*

<b>NAME</b>	apinst – identify disk host adapter instances, /dev/dsk targets
<b>SYNOPSIS</b>	<b>apinst</b>
<b>DESCRIPTION</b>	The /usr/sbin/apinst program identifies all disk host bus adapters and provides the name, instance number, and /dev/dsk targets attached to each.
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> Identifying Disk, Host Adapter Instances</p> <pre> % 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  pln1     /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 </pre>

<b>NAME</b>	apnet - manage network pathgroups
<b>SYNOPSIS</b>	<p><b>apnet</b> -c -p <i>ntwk_intrfc</i> [-a <i>ntwk_intrfc</i>]</p> <p><b>apnet</b> -d <i>meta_ntwk_intrfc</i></p> <p><b>apnet</b> -z <i>meta_ntwk_intrfc</i></p> <p><b>apnet</b> -f <i>ntwk_intrfc</i></p> <p><b>apnet</b> -n <i>ntwk_intrfc</i></p> <p><b>apnet</b> -m <i>meta_ntwk_intrfc</i> -a <i>ntwk_intrfc</i></p> <p><b>apnet</b> -m <i>meta_ntwk_intrfc</i> -r <i>ntwk_intrfc</i></p> <p><b>apnet</b> -t <i>meta_ntwk_intrfc</i></p> <p><b>apnet</b> -w <i>meta_ntwk_intrfc</i></p>
<b>DESCRIPTION</b>	The /usr/sbin/apnet command helps you manage network pathgroups in the Alternate Pathing (AP) system.
<b>OPTIONS</b>	<p>The following options are supported:</p> <hr/> <p><b>Note</b> - The parameters <i>meta_ntwk_intrfc</i> and <i>ntwk_intrfc</i> are strings of the form <i>type instance_#</i>, but with no space between them; for example, m1e0 and m1e1.</p> <hr/> <p>-c -p <i>ntwk_intrfc</i> [-a <i>ntwk_intrfc</i>]</p> <p>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.)</p> <p>-d <i>meta_ntwk_intrfc</i></p> <p>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.</p> <p>-z <i>meta_ntwk_intrfc</i></p>

Undelete the specified metanetwork and pathgroup. This option cancels a previous `apnet -d` request that marked a committed pathgroup for deletion.

`-f ntwk_intrfc`

Mark the specified network interface as offline and therefore inaccessible through its metanetwork interface.

---

**Note** - An interface cannot be marked as offline if it is active.

---

`-n ntwk_intrfc`

Mark the specified network interface as online and therefore accessible through its metanetwork interface.

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

`-m meta_ntwk_intrfc -a ntwk_intrfc`

Add the specified network interface as the alternate for the specified network pathgroup. You can use this option only if the network pathgroup currently contains exactly one path.

`-m meta_ntwk_intrfc -r ntwk_intrfc`

Remove the specified network interface from the specified network pathgroup.

Note: When an alternate is added (`-a`) or removed (`-r`) from a committed network 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 one is created without affecting usage of the existing interface.

`-t meta_ntwk_intrfc`

Make the next alternate path (after the primary) 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_ntwk_intrfc`



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, `m1e0`, which has `1e0` as its primary physical network interface and `1e1` as its alternate physical network interface.

```
# apnet -c -p 1e0 -a 1e1
# apdb -C
```

**EXAMPLE 2** Deleting a Network Pathgroup and Metanetwork Interface

The following example deletes the network pathgroup and metanetwork interface `m1e0`:

```
# apnet -d m1e0
# apdb -C
```

**SEE ALSO**

*Sun Enterprise Server Alternate Pathing User's Guide*

`apconfig(1M)`, `apdb(1M)`, `apdisk(1M)`

<b>NAME</b>	apssp – client of AP SSP daemon
<b>SYNOPSIS</b>	<b>apssp</b>
<b>AVAILABILITY</b>	Sun Enterprise 10000 servers only
<b>DESCRIPTION</b>	apssp is a client of the Alternate Pathing SSP daemon, <b>ap_ssp_daemon(1M)</b> . It takes information from <b>ap_ssp_daemon</b> and passes it to <b>ap_reboot_host(1M)</b> .
<b>CAUTION</b>	Do not execute <b>apssp</b> on the command line; it is intended for use only by other commands.
<b>SEE ALSO</b>	<b>ap_ssp_daemon(1M)</b> , <b>ap_reboot_host(1M)</b>

# Device and Network Interfaces

<b>NAME</b>	Intro – AP special files
<b>DESCRIPTION</b>	This section describes AP files for your Sun Enterprise server.
<b>LIST OF FUNCTIONS</b>	<p>The following functions are supported:</p> <p><b>ap(7)</b>                 alternate pathing librarian driver, /dev/ap</p> <p><b>ap_dmd(7)</b>            AP disk meta-driver</p> <p><b>ap_nmd(7)</b>            AP network meta-driver group</p> <p><b>mhme(7)</b>             SunFastEthernet 2.0</p> <p><b>mle(7)</b>                Lance Ethernet special character device</p> <p><b>mnf(7)</b>                FDDI 3.0.x and 4.x special character device</p> <p><b>mqe(7)</b>                Quad Ethernet special character device</p> <p><b>mqfe(7)</b>             Quad FastEthernet special character device</p> <p><b>mge(7)</b>                Sun Gigabit Ethernet special character device</p>

<b>NAME</b>	ap – alternate pathing librarian driver, /dev/ap
<b>DESCRIPTION</b>	The AP driver provides a pseudo-driver interface to the kernel Alternate Pathing (AP) Librarian features.
<b>FILES</b>	The following files are used: /kernel/drv/ap            AP driver module /kernel/drv/ap.conf     AP driver configuration file
<b>SEE ALSO</b>	<i>Sun Enterprise Alternate Pathing User's Guide</i> ap(1M), ap_daemon(1M)

<b>NAME</b>	ap_dmd – AP disk meta-driver
<b>SYNOPSIS</b>	<b>ap_dmd@</b> <i>target,lun:partition</i>
<b>DESCRIPTION</b>	<p>The <code>ap_dmd</code> driver works with the AP software to support Alternate Pathing for physical devices handled by the <code>ssd</code> SCSI disk driver. See <code>ssd(7)</code> in <i>man Pages(7): Device and Network Interfaces</i> in the <i>SunOS Reference Manual</i>.</p> <p>The AP feature lets you configure alternate SCSI paths to a physical device. These paths are associated with a <i>metadisk device</i>, which is one of the file system special nodes associated with a particular metadriver.</p> <p><code>ap_dmd</code> allows the AP Librarian, <code>ap(7)</code>, to configure or unconfigure physical paths to a SCSI device by using an interface that allows <code>APSET</code>, <code>APUNSET</code>, and <code>APSWITCH</code> commands. These commands are issued by <code>ap(7)</code> at the behest of the user-invoked AP commands and AP daemon. To change the SCSI path information associated with a particular <code>ap_dmd</code> device, use <code>apconfig(1M)</code>, <code>apdb(1M)</code> and <code>apdisk(1M)</code>. For more information, see the <i>Sun Enterprise Server Alternate Pathing User's Guide</i>.</p> <p>All device operations supported by the <code>ssd</code> driver are also valid on <code>ap_dmd</code> 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 <code>ssd(7)</code> man page for information about block/character file accesses, I/O requests, disk partitioning schemes, CD-ROM support, and <code>ioctl</code>s.</p>
<b>ERRORS</b>	<p>The <code>ENXIO</code> function sets <code>errno</code> as listed for the following conditions:</p> <p><code>ENXIO</code>            No physical SCSI path to the target device exists.</p> <p><b>Other</b>            For information on other errors, see <code>sd(7)</code>.</p>
<b>FILES</b>	<p>The following files are used by this utility:</p> <p><code>apdmd.conf</code>                            driver configuration file</p> <p><code>/dev/ap/dsk/mn cn tn dn sn</code>            block files</p> <p><code>/dev/ap/rdsk/mn cn tn dn sn</code>           raw files</p> <p>where <code>m</code> identifies the device as a metadvice and:</p> <p><code>cn</code>                    Controller number</p> <p><code>tn</code>                    Target number</p> <p><code>dn</code>                    Logical unit number</p> <p><code>sn</code>                    Slice (partition) number</p>

**DIAGNOSTICS**

See *ssd(7)* in *man Pages(7): Device and Network Interfaces* in the *SunOS Reference Manual*.

**SEE ALSO**

*Sun Enterprise Server Alternate Pathing User's Guide*

*apconfig(1M)*, *apdb(1M)*, *ap\_disk(1M)*, *apnet(1M)*, *ap\_daemon(1M)*, *ap(7)*, and *ap\_nmd(7)* in this reference manual

*ssd(7)* in *man Pages(7): Device and Network Interfaces* in the *SunOS Reference Manual*

<b>NAME</b>	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
<b>SYNOPSIS</b>	<pre> /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 </pre>
<b>DESCRIPTION</b>	<p>The <code>ap_nmd</code> group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, <code>dlpi(7)</code>, for <code>hme(7)</code> (Sun FastEthernet 2.0), <code>le(7)</code> (Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Quad FastEthernet), and <code>ge</code> (Sun GigabitEthernet 2.0).</p> <hr/> <p><b>Note</b> - SunOS man pages that describe drivers for optional packages, such as FDDI and Sun FastEthernet, are available only on systems that have those packages installed.</p> <hr/> <p><code>ap_nmd</code> works with the AP software to support Alternate Pathing for physical network devices.</p> <p>Device operations of <code>ap_nmd</code> are an extension of the operations of the underlying network drivers. <code>ap_nmd</code> 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 <code>DL_ATTACH_REQ</code> and <code>DL_INFO_ACK</code> messages.</p> <p><code>DL_ATTACH_REQ</code> messages are captured and used to drive the initial connection between logical and physical devices. <code>DL_INFO_ACK</code> 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.</p> <p>The cloning character-special device <code>/dev/mxx</code> is used to access all device-specific instances of the <code>ap_nmd</code> driver within the system.</p>
<b>ap_nmd and AP</b>	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provide an interface to remove a physical-to-logical path mapping and <code>APSWITCH</code> provides a mechanism to switch a logical path from its existing physical path to a new physical path. For a more complete</p>



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)`, `ap_dmd(7)`, in this reference manual

`driver.conf(4)` *man Pages(4): File Formats of the SunOS Reference Manual*

*man Pages(7): Device and Network Interfaces of the SunOS Reference Manual and other optional reference manuals (for example, the FDDI Reference Manual), as appropriate*

<b>NAME</b>	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
<b>SYNOPSIS</b>	<pre> /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 </pre>
<b>DESCRIPTION</b>	<p>The <code>ap_nmd</code> group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, <code>dlpi(7)</code>, for <code>hme(7)</code> (Sun FastEthernet 2.0), <code>le(7)</code> (Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Quad FastEthernet), and <code>ge</code> (Sun GigabitEthernet 2.0).</p> <hr/> <p><b>Note</b> - SunOS man pages that describe drivers for optional packages, such as FDDI and Sun FastEthernet, are available only on systems that have those packages installed.</p> <hr/> <p><code>ap_nmd</code> works with the AP software to support Alternate Pathing for physical network devices.</p> <p>Device operations of <code>ap_nmd</code> are an extension of the operations of the underlying network drivers. <code>ap_nmd</code> 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 <code>DL_ATTACH_REQ</code> and <code>DL_INFO_ACK</code> messages.</p> <p><code>DL_ATTACH_REQ</code> messages are captured and used to drive the initial connection between logical and physical devices. <code>DL_INFO_ACK</code> 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.</p> <p>The cloning character-special device <code>/dev/mxx</code> is used to access all device-specific instances of the <code>ap_nmd</code> driver within the system.</p>
<b>ap_nmd and AP</b>	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provide an interface to remove a physical-to-logical path mapping and <code>APSWITCH</code> provides a mechanism to switch a logical path from its existing physical path to a new physical path. For a more complete</p>

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)`, `ap_dmd(7)`, in this reference manual

`driver.conf(4)` *man Pages(4): File Formats of the SunOS Reference Manual*

*man Pages(7): Device and Network Interfaces of the SunOS Reference Manual and other optional reference manuals (for example, the FDDI Reference Manual), as appropriate*

<b>NAME</b>	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
<b>SYNOPSIS</b>	<pre> /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                 </pre>
<b>DESCRIPTION</b>	<p>The <code>ap_nmd</code> group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, <code>dlpi(7)</code>, for <code>hme(7)</code> (Sun FastEthernet 2.0), <code>le(7)</code> (Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Quad FastEthernet), and <code>ge</code> (Sun GigabitEthernet 2.0).</p> <hr/> <p><b>Note</b> - SunOS man pages that describe drivers for optional packages, such as FDDI and Sun FastEthernet, are available only on systems that have those packages installed.</p> <hr/> <p><code>ap_nmd</code> works with the AP software to support Alternate Pathing for physical network devices.</p> <p>Device operations of <code>ap_nmd</code> are an extension of the operations of the underlying network drivers. <code>ap_nmd</code> 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 <code>DL_ATTACH_REQ</code> and <code>DL_INFO_ACK</code> messages.</p> <p><code>DL_ATTACH_REQ</code> messages are captured and used to drive the initial connection between logical and physical devices. <code>DL_INFO_ACK</code> 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.</p> <p>The cloning character-special device <code>/dev/mxx</code> is used to access all device-specific instances of the <code>ap_nmd</code> driver within the system.</p>
<b>ap_nmd and AP</b>	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provide an interface to remove a physical-to-logical path mapping and <code>APSWITCH</code> provides a mechanism to switch a logical path from its existing physical path to a new physical path. For a more complete</p>

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)`, `ap_dmd(7)`, in this reference manual

`driver.conf(4)` *man Pages(4): File Formats of the SunOS Reference Manual*

*man Pages(7): Device and Network Interfaces of the SunOS Reference Manual and other optional reference manuals (for example, the FDDI Reference Manual), as appropriate*



<b>NAME</b>	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
<b>SYNOPSIS</b>	<pre> /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 </pre>
<b>DESCRIPTION</b>	<p>The <code>ap_nmd</code> group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, <code>dlpi(7)</code>, for <code>hme(7)</code> (Sun FastEthernet 2.0), <code>le(7)</code> (Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Quad FastEthernet), and <code>ge</code> (Sun GigabitEthernet 2.0).</p> <hr/> <p><b>Note</b> - SunOS man pages that describe drivers for optional packages, such as FDDI and Sun FastEthernet, are available only on systems that have those packages installed.</p> <hr/> <p><code>ap_nmd</code> works with the AP software to support Alternate Pathing for physical network devices.</p> <p>Device operations of <code>ap_nmd</code> are an extension of the operations of the underlying network drivers. <code>ap_nmd</code> 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 <code>DL_ATTACH_REQ</code> and <code>DL_INFO_ACK</code> messages.</p> <p><code>DL_ATTACH_REQ</code> messages are captured and used to drive the initial connection between logical and physical devices. <code>DL_INFO_ACK</code> 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.</p> <p>The cloning character-special device <code>/dev/mxx</code> is used to access all device-specific instances of the <code>ap_nmd</code> driver within the system.</p>
<b>ap_nmd and AP</b>	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provide an interface to remove a physical-to-logical path mapping and <code>APSWITCH</code> provides a mechanism to switch a logical path from its existing physical path to a new physical path. For a more complete</p>

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)`, `ap_dmd(7)`, in this reference manual

`driver.conf(4)` *man Pages(4): File Formats of the SunOS Reference Manual*

*man Pages(7): Device and Network Interfaces of the SunOS Reference Manual and other optional reference manuals (for example, the FDDI Reference Manual), as appropriate*

<b>NAME</b>	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
<b>SYNOPSIS</b>	<pre> /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 </pre>
<b>DESCRIPTION</b>	<p>The <code>ap_nmd</code> group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, <code>dlpi(7)</code>, for <code>hme(7)</code> (Sun FastEthernet 2.0), <code>le(7)</code> (Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Quad FastEthernet), and <code>ge</code> (Sun GigabitEthernet 2.0).</p> <hr/> <p><b>Note</b> - SunOS man pages that describe drivers for optional packages, such as FDDI and Sun FastEthernet, are available only on systems that have those packages installed.</p> <hr/> <p><code>ap_nmd</code> works with the AP software to support Alternate Pathing for physical network devices.</p> <p>Device operations of <code>ap_nmd</code> are an extension of the operations of the underlying network drivers. <code>ap_nmd</code> 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 <code>DL_ATTACH_REQ</code> and <code>DL_INFO_ACK</code> messages.</p> <p><code>DL_ATTACH_REQ</code> messages are captured and used to drive the initial connection between logical and physical devices. <code>DL_INFO_ACK</code> 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.</p> <p>The cloning character-special device <code>/dev/mxx</code> is used to access all device-specific instances of the <code>ap_nmd</code> driver within the system.</p>
<b>ap_nmd and AP</b>	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provide an interface to remove a physical-to-logical path mapping and <code>APSWITCH</code> provides a mechanism to switch a logical path from its existing physical path to a new physical path. For a more complete</p>

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:

- |               |   |
|---------------|---|
| <b>EBUSY</b>  | An attempt was made to unload a busy device, or to APUNSET an active device.  |
| <b>EEXIST</b> | An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory |
| <b>EIO</b>    | An attempt to switch between physical devices failed  |
| <b>ENODEV</b> | No physical mapping exists  |
| <b>ENOMEM</b> | 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)`, `ap_dmd(7)`, in this reference manual

`driver.conf(4)` *man Pages(4): File Formats of the SunOS Reference Manual*

*man Pages(7): Device and Network Interfaces of the SunOS Reference Manual and other optional reference manuals (for example, the FDDI Reference Manual), as appropriate*

<b>NAME</b>	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
<b>SYNOPSIS</b>	<pre> /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 </pre>
<b>DESCRIPTION</b>	<p>The <code>ap_nmd</code> group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, <code>dlpi(7)</code>, for <code>hme(7)</code> (Sun FastEthernet 2.0), <code>le(7)</code> (Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Quad FastEthernet), and <code>ge</code> (Sun GigabitEthernet 2.0).</p> <hr/> <p><b>Note</b> - SunOS man pages that describe drivers for optional packages, such as FDDI and Sun FastEthernet, are available only on systems that have those packages installed.</p> <hr/> <p><code>ap_nmd</code> works with the AP software to support Alternate Pathing for physical network devices.</p> <p>Device operations of <code>ap_nmd</code> are an extension of the operations of the underlying network drivers. <code>ap_nmd</code> 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 <code>DL_ATTACH_REQ</code> and <code>DL_INFO_ACK</code> messages.</p> <p><code>DL_ATTACH_REQ</code> messages are captured and used to drive the initial connection between logical and physical devices. <code>DL_INFO_ACK</code> 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.</p> <p>The cloning character-special device <code>/dev/mxx</code> is used to access all device-specific instances of the <code>ap_nmd</code> driver within the system.</p>
<b>ap_nmd and AP</b>	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provide an interface to remove a physical-to-logical path mapping and <code>APSWITCH</code> provides a mechanism to switch a logical path from its existing physical path to a new physical path. For a more complete</p>

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)`, `ap_dmd(7)`, in this reference manual

`driver.conf(4)` *man Pages(4): File Formats of the SunOS Reference Manual*



*man Pages(7): Device and Network Interfaces of the SunOS Reference Manual and other optional reference manuals (for example, the FDDI Reference Manual), as appropriate*

<b>NAME</b>	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
<b>SYNOPSIS</b>	<pre> /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 </pre>
<b>DESCRIPTION</b>	<p>The <code>ap_nmd</code> group of multi-threaded, loadable, clonable, STREAMS metanetwork device drivers that support the connectionless Data Link Provider Interface, <code>dlpi(7)</code>, for <code>hme(7)</code> (Sun FastEthernet 2.0), <code>le(7)</code> (Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Quad FastEthernet), and <code>ge</code> (Sun GigabitEthernet 2.0).</p> <hr/> <p><b>Note</b> - SunOS man pages that describe drivers for optional packages, such as FDDI and Sun FastEthernet, are available only on systems that have those packages installed.</p> <hr/> <p><code>ap_nmd</code> works with the AP software to support Alternate Pathing for physical network devices.</p> <p>Device operations of <code>ap_nmd</code> are an extension of the operations of the underlying network drivers. <code>ap_nmd</code> 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 <code>DL_ATTACH_REQ</code> and <code>DL_INFO_ACK</code> messages.</p> <p><code>DL_ATTACH_REQ</code> messages are captured and used to drive the initial connection between logical and physical devices. <code>DL_INFO_ACK</code> 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.</p> <p>The cloning character-special device <code>/dev/mxx</code> is used to access all device-specific instances of the <code>ap_nmd</code> driver within the system.</p>
<b>ap_nmd and AP</b>	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provide an interface to remove a physical-to-logical path mapping and <code>APSWITCH</code> provides a mechanism to switch a logical path from its existing physical path to a new physical path. For a more complete</p>

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)`, `ap_dmd(7)`, in this reference manual

`driver.conf(4)` *man Pages(4): File Formats of the SunOS Reference Manual*

*man Pages(7): Device and Network Interfaces of the SunOS Reference Manual and other optional reference manuals (for example, the FDDI Reference Manual), as appropriate*