



Sun Enterprise Server Alternate Pathing Reference Manual

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

Part No: 805-5986-10
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, Californie 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
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

ap_nmd(7) 36

ap_nmd(7) 38

ap_nmd(7) 40

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	<i>Sun Enterprise Server Alternate Pathing User's Guide</i>

NAME	apboot – set up system files for boot metadisk
SYNOPSIS	apboot [-n] [-k <i>system-name</i>] [-v <i>vfstab-name</i>] <i>device</i> apboot [-m <i>metadevice-name</i>] apboot [-u <i>metadevice-name</i>]
DESCRIPTION	<p>Use <code>/usr/sbin/apboot</code> to edit <code>/etc/vfstab</code> and <code>/etc/system</code> 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.</p> <p>In addition to editing <code>/etc/vfstab/</code> and <code>/etc/system</code>, apboot checks the current configuration of system swap and dump devices. If either is configured as a partition of the boot disk, apboot calls <code>swap(1M)</code> or <code>dump(1M)</code>, as appropriate, to ensure that swap and dump devices are consistent with the boot device.</p>
OPTIONS	<p>The following options are supported:</p> <p>-n</p> <p>Print what would be done without actually doing it.</p> <p>-k <i>system-name</i></p> <p>Edit <i>system-name</i> instead of the default <code>/etc/system</code> file.</p> <p>-v <i>vfstab-name</i></p> <p>Edit <i>vfstab-name</i> instead of the default <code>/etc/vfstab</code> table of file system defaults.</p> <p>-m <i>metadevice-name</i></p> <p>Enable boot mirror support for the specified AP metadevice.</p> <p>-u <i>metadevice-name</i></p> <p>Disable boot mirror support for the specified AP metadevice.</p>
EXAMPLES	<p>EXAMPLE 1 Using apboot with Metadisks</p> <p>The following command edits <code>/etc/system</code> and <code>/etc/vfstab</code> to specify that the boot-disk file systems are now on metadisk <code>mc3t0d0</code>.</p>


```
apboot c3t0d0
```

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 mc3t0d0
```

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

NAME	apconfig – display and manage AP configuration
SYNOPSIS	<p>apconfig -D</p> <p>apconfig -F</p> <p>apconfig -N [-u]</p> <p>apconfig -P <i>meta_network</i> -a <i>new_physical_path</i></p> <p>apconfig -P <i>meta_disk</i> -a <i>new_physical_path</i></p> <p>apconfig -R</p> <p>apconfig -S [-u]</p>
DESCRIPTION	The /usr/sbin/apconfig command displays and helps you manage the Alternate Pathing (AP) system configuration.
OPTIONS	<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_network</i> -a <i>new_physical_path</i></p>

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/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 "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 <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).

Character flags after physical device paths.

X	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 meta-disk 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 <i>Sun Enterprise Server Alternate Pathing User's Guide</i> .
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`.

```
# 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
syncd: 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	<p>The <code>/usr/sbin/ap_daemon</code> is an RPC program that provides the interface to the Alternate Pathing (AP) driver.</p>
Configuration Information	<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>
SEE ALSO	<p><i>Sun Enterprise Server Alternate Pathing User's Guide</i></p> <p>apconfig(1M), apdb(1M), apdisk(1M), apnet(1M)</p>

NAME	apdb – manage AP database
SYNOPSIS	<p>apdb -c <i>raw_disk_slice</i> [-k <i>system_file</i>] [-f]</p> <p>apdb -d <i>raw_disk_slice</i> [-k <i>system_file</i>] [-f]</p> <p>apdb -m <i>major</i> -n <i>minor</i> [-f]</p> <p>apdb -C</p> <p>apdb -Z</p>
DESCRIPTION	The /usr/sbin/apdb command helps you manage the AP database.
OPTIONS	<p>The following options are supported:</p> <p>-c <i>raw_disk_slice</i> 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 <i>raw_disk_slice</i> 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 <i>system_file</i> Patch the the database copy information to the kernel file <i>system_file</i>, rather than the default file, /etc/system.</p> <p>-m <i>major</i> -n <i>minor</i> 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)**

NAME	apdisk – manage disk pathgroups
SYNOPSIS	<p>apdisk -c -p <i>primary_path</i> -a <i>alternate_path</i></p> <p>apdisk d <i>primary_path</i></p> <p>apdisk -z <i>primary_path</i></p> <p>apdisk -f <i>io_controller_path</i></p> <p>apdisk -n <i>io_controller_path</i></p> <p>apdisk -u -p <i>primary_path</i> -a <i>alternate_path</i></p> <p>apdisk -w <i>io_controller_path</i></p>
DESCRIPTION	The /usr/sbin/apdisk command helps you manage disk pathgroups in the Alternate Pathing (AP) system.
OPTIONS	<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 disk arrays connected to two I/O ports. Give the I/O port names (for example, sf0 and sf1) 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_controller_path</i></p> <p>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.</p>

`-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	<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>

NAME	apnet – manage network pathgroups
SYNOPSIS	<p>apnet -c -p <i>network_controller_path</i> [-a <i>network_controller_path</i>]</p> <p>apnet -d <i>meta_network</i></p> <p>apnet -z <i>meta_network</i></p> <p>apnet -f <i>network_controller_path</i></p> <p>apnet -n <i>network_controller_path</i></p> <p>apnet -m <i>meta_network</i> -a <i>network_controller_path</i></p> <p>apnet -m <i>meta_network</i> -x <i>network_controller_path</i></p> <p>apnet -t <i>meta_network</i></p> <p>apnet -w <i>meta_network</i></p>
DESCRIPTION	The <code>/usr/sbin/apnet</code> command helps you manage network pathgroups in the Alternate Pathing (AP) system.
OPTIONS	<p>The following options are supported:</p> <p>-c -p <i>network_controller_path</i> [-a <i>network_controller_path</i>]</p> <p>Create a metanetwork and network pathgroup for the network connected to the specified network controller paths. If -a is given, <code>apnet</code> 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 <code>apnet -m</code>.)</p> <p>Note: An example of a <i>meta_network</i> is <code>m1e0</code>. An example of a <i>network_controller_path</i> is <code>l1e0</code>.</p> <p>-d <i>meta_network</i></p> <p>Delete the specified metanetwork and corresponding network pathgroup. If the pathgroup is currently uncommitted, <code>apnet</code> 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_network</i></p>

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.

```
# 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`:

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

SEE ALSO

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

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

NAME	apssp – client of AP SSP daemon
SYNOPSIS	apssp
AVAILABILITY	Sun Enterprise 10000 servers only
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	<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>
SEE ALSO	<code>ap_daemon(1M)</code> , <code>ap_reboot_host(1M)</code> , <code>apssp(1M)</code>

Device and Network Interfaces

NAME	Intro – AP special files	
DESCRIPTION	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@ <i>target,lun:partition</i>														
DESCRIPTION	<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 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>The <code>ap_dmd</code> driver enables 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 request 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>														
ERRORS	<p>The <code>ENXIO</code> function sets <code>errno</code> as listed for the following conditions:</p> <table> <tr> <td><code>ENXIO</code></td><td>No physical SCSI path to the target device exists.</td></tr> <tr> <td><i>Other</i></td><td>For information on other errors, see <code>sd(7)</code>.</td></tr> </table>	<code>ENXIO</code>	No physical SCSI path to the target device exists.	<i>Other</i>	For information on other errors, see <code>sd(7)</code> .										
<code>ENXIO</code>	No physical SCSI path to the target device exists.														
<i>Other</i>	For information on other errors, see <code>sd(7)</code> .														
FILES	<p>The following files are used by this utility:</p> <table> <tr> <td><code>apdmd.conf</code></td><td>driver configuration file</td></tr> <tr> <td><code>/dev/ap/dsk/<i>mn cn tn dn</i></code></td><td>block files</td></tr> <tr> <td><code>/dev/ap/rdisk/<i>mn cn tn dn sn</i></code></td><td>raw files</td></tr> </table> <p>where <i>m</i> identifies the device as a metadvice and:</p> <table> <tr> <td><i>cn</i></td><td>Controller number</td></tr> <tr> <td><i>tn</i></td><td>Target number</td></tr> <tr> <td><i>dn</i></td><td>Logical unit number</td></tr> <tr> <td><i>sn</i></td><td>Slice (partition) number</td></tr> </table>	<code>apdmd.conf</code>	driver configuration file	<code>/dev/ap/dsk/<i>mn cn tn dn</i></code>	block files	<code>/dev/ap/rdisk/<i>mn cn tn dn sn</i></code>	raw files	<i>cn</i>	Controller number	<i>tn</i>	Target number	<i>dn</i>	Logical unit number	<i>sn</i>	Slice (partition) number
<code>apdmd.conf</code>	driver configuration file														
<code>/dev/ap/dsk/<i>mn cn tn dn</i></code>	block files														
<code>/dev/ap/rdisk/<i>mn cn tn dn sn</i></code>	raw files														
<i>cn</i>	Controller number														
<i>tn</i>	Target number														
<i>dn</i>	Logical unit number														
<i>sn</i>	Slice (partition) number														

DIAGNOSTICS

See `ssd(7)` in the *SunOS Reference Manual*.

SEE ALSO

`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 the *SunOS Reference Manual*

NAME	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
SYNOPSIS	<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>
DESCRIPTION	<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> (SunFastEthernet 2.0), <code>le(7)</code> (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Sun Quad FastEthernet), and <code>ge</code> (GigabitEthernet 2.0).</p> <hr/> <p>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.</p> <hr/> <p>The <code>ap_nmd</code> driver 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. The <code>ap_nmd</code> 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 <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>
ap_nmd and AP	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. The <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provides 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.</p>

	For a more complete description of this AP capability, see the <i>Sun Enterprise Server Alternate Pathing User's Guide</i> .
ap_nmd and DLPI	The <code>ap_nmd</code> 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	<p>The <code>ap_nmd()</code> function sets <code>errno</code> as listed for the following conditions:</p> <p>EBUSY An attempt was made to unload a busy device or to APUNSET an active device.</p> <p>EEXIST An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory.</p> <p>EIO An attempt to switch between physical devices failed.</p> <p>ENODEV No physical mapping exists.</p> <p>ENOMEM System memory was exhausted during an attempt to create a mapping between a physical path and a logical path.</p>
FILES	<p>The following files are used by this utility:</p> <pre> 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 </pre>
DIAGNOSTICS	See <code>le(7)</code> and <code>qe(7)</code> in the <i>SunOS Reference Manual</i> .
SEE ALSO	<p><code>ap_daemon(1M)</code>, <code>apconfig(1M)</code>, <code>apdb(1M)</code>, <code>apnet(1M)</code>, <code>ap(7)</code>, and <code>ap_dmd(7)</code> in this reference manual</p> <p><code>driver.conf(4)</code> in the <i>SunOS Reference Manual</i></p> <p>The <i>SunOS Reference Manual</i> and other optional reference manuals (for example, the <i>SunFDDI Reference Manual</i>), as appropriate.</p>

NAME	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
SYNOPSIS	<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>
DESCRIPTION	<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> (SunFastEthernet 2.0), <code>le(7)</code> (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Sun Quad FastEthernet), and <code>ge</code> (GigabitEthernet 2.0).</p> <hr/> <p>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.</p> <hr/> <p>The <code>ap_nmd</code> driver 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. The <code>ap_nmd</code> 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 <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>
ap_nmd and AP	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. The <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provides 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.</p>

	For a more complete description of this AP capability, see the <i>Sun Enterprise Server Alternate Pathing User's Guide</i> .
ap_nmd and DLPI	The <code>ap_nmd</code> 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	<p>The <code>ap_nmd()</code> function sets <code>errno</code> as listed for the following conditions:</p> <p>EBUSY An attempt was made to unload a busy device or to APUNSET an active device.</p> <p>EEXIST An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory.</p> <p>EIO An attempt to switch between physical devices failed.</p> <p>ENODEV No physical mapping exists.</p> <p>ENOMEM System memory was exhausted during an attempt to create a mapping between a physical path and a logical path.</p>
FILES	<p>The following files are used by this utility:</p> <pre> 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 </pre>
DIAGNOSTICS	See <code>le(7)</code> and <code>qe(7)</code> in the <i>SunOS Reference Manual</i> .
SEE ALSO	<p><i>Sun Enterprise Server Alternate Pathing User's Guide</i></p> <p><code>ap_daemon(1M)</code>, <code>apconfig(1M)</code>, <code>apdb(1M)</code>, <code>apnet(1M)</code>, <code>ap(7)</code>, and <code>ap_dmd(7)</code> in this reference manual</p> <p><code>driver.conf(4)</code> in the <i>SunOS Reference Manual</i></p> <p>The <i>SunOS Reference Manual</i> and other optional reference manuals (for example, the <i>SunFDDI Reference Manual</i>), as appropriate.</p>

NAME	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
SYNOPSIS	<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>
DESCRIPTION	<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> (SunFastEthernet 2.0), <code>le(7)</code> (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Sun Quad FastEthernet), and <code>ge</code> (GigabitEthernet 2.0).</p> <hr/> <p>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.</p> <hr/> <p>The <code>ap_nmd</code> driver 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. The <code>ap_nmd</code> 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 <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>
ap_nmd and AP	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. The <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provides 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.</p>

	For a more complete description of this AP capability, see the <i>Sun Enterprise Server Alternate Pathing User's Guide</i> .
ap_nmd and DLPI	The <code>ap_nmd</code> 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	<p>The <code>ap_nmd()</code> function sets <code>errno</code> as listed for the following conditions:</p> <p>EBUSY An attempt was made to unload a busy device or to APUNSET an active device.</p> <p>EEXIST An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory.</p> <p>EIO An attempt to switch between physical devices failed.</p> <p>ENODEV No physical mapping exists.</p> <p>ENOMEM System memory was exhausted during an attempt to create a mapping between a physical path and a logical path.</p>
FILES	<p>The following files are used by this utility:</p> <pre> 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 </pre>
DIAGNOSTICS	See <code>le(7)</code> and <code>qe(7)</code> in the <i>SunOS Reference Manual</i> .
SEE ALSO	<p><i>Sun Enterprise Server Alternate Pathing User's Guide</i></p> <p><code>ap_daemon(1M)</code>, <code>apconfig(1M)</code>, <code>apdb(1M)</code>, <code>apnet(1M)</code>, <code>ap(7)</code>, and <code>ap_dmd(7)</code> in this reference manual</p> <p><code>driver.conf(4)</code> in the <i>SunOS Reference Manual</i></p> <p>The <i>SunOS Reference Manual</i> and other optional reference manuals (for example, the <i>SunFDDI Reference Manual</i>), as appropriate.</p>

NAME	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
SYNOPSIS	<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>
DESCRIPTION	<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> (SunFastEthernet 2.0), <code>le(7)</code> (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Sun Quad FastEthernet), and <code>ge</code> (GigabitEthernet 2.0).</p> <hr/> <p>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.</p> <hr/> <p>The <code>ap_nmd</code> driver 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. The <code>ap_nmd</code> 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 <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>
ap_nmd and AP	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. The <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provides 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.</p>

	For a more complete description of this AP capability, see the <i>Sun Enterprise Server Alternate Pathing User's Guide</i> .
ap_nmd and DLPI	The <code>ap_nmd</code> 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	<p>The <code>ap_nmd()</code> function sets <code>errno</code> as listed for the following conditions:</p> <p>EBUSY An attempt was made to unload a busy device or to APUNSET an active device.</p> <p>EEXIST An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory.</p> <p>EIO An attempt to switch between physical devices failed.</p> <p>ENODEV No physical mapping exists.</p> <p>ENOMEM System memory was exhausted during an attempt to create a mapping between a physical path and a logical path.</p>
FILES	<p>The following files are used by this utility:</p> <pre> 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 </pre>
DIAGNOSTICS	See <code>le(7)</code> and <code>qe(7)</code> in the <i>SunOS Reference Manual</i> .
SEE ALSO	<p><i>Sun Enterprise Server Alternate Pathing User's Guide</i></p> <p><code>ap_daemon(1M)</code>, <code>apconfig(1M)</code>, <code>apdb(1M)</code>, <code>apnet(1M)</code>, <code>ap(7)</code>, and <code>ap_dmd(7)</code> in this reference manual</p> <p><code>driver.conf(4)</code> in the <i>SunOS Reference Manual</i></p> <p>The <i>SunOS Reference Manual</i> and other optional reference manuals (for example, the <i>SunFDDI Reference Manual</i>), as appropriate.</p>

NAME	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
SYNOPSIS	<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>
DESCRIPTION	<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> (SunFastEthernet 2.0), <code>le(7)</code> (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Sun Quad FastEthernet), and <code>ge</code> (GigabitEthernet 2.0).</p> <hr/> <p>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.</p> <hr/> <p>The <code>ap_nmd</code> driver 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. The <code>ap_nmd</code> 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 <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>
ap_nmd and AP	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. The <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provides 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.</p>

	For a more complete description of this AP capability, see the <i>Sun Enterprise Server Alternate Pathing User's Guide</i> .
ap_nmd and DLPI	The <code>ap_nmd</code> 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	<p>The <code>ap_nmd()</code> function sets <code>errno</code> as listed for the following conditions:</p> <p>EBUSY An attempt was made to unload a busy device or to APUNSET an active device.</p> <p>EEXIST An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory.</p> <p>EIO An attempt to switch between physical devices failed.</p> <p>ENODEV No physical mapping exists.</p> <p>ENOMEM System memory was exhausted during an attempt to create a mapping between a physical path and a logical path.</p>
FILES	<p>The following files are used by this utility:</p> <pre> 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 </pre>
DIAGNOSTICS	See <code>le(7)</code> and <code>qe(7)</code> in the <i>SunOS Reference Manual</i> .
SEE ALSO	<p><i>Sun Enterprise Server Alternate Pathing User's Guide</i></p> <p><code>ap_daemon(1M)</code>, <code>apconfig(1M)</code>, <code>apdb(1M)</code>, <code>apnet(1M)</code>, <code>ap(7)</code>, and <code>ap_dmd(7)</code> in this reference manual</p> <p><code>driver.conf(4)</code> in the <i>SunOS Reference Manual</i></p> <p>The <i>SunOS Reference Manual</i> and other optional reference manuals (for example, the <i>SunFDDI Reference Manual</i>), as appropriate.</p>

NAME	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
SYNOPSIS	<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>
DESCRIPTION	<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> (SunFastEthernet 2.0), <code>le(7)</code> (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Sun Quad FastEthernet), and <code>ge</code> (GigabitEthernet 2.0).</p> <hr/> <p>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.</p> <hr/> <p>The <code>ap_nmd</code> driver 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. The <code>ap_nmd</code> 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 <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>
ap_nmd and AP	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. The <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provides 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.</p>

	For a more complete description of this AP capability, see the <i>Sun Enterprise Server Alternate Pathing User's Guide</i> .
ap_nmd and DLPI	The <code>ap_nmd</code> 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	<p>The <code>ap_nmd()</code> function sets <code>errno</code> as listed for the following conditions:</p> <p>EBUSY An attempt was made to unload a busy device or to APUNSET an active device.</p> <p>EEXIST An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory.</p> <p>EIO An attempt to switch between physical devices failed.</p> <p>ENODEV No physical mapping exists.</p> <p>ENOMEM System memory was exhausted during an attempt to create a mapping between a physical path and a logical path.</p>
FILES	<p>The following files are used by this utility:</p> <pre> 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 </pre>
DIAGNOSTICS	See <code>le(7)</code> and <code>qe(7)</code> in the <i>SunOS Reference Manual</i> .
SEE ALSO	<p><i>Sun Enterprise Server Alternate Pathing User's Guide</i></p> <p><code>ap_daemon(1M)</code>, <code>apconfig(1M)</code>, <code>apdb(1M)</code>, <code>apnet(1M)</code>, <code>ap(7)</code>, and <code>ap_dmd(7)</code> in this reference manual</p> <p><code>driver.conf(4)</code> in the <i>SunOS Reference Manual</i></p> <p>The <i>SunOS Reference Manual</i> and other optional reference manuals (for example, the <i>SunFDDI Reference Manual</i>), as appropriate.</p>

NAME	ap_nmd, mhme, mle, mnf, mqe, mqfe, mge – AP network meta-driver group
SYNOPSIS	<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>
DESCRIPTION	<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> (SunFastEthernet 2.0), <code>le(7)</code> (SCSI-2/Buffered Ethernet FSBE/S and DSBE/S Lance Ethernet), <code>nf(7)</code> (SunFDDI 5.x), <code>qe(7)</code> (Quad Ethernet), <code>qfe</code> (Sun Quad FastEthernet), and <code>ge</code> (GigabitEthernet 2.0).</p> <hr/> <p>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.</p> <hr/> <p>The <code>ap_nmd</code> driver 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. The <code>ap_nmd</code> 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 <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>
ap_nmd and AP	<p>The <code>ap_nmd</code> driver provides an interface to support Alternate Pathing. The <code>APSET</code> interface enables a user to provide a mapping between physical path and logical path. <code>APUNSET</code> provides 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.</p>

	For a more complete description of this AP capability, see the <i>Sun Enterprise Server Alternate Pathing User's Guide</i> .
ap_nmd and DLPI	The <code>ap_nmd</code> 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	<p>The <code>ap_nmd()</code> function sets <code>errno</code> as listed for the following conditions:</p> <p>EBUSY An attempt was made to unload a busy device or to APUNSET an active device.</p> <p>EEXIST An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory.</p> <p>EIO An attempt to switch between physical devices failed.</p> <p>ENODEV No physical mapping exists.</p> <p>ENOMEM System memory was exhausted during an attempt to create a mapping between a physical path and a logical path.</p>
FILES	<p>The following files are used by this utility:</p> <pre> 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 </pre>
DIAGNOSTICS	See <code>le(7)</code> and <code>qe(7)</code> in the <i>SunOS Reference Manual</i> .
SEE ALSO	<p><i>Sun Enterprise Server Alternate Pathing User's Guide</i></p> <p><code>ap_daemon(1M)</code>, <code>apconfig(1M)</code>, <code>apdb(1M)</code>, <code>apnet(1M)</code>, <code>ap(7)</code>, and <code>ap_dmd(7)</code> in this reference manual</p> <p><code>driver.conf(4)</code> in the <i>SunOS Reference Manual</i></p> <p>The <i>SunOS Reference Manual</i> and other optional reference manuals (for example, the <i>SunFDDI Reference Manual</i>), as appropriate.</p>