Sun Enterprise Server ™ Alternate Pathing 2.3.1 User Guide

Sun Enterprise 10000 SSP Attributions:

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

Please
Recycle

™
Adobe PostScript

Contents

Figures

Preface

The *Sun Enterprise Server Alternate Pathing 2.3.1 User Guide* describes the alternate pathing (AP) component of the Sun Enterprise™ server product line. Some AP features apply only to the Sun Enterprise 10000 server, and are noted throughout this guide.

How This Book Is Organized

This guide contains the following chapters:

Chapter 1 introduces AP.

Chapter 2 covers the AP database operations.

Chapter 3 describes metadisks and disk path groups, and explains how to use them.

Chapter 4 covers unattended system boot issues.

Chapter 5 describes metanetworks and network path groups, and explains how to use them.

Chapter 6 describes how dynamic reconfiguration (DR) and AP work together.

Appendix A provides a list of all AP commands.

Appendix B provides an overview of the underlying AP architecture.

Appendix C provides an overview of the underlying AP drivers.

Before You Read This Book

This manual is intended for the Sun Enterprise system administrator, who has a working knowledge of UNIX® systems, particularly those based on the Solaris™ operating environment. If you do not have such knowledge, read the Solaris User and System Administrator AnswerBook™ documentation provided with this system, and consider UNIX system administration training.

Using UNIX Commands

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

See one or more of the following for this information:

- AnswerBook online documentation for the Solaris software environment, particularly those dealing with Solaris system administration
- Other software documentation that you received with your system

Typographic Conventions

TABLE P-1   Typographic Conventions

Typeface or Symbol	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output.	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output.	`%` **`su`** `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized. Command-line variable; replace with a real name or value.	Read Chapter 6 in the *User Guide*. These are called *class* options. You *must* be `root` to do this. To delete a file, type `rm` *filename*.

Shell Prompts

TABLE P-2   Shell Prompts

Shell	Prompt
C shell	*machine_name*`%`
C shell superuser	*machine_name*`#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell superuser	`#`

Ordering Sun Documentation

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at:

```
http://www1.fatbrain.com/documentation/sun
```

Related Documentation

TABLE P-3 Related Documentation

Application	Title
Installation	*Solaris 8 01/01 Sun Hardware Platform Guide*
Reference (man pages)	*Sun Enterprise Server Alternate Pathing 2.3.1 Reference Manual*
Release Notes	*Release Notes Supplement Solaris 8 01/01*
Other	*Sun Enterprise 10000 Dynamic Reconfiguration User Guide* *Sun Enterprise 6x00, 5x00, 4x00, 3x00 Dynamic Reconfiguration User Guide* *Sun Enterprise 10000 Dynamic Reconfiguration Reference Manual*

Sun Documentation on the Web

The `docs.sun.com`^{SM} web site enables you to access Sun technical documentation on the Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at:

```
http://docs.sun.com
```

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

docfeedback@sun.com

Please include the part number of your document in the subject line of your email.

Introduction to Alternate Pathing

This chapter describes the basic purpose of alternate pathing and provides an overview of alternate pathing concepts and terminology.

Purpose of Alternate Pathing

Alternate pathing (AP) supports high availability of I/O controllers—the hardware components that reside on system boards and enable the Sun Enterprise server to communicate with I/O devices such as disks and networks. With AP, each I/O device connects to *two* I/O controllers.
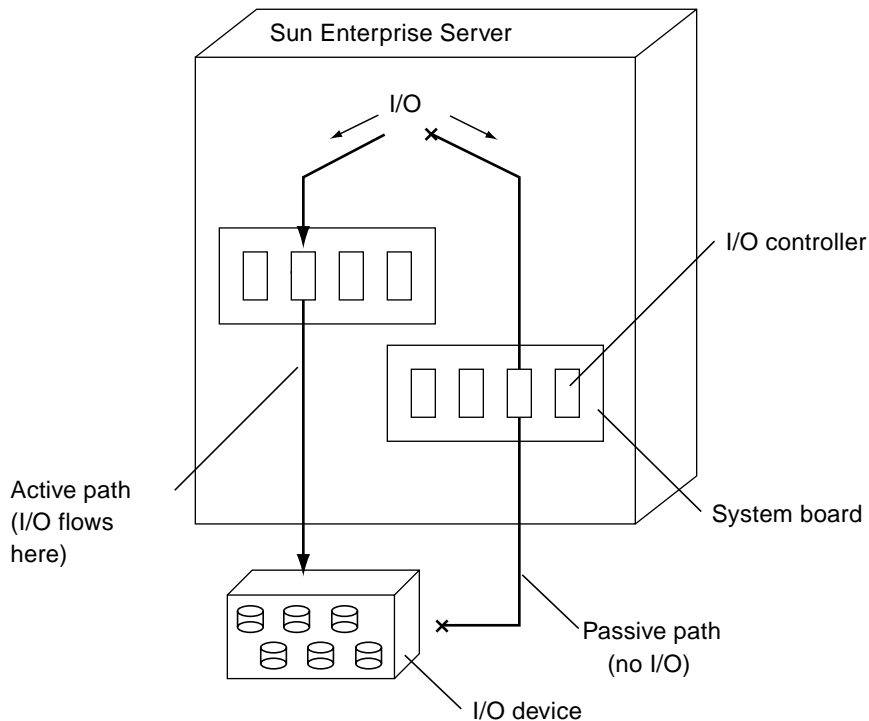
FIGURE 1-1   Alternately Pathed I/O Device

The I/O controllers are part of two separate electrical pathways to the I/O device, known as *alternate paths*. AP enables you to set up and use alternate paths on the Sun Enterprise servers.

For disk controllers, this switch occurs *automatically* whenever a path failure is detected during normal operation. For network controllers, you must manually switch paths (using a single AP command).

There are two purposes for AP. One purpose is to help protect against I/O controller failures. With AP, if one I/O controller fails, you can switch to the alternate controller.

For disk metadevices on a Sun StorEdge™ T3 disk; where two viable physical I/O device paths are present, path optimization is applied to a disk pathgroup. Path optimization refers to the efficient distribution of I/O traffic for a particular device. If one of the physical I/O device paths becomes unavailable, whether due to device failure or user action, path optimization is disabled.
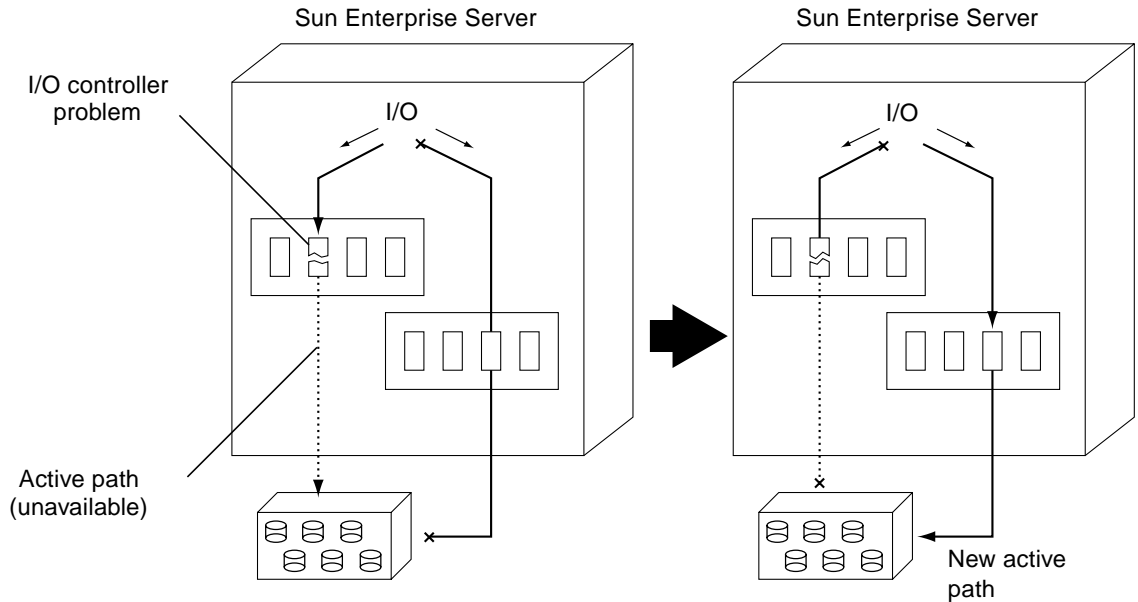
FIGURE 1-2   Switching Paths After an I/O Controller Failure

Note – Automatic switching, on a T3, disables the path optimization algorithm, since only one path is available.

The second purpose of AP is to support dynamic reconfiguration (DR). DR is used to logically attach and detach system boards from the operating system without having to halt and reboot.

For example, with DR you can detach a board from the operating system, physically remove and service the board, and then re-insert the board and attach it to the operating system again. You can do all of this without halting the operating system or terminating any user applications.

If you want to detach a board that is connected to an I/O device, and if that I/O device is alternately pathed, you must first use AP to redirect the I/O flow to a controller on a different board.

For T3 disk path groups this action disables path optimization, allowing the DR detach of the inactive path. You can then use DR to detach the system board without interrupting the I/O flow.

On the Sun Enterprise 10000, the switch can occur automatically during the DR operation (for both disk and network devices), assuming a viable alternate controller exists on another board. However, a manual switch is preferred to disable path optimization prior to issuing a DR detach.

On all other servers, the switch must be performed manually.

The following figure shows the relationship between AP and DR.
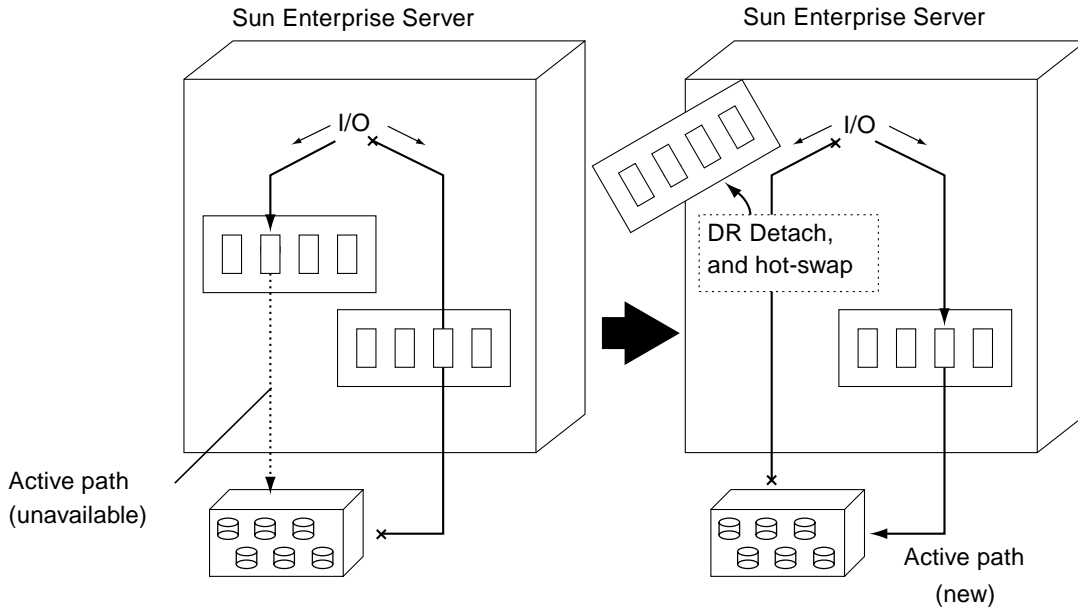
FIGURE 1-3   Switching Paths for a DR Detach Operation

Basic Alternate Pathing Concepts

This section discusses basic AP concepts and introduces the terminology that is used throughout this chapter.

Physical Path

For the purposes of AP, an *I/O device* is either a disk or network. An *I/O controller* is the controller card for an I/O device. An *I/O port* is a connector on a controller card. Often there are two or more ports per controller card. A *device node* is a path in the `/devices` or `/dev` directories that is used to specify a physical device, for example, `/dev/dsk/c0t0d1s0`. The term *physical path* refers to the electrical path from the host to a disk or network.
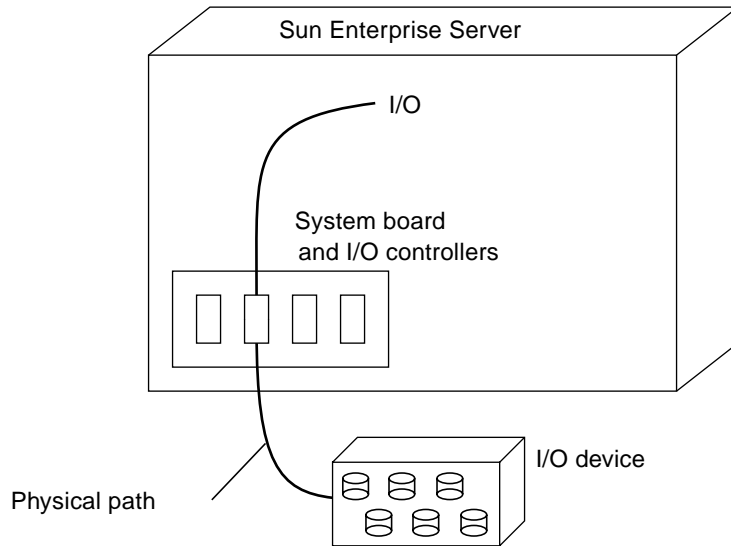
FIGURE 1-4   Physical Path

You reference a physical device by means of a device node, for example, `/dev/dsk/c0t1d1s0`.

Metadisk

A *metadisk*, as illustrated in FIGURE 1-5, is a construct that enables you to access a disk by using either of two physical paths without having to reference either path explicitly within your scripts and programs.

You reference a metadisk, in your scripts and programs, using an AP-specific device node such as `/dev/ap/dsk/mc0t1d1s0`. See "Device Nodes for Metadisks" on page 21 for more information.

In the following figure, an AP-specific device node is used to perform disk I/O, regardless of which pln port (pln:2 or pln:9) is currently handling I/O.
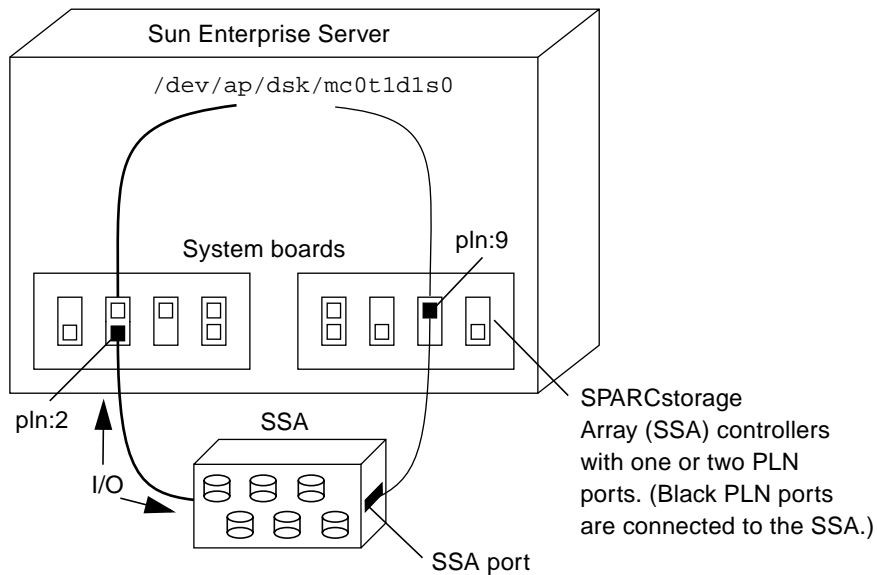
FIGURE 1-5   Metadisk Example

Path Optimization on a Sun StorEdge T3 Disk

At system startup for a T3, a path optimization algorithm is executed for disk path groups, whenever two functioning physical paths to the device are available. Disabling one of the physical paths through device failure or user action turns off path optimization for the affected disk path group. Users restart the path optimization algorithm using the apconfig(1M) command, or by performing a system reboot. Path optimization cannot be re-enabled until both physical I/O paths are available. For more information see "Working With Disk Path Groups and Metadisks" on page 29.

Metanetwork

A *metanetwork*, as illustrated in FIGURE 1-6, is a construct that enables you to access a network by using either of two physical paths without having to reference either path explicitly within your scripts and programs. You reference a metanetwork, in your scripts and programs, using a *metanetwork interface* name such as mether1. See "Metanetwork Interfaces" on page 45 for more information.

In the following figure, mether1 is used to access a metanetwork, regardless of which controller (hme1 or qfe3) is currently processing I/O for the metanetwork.
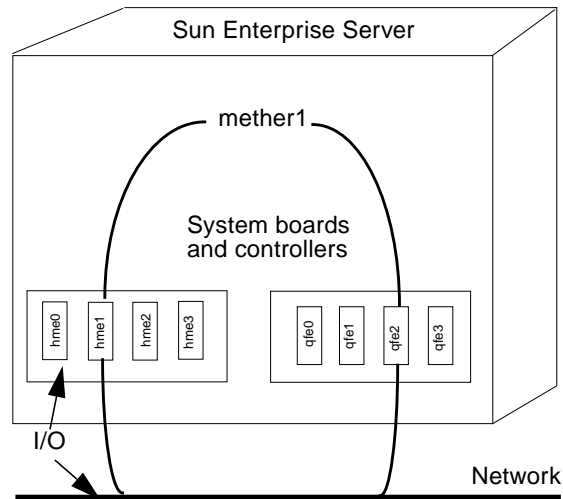
FIGURE 1-6   Metanetwork

Disk Path Group

A *disk path group*, as illustrated in FIGURE 1-7, consists of two physical paths leading to the same disk array. When a physical path is part of a path group, it is called an *alternate path*. An alternate path to a disk can be uniquely identified by the pln port or sf port that the alternate path uses. The alternate path that is currently handling I/O is called the *active alternate*.

Note – When a T3 disk path group is actively executing the path optimization algorithm, both physical paths are marked as *active alternate*s. The loss of a physical path, for whatever reason, disables path optimization. In that situation, only one path will be marked as an *active alternate*.

Note that whereas a metadisk (for example, /dev/ap/[r]dsk/mc?t?d?s?) provides a means for you to access an *individual disk*, in your scripts and programs, a disk path group provides a means for you to manipulate the *path* to that disk when you run AP commands. For example, to perform a *switch* operation, that is, change the active alternate from one alternate path to another, you reference a disk path group with an apconfig(1M) command.

Note – Initiating a switch operation automatically disables path optimization on the T3.

One of the alternate paths is designated as the *primary path*. Although the active alternate changes when you perform a switch operation, the primary path remains constant. You reference a disk path group by specifying the `pln` port (for example, `pln:2`) or `sf` port (for example, `sf:2`) that corresponds to the primary path. For information about determining the `pln` or `sf` port name, see "Device Nodes for Metadisks" on page 21.

To switch the active alternate of a disk path group, use:

```
# apconfig -P sf:2 -a sf:9
```

The following figure is an example of the results of using the `apconfig`(1M) command to switch the active alternate of a disk path group.
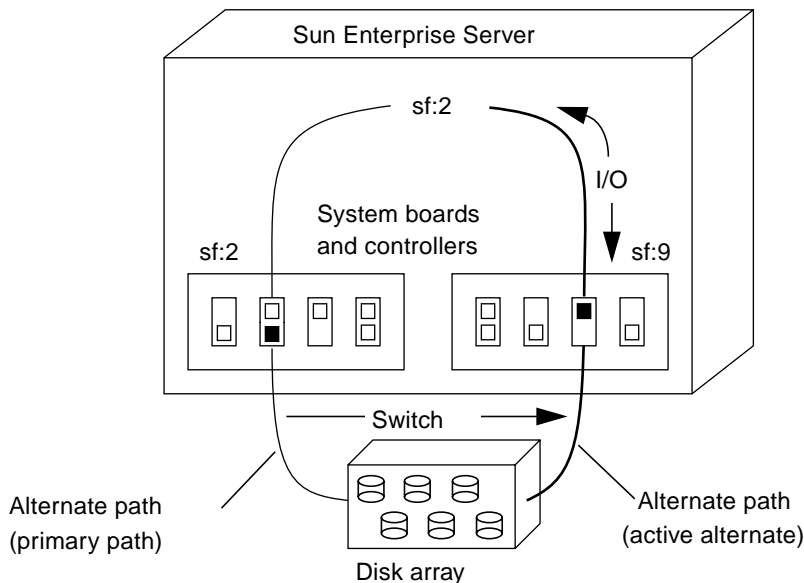
FIGURE 1-7 Disk Path Group Switch

Note – This action disables path optimization on the T3. To re-enable path optimization use:

```
# apconfig -P sf:2 -a sf:2 -a sf:9
```

Network Path Group

A *network path group*, as illustrated in FIGURE 1-8 consists of two network controllers connected to the same physical network. The terms *alternate path*, *active alternate*, and *switch* have basically the same meaning as they do for disk path groups. However, there is no *primary path* in a network path group.

To specify a network path group, reference the corresponding metanetwork interface name, for example, mether1. Metanetwork interface names are described in "Metanetwork Interfaces" on page 45. To switch the active alternate of a network pathgroup, use:

```
#apconfig -a mether1 -a hme1
```

For example, FIGURE 1-8 shows the results of using the apconfig(1M) command to switch the active alternate of a network path group.
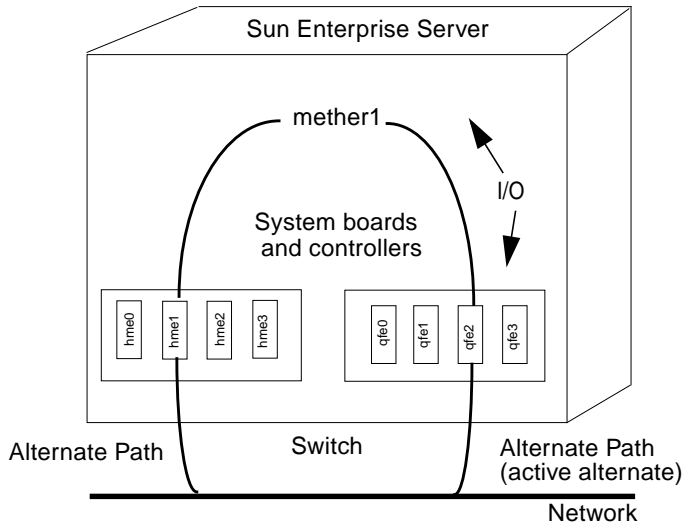
FIGURE 1-8   Network Path Group

Supported Software Versions

AP 2.3.1 supports the Solaris 2.6, Solaris 7, and Solaris 8 operating environments.

The disks, network devices and third party software products supported by AP are listed in the Release Notes Supplement Solaris 8 01/01.

If you have created alternate paths to your disks, and you also use a volume manager with those disks, the disks must be known to the volume manager, *exclusively*, by their AP metadisk names. This requirement enables AP to switch the active path without interfering with the volume manager.

You can place the boot disk and the primary network interface under AP control. AP makes it possible for the system to boot unattended even if the primary network or boot disk controller is not accessible, as long as viable alternate paths for these devices are defined.

AP Configuration Examples

FIGURE 1-9 shows how you can use AP to support an Ethernet network and a disk array.
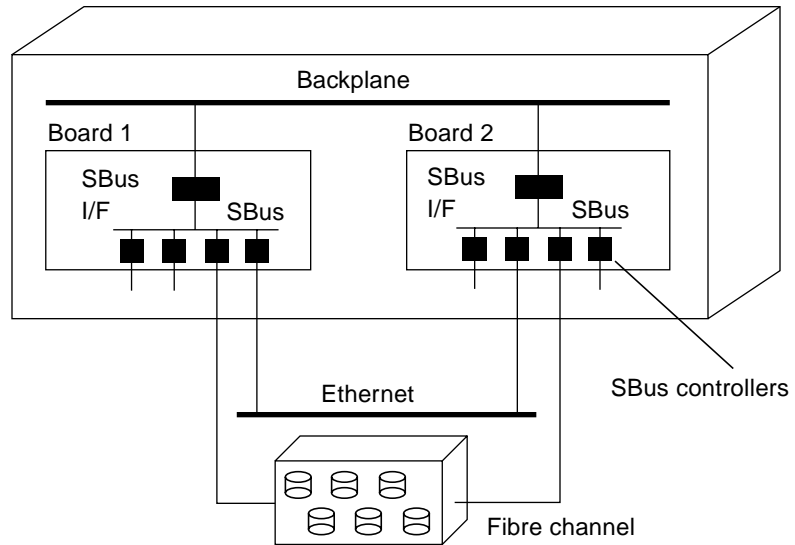
FIGURE 1-9 Typical AP Configuration

In this example, two network controllers—one each on Board 1 and Board 2—are connected to the same network. Similarly, two SSA controllers on the two boards are connected to the same SSA. In this situation, if Board 1 is detached with a DR Detach operation, AP can switch usage to Board 2 without interfering with any I/O operations that may be in progress.

AP is not equivalent to disk mirroring. Disk mirroring primarily achieves *data* redundancy although two paths are available, one for each side of the mirror.

AP achieves true *pathing* redundancy, by making two paths available for each side of the mirror. To use AP and disk mirroring together, you must configure your volume manager software (such as Sun Enterprise Volume Manager™ (SEVM)) so that it uses the AP metadisk paths.

The following figure shows an example of how AP can be used in conjunction with disk mirroring.
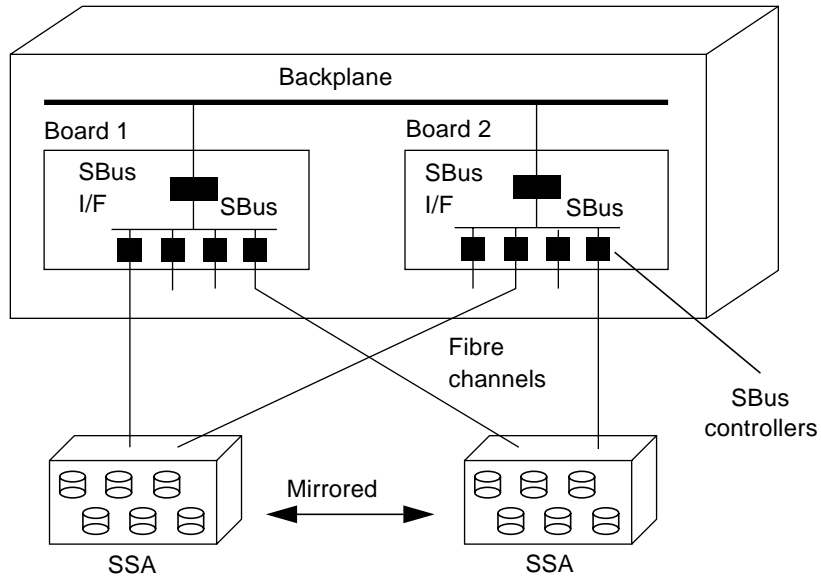
FIGURE 1-10 AP and Disk Mirroring

This type of configuration enables you to switch the paths used to implement the mirror from one board to another board, without disrupting the disk mirroring or any active I/O.

AP and Domains

All Sun Enterprise servers support domains. The Sun Enterprise 10000 server supports dynamic system domains. However, AP cannot be used across two domains.

For example, suppose a board contains a controller that is part of a path group, and you move that board into a different domain using DR. You can do this only if the alternate path on that board is not currently active. In this case, you can no longer switch to the alternate path on that board.

Alternate Pathing Database

This chapter describes how to create and manage the AP database, which maintains the state of the AP configuration.

Managing Copies of the Database

AP maintains a database that contains information about all defined metadisks and metanetworks, and their corresponding alternate paths and properties. Always set up multiple copies of the database. In this way, if a given database copy is not accessible or becomes corrupted, AP can automatically begin to use a current, noncorrupted database copy.

Caution – At least one current noncorrupted AP database must be available to an AP boot disk or the system will not boot.

You must dedicate an entire disk partition with at least 300 Kbytes to each database copy. Larger partitions waste disk space. Keep the following information in mind when selecting disk partitions for the AP database:

- Set up three to five database copies.
- As configured at the factory, partition four of the root disk is appropriately sized for the AP database and is not allocated for any other purposes. This partition is a good choice for an AP database copy, assuming you are not using it for other purposes.
- The database copies should have no I/O controllers in common with each other. Following this rule allows maximum availability in case of controller failure.
- If you have configured your system to make use of DR, the database copies should be hosted by I/O controllers on different system boards so that a database copy is accessible if one of the system boards is detached.

■ If you want to place an AP database copy in a partition of an alternately pathed disk, create one copy of the database twice, using each of the *physical path*s utilized by the AP metadisk to access the partition. AP behaves as if two copies of the database exist, when actually there is only one, since the disk is accessible by two paths. This behavior does not result in database inconsistencies, however, since AP always updates and accesses database copies sequentially. This behavior also does not result in performance problems since the AP database is not accessed frequently.

On the Sun Enterprise 10000 server versions of AP prior to AP 2.3, contained a subset of the information in the AP database on the SSP for use at boot time. This database contained AP information for the boot disk. If you are going to continue using versions of AP older than AP2.3:

1. DO NOT remove the SUNWapssp package from the SSP.

2. Verify that the version of SUNWapssp is the latest version for the latest version of AP you have prior to AP 2.3. For example, if you are running AP 2.0 in one domain and AP 2.1 in another then your SUNWapssp package should be for AP 2.1. Failure to have the latest version of your previous software running could result in losing the ability to boot an alternate path for an AP controlled boot disk, prior to UNIX boot.

Locating Databases to Maximize RAS

You should consider how you expect to use the system boards that host the I/O controllers for the disks where the AP databases will be stored. If you expect to detach a board often, perhaps because you intend to migrate it between domains, you should probably not place an AP database on any disk attached to a controller hosted by that board. If you do find it necessary to detach such a board, error messages will be sent to the console whenever AP attempts to write to the unavailable database. This does not pose a serious problem. You can reattach the board at any time, and the stale database is resynchronized immediately. However, if you attach the board to other domains in the meantime, those domains can write data to the slice that is reserved for the database.

Creating and Deleting Databases

The following AP examples assume that your command search path includes the directory where the commands are installed. See "Using Single-User Mode" on page 43.

▼ To Create a Copy of the AP Database

1. **Use** `apdb`**(1M):**

```
# apdb -c /dev/rdsk/c0t1d0s4
```

where:

`-c` specifies the raw disk slice (under `/dev/rdsk`) where you want to create the database copy. You must dedicate an entire disk partition to each database copy. The disk partition must have at least 300 Kbytes.

▼ To Delete a Copy of the AP Database

1. Use `apdb`**(1M):**

```
# apdb -d /dev/rdsk/c0t1d0s4 -f
# apconfig -D
#
```

where:

`-d` specifies the raw disk slice (under `/dev/rdsk`) where the copy of the database that you want to delete is located.

`-f` (force) is necessary only to delete the second-to-last copy and the last copy of the database.

In this example, `apconfig -D` is used to verify that the last database copy has been deleted. Normally, `apconfig -D` is used to view information about the existing AP database copies. Since no information is returned, the `apdb`(1M) command must have deleted the last database copy.

If you reboot after removing the last database, all AP metadevices will be unavailable. It's a good idea to deconfigure all AP metadevices before reboot, otherwise references to them (for example, `/etc/vfstab`) will be broken when the system comes back up. For more information see "To Deconfigure a Metadisk" on page 38 or "To Deconfigure a Metanetwork" on page 51.

Caution – If you delete your last database and your boot disk is alternately pathed, this means your system will become unbootable in the event of a system crash or reboot! Therefore, before you delete your last database, make sure you have taken your boot disk out from under AP control using apboot(1M) before rebooting. See "To Remove the Boot Disk From AP Control" on page 42.

Viewing Database Information

You can view information in the database, including information about the database copies, the disk entries within the database, and the network entries within the database.

▼ To View Information About Database Copies

1. Use `apconfig -D`:

```
# apconfig -D

path: /dev/rdsk/c0t1d0s4
major: 32
minor: 12
timestamp: Thu Jul 27 16:24:27 1995
checksum: 687681819
corrupt: No
inaccessible: No
```

In this example, there is only one AP database. The command shows the path to this database, along with its major number, minor number, timestamp, and checksum. The `corrupt` field indicates whether the database is corrupt. (If `corrupt` is set to `Yes`, the data did not validate properly against the checksum.) The `inaccessible` field indicates whether the device that holds the database can be accessed.

Viewing Path Group Information

The AP database contains information about disk and network path groups. When a path group is initially defined (as described in Chapter 3 and Chapter 5), its path group definition is an *uncommitted* database entry. The metadisk or metanetwork associated with an uncommitted entry is not available until the path group definition is *committed*. Conversely, when a path group definition is deleted, the deletion must be committed before it takes effect. The two states (uncommitted and committed) enable you to review the effects of an operation before allowing the operation to proceed. To commit the uncommitted database entries, use the `apdb -C` command.

Note – Uncommitted entries remain in the database indefinitely, until you either commit them or remove them. Upgrades are an exception. Upgrading AP software removes uncommitted entries.

▼ To View Uncommitted Disk Entries

1. Use `apconfig`**(1M) with the** `-S` **and** `-u` **options as follows, where** `-S` **stands for storage and** `-u` **stands for uncommitted:**

```
# apconfig -S -u

c1      sf:0  P A
c2      sf:1
        metadiskname(s):
                mc1t5d0  U
                mc1t4d0  U
                mc1t3d0  U
                mc1t2d0  U
                mc1t1d0  U
                mc1t0d0  U
```

For more information see Chapter 3.

▼ To View Committed Disk Entries

1. Use `apconfig`**(1M) with the** `-S` **option, as follows, where** `-S` **stands for storage:**

```
# apconfig -S

c1      pln:0  P A
c2      pln:1  A
        metadiskname(s):
                mc1t5d0  R
                mc1t4d0
                mc1t3d0
                mc1t2d0
                mc1t1d0
                mc1t0d0
```

For more information see Chapter 3.

▼ To View Uncommitted Network Entries

1. Use apconfig**(1M) with the** -N **and** -u **options, as follows, where** -N **stands for network and** -u **stands for uncommitted:**

```
# apconfig -N -u

metanetwork:    mether0  U
physical devices:
                hme2  A
                qfe0
```

For more information see Chapter 5.

▼ To View Committed Network Entries

1. Use apconfig**(1M) with the** -N **option, as follows:**

```
# apconfig -N

metanetwork:    mether3
physical devices:
                hme4 A
                qfe2
```

For more information see Chapter 5.

Using Metadisks and Disk Path Groups

You can create metadisks and disk path groups only for disks that are accessible through two paths. Generally, you use two separate controllers on different system boards.

Note – AP does not modify the data on a disk when that disk is placed under AP control or when a path group is deleted (except for the data on the slices that contain AP database copies). AP does not repartition a disk. If a path group is deleted, you can continue to access the data by using its physical device name.

Device Nodes for Metadisks

Here are two examples of *physical* device nodes for disk devices:

- `/dev/dsk/c0t0d0s0`
- `/dev/rdsk/c0t0d0s0`

where:

`c` references the I/O port on the host (not the disk array)

`t` is the bus within the disk array

`d` is the target ID of the disk on that bus

`s` is the slice number on the disk

These physical device nodes represent a particular physical path to a partition on a SCSI disk.

where:

`c` is the host adapter number

`t` is the target number of a disk tray

`d` is the disk number

`s` is the slice number on the disk

Each controller port has both a port number (such as `c0`) and a port name (such as `pln:2` or `sf:2`). The port name consists of the port type and *instance* number, delimited by a colon (:). This is a change from previous versions of AP port names. This change applies *only* to the disk driver naming convention and not to networks. Refer to `path_to_inst`(4) for more information on instance numbers.

When a disk array is connected to two ports, it can potentially be accessed from either path by using the physical device node, for example, `/dev/dsk/c0t0d0s0` or `/dev/dsk/c1t0d0s0`.

The device node for a metadisk is derived from the physical device node of the primary path for a path group. Here are two examples of metadisk device nodes:

- `/dev/ap/dsk/mc0t0d0s0`
- `/dev/ap/rdsk/mc0t0d0s0`

As you can see, an `ap` directory has been added, and an `m` (for "meta") is prepended to the device specification. The device node for a metadisk has the ability to access the underlying physical disk drive from multiple paths.

Automatic Switching of Metadisks

Metadisks can be automatically switched from an active path to an alternate path in two situations:

- The active path fails.
- The board containing the controller for the active path is detached by using a DR Detach operation. Automatic switching during a DR Detach is available only on the Sun Enterprise 10000 server.

For the Sun StoreEdge T3 disks, prior to any attempted DR activity, a manual switch should be initiated. This disables path optimization on a T3 disk. Later, when both physical paths are once again available, you can re-enable path optimization using:

```
# apconfig -P sf:2 -a sf:2 -a sf:9
```

When the active path fails, an automatic switch is attempted only if an alternate path is available. The failed path is then marked unavailable, or *tried*. You can identify the tried paths with apconfig-S:

```
# apconfig -S

c1       sf:0   P A
c2       sf:1   T
         metadiskname(s):
                  mc1t5d0
                  mc1t4d0
                  mc1t3d0
                  mc1t2d0
                  mc1t1d0
                  mc1t0d0
```

In this example, the currently inactive path, sf:1, is marked with a T, which indicates that path was tried but failed.

The tried flag is only significant for automatic switch operations, not for manual switch operations. Generally, AP 2.3.1 does not attempt to automatically switch to a tried path. This prevents thrashing in the case where both paths may have failed.

You can reset the tried flag with any of the following actions:

■ Rebooting the corresponding domain.
■ Performing a DR Detach followed by a DR Attach of a board that contains the controller marked as tried.
■ Manually resetting the tried flag for a particular controller.

You can manually reset the tried flag as shown in the following example:

```
# apdisk -w sf:1
```

In this example, sf:1 is a controller with the tried flag set to true. The apdisk-w feature should be used judiciously. This command merely clears the tried flag; it does not address any potential problems with the controller or device. This command should only be used in situations where the failed path has been restored without an intervening DR operation or reboot. Note that you can attempt to perform a manual switch to a path that is marked as tried.

Disk Availability and Performance Trade-Offs

Before you configure your disk arrays and controllers, you need to establish your disk usage priorities. You can achieve greater availability of your disk resources by trading off performance, or perhaps by investing in more hardware.

Consider a dual-ported SSA disk array. This type of device can be connected to either one or two Fibre Channel disk controllers (SOC controllers). Within an SSA, there are multiple targets. Each target contains multiple disks. And each disk is divided into multiple slices. Depending upon how you set up your system, there can be several different levels of contention for these disk I/O resources.

- Disk contention
- Target contention (I/O bus contention)
- Controller contention

For example, suppose you divide a disk into four slices, and create a file system out of the four slices. Although the file system spans several slices, those slices reside on the same disk and you may as well have simply placed the file system on a single slice. This is considered a poor configuration and it results in disk-level contention, since every read or write to that file system requires access to the same disk.

You can place a file system across several disks in the same target. In this situation you have target-level contention, since each read or write to the file system must access the same target. Target-level contention is not as severe as disk-level contention, but it is still a poor configuration.

If you create a file system across too many targets in the same SSA, you will have controller-level contention. This is because each read or write to the file system must use the same controller.

Generally, it is best to create a file system across multiple SSA disk arrays, using multiple controllers. However, there is a trade-off between disk access speed and system availability. The more disk arrays that you use for your file systems, the faster your disk access times will be. However, if any component fails in any of the disk arrays, your file systems will become unavailable. If you limit the number of disk arrays for a file system, perhaps to a single disk array, performance will decrease, but overall system availability should increase. This is because there are fewer components that could fail for a given file system.

Suppose you have six disk controllers connected to three dual-ported SSA disk arrays.

System Boards and Disk Controllers
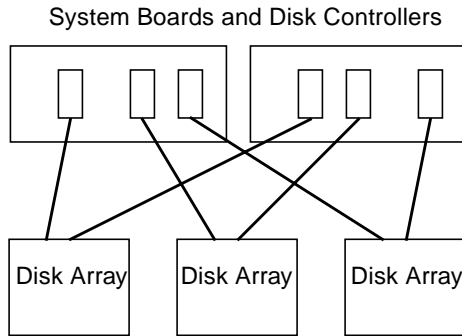
FIGURE 3-1 System Boards and Disk Controllers

If you want to maximize availability, you can alternately path each SSA using AP. The advantage of this is that you can use DR to attach and detach system boards (possibly to service or upgrade those boards) without losing access to the file systems on the disk arrays. Of course, to do this, you must place the alternate disk controllers (SOC controller) on different system boards. One arrangement is to use two system boards with three disk controllers on each board. This is a simple and clean arrangement, and it enables you to switch to the controllers on one of the boards when you need to detach the other board. It also enables you to migrate the disk resources among domains fairly easily, by detaching and attaching a single board.

Two SOC controllers must be purchased for each SSA. Also, in very large installations, it is possible to run into limitations in terms of the number of SBus slots that are available to host all the SOC controllers needed to dual path a large number of SSAs

Disk Mirroring Considerations

If you use a third party volume manager, such as SDS or VERITAS Volume Manager™(VxVM), to mirror your disks, and you also want to detach system boards with DR, you must configure your volumes and mirrors so that they will work properly with AP and DR.

For example, suppose you have 12 system boards, each of which has one host adapter (called a "controller" in the following diagram):
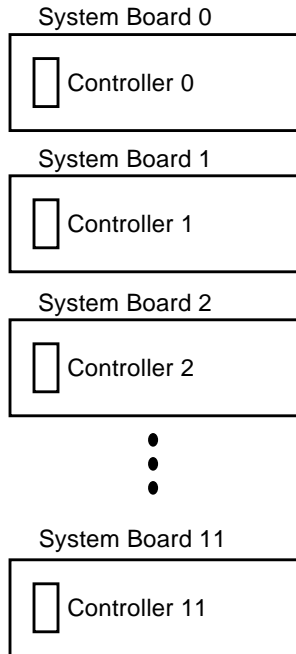
FIGURE 3-2 System Boards and Controllers

You may need to create a mirrored volume. Consider the following configuration:
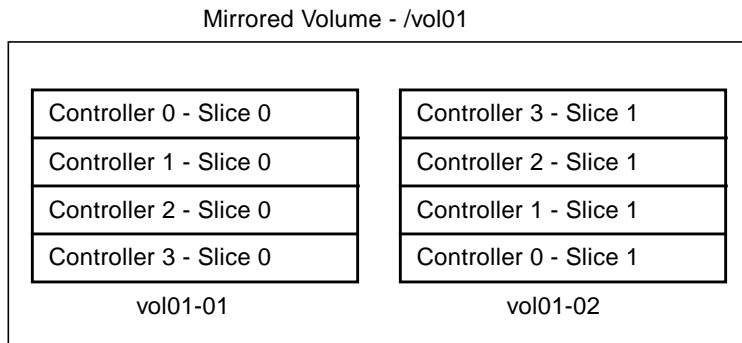
FIGURE 3-3 Mirrored Volumes Example 1

In Example 1, vol01-01 consists of a four-way slice that is accessed through four separate controllers that reside on four separate system boards. vol01-01 is mirrored to vol01-02 which also consists of a four-way slice. For example, Controller 0 Slice 0 is mirrored to Controller 3 Slice 1, and so forth.

Consider the situation where you need to detach a board that contains one of these four controllers. Before you can detach the board, you must deactivate the half of the mirror that uses controllers on that board. This is impossible with the configuration shown above. For example, if you wish to detach Board 0 (which contains Controller 0), you would have to deactivate *both* sides of the mirror, which makes the file system inaccessible. The result is that you cannot use DR on any of the boards in the configuration shown above.

One way around this situation is to mirror your volumes so that controllers from the same system board do not appear on both sides of the mirror, for example:

Mirrored Volume - /vol01

Controller 0 - Slice 0		Controller 4 - Slice 0
Controller 1 - Slice 0		Controller 4 - Slice 1
Controller 2 - Slice 0		Controller 5 - Slice 0
Controller 3 - Slice 0		Controller 5 - Slice 1
vol01-01		vol01-02
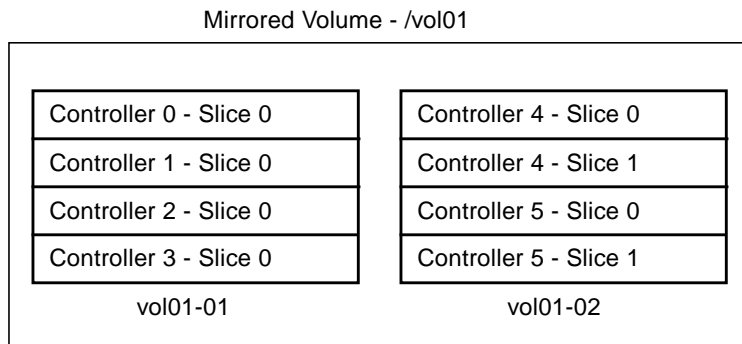
FIGURE 3-4   Mirrored Volumes Example 2

In Example 2, you can detach any board (Board 0 through Board 5) by first deactivating the half of the mirror that uses a controller on that board. For example, to detach Board 4 (which hosts Controller 4), simply deactivate vol01-02 first. You do not lose access to the file system, since it is still available through vol01-01. Later when you reattach Board 4, you can add vol01-02 to the mirror again.

However, the problem with this solution is that your system is vulnerable to single points of failure while the mirror is down. If a disk fails, there is no mirrored backup disk. You can protect against this by using AP. You might set up the following AP metadevices:

- `mc0` is the metadevice for Controller 0 and Controller 6
- `mc1` is the metadevice for Controller 1 and Controller 7
- `mc2` is the metadevice for Controller 2 and Controller 8
- `mc3` is the metadevice for Controller 3 and Controller 9
- `mc4` is the metadevice for Controller 4 and Controller 10

■ `mc5` is the metadevice for Controller 5 and Controller 11

Abbreviations are used above to simplify this discussion. For example, the full metadevice name may be `mc0t0d0s0`, and it may encapsulate the physical devices `c0t0d0s0` and `c6t0d0s0` as the alternate paths.

Now consider the following configuration:

Mirrored Volume - /vol01

Metadevice mc0	Metadevice mc4
Metadevice mc1	Metadevice mc5
Metadevice mc2	Metadevice mc4
Metadevice mc3	Metadevice mc5
vol01-01	vol01-02
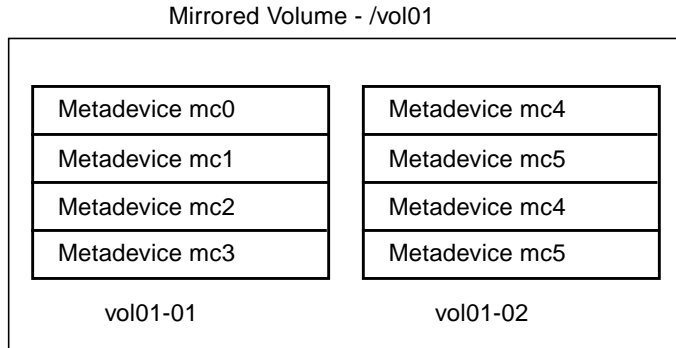
FIGURE 3-5   Mirrored Volumes Example 3

In Example 3, you can detach any of the boards (Board 0 to Board 11) without taking down the mirror. This reduces your exposure to single points of failure. For example, to detach Board 4, which contains Controller 4, you would first switch the metadevice `mc4` so that it uses Controller 10 on Board 10. (You do this with a single AP command, `apconfig -P`.)

In this example, as you increase the level of RAS support (that is, the availability of disk I/O resources and the level of protection against single points of failure), you have to increase the number of controllers and boards in the configuration. This amounts to an increase in the cost of the system to provide greater support for RAS features.

Although, this is a hypothetical example, the main point is that you must consider AP and DR when you set up your volumes and mirrors. Otherwise, you may end up in a situation where you cannot use AP and DR. If you use a third party volume manager, keep track of the *physical* controllers and slices that make up your volumes. Volume managers can be used in such a way that it chooses the physical components automatically, but this selection process is not aware of AP and DR considerations. Explicitly choose the physical components that make up your volumes if you want to assure compatibility with AP and DR.

Working With Disk Path Groups and Metadisks

Note – The example commands in this section use `pln` ports (for SSA disk arrays). If you have Sun StorEdge™ A5000 or T3 disk arrays/trays, you specify `sf` ports or `fp` ports (Solaris 8 environment only) wherever `pln` ports are shown. Some examples using `sf` ports for the T3 have been provided. For a list of Sun supported devices refer to *Sun Enterprise Server Alternate Pathing 2.3.1 Release Notes*

▼ To Create a Disk Path Group and Metadisk

1. **Decide which two ports will make up the alternate paths for the path group.**

   a. **You can use the** `apinst`**(1M) command to display all ports (for example,** `pln:0` **and** `pln:1`**) and their disk device nodes (for example,** `/dev/dsk/c1t0d0`**):**

```
# apinst
pln:0
        /dev/dsk/c1t0d0
        /dev/dsk/c1t1d0
        /dev/dsk/c1t2d0
        /dev/dsk/c1t3d0
        /dev/dsk/c1t4d0
        /dev/dsk/c1t5d0
pln:1
        /dev/dsk/c2t0d0
        /dev/dsk/c2t1d0
        /dev/dsk/c2t2d0
        /dev/dsk/c2t3d0
        /dev/dsk/c2t4d0
        /dev/dsk/c2t5d0
```

   b. **You must know your system hardware configuration to recognize when two ports are connected to the same disk array.**

   In this example, it is assumed that the SSA contains six disks and two SSA ports. One SSA port is connected to `pln` port `c1`, and the other SSA port is connected to `pln` port `c2`.

2. **Use** `apdisk`**(1M)) with the** `-c` **,** `-p`**, and** `-a` **options to create an uncommitted disk path group:**

```
# apdisk -c -p pln:0 -a pln:1
```

where:

`-p` specifies the primary path
`-a` specifies the alternate path
`-c` specifies that this information is to be created

This `apdisk`(1M) command creates a metadisk name, as well as all of the necessary information in the AP database for maintaining the two alternate paths for all six disks.

3. **Verify the results:**

```
# apconfig -S -u

c1      pln:0  P A
c2      pln:1
        metadiskname(s):
                mc1t5d0  U
                mc1t4d0  U
                mc1t3d0  U
                mc1t2d0  U
                mc1t1d0  U
                mc1t0d0  U
```

The `apconfig -S -u` command lists the uncommitted metadisks.

where:

`-S` lists storage devices only (that is, disks rather than networks)
`-u` lists uncommitted devices only

The `U` next to each metadisk name indicates that the metadisk entry is uncommitted.

The `P` next to `pln:0` indicates that `pln:0` is the primary path `A` indicates that `pln:0` is the active alternate.

Although you can change the active alternate, the primary path always remains constant. The significance of the primary path is that it is initially the active alternate, it is used when the metadisk is named, and it is used to identify the metadisk. In this case, `c1t0d0` from the primary path name becomes part of `mc1t0d0` in the metadisk name.

4. **If you are satisfied with the results shown in the previous step, use** apdb**(1M)**
   with the -C **option to commit the uncommitted database entries:**

```
# apdb -C
```

5. **Before continuing, verify the results by using** apconfig **(1M) with the** -S **option**
   to view the committed storage entries in the database:

```
# apconfig -S

c1      pln:0  P A
c2      pln:1
        metadiskname(s):
                mc1t5d0
                mc1t4d0
                mc1t3d0
                mc1t2d0
                mc1t1d0
                mc1t0d0
```

Note – On T3 disk, when both paths are available, path optimization for the path
group is created by default. The above command would output the following
results:

```
# apconfig -S

c1      sf:0  P A
c2      sf:1 A
        metadiskname(s):
                mc1t5d0
                mc1t4d0
                mc1t3d0
                mc1t2d0
                mc1t1d0
                mc1t0d0
```

If a partition is currently mounted under a physical path name, it should be
unmounted and remounted under the metadisk path name.

If you do not want to unmount a partition, perhaps because it is heavily used, you
can delay placing the partition under AP control until you are ready to bring the
system down for maintenance and then reboot. In this scenario, you modify the

/etc/vfstab file so that when the system is rebooted, the partition comes up under an AP device. If you are placing the boot disk under AP control, you also need to modify /etc/vfstab using apboot(1M) as described in Chapter 4.

Use apconfig-S to list the committed storage entries in the database. As shown in the example, this listing is exactly the same as the previous listing, except the U no longer appears after the metadisk names, indicating that the metadisks are no longer uncommitted.

6. **Use the following command to view the symbolic links and verify the results:**

```
# ls -l /dev/ap/dsk
total 8
lrwxrwxrwx  1 root              40 Jul 27 16:47 mc1t0d0s0 ->
    ../../../devices/pseudo/ap_dmd@0:128,blk
lrwxrwxrwx  1 root              40 Jul 27 16:47 mc1t0d0s1 ->
    ../../../devices/pseudo/ap_dmd@0:129,blk
lrwxrwxrwx  1 root              40 Jul 27 16:47 mc1t0d0s2 ->
    ../../../devices/pseudo/ap_dmd@0:130,blk
```

The device nodes that you need—under /dev/ap/dsk as well as /dev/ap/rdsk—are now ready to be used.

7. **Modify every reference that uses a physical device node (that is, a path that begins with** /dev/dsk **or** /dev/rdsk**) to use the corresponding metadisk device node (that is, a path that begins with** /dev/ap/dsk **or** /dev/ap/rdsk**).**

▼ To Switch From the Primary Path to the Alternate Path

You can perform a switch at any time, even while I/O is occurring on the device. You can experiment with the switching process to verify that you understand it and that your system is set up properly, rather than wait until a critical situation occurs.

Caution – When you switch paths, AP does not check to verify that data can be transferred over the path to which you are switching, although it does determine whether or not that path is detached or offline. Verify the status of the path before switching to it by performing an I/O operation such as prtvtoc(1M). AP does not produce any error or warning messages if you switch to a path that is not functioning properly. If you switch to a nonfunctioning path for your boot disk, your system may crash if the path is not switched back immediately.

1. **Use the** `apconfig`**(1M) with the** `-S` **option to view the current configuration:**

```
# apconfig -S

c1      pln:0   P A
c2      pln:1
        metadiskname(s):
                mc1t5d0
                mc1t4d0
                mc1t3d0
                mc1t2d0
                mc1t1d0
```

In this example, `pln:0` is the active alternate since it is followed by an `A`. It is also the primary path, since it is followed by a `P`.

Output for the T3 would look like this:

```
# apconfig -S

c1      sf:0   P A
c2      sf:1
        metadiskname(s):
                mc1t5d0
                mc1t4d0
                mc1t3d0
                mc1t2d0
                mc1t1d0
```

In this example, `sf:0` and `sf:1` are active alternates since they are followed by an `A`. The primary path is also `sf:0`, since it is followed by a `P`.

Note – Path optimization on a T3 disk is initiated by default when two paths are available. It is possible to have a single path, in which case, in the above example,`sf:1` would not be listed as an active alternate.

---

2. **To perform the switch, use** `apconfig`**(1M) with the** `-P` **and** `-a` **options:**

```
# apconfig -P pln:0 -a pln:1
```

Note – This action will disable path optimization on the T3s, if it was previously enabled.

-P specifies the primary path and thereby *identifies the path group* for which you want to change the active alternate. Thus, -P pln:0 in the example above identifies the path group for which pln:0 is the primary path.

-a specifies the alternate that you want to make active.

3. **You can verify the results using** apconfig**(1M) with the** -S **option to view the committed metadisks in the database:**

```
# apconfig -S

c1      pln:0  P
c2      pln:1  A
        metadiskname(s):
                mc1t5d0
                mc1t4d0
                mc1t3d0
                mc1t2d0
                mc1t1d0
```

Note – Only one path is active after an AP switch. Path optimization for the T3 has been disabled.

The active alternate has been switched to pln:1.

Note that you do not have to commit a switch operation.

▼ To Switch Back to the Primary Path

1. You can switch back to the primary path using the following commands:

```
# apconfig -P pln:0 -a pln:0
# apconfig -S

c1      pln:0   P A
c2      pln:1
        metadiskname(s):
                mc1t5d0
                mc1t4d0
                mc1t3d0
                mc1t2d0
                mc1t1d0
```

Note – Path optimization for a T3 is still disabled. Output for the previous command would look like this:

```
# apconfig -P pln:0 -a pln:0
# apconfig -S

c1      sf:0    P A
c2      sf:1
        metadiskname(s):
                mc1t5d0
                mc1t4d0
                mc1t3d0
                mc1t2d0
                mc1t1d0
```

The first apconfig(1M) command switches the active alternate for the path group that has the primary controller pln:0. The active alternate becomes pln:0.

2. To re-enable path optimization on a T3 use:

```
# apconfig -P sf:0 -a sf:0 -a sf:1
```

▼ To Delete Disk Path Groups and Metadisks

1. **If your boot disk is under AP control, use** `apboot`**(1M) to remove it from AP control, as described in "To Remove the Boot Disk From AP Control" on page 42.**

   You do not need to unmount any file systems mounted from the boot disk, since `apboot`(1M) places those file systems on top of physical devices without requiring you to unmount them.

2. **Unmount any file systems that are built on top of AP metadisks (other than file systems mounted from the boot disk).**

   Your scripts and programs may contain references to metadisks of the form:

   `/dev/ap/dsk/mc?t?d?s?` and `/dev/ap/rdsk/mc?t?d?s?`

   These references must be converted to appropriate physical device references of the form, respectively:

   `/dev/dsk/c?t?d?s?` and `/dev/rdsk/c?t?d?s?`

   Typically, references to metadisks exist in the following locations:

   `/etc/vfstab`

   `/etc/system`

   `/etc/dumpadm.conf`

   Any application or script that references disks

3. **Use** `apdisk`**(1M) with the** `-d` **option to specify the primary path of the path group you intend to delete:**

   ```
   # apdisk -d pln:0
   ```

4. **To verify the results, use** `apconfig`**(1M) with the** `-S` **option to view the committed disk entries in the database:**

```
# apconfig -S

c1      pln:0  P A
c2      pln:1
        metadiskname(s):
                mc1t5d0  D
                mc1t4d0  D
                mc1t3d0  D
                mc1t2d0  D
                mc1t1d0  D
                mc1t0d0  D
```

If the path group was not previously committed, the `apdisk-d` command deletes it from the database. However, if the path group was previously committed, the `apdisk-d` command simply marks it as deleted, but the deletion is not completed until the next time you commit the entries in the database. In the example above, the `pln:0` path group was previously committed, so the letter `D` indicates that it is marked for deletion.

5. **Use** `apdb`**(1M) to commit the database entries, thereby completing the deletion:**

```
# apdb -C
```

6. **You can verify the deletion with** `apconfig`**(1M) using the** `-S` **option:**

```
# apconfig -S
```

Note – You can undo a deletion if the deletion is uncommitted. To undo a deletion, use `apdisk -z`, specifying the same port that you previously specified.

▼ To Deconfigure a Metadisk

1. Convert your script references from this form:

`/dev/ap/dsk/mc?t?d?s?` and `/dev/ap/rdsk/mc?t?d?s?`

To this form respectively:

`/dev/dsk/c?t?d?s?`and `/dev/rdsk/c?t?d?s?`

Typically, references to metadisks can exist in the following locations:

`/etc/vfstab`

`/etc/system`

`/etc/dumpadm.conf`

And any application or script that references your configured metadisks

▼ To Reconfigure a Metadisk

This procedure assumes that you previously created a disk path group and metadisk, and subsequently deconfigured the metadisk references. If you simply want to reconfigure the metadisk interface, use this procedure.

1. Convert physical device references back into metadisk references; from this form:

`/dev/dsk/c?t?d?s?` and `/dev/rdsk/c?t?d?s?`

To this form, respectively:

`/dev/ap/[r]dsk/mc?t?d?s?`

Typically, references to disk devices can exist in the following locations:

`/etc/vfstab`

`/etc/system`

`/etc/dumpadm.conf`

And any application or script that references disks or any other application or script that references physical disk devices now under the control of the newly configured metadevice

Using AP Boot Devices

This chapter describes how you can alternately path the boot disk.

Placing the Boot Disk Under AP Control

You can now allow for unattended system boot, on all Sun Enterprise servers, even if the controller for the boot disk fails, by placing your boot disk under AP control.

You can use dynamic reconfiguration (DR) to detach a system board, on all Sun Enterprise servers, even if that system board hosts a controller for the boot disk. To do this, you must alternately path the boot disk using controllers from two different system boards. Note, however, if a controller for the primary network is hosted on the same system board as a controller for the boot disk, you must alternately path the primary network as well. Otherwise, you will not be able to use DR to detach that board.

▼ To Place a Boot Disk Under AP Control

1. **Create an AP path group for the boot disk.**

   This process is described in Chapter 3.

2. **Use** `apboot`**(1M) to define the new AP boot device.**

   `apboot`(1M):

   - Modifies `/etc/vfstab`, `/etc/system` and `/etc/dumpadm.conf`.

   For example:

   ```
   # apboot mc2t0d0
   ```

where:

`mc2t0d0` is the metadisk name of the boot disk.

- Examines `/etc/vfstab` and replaces the physical device name of the disk (for example, `/dev/ap/dsk/c2t0d0*` or `/dev/dsk/c1t0d0*`) with the metadisk name `/dev/ap/dsk/mc2t0d0*`.

Do not manually replace the physical devices in `/etc/vfstab` with metadisks for the boot disk. Instead, use `apboot`(1M) to ensure that all required changes are made.

- Checks `/etc/vfstab` to determine if the swap device is to be changed to a metadevice. If so, it converts the swap device to a metadevice.
- Edits `/etc/system` so that the kernel drivers required for AP boot disk usage are loaded at the proper time.
- Checks the configuration of the dump device, and calls `dumpadm`(1M), if necessary, to configure the dump device as a metadevice.
- Updates the boot device property of OpenBoot™ PROM to list the physical paths for each alternate.

Note – If you choose to suppress this feature, (using `apboot -o`) an automatic reselection of the alternate path of a UNIX controlled boot disk prior to UNIX boot will be disabled.

---

3. Place boot-mounted file systems under AP control.

Normally, the file systems that are mounted as part of the boot process are split across two separate disks (because of disk space requirements). If you place the boot disk under AP control (using `apboot`(1M)), you *must* manually edit the `/etc/vfstab` file to also place other file systems that are mounted during the boot process under AP control.

In the `/etc/vfstab` file, you must change the `device to mount` and `device to fsck` paths for all other mount points that you want to place under AP control.

For example:

```
# device               device               mount    FS    fsck
mount      mount
# to mount             to fsck              point    type  pass
at boot    options
#...
/dev/ap/dsk/mc1t34d0s1  -                    -        swap   -
no         -
/dev/ap/dsk/mc1t34d0s0  /dev/ap/rdsk/mc1t34d0s0 /    ufs    1
no         -
/dev/ap/dsk/mc1t34d0s6  /dev/ap/rdsk/mc1t34d0s6 /usr ufs    1
no         -
/dev/ap/dsk/mc1t34d0s7  /dev/ap/rdsk/mc1t34d0s7 /export/home
ufs    2    yes   -
swap                    -                    /tmp     tmpfs  -
yes        -
#...
```

4. At this point, reboot the system to begin using the AP boot device.

▼ To Alternately Path a Mirrored Boot Disk

Mirroring the boot disk is primarily a function of your disk management software. The purpose of this procedure is to notify AP about a mirrored boot disk. When you use mirrored boot disks that are also alternately pathed, you have four potential physical paths to the boot disk, two for each side of the mirror. This is the suggested configuration if you want to maximize protection against controller failure. There are two benefits to performing the procedure that follows.

- AP assures that the appropriate alternate path is always designated as the active path, even if you boot using a different boot device path. For this to work, you must first put the boot disk under AP control. Then you create a path group for the mirror of the boot disk.

- AP also assures that all four paths are available as alternates should an automatic switch be required at boot time.The default for the paths in a mirrored system are: primary1, mirror1, primary2, mirror2. This is a change from version AP 2.2 and earlier to improve redundancy and serviceability. The default order in an alternately pathed, nonmirrored system is: primary root, alternate root.

1. **Place the boot disk under AP control, as described in "To Place a Boot Disk Under AP Control".**

2. **Create an AP path group for the mirror of the boot disk.**

   This process is described in Chapter 3.

3. **Notify AP about the boot disk mirror.**

   ```
   # apboot -m mc3t0d0
   ```

   In this example, `mc3t0d0` is the metadisk for the mirror of the boot disk.

4. **Create a mirror of your boot disk (using the two metadisks) with your disk management software.**

▼ To Remove a Mirrored Boot Disk From AP Control

1. **Use `apboot`(1M) to undefine the AP mirrored boot device.**

   ```
   # apboot -u mc3t0d0
   ```

▼ To Remove the Boot Disk From AP Control

1. **Use `apboot`(1M) to specify an appropriate physical device node.**

   ```
   # apboot c2t0d0
   ```

   In this command, `c2t0d0` is the physical device node of an alternate path for the boot disk (as currently specified in `/etc/vfstab`).

   - `apboot`(1M) edits the `/etc/system` file to remove force loading of AP kernel driver modules, since they are no longer needed when the boot disk is not an AP device.
   - `apboot`(1M) reconfigures swap, dump devices, and the OpenBoot PROM `boot-device` property to use appropriate physical device paths, if necessary.

> **⚠**
>
> **Caution –** If you place the boot disk under AP control, and later decide to remove the AP package (using pkgrm(1M)), you must first use apboot(1M) to remove the boot disk from AP control. If you do not remove the boot disk from AP control prior to the pkgrm of AP, the disk becomes unbootable.

AP Boot Sequence

This section briefly describes the flow of events that occur when a Sun server is booted on an alternately pathed boot disk. This sequence of events illustrates how auto-switching of the boot disk controller is achieved during the boot process, if such a switch is necessary. The boot sequence proceeds as follows:

1. By default, the system is booted from the first device specified in the OpenBoot™ (OBP) `boot-device` property. Note that this device may be different from the last active alternate for the boot disk.

2. A failure to boot from the first device is detected after a few seconds up to a few minutes (less than three) depending on your firmware. OBP then goes to the next listed boot device. This process continues until a device boots or OBP runs out of devices.

3. After the reboot succeeds, AP makes the successful device the active alternate.

Using Single-User Mode

Normally, when the Sun Enterprise server is fully booted, you use the AP command versions located in `/usr/sbin`. However, if your server comes up in single-user mode because the boot process did not fully complete, you can use the AP commands in `/sbin`. The AP command versions under `/sbin` do not rely on the AP daemon services (which are not available in single-user mode). If the system enters single-user mode because of a problem related to AP, you may be able to resolve the problem by using the `/sbin` commands to perform needed AP operations.

Two AP-related problems can cause the system to come up in single-user mode.

- If two paths are supposed to lead to the same disk (according to the AP database) but those paths actually lead to different disks, and if that disk needs to be mounted during the boot process, the server will not find it and come up in single user mode. This can happen only if you change the physical configuration of the path group without running AP commands to update the database.

- If an active alternate for a disk turns out to be inaccessible and that disk is required during the boot process the server will come up in single user mode. A disk is required during the boot process if it has file systems that are mounted during the boot process; that is, it has entries in the `/etc/vfstab` file.

These situations arise only with respect to disks, not networks. In either case, you can use the AP commands under `/sbin` to resolve the problem.

Using Metanetworks and Network Path Groups

To use AP metanetworks, both physical networks within a network path group must be of the same media type and on the same subnet. For example, a network path group can consist of two Ethernet networks or two FDDI networks, but not one of each. Within an Ethernet network you can have different types of Ethernet. For example, `hme` and `qfe` can belong to the same path group.

Both alternates in a network path group must be physically connected to the same network. For example, Ethernet controllers must be connected to the same subnet.

While multiple physical network connections exist, only one controller at a time is active. The controllers must be on different system boards for DR operations (such as DR Detach) to be performed without affecting all potential active alternates.

The AP switch procedures in this section show how to switch the active alternate.

Metanetwork Interfaces

A metanetwork interface name is derived from the type of network to which the alternates belong. An Ethernet metanetwork interface name has the form `mether`*x*, where *x* is the instance number; for example, `mether0`. An FDDI metanetwork interface name has the form `mfddi`*x*, where *x* is the instance number; for example, `mfddi0`.

You must use two network interfaces of the same media type when creating a metanetwork interface; for example, you can use `hme0` and `qfe2`, or `nf0` and `nf1`. However, you cannot use `hme0` and `nf1`. Some examples follow.

- Assume the network controllers `hme0` and `qfe1` connect to the same Ethernet subnet. A metanetwork `mether0` can encompass these two controllers. Ethernet controllers of all types can be mixed and matched: `hme`, `qfe`, `le`, and so on, as long as they are on the same subnet.
- FDDI networks can be either SAS or DAS. SAS and DAS configurations can be mixed when creating a metanetwork interface.

Working With Network Path Groups

▼ To Create a Network Path Group and Metanetwork

Note – This procedure should *not* be used for the primary network. To alternately path the primary network, see "Alternately Pathing the Primary Network Interface" on page 53.

---

1. **Use** `apnet(1M)` **with the** `-c` **option:**

```
# apnet -c -a hme0 -a qfe2
# apconfig -N -u

metanetwork:    mether0  U
physical devices:
                hme0 A
                qfe2
```

This `apnet`(1M) command creates the network path group as well as the metanetwork interface name `mether0` for the two physical devices `hme0` and `qfe2`.

`apconfig`(1M) lists the uncommitted network entries in the database.

`-N` specifies that network database entries should be listed.

`-u` specifies that uncommitted entries should be listed.

2. **If you are satisfied with the network path group, commit the entry:**

```
# apdb -C
# apconfig -N

metanetwork:     mether0
physical devices:
                 hme0 A
                 qfe2
```

apdb –C commits the database entries.

apconfig –N lists the committed network entries in the database.

The listing appears exactly as it did before, except that the U no longer appears after mether0.

3. **Remove all direct usage of both members of the path group (see** ifconfig**(1M)).**

a. **If the physical interface is currently 'plumbed',** unplumb **the physical interface unless:**

■ It is not the primary network interface.

■ It is not the interface that you will be using to configure the metanetwork.

If the interface you will unplumb is the primary network interface, or if it is the interface that you will be using to configure the metanetwork, follow one of the procedures in "Alternately Pathing the Primary Network Interface" on page 53.

You can unplumb the physical interface as shown in the following example:

```
# ifconfig hme0 down unplumb
```

Usually network interfaces are configured during system boot using the file /etc/hostname.*xxxx*, where *xxxx* is the interface name (such as hme0). This file contains the IP address or the hostname associated with the interface. You should remove or rename the /etc/hostname.*xxxx* for all interfaces that have been made AP alternates, since direct usage of the alternate must not occur.

Note – IPv6: Regarding AP, you can use hostname6.*xxxx* files wherever hostname.*xxxx* is used. If you have both IPv4 and IPv6 on your system, you need to make sure that the entries in each file are consistent with one another. For more information on IPv6, refer to *System Administration Guide, Volume 3*

---

b. Create an `/etc/hostname.mether`*x* **file, (such as** `/etc/hostname.mether0`**)**
for any metanetworks that you want to configure at system reboot.

This file should contain the metanetwork IP address or the hostname for the
interface. You can simply rename the files:

```
# mv /etc/hostname.hme0 /etc/hostname.mether0
```

The normal operating state of a network interface is plumbed up when in use,
and unplumbed when not in use. When you automatically configure network
interfaces through `/etc/hostname.*`, the interfaces are left in one of these
states. It is possible to leave a network interface in a transitory plumbed state
when you manually configure your network interface. As this is not a normal
operational mode, it is unlikely that network interfaces will be left in this state.

Do not leave metanetworks in this state during AP network configuration.

A network metadevice can be deleted only if it and all other network metadevices
of that device type are either in the unplumbed state or the plumbed state.
Otherwise, AP ignores the delete request, and depending on your configuration,
may display warning messages of the following form:

```
WARNING:mether_setphyspath: APUNSET busy
WARNING:ap_db_commit: mfddi3 not deleted, metadevice returned
error 16
```

**c. If you are using FDDI, you must specify a unique MACID for the
metanetwork.**

The MACID is set through the `ether` parameter to the `ifconfig`(1M) command.
First examine the MACID for each of the alternates. You can do this by bringing
up each alternate and examining the `ether` field. Then, fabricate a MACID that
does not match any of the alternates.

Note – The allocation of Media Access Control Identifiers (MACIDs) is described in IEEE Std. 802-1990, as well as, RFC1340, "Assigned Numbers", July 1992. When generating a MACID for your AP network interface, the new 48-bit hardware address should be acquired from the IEEE Standards Office, 345 East 47th Street, New York, N.Y. 10017. However, it is possible to "create" a number by adding 2 to the first byte of an existing MACID of one of the metainterface alternate elements, (for example, `8:0:20:xx:xx:xx` becomes `A:0:20:xx:xx:xx`.) After creating a number, it is important to verify that there is no other hardware on the same subnet that is a legitimate user of the created address.

The following is an example:

```
#!/sbin/sh
/sbin/ifconfig mfddi0 ether A:0:20:68:6d:62
```

Once you have created `S19macid`, set the attributes to 744(rwxr--r--) using the `chmod` command.

This metanetwork MACID is used to configure the active physical interface of the metanetwork. The use of this MACID is necessary to prevent duplication of MACIDs on the network when combining AP switching of interfaces and DR board insertion activities.

The metanetwork defaults to the MACID of the active alternate at boot time. To ensure that the MACID is set properly at boot time, as superuser, create the following file: `/etc/rcs.d/S19macid`.

```
#!/sbin/sh
/sbin/ifconfig mfddix ether mfddix_macid
```

Replace `mfddix` with the correct metanet device number (use `apconfig-N` to obtain it).

Replace `mfddix_macid` with actual Ethernet numbers.

4. **Bring up the metanetwork in the usual manner, but use the metanetwork name instead of the physical network name. You can do this by either rebooting the machine or manually configuring the network as in the following example:**

```
# ifconfig mether0 plumb
# ifconfig mether0 inet 136.162.65.30 up netmask + broadcast +
Setting netmask of mether0 to 255.255.255.0
# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
mether0: flags=843<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 136.162.65.30 netmask ffffff00 broadcast 136.162.65.255
    ether 0:0:be:0:8:c5
```

At this point, the device node, such as `/dev/mether` can be used to access the network from Solaris commands such as `snoop`(1M).

▼ To Switch a Network Path Group

Note – You can switch a network path group even while the network sustains traffic.

---

1. **Use the `apconfig`(1M) command:**

```
# apconfig -P mether0 -a hme2
# apconfig -N

metanetwork:    mether0
physical devices:
                hme0
                hme2  A
```

where:

`-P` specifies the path group

`-a` specifies the alternate that you want to become active.

The listing above shows that the active alternate has been switched to `hme2`, as indicated by the `A` following `hme2`.

You do not have to commit a switch operation.

▼ To Delete a Network Path Group and Metanetwork

1. **Remove all usage of the corresponding metanetwork, and use** `apnet`**(1M) with the** `-d` **option:**

```
# ifconfig mether0 down unplumb
# apnet -d mether0
# apconfig -N

metanetwork:     mether0  D
physical devices:
                 hme0
                 hme2  A
```

In the listing produced by `apconfig -N`, a D follows `mether0`, indicating that the path group is marked as deleted.

2. **Commit the entries in the database using** `apdb`**(1M) with** `-C` **option:**

```
# apdb -C
# apconfig -N
#
```

The `apconfig -N` command produces no listing, indicating that the network path group (the only one that existed previously in this example) has been deleted.

You can undo a deletion if the deletion is uncommitted. To undo a deletion, use `apnet -z`, specifying the same metanetwork interface that you previously deleted.

When an `apnet -m -r` or `apnet -m -a` command is used, AP marks the current path group configuration as deleted and creates a new uncommitted path group definition.

Once the database change is committed using `apdb -C`, the new definition replaces the old.

3. **Remove the** `/etc/hostname.mether`*x* **file, as described in "To Deconfigure a Metanetwork".**

▼ To Deconfigure a Metanetwork

Caution – This procedure requires you to reboot the machine. If you are not ready to reboot the machine, do not perform this procedure.

1. Verify that the primary network interface is mether0 **(in this example):**

```
# cat /etc/nodename
eng2
# cat /etc/hostname.mether0
eng2
#
```

2. Rename the hostname.*xxxx* **file so the network will be automatically configured at boot time:**

```
# mv /etc/hostname.mether0 /etc/hostname.qfe0
```

3. Reboot.

▼ To Reconfigure a MetaNetwork

> **Caution –** This procedure requires you to reboot the machine. If you are not ready to reboot the machine, do not perform this procedure.

1. Verify that the primary network interface is qfe0 **(in this example):**

```
# cat /etc/nodename
eng2
# cat /etc/hostname.qfe0
eng2
#
```

2. Rename the hostname.*xxxx* **file so the network will be automatically configured at boot time:**

```
# mv /etc/hostname.qfe0 /etc/hostname.mether0
```

3. Reboot.

Alternately Pathing the Primary Network Interface

The primary network interface between your Sun Enterprise server and the other machines on the network is the interface that carries the address associated with the hostname of that server. One way to identify the primary network is to look in the `/etc/hostname.mether`*x* files until you find the one that contains the host that matches the hostname found in the file `/etc/nodename`. The corresponding `mether`*x* network (for example, `mether0`) is the primary network.

You may alternately path the primary network. The primary network is the only network interface that can be auto-switched at boot-time. During the boot process, if the active alternate for the primary network fails, the system attempts to find a working alternate for the network.

When you configure an alternately pathed network, you must not configure the metanetwork while the underlying driver is still active.

When you configure AP for a network that you are currently using, the transition period between configuring the physical interface down and the AP interface up generates a loss of network service for your Sun Enterprise server.

To perform this transition, you should use one of the following procedures, which are shown in order of preference:

- Create the appropriate AP database entries, create a new `/etc/hostname.`*xxx* file, remove (or rename) the corresponding `/etc/hostname.`*xxx* file, and then reboot your Sun Enterprise server. This approach is shown as detailed examples in the following section, "Configuring AP for a Current Network".
- Set up a script file to perform the transition on your Sun Enterprise server.
- Log in to your Sun Enterprise server by another network interface to enable commands to be issued when the network service is lost on the network interface that you are bringing up under AP.

Configuring AP for a Current Network

The following examples show the preferred way to configure AP for the primary network you are currently using. This example assumes you have an Sun Enterprise server named `eng5` with a primary network interface on `mether0`, and that you want to have a metanetwork interface composed of `qfe0` and `hme2`. If you are unsure which network interfaces should be combined into a metanetwork, you can use `snoop -d` to determine which of your configured networks are on the same subnet.

▼ To Create a Network Path Group and Metanetwork for the Primary Network

> **Caution –** This procedure requires you to reboot the machine. If you are not ready to reboot the machine, do not perform this procedure.

> **Note –** IPv6: replace `hostname.`*xxxx* with `hostname6.`*xxxx* in all examples.

1. **Verify that the primary network interface is** `qfe0`:

```
# cat /etc/nodename
eng5
# cat /etc/hostname.qfe0
# eng5
```

2. **Create the new network path group and commit the changes:**

```
# apnet -c -a qfe0 -a hme2
# apdb -C
```

3. **Verify the new path group by looking at committed network entries in the AP database:**

```
# apconfig -N
metanetwork:    mether0
physical devices:
                qfe0  A
                hme2
```

4. **Rename the** hostname.*xxxx* **file so the network will be automatically configured at boot time:**

```
# mv /etc/hostname.qfe0 /etc/hostname.mether0
```

5. **Bring down the physical network interfaces and bring up the metanetwork interface by rebooting the machine.**

▼ To Delete the Network Path Group and Metanetwork for the Primary Network

> ⚠ **Caution –** This procedure requires you to reboot the machine. If you are not ready to reboot the machine, do not perform this procedure.

> **Note –** IPv6: replace hostname.*xxxx* with hostname6.*xxxx* in all examples.

1. **Verify that the primary network interface is** mether0 **(in this example):**

```
# cat /etc/nodename
eng5
# cat /etc/hostname.mether0
eng5
```

2. **Rename the configuration files for the metanetwork interface:**

```
# mv /etc/hostname.mether0  /etc/hostname.qfe0
```

3. **Reboot.**

4. **Delete the entry in the AP database:**

```
# apnet -d mether0
# apdb -C
# apconfig -N
#
```

▼ To Deconfigure the Metanetwork for the Primary Network

⚠ **Caution –** This procedure requires you to reboot the machine. If you are not ready to reboot the machine, do not perform this procedure.

Note – IPv6: replace hostname.*xxxx* with hostname6.*xxxx* in all examples.

1. **Verify that the primary network interface is** me0 **(in this example):**

```
# cat /etc/nodename
eng5
# cat /etc/hostname.mether0
eng5
```

2. **Rename the** hostname.*xxxx* **file so the network will be automatically configured at boot time:**

```
# mv  /etc/hostname.mether0 /etc/hostname.qfe0
```

3. **Reboot.**

▼ To Reconfigure the Metanetwork for the Primary Network

⚠ **Caution –** This procedure requires you to reboot the machine. If you are not ready to reboot the machine, do not perform this procedure.

> **Note –** IPv6: replace `hostname.`*xxxx* with `hostname6.`*xxxx* in all examples.

1. **Verify that the primary network interface is** `qfe0` **(in this example):**

```
# cat /etc/nodename
eng5
# cat /etc/hostname.qfe0
eng5
```

2. **Rename the** `hostname.`*xxxx* **file so the network will be automatically configured at boot time:**

```
# mv /etc/hostname.qfe0 /etc/hostname.mether0
```

3. **Reboot.**

Interaction Between AP and DR

This chapter describes the relationship between alternate pathing (AP) and dynamic reconfiguration (DR).

Using DR and AP Together

Dynamic reconfiguration and alternate pathing are designed to work closely together. DR enables you to attach and detach system boards without halting the operating system, as described in the *Sun Enterprise 10000 Dynamic Reconfiguration User Guide*. AP enables you to switch usage *away from* a controller on a board that is being detached or possibly *to* a controller on a board that has been attached.

On the Sun Enterprise 10000 server, AP *automatically* switches each disk and network metadevice that has an active controller on a board being detached (assuming a viable alternate path exists on another board). Also, on the Sun Enterprise 10000 server, AP prevents you from manually switching to controllers on a board that is in the drain state of a DR detach operation.

Note – Prior to any DR operation on a T3 disk, *manually* disable path optimization using an AP switch command such as: `apconfig`(1M).

On Sun Enterprise servers other than the Sun Enterprise 10000 server, you must *manually* switch disk and network metadevices (if necessary) before detaching a board.

The following AP command is an example of a `sf:1` controller on a board that is detached (as indicated by the `DE` flag) and, thus, you cannot switch to that controller:

```
# apconfig -S

c1      sf:0  P A
c2      sf:1  DE
        metadiskname(s):
                mc1t5d0
                mc1t4d0
                mc1t3d0
                mc1t2d0
                mc1t1d0
```

Similarly, the following AP command shows that the `sf:1` controller is on a board that is in the drain state (as indicated by the `DR` flag) and, thus, you cannot switch to that controller:

```
# apconfig -S

c1      sf:0  P A
c2      sf:1  DR
        metadiskname(s):
                mc1t5d0
                mc1t4d0
                mc1t3d0
                mc1t2d0
                mc1t1d0
```

AP is notified that a board is in the DR drain state only on the Sun Enterprise 10000 server.

When you plan to detach a board that hosts the active controller of a path group, you can manually switch to a controller on another board before the DR Detach operation, or even during the DR Detach operation.

On machines other than the Sun Enterprise 10000 server, however, you *must* perform any such switch before the detach operation attempts to complete, or the detach operation will fail. In this case, you can perform the switch and then retry the detach operation.

Note – The DR Attach operation can complete without the board being immediately accessible to AP. You must verify that the physical device is present before switching to the new board using `apconfig`(1M)

For more information about DR, refer to the *Sun Enterprise 10000 Dynamic Reconfiguration User Guide* or the *Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems Dynamic Reconfiguration User Guide.*

Maintaining the Correct AP State

On machines other than the Sun Enterprise 10000 server, if you attach or detach a board that hosts an I/O controller for a disk or network path group, you must run `apconfig -F`. This command sets or clears the detach flag (DE) for that board so that it correctly indicates whether the board is attached or detached.

On the Sun Enterprise 10000 server, you do not need to use `apconfig -F` after an attach or detach operation, since the DE flag is automatically set or cleared after the DR operation completes.

If you detach a board that hosts a network controller, and that network device has not been used since the previous boot, you must run `apconfig -F` to notify the system that the network device is no longer available.

The `apconfig -N` command may incorrectly indicate that a network controller resides on a board that is detached (or incorrectly indicate that it resides on a board that is present) if the corresponding AP metadriver has not been loaded. Use `apconfig -F` to ensure the correct information is shown by `apconfig -N`.

AP Components

AP consists of the following components:

- AP commands – Program instructions that control the various processes and options of AP.
- AP librarian – `ap`(7D) manages the AP database and interacts with the metadrivers as necessary. It receives requests through `ioctls` and acts on them by updating the database or calling entry points in the metadrivers.
- AP metadrivers – The low-level capability for rerouting I/O accesses to alternate paths is implemented in the metadrivers.

All application I/O requests that use the appropriate metadisk go through a metadriver that passes them to the physical device drivers. As a result, the metadrivers can decide which physical path to use, whether a given path is no longer functioning, and so forth. The information on which the metadrivers base their decisions comes from the AP librarian and AP database.

AP man pages

The AP man pages are in the *Alternate Pathing 2.3.1 Reference Manual* portion of your Sun Enterprise server documentation set, as well as online (after you have installed the AP packages). Following is a list of the AP man pages:

- `ap(1M)` – alternate pathing overview
- `apboot(1M)` – define an AP boot device
- `apcheck(1M)` – determine accessibility of AP SCSI meta–device
- `apconfig(1M)` – display and manage AP configuration
- `apdb(1M)` – manage replicas of AP database
- `apdisk(1M)` – manage AP for SCSI disks
- `apinst(1M)` – identify SCSI bus controller
- `apnet(1M)` – manage AP for networks
- `ap(7D)` – AP driver
- `ap_dmd(7D)` – AP disk metadriver
- `mether(7D)` – AP network metadriver
- `mfddi(7D)` – AP network metadriver

Driver Layers

The following figure (with examples in parenthesis) illustrates the driver layers that are used when AP controls disk devices.
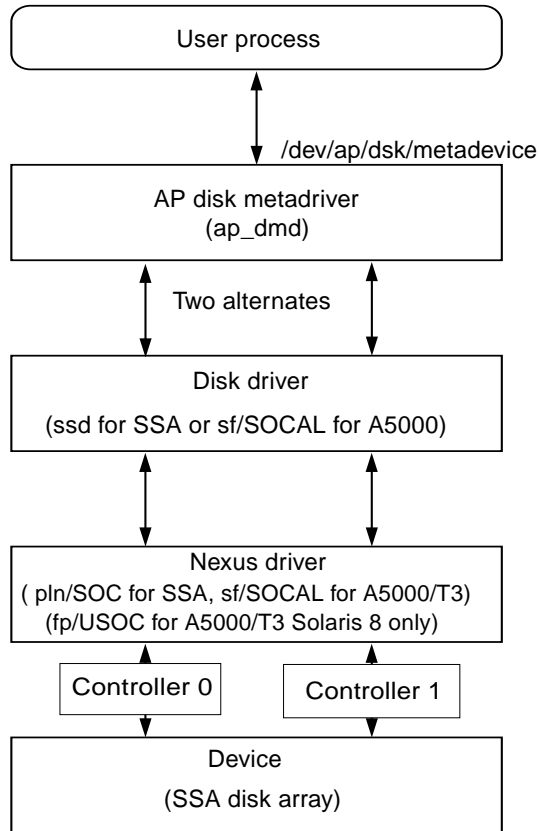
FIGURE C-1   AP Disk Driver Layers

A user process references a metadisk, which provides access to the AP disk metadriver. The AP disk metadriver controls two instances of the physical disk driver which, in turn, controls two instances of the nexus driver (or controller driver). The nexus driver controls the physical device.

FIGURE C-2 illustrates the driver layers that are used when AP controls networks. Examples are given in parenthesis.
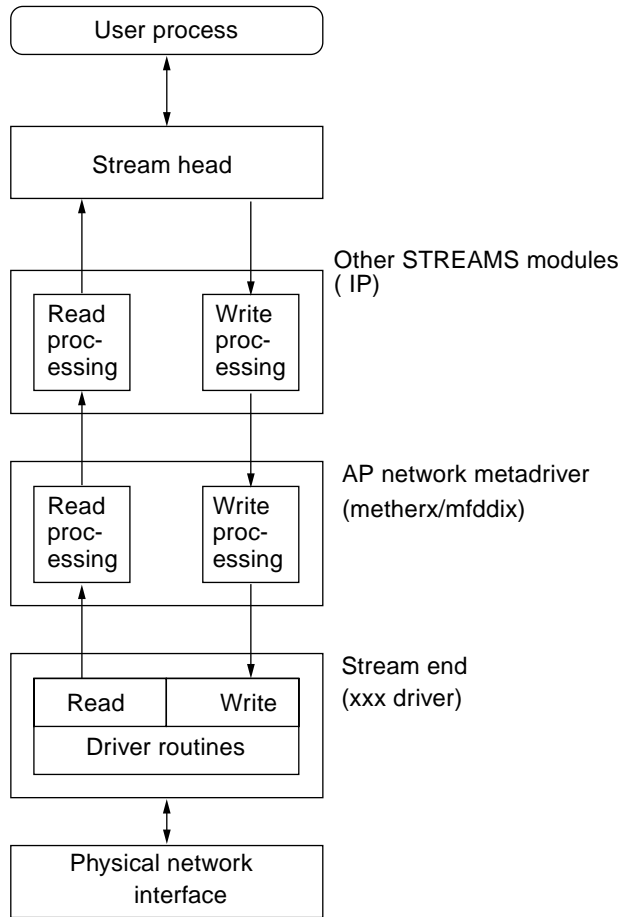
FIGURE C-2   AP Network Driver Layers

A user process references a metanetwork, which provides access to the stream head. The AP network metadriver is inserted into the stream between the high-level read/write processing components and the physical driver routines.

Glossary

active alternate   The alternate path that is currently handling I/O for a path group.

AP database (or simply database)   A database that is maintained by the AP subsystem. The AP database contains all of the information needed to maintain the configuration alternate paths.

alternate path   One of the physical paths within a path group. See primary path.

committed database entry   An entry in the AP database that is currently being used by AP to manage access to a disk or network. (Compare with *uncommitted database entry*.)

disk array   A collection of disks within a hardware peripheral. The disk array provides access to each of its housed disks through one or two Fibre Channel modules.

disk array controller   A controller that resides on the host system and has one or two Fibre Channel modules.

disk array port   A Fibre Channel module that can be connected to a disk array controller that is serviced by a driver pair; for example; `soc/pln` for SSAs.

Fibre Channel module   An optical link connection (OLC) module on a disk array controller that can be connected to a disk array port.

metadisk   A disk abstraction that provides access to an underlying group of two physical paths to a disk.

metanetwork   A network abstraction that provides access to an underlying group of two physical paths to a network.

most favored path   See *path optimization*.

path group   A set of two alternate paths that provide access to the same device or set of devices.

path optimization   The efficient distribution of I/O traffic for a particular device.

physical path   The electrical path from the host to a disk or network.

primary path   The alternate path in a disk path group that is initially the active alternate. The name of the primary path is used when constructing the name of the metadisk. The primary path does not change when a switch occurs. See alternate path.

SSA   A SPARCstorage Array, a collection of disks within a hardware peripheral. The SSA provides access to each of its housed disks using two ports.

switch   The act of establishing a new active path as the path to be used for a given path group. Switching does not change the primary path.

T3   A Sun StorEdge tray attached to a host via an FC-AL connection to a PCI HBA or to an SBus HBA using a GBIC adapter. AP 2.3.1 has optimized I/O path distribution for the T3.

uncommitted database entry   An entry in the AP database that has not been committed and is therefore not currently in effect. If a path group has been created but the database entry has not been committed, that path group is not currently used by AP to manage access to a disk or network. If a previously committed path group has been deleted, but that database entry has not been committed, that path group is still being used by AP to manage access to the disk or network.

Index

R

raw disk slice
  for creating database, 15
  for deleting database, 16
references to device nodes, modifying for AP, 32
remove
  alternate pathing of primary network, 51, 52, 55, 56
  boot disk from AP control, 42
  mirrored boot disk from AP control, 42
removing
  AP packages, 43
  configuration files for physical network interfaces, 55
  direct usage of physical paths (for networks), 47
repartition, not done by AP, 21
resetting tried flag, 23

S

single user mode
  and AP, 43
  reasons it is invoked, 43
Solaris version supported, 10
state of AP, managing, 61
switch
  automatic switch at boot time, 43
  automatic switching and DR, 59
  during DR drain state, 59
  example (for disks), 33
  manual switching and DR, 22
  primary path to alternate path, 32
  switch operation (defined), 7
  switching metadisks, automatic, 22
  switching network path group (Ethernet or FDDI), 50
switching
  automatic and DR, 59
  verifying path, 32
switching path during DR, 4
system (/etc/system), modifying, 39

T

T (tried) flag, 23

timestamp on database, viewing, 17
tried flag, 23
  resetting tried flag, 23
typical AP configuration, 11

U

unattended boot, overview, 10
unavailable (tried) paths, 23
uncommitted database entries, 17
  deleting, 37
  viewing, 18
    for metadisks, 30
    for networks, 19
undo deletion, 37, 51
unplumb network, 51

V

verifying path before switching to, 32
vfstab, modifying, 39
viewing
  committed database entries
    for disks, 18, 31
    for networks, 19
  database information, 17
  path group information, 17
  uncommitted database entries
    for disks, 18, 30
    for networks, 19