Part Number: 313421803

Shared Virtual Array

SVA Path

Version 3.2

for Solaris

User’s Guide

Proprietary Information Statement

The information in this document is confidential and proprietary to Storage Technology Corporation and may be used only under the terms of the product license or nondisclosure agreement. The information in this document, including any associated software program, may not be disclosed, disseminated, or distributed in any manner without the written consent of Storage Technology Corporation.

Limitations on Warranties and Liability

This document neither extends nor creates warranties of any nature, expressed or implied. Storage Technology Corporation cannot accept any responsibility for your use of the information in this document or for your use of any associated software program. You are responsible for backing up your data. You should be careful to ensure that your use of the information complies with all applicable laws, rules, and regulations of the jurisdictions in which it is used.

Warning: No part or portion of this document may be reproduced in any manner or in any form without the written permission of Storage Technology Corporation.

Restricted Rights

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227–7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software—Restricted Rights at 48 CFR 52.227–19, as applicable.

Trademarks

StorageTek is a registered trademark, and Shared Virtual Array and SVA are trademarks of Storage Technology Corporation.

Solaris is a trademark of Sun Microsystems, Inc.

UNIX is a registered trademark of The Open Group.

Other terms and product names used herein may be trademarks of their respective owners.

Fourth Edition December 2001

This edition applies to Version 3.2 of the SVA Path for Solaris product and to all subsequent modifications of that program until otherwise indicated in new editions or revision pages. If there are changes in the program or improvements in the information about the program, this document will be revised and reissued.

EC Number 123430

Comments concerning the contents of this manual should be directed to:

Storage Technology Corporation
Manager, Enterprise Disk Information Development
One StorageTek Drive
Louisville, CO 80028-2201

© Copyright Storage Technology Corporation 2001. All rights reserved.

© Copyright Dot Hill Systems Corp. 2001. All rights reserved.

Table of Contents

Preface

This guide describes how to use the SVA Path Intelligent Data Path Management software. SVA Path provides improved performance and data accessibility for the StorageTek Shared Virtual Array (SVA).

Who Should Read This Guide

This guide is for data administrators, capacity planners, performance specialists, and system administrators. This guide assumes that you are familiar with Shared Virtual Array operations and UNIX system administration.

Shared Virtual Array Documentation

This section lists software and hardware documentation for the Shared Virtual Array products.

How to Obtain Software Documentation

All of the Shared Virtual Array software publications are available from the following sources:

- On the SVA Software Publications CD-ROM (part number 3134524*nn*). To order a copy, contact StorageTek Publication Sales and Service at 800-436-5554 or send a fax to 303-661-7367.

- Online (for viewing and printing), at the StorageTek Customer Resource Center (CRC) website at: www.support.storagetek.com. Click on Software and go to the Shared Virtual Array Software list.

Note: Access to the CRC site requires a password. To obtain a password, call StorageTek Customer Support at 800-678-4430.

SVA Administrator for Solaris Library

- *Shared Virtual Array Administrator for Solaris Command Quick Reference*
- *Shared Virtual Array Administrator for Solaris Installation Guide*
- *Shared Virtual Array Administrator for Solaris Messages*
- *Shared Virtual Array Administrator for Solaris Quick Start Guide*
- *Shared Virtual Array Administrator for Solaris User's Guide*

Related SVA Software Publications

For SVA SnapShot for Solaris:

- *Shared Virtual Array SnapShot for Solaris User's Guide*
- *Shared Virtual Array SnapShot for Solaris Quick Start Guide*

For Shared Virtual Array Administrator for Solaris:

- *Shared Virtual Array Administrator for Solaris Command Quick Reference*
- *Shared Virtual Array Administrator for Solaris Installation Guide*
- *Shared Virtual Array Administrator for Solaris Messages*
- *Shared Virtual Array Administrator for Solaris Quick Start Guide*
- *Shared Virtual Array Administrator for Solaris User's Guide*

For SnapShot for Solaris:

- *SnapShot for Solaris User's Guide*
- *SnapShot for Solaris Quick Start Guide*

For SVA Administrator for OS/390:

- *Shared Virtual Array Administrator for OS/390 Configuration and Administration*
- *Shared Virtual Array Administrator for OS/390 Installation, Customization, and Maintenance*
- *Shared Virtual Array Administrator for OS/390 Messages and Codes*
- *Shared Virtual Array Administrator for OS/390 Reporting*

For SVA SnapShot for OS/390:

- *SVA SnapShot for OS/390 Installation, Customization, and Maintenance*
- *SVA SnapShot for OS/390 User's Guide*

For SVA Console for Windows NT (SVAC):

- *Shared Virtual Array Console for Windows NT Quick Start Guide*

For any StorageTek software:

- *Requesting Help from Software Support*

SVA Hardware Publications

Shared Virtual Array hardware publications are available from the following sources:

- On the SVA Hardware Publications CD-ROM (part number 3118447*nn*). To order a copy, contact StorageTek Publication Sales and Service at 800-436-5554 or send a fax to 303-661-7367.

- Online (for viewing and printing), at the StorageTek Customer Resource Center (CRC) website at: www.support.storagetek.com. Click on Disk Subsystems.

☞ **Note:** Access to the CRC site requires a password. To obtain a password, call StorageTek Customer Support at 800-678-4430.

The V960 Shared Virtual Array (SVA) library consists of:

- *V960 Shared Virtual Array Introduction*
- *V960 Shared Virtual Array Operation and Recovery*
- *V960 Shared Virtual Array Physical Planning*
- *V960 Shared Virtual Array Planning, Implementation, and Usage*
- *V960 Shared Virtual Array Reference*
- *V960 Shared Virtual Array System Assurance*
- *Peer to Peer Remote Copy Configuration Guide*

The 9500 SVA library consists of:

- *9500 Shared Virtual Array Introduction*
- *9500 Shared Virtual Array Operation and Recovery*
- *9500 Shared Virtual Array Physical Planning*
- *9500 Shared Virtual Array Planning, Implementation, and Usage*
- *9500 Shared Virtual Array Reference*
- *9500 Shared Virtual Array System Assurance*

Chapter 1. SVA Path Overview

This chapter provides an overview of SVA Path Intelligent Data Path Management software and its features.

SVA Path offers a new level of data accessibility and improved performance for the SVA. It eliminates the point of failure represented by a single input/output (I/O) path between servers and storage systems and permits I/O to be distributed across multiple paths.

SVA Path suports up to 1024 VDEVs per LUN, and up to 1024 individual paths to up to 1024 LUNs. This means that you can have 512 LUNs with two paths each, or up to 32 LUNs with 32 paths each.

Failover/Failback Data Paths

By providing alternate I/O paths from the server to the SVA, SVA Path provides uninterrupted access to mission-critical data. This substantially insulates server applications from I/O path failures.

In the event of a failed host bus adapter (HBA), interface cable, or channel I/O card within the SVA, SVA Path automatically reroutes I/O traffic to an alternate data path. Failover is essentially transparent, ensuring continuous access to data stored on the SVA. When configured in the recommended failback mode, SVA Path automatically restores the primary data path and system redundancy once the defective component is replaced.

Load Balancing

SVA Path supports up to 32 data paths between a host and any SVA logical device. Since only two data paths are required for path failover capability, multiple data paths can be used to balance the I/O load evenly across all I/O channels. SVA Path measures the I/O load being sent to each logical disk device and can reassign the primary paths of one or more devices to evenly distribute the load among all available HBAs.

How SVA Path Works

SVA Path's filter driver resides between the file system drivers and the device drivers, within the layered driver architecture. I/O requests are passed from the file system through the drivers and ultimately to the hardware.

SVA Path monitors the flow of I/O requests through the layered driver architecture. When it detects a failure along an I/O path, it automatically reroutes the request to an alternate path. Failover to the redundant I/O path is transparent to server applications and permits continuous access to the information stored on the disk array(s). To the operating system, there is only a slight delay in normal I/O operations during path failover; existing drive numbers and device access functions continue to work as expected.

Note: Boot disk failover support is not provided due to the loading sequence of the filter driver.

Supported SVA Path Configurations

SVA Path supports single server configurations.

System Requirements

Before proceeding to the next chapter, you should verify that your site meets the following minimum requirements (Table 1-1, "SVA Path Minimum System Requirements").

Table 1-1 SVA Path Minimum System Requirements

Host hardware:	Sparc 2 with 128MB memory
Host software:	Solaris 2.6 (with Patch 105181-13 or later), or 7 (32, 64 bit), plus the applicable recommended patch cluster as of August 1, 1999 (the recommended cluster contains all patches needed for running the SVAA server), or 2.8 or greater
HBA (Fibre)	JNI FC64-1063 (fcaw driver) 2.6, 7 SBUS JNI FC64-1063 2.6, 7 PCI Emulex LP8000
Host disk space:	20MB free space in $TEMP and 20MB free space in the server installation directory.
SVA Subsystem Microcode:	K05.04.08 or greater

Summary of SVA Path Benefits

- Increases potential subsystem throughput by directing I/O through multiple host adapters and SVA channels. Logical drives can be assigned to host bus adapters, manually balancing the I/O load across paths.
- Provides continuous access to mission-critical data by insulating server applications from I/O path failures.
- Installs easily and is transparent to server applications.

Document Overview

This manual describes how to install and configure SVA Path on systems running the Solaris operating system.

- Chapter 2 describes configuring your hardware in preparation for installing SVA Path.
- Chapter 3 contains the instructions for installing SVA Path.
- Chapter 4 explains configuring SVA Path.
- Chapter 5 details the syntax and usage of SVA Path commands.
- Chapter 6 offers assistance in diagnosing error messages.
- Appendix A provides information regarding attachment to an SVA with SCSI interface.

Note that user documentation for products used with SVA Path, including Solaris documentation, is referenced throughout this manual. Have your hardware and operating system manuals available for quick reference.

Chapter 2. SVA Path Hardware Setup

Fibre Addressing Concepts

Host Bus Adapters/Initiators

The terms “host bus adapter” and “initiator” mean essentially the same thing. Typically, the HBA is a card within the host that, in its role as initiator, issues commands on the Fibre channel.

Domains in Fibre Channel Connection

StorageTek uses the concept of “domains” to allow open systems hosts access to blocks of logical devices (the domains) within an SVA. A domain is an additional layer of device addressing, but one that is manually configured by the Customer Service Engineer (CSE) in the SVA. This layer of addressing divides the SVA into “domains of access.” There can be up to 16 (0–15) domains per SVA, with each domain having one target with 256 LUNs. There is a limit of 1024 total devices available within an SVA.[1]

Each open systems host initiator is connected with Fibre cables to a controller card port, giving it access to the devices that have been configured within its domain. (An open systems host cannot see devices in domains other than the one to which it is attached.)

Full SVA Path functionality requires that redundant initiators can access the SVA over redundant data paths.

Figure 2-1 shows two data paths connecting the open systems platform to the attached SVA using Fibre cables.

1. Using all allowed domains, targets, and LUNs, there are more than 1024 logical devices, but the SVA has a limit of 1024 logical devices.

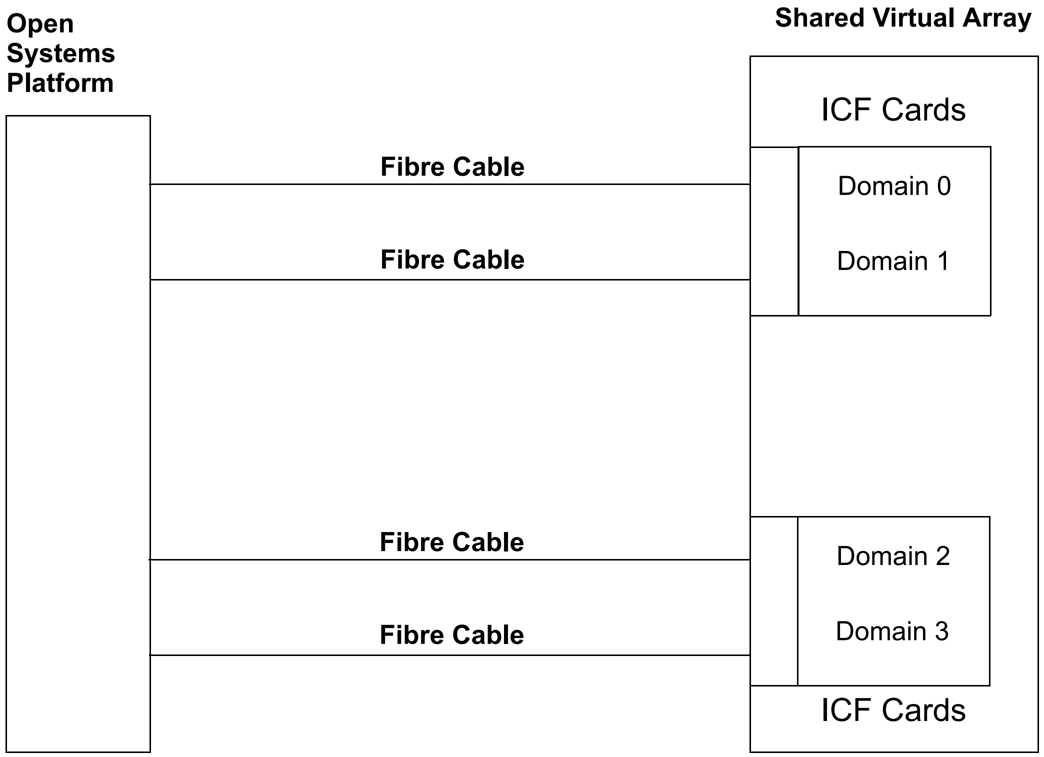

Figure 2-1 Fibre Paths from the Host to the SVA

Note: In the above figure, a domain can represent up to 256 logical devices (1 Target × 256 LUNs = 256 Logical Devices). An SVA is limited to 1024 logical devices.

Domain Numbering

Using Fibre connection, the domain number is configured at the SVA operator panel by the StorageTek Customer Service Engineer.

The domain number is never seen by the open systems host; from its point of view, just a target and logical unit number are involved in an I/O operation.

Domain Numbering with Fibre and SVA Path

It is not normally advisable to have more than one path from a single host set to the same domain number to a given SVA (in case two hosts attempt to share the same LUN and so corrupt the data stored on it). SVA Path requires exactly this configuration for failover to work.

SVA Path manages multiple paths from a single host, using identical domain numbers.

Hardware Preparation

Configuring a Solaris System to See New Devices

Perform the following steps on the Solaris host:

> **Note:** If any of the following steps do not produce the expected result, refer to your Solaris system administrator documentation for instructions on setting up the host platform correctly.

1. Set the domain addresses at the SVA Operator Panel.
2. Log in as `root`.
3. Open a console or terminal window.
4. Determine which controllers are being used to attach the Solaris host to the Shared Virtual Array (SVA).
5. Define the LUNs (logical unit numbers) on the SVA you want to be able to access now and in the foreseeable future.
6. Open the `/kernel/drv/sd.conf` file and scan its contents.

    a. If the file lists all the functional devices you want to be able to access on the SVA, and you know the devices were added to the file before the host was rebooted last, you will not need to reboot again. Perform the following steps:

    Exit the `sd.conf` file.

    Reconfigure the `/device` entries on the host:

drvconfig

Reconfigure the `/dev` entries on the host:

disks

b. If the file does not list all the functional devices you want to be able to access on the SVA, or if the file lists them all but you are not sure whether they were added before the last reboot, perform the following steps:

   Add any missing functional devices to the file. Be sure to add all the targets and LUNs necessary, as the `sd.conf` file is only consulted at boot time, and later additions will require a reboot of the host. Each new entry for fibre should look like:

```
name="sd" parent="nnnn" target= nn lun= n;
```

   or for scsi should look like:

```
name="sd" class="scsi" target= nn lun= n;
```

   Where *nnnn* is a 4-character string that indicates the card manufacturer/type (for example, lpfc=Emulex, jnic=JNI, etc.). The target *nn* is an integer from 0 to 15. The LUN *n* is an integer from 0 to 255 that corresponds to the device LUN.

   Save and exit the `sd.conf` file.

   Reboot the system, telling it to reread the attached devices and reconfigure the `/device` and `/dev` entries:

```
# init 0
```

```
> boot -r
```

7. Log in as `root`.

8. Perform the following steps to verify that the new devices are configured correctly on the Solaris host:

   Display all the SCSI devices the host recognizes:

```
# format
```

The display for a dual-path system would appear something like the following:

```
AVAILABLE DISK SELECTIONS:

0. c0t0d0 <ST39140A cyl 17660 alt 2 hd 16 sec 63>
/pci@1f,0/pci01/ide@3/dad@0,0

1. c1t0d0 <STK-V960-0401 cyl 10013 alt 2 hd 15 sec 48>
/pci@1f,0/pci@1/scsi@1,1/sd@0,0

2. c1t1d0 <STK-V960-0401 cyl 10013 alt 2 hd 15 sec 48>
/pci@1f,0/pci@1/scsi@1,1/sd@1,0

3. c1t2d0 <STK-V960-0401 cyl 3335 alt 2 hd 15 sec 48>
/pci@1f,0/pci@1/scsi@1,1/sd@2,0

4. c2t0d0 <STK-V960-0401 cyl 10013 alt 2 hd 15 sec 48>
/pci@1f,0/pci@1/scsi@2,1/sd@0,0

5. c2t1d0 <STK-V960-0401 cyl 10013 alt 2 hd 15 sec 48>
/pci@1f,0/pci@1/scsi@2,1/sd@1,0

6. c2t2d0 <STK-V960-0401 cyl 3335 alt 2 hd 15 sec 48>
/pci@1f,0/pci@1/scsi@2,1/sd@2,0

Specify disk (enter its number):
```

Figure 2-2 Example format Output for a Dual -Path System

Each of the three disk devices in our example is seen twice, once on the first controller (`c1`) and again on the second controller (`c2`).

9. Verify that all of the new device(s) you have defined are included in the display. They are identified with a type of `STK-V960-`*nnnn*.

10. Write down the first six characters (the controller, target, and device numbers) of the new device(s). This information looks something like:

    `c1t0d0`

11. Press Ctl-d to exit the `format` utility.

The Solaris host is now ready for SVA Path installation.

Note: Each LUN defined above must be labeled and formatted prior to use, using the `format` command. For help, consult the `man` pages or the appropriate system user documentation.

Once the new virtual devices have been labeled, initialize and mount them in standard fashion to prepare them for data storage.

Chapter 3. SVA Path Installation

This chapter describes how to install SVA Path for use with SVA.

Note: On systems that will run SVA Administrator (SVAA) with SVA Path, SVAA can be installed *before* SVA Path, but a few additional steps must be performed after SVA Path installation. See the section beginning on page 14 for details.

Note: On systems running Veritas Volume Manager, some additional procedures must be followed in advance of installation. Refer to the section beginning on page 16 for instructions.

Note: References to product version numbers (such as "VxVM 3.1.1") imply that version "or greater" unless a range is specified.

Note: SVA Path ignores the boot disk; this ensures a normal boot sequence.

Installing SVA Path on Solaris

Follow these steps to install the SVA Path driver and its supporting files.

1. Set the domain addresses on the SVA (refer to "Hardware Preparation," on page 7).
2. Log in as `root` on the host on which you are installing SVA Path.

3. If you chose to disable SVA Path LUN exclusion (default behavior) edit the `/etc/system` file to include the following line:

   ```
   set spd:spd_exclusion_enable = 0
   ```

   This is not recommended, however. The preferred method is shown in step 10, below.

4. Reboot the host. When the host is back up log in as root.

5. Insert the SVA Path installation CD (refer to Sun documentation for CD-ROM operation).

6. From the `root` directory:

   a. Change to the CD directory (for example):

   ```
   # cd /cdrom/cdrom0
   ```

   Note: If the directory tree opens, close it.

   b. Enter:

   ```
   # ./install
   ```

7. Follow the onscreen instructions to complete the installation.

8. Eject the CD, by entering the appropriate system command, for example:

   ```
   # eject
   ```

   `or`

   ```
   # eject cdrom
   ```

9. After installation, reboot Solaris.

   ```
   # boot -- -r
   ```

   If this is a new installation of SVA Path, a second reboot will be automatically initiated when the first has completed. Configuration files are written during the first boot that must then be read and implemented during the second.

10. After re-booting, ALL devices will be excluded by default. Issue the following command to enable them all:

    ```
    # setsp -e0 -l all
    ```

    To selectively enable devices, use:

    ```
    # setsp -e0 -l <spd#>
    ```

    where <spd#> is the number of the spd device you wish to enable.

11. Run `format` to verify that all appropriate devices are listed as SVA Path devices, which are shown in the format:

    ```
    /pseudo/spn@<n>/spd@<tgt>,<lun>
    ```

12. To correlate the pseudo-devices to the original device names, run `setsp -a`. The new device name appears in the first column of the same entry, below the `spd` number (See Figure 4-1, “Sample Output of `setsp -a Command`” on page 25).

13. If you are using fibre channel HBAs (either the Emulex LP8000 or JNI FC64-1063 HBA cards), their configuration files must be manually edited to allow 10 seconds for loop recovery before allowing commands to fail and triggering path failover.

    For Emulex LP8000, edit the `/kernel/drv/lpfc.conf` file to include the following:

    ```
    linkdown-tmo = 10;
    ```

    For JNI, edit the `/kernel/drv/fcaw.conf` file to include the following:

    ```
    failover = 10;
    ```

Note: Changing this timeout value to a number higher than 10 is not advisable as it’s value can be additive to other system timeout and retry values and may cause extremely long delays before path failover will occur.

14. Edit any other application-specific files (`/etc/vfstab`, for example) to reflect the new device names. New device files— identical to the pre-installation device files, except for their controller numbers—are generated during SVA Path installation for all SVA devices (including those accessible by only a single path). Any applications already configured to use the older device files to access SVA LUNs must be reconfigured to use the new pathnames (an example of this is given in the instructions below for installing SVA Path with SVA Administrator).

Note: If you are using Veritas Volume Manager and follow the directions beginning on page 16, VxVM should require no reconfiguration.

The installation is now complete.

Installing SVA Path with SVA Administrator

SVAA uses a designated LUN on the SVA for administrative commands (rather than data storage) and a particular character special device file (e.g., `/dev/rdsk/c1t0d0s2`) to access that LUN. This is SVAA's Extended Control and Monitoring (ECAM) facility, and the LUN designated for its use is the ECAM device. *Before* SVA Path is installed, make a note of this device name.

Note: If SVA Path is installed after SVAA, SVA Path will claim the SVAA device. Either run the following commands, which will cause SVA Path to ignore the specified device, or run the SVAA `sibconfig` utility to identify the new ECAM device path:

```
# sppath -IcXtYdZ
```

```
# setsp -T -l<n>
```

```
# setsp -g
```

Changes will take effect only after the next reboot of the host.

After SVA Path is installed, in addition to its physical path(s) (e.g., `/dev/rdsk/c1t0d0s2` and `/dev/rdsk/c2t0d0s2`), a virtual path to the ECAM device will have been created along with an additional device name (e.g., `/dev/rdsk/c3t0d0s2`), which SVAA must be configured to use instead of the original ECAM device name. To discover the new name for the ECAM device, run `setsp -a` and look for the original device name among the listings in the second column; the new device name appears in the first column of the same entry, below the `spd` number. (An example of `setsp -a` output is shown in Figure 4-1, on page 25.)

Consult your SVAA documentation for details on running SVAA's `sibconfig` command to reconfigure the ECAM device name.

Excluding the ECAM Device from SVA Path and VxVM Control

When using SVA Path and VxVM 3.1.1 (or greater), the ECAM device for an SVAA server must be excluded from both SVA Path and VxVM DMP control.

Note: The following example assumes two paths (c1t0d0 and c2t0d0) to an ECAM device.

1. Do the following to exclude the ECAM device from SVA Path control:

   a. Use `setsp` command to terminate the spd device the ECAM device: `setsp -T -l spd #`

   b. Use `sppath` command to make the ECAM device paths to be ignored by SVA Path:

   c. `sppath -I c1t0d0`

   d. `sppath -I c2t0d0`

   e. Use setsp command to update the SVA Path configuration: `setsp -g`

2. Do the following to exclude the ECAM device from VxVM DMP control:

Note: If both SVA Path and VxVM 3.1.1 are installed, refer to “Version 3.1.1 and Higher,” on page 18 to exclude the paths for the ECAM device from SVA Path control

Note: If only VxVM 3.1.1 is installed and you want the SVA devices except the ECAM device to be controlled by DMP, do the following.

a. From " Volume Manager Support Operations," select 17 to prevent multipathing /Suppress devices from VxVM's view.

b. From the "VolumeManager/Disk/ExcludeDevices" menu, select 6 to prevent multipathing of a disk by VxVM.

c. Enter the device path, ex. c1t0d0.

d. Repeat 3 to exclude all paths to the ECAM device from DMP.

Installing SVA Path with Veritas Volume Manager

Versions 2.4 and higher of the Veritas Volume Manager (VxVM) and the Sun Enterprise Volume Manager are compatible with SVA Path.

If you are running VxVM version 2.4 to 3.1.0, continue with the following section; If you are running VxVM version 3.1.1 or higher, go to “Version 3.1.1 and Higher,” on page 18.

Version 2.4 to 3.10

If you are already running VxVM (version 2.4 to 3.1.0), you must disable the Veritas DMP driver before installing SVA Path. The DMP driver duplicates some SVA Path driver functionality and is not compatible with it.

If you are installing SVA Path before installing VxVM, SVA Path will prevent VxVM from installing its DMP driver. You can use the normal installation described above and then install VxVM.

The instructions below for removing the DMP driver summarize those given for users of Sun's AP driver in the DMP Issues section of the VERITAS Volume Manager Release Notes, Release 2.5.1, page 17; the process is the same for SVA Path users. If you are running the Sun Enterprise Volume Manager, please consult that product's release notes for instructions for removing the DMP driver.

1. Use `umount` to unmount all filesystems that have been created on VxVM volumes.

2. Stop VxVM with the command:

   ```
   # vxdctl stop
   ```

3. Edit `/etc/system` and remove the following line:

   ```
   forceload: drv/vxdmp
   ```

4. Remove the `vxdmp` driver from the `/kernel/drv` directory:

   ```
   # rm /kernel/drv/vxdmp /kernel/drv/sparcv9/vxdmp
   ```

5. Remove the VxVM DMP files:

   ```
   # rm -rf /dev/vx/dmp /dev/vx/rdmp
   ```

6. Link the dmp device files to `/dev/dsk` and `/dev/rdsk`:

   ```
   # ln -s /dev/dsk /dev/vx/dmp
   # ln -s /dev/rdsk dev/vx/rdmp
   ```

7. Reboot the system.

You can now proceed with the normal SVA Path installation.

Note: For the duplicate paths to operate properly, care must be taken not to use the new devices until the system is completely configured: proceed to the next chapter.

Version 3.1.1 and Higher

With Veritas VxVM version 3.1.1 (and higher), DMP cannot be removed.

> **Note:** vxinstall or vxdiskadm can be used to prevent multipathing of all the devices on the controllers that will be controlled by SVA Path, preventing conflict between SVA Path and Veritas DMP.

> **Note:** If VxVM is installed after SVA Path, perform the following procedure *after* installing VxVM and re-boot the system. If SVA Path is installed after VxVM, perform the procedure *before* installing SVA Path.

The following example uses vxdiskadm to prevent conflict between SVA Path and Veritas DMP:

1. From "Volume Manager Support Operations," select 17 to prevent multipathing/Suppress devices from VxVM's view.
2. From the "VolumeManager/Disk/ExcludeDevices" menu, select 5 to prevent multipathing of all disks on a controller by VxVM.
3. Enter the controller number; then relay to the questions to make the devices on the controller to be non-multipathed devices.
4. Repeat Step 3, as needed.
5. Return to the "VolumeManager/Disk/ExcludeDevices" menu, and select 8 to list the currently-suppressed/non-multipathed devices to verify that the controllers are excluded.
6. Proceed with the normal SVA Path installation.

Uninstalling SVA Path

1. To remove SVA Path you need only enter the package remove command from the root directory:

```
# pkgrm SVAPath
```

2. Restore any application-specific files that were modified during the installation procedure (such as `/etc/vfstab`).

3. Restore the hardware configuration (XSA Domain switches).

4. Reboot the system.

☞ To remove the install files, remove the directory `/opt/storagetek/svapath`.

When SVA Path is uninstalled, the configuration files are saved as (for example):

`var/tmp/sp-configuration-backup-200101060215.tar`

where the number represents a year-month-day-hour-minute time stamp. Included files are:

```
/etc/sppath.conf

/kernel/drv/spd.conf

/kernel/drv/spn.conf
```

Uninstalling SVA Path with Veritas Volume Manager

Uninstall SVA Path normally (See “Uninstalling SVA Path” on page 18). After the reboot, Volume Manager will continue to function normally. (VxVM will function normally even after the package is removed and before the reboot, as long as no attempt is made to reconfigure it, in which case it would try to open SVA Path device files no longer present on the system.)

Volume Manager will also function normally with DMP disabled (as directed in “Installing SVA Path with Veritas Volume Manager” on page 16). If DMP is required, however, refer to the Volume Manager documentation for instructions on re-enabling it.

If SVA Path is permanently uninstalled, remove `/kernel/drv/ap` if—and only if—it is zero-length file. It is this file that prevents VxVM from installing DMP.

Installed Files

During installation, the files listed in Table 3-1, "Installed SVA Path Files" are placed in your system.

Table 3-1 Installed SVA Path Files

File	Description
/kernel/drv/spn	nexus (virtual HBA) driver, 32-bit mode
/kernel/drv/sparcv9/spn	nexus driver, 64-bit mode
/kernel/drv/spn.conf	spn configuration file
/kernel/drv/spd	SVA Path driver, 32-bit mode
/kernel/drv/sparcv9/spd	SVA Path driver, 64-bit mode
/kernel/drv/spd.conf	spd configuration file
/kernel/drv/ap	blocks installation of DMP driver. see "Uninstalling SVA Path with Veritas Volume Manager"
/usr/sbin/sppath	qualifies and claims SVA storage devices for SVA Path control
/usr/sbin/setsp	configures SVA Path parameters
/etc/sppath.conf	sppath configuration file
/etc/spmon.conf	configures load balancing parameters
/etc/spd/bin/spmon	monitors device paths and implements load balancing
/etc/spd/bin/badlinks	daemon that removes inactive links during uninstall process
/etc/spd/bin/forceload_add	adds forceload statements to /etc/system (invoked by setsp)
/etc/spd/bin/forceload_rm	removes forceload statements from /etc/system (invoked by setsp)
/etc/init.d/spconfigure	configures SVA Path devices at reconfiguration reboot
/etc/init.d/spdaemon	stops and starts spmon daemon

SVA Path Device Naming on Solaris

In a typical, single-path disk storage system, each physical disk is represented in the host's `/dev` directory by a set of special device files representing the raw and block mode device instances for each possible disk partition on that device. For simplicity, we will discuss this set of device files as a single device filename and disregard both the first parts of the pathnames that distinguish between raw and block modes and the suffixes that identify individual partitions.

As a layered driver that runs on top of the standard Solaris disk driver (`sd` or `ssd`), SVA Path creates its own device files for physical devices, and it is through these filenames that applications access the devices. When multiple paths to a single device exist, so would multiple device files be present, one for each path via the native disk driver. In these cases, which are typical of SVA Path installations, SVA Path creates a single additional device filename for applications to access the device and manages the original device files transparently to those applications.

For example, two connections to a LUN on an SVA might present the device to the Solaris host as `c1t0d0` and `c2t0d0`. When SVA Path is installed, it creates a third set of device files named `c3t0d0`, and blocks applications' access to the original two device files.

How Device Filenames Are Chosen

In order to provide interoperability with complementary storage management software (e.g., Solstice Disk Suite and Veritas Volume Manager), SVA Path uses standard Solaris device names in the form c*X*t*Y*d*Z*s*P*, where *X* represents a controller or HBA number, *Y* represents a SCSI target number, *Z* represents a SCSI logical unit number, and *P* represents a slice or partition number.

When SVA Path adds new device files to the system and changes the device names by which pre-existing devices must be accessed, the new device files, in order to be as easily understood as possible, retain the SCSI target and LUN (logical unit) numbers from the original device files.

For example, a set of physical devices might originally be accessible via HBAs `c1` and `c2`. When SVA Path is installed, a new, virtual HBA, `c3`, will be created, along with virtual disk device files whose names start with `c3` and which have the same target and LUN numbers as the original device files. Therefore, a device originally accessible via the fibre channel or SCSI disk driver device files `c1t4d0` and `c2t4d0` will, after SVA Path is installed, be accessed through the SVA Path device file `c3t4d0` (Figure 3-1, "SVA Path Device Filename Management").

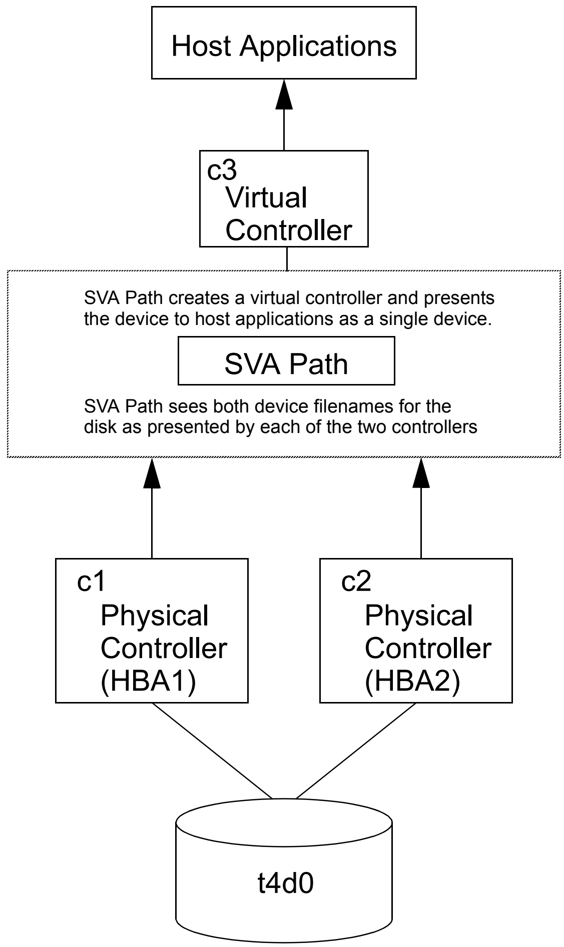

Figure 3-1   SVA Path Device Filename Management

If the target and LUN numbers assigned to the device are different on each physical path, SVA Path uses the target and LUN number associated with the lowest numbered HBA. If a device is originally accessible as `c1t4d0` and `c2t2d0`, for example, the `c1t4d0` device name has the lower controller number (`c1`) and thus SVA Path takes the target (`t4`) and LUN (`d0`) number from it to construct the new device name, `c3t4d0`.

If there are more than one set of redundant paths, there could be more than one device with a given target and LUN number. For this reason, SVA Path creates a different virtual HBA for each set of redundant physical HBAs. For example, if one device is accessible via `c1t0d0` and `c2t0d0`, and another device is accessible via `c3t0d0` and `c4t0d0`, SVA Path would create two new device files `c5t0d0` (for `c1t0d0` and `c2t0d0`) and `c6t0d0` (for `c3t0d0` and `c4t0d0`).

SVA Path does not select the controller number used; instead, the number is assigned by the `disks(1m)` program automatically during the reconfiguration reboot process, just as it would be for any new physical HBA added to the system.

Devices without Redundant Paths to the Host

SVA Path creates virtual device files for all supported devices and blocks access to them through their original device filenames whether or not they are accessible via redundant paths. Virtual device files are created for devices that have only one physical path for two reasons:

1. a device could actually have redundant paths but only one was functional at the time that SVA Path was installed

2. the device could be part of a SAN configuration where it is necessary to prohibit applications on the local host from accessing the device because the device was assigned to another host on the SAN.

Reconfiguration for Existing Applications

As explained above, any physical device supported by SVA Path and in use prior to SVA Path installation will, from the perspective of host applications, be renamed during installation. This has no effect on storage encapsulated by the Veritas Volume Manager or Sun Enterprise Volume Manager, but other applications will need to be redirected to the new filenames, by either editing `/etc/vfstab` or modifying the configuration of the individual application to reflect the new device names.

Underlying Device Pathnames

Following Solaris convention, the device files in `/dev/dsk` and `/dev/rdsk` are actually symbolic links to real device files in the `/devices` directory tree. A typical SCSI disk driver device file is `/devices/pci@1f,4000/scsi@3/sd@4,0`, which refers to a disk at target 4 LUN 0 attached to the SCSI HBA in PCI slot 3. In some hardware configurations, the word `scsi` may be replaced by the name of a specific HBA driver, such as `isp` or `fas`.

The corresponding SVA Path device file is similar. For example, `/devices/pseudo/spn@1/spd@4,0` indicates an `spd` (SVA Path driver) device with target 4 and LUN 0, attached to SVA Path virtual HBA (`spn`) number 1. These device paths are also displayed by `format(1m)` and certain other utilities.

Chapter 4. SVA Path Configuration

This chapter explains the default configuration after installation and the procedures for reconfiguring SVA Path.

The Default Configuration

During installation, in addition to device files for the redundant physical paths to each SVA LUN, SVA Path creates a virtual device file for each SVA LUN. It is the virtual device file that will actually be used by applications to access that LUN (the original and redundant data paths should never be used to access the device, or data on the LUN could be corrupted).

Use the `setsp -a` command to display the default path configurations.

```
# setsp -a
==============================================================================
 spd   Path/disk     Status    Pri   Exc    Buf Balance RtrCnt  RtrDly FailBack
==============================================================================
   0   c7t0d0/sd66    Excluded   X     X     32    1      20     3000      1
       c8t0d0/sd88    Excluded         X
 spd0 = c3t0d0                                 ID = "STK V960 0000000010390000"
==============================================================================
   1   c7t0d1/sd67    Excluded   X     X     32    1      20     3000      1
       c8t0d1/sd89    Excluded         X
 spd1 = c3t0d1                                 ID = "STK V960 0000000010390001"
==============================================================================
   2   c7t0d2/sd68    Excluded   X     X     32    1      20     3000      1
       c8t0d2/sd90    Excluded         X
 spd2 = c3t0d2                                 ID = "STK V960 0000000010390002"
==============================================================================
```

Figure 4-1 Sample Output of setsp -a Command

Note: Default path configuration has all paths excluded.

The column headings identify the various fields on this screen, the last seven of which are user-configurable parameters for the device. The `setsp -a` output fields (with their default values, when applicable) are described below. The commands used to change the default settings are described in the following section.

- **`spd`** is the SVA Path driver number, an ID assigned to the device by SVA Path and used in `/var/adm/messages` (*X* in the format spd*X*). The spd*X* = field gives the virtual device name created by SVA Path to access the device.

- **`Path/disk`** shows the device names and Solaris SCSI or fibre channel disk numbers (in the format, sd*nn* or ssd*nn*) for the redundant physical paths to the device.

- **`Status`** shows the current state of the path. `Good` paths are functioning normally. `Bad` paths have failed. `Excluded` paths are unavailable to applications on the host.

- **`Primary`** shows which of the physical I/O paths connecting the device to the host's host bus adapters (or controllers) is defined as primary (marked by an `X`). In the default configuration, primary path assignments, from device to device, cycle sequentially through the available data paths. Figure 4-1, "Sample Output of `setsp -a Command`" shows this in an alternating, dual-path configuration: `spd0` has a primary path to controller 7, `spd1` uses its path to controller 8 as primary, and `spd2` alternates back to controller 7. The device's duplicate path(s) are "hidden" unless the primary path fails over to one of them.

  Note: When Load-balancing is enabled all paths are active at all times, but there is no indication on this display.

- **`Exclude`** indicates the exclusion setting, which is used to keep particular servers from reading particular logical drives. LUN exclusion is set to 1 (exclusion) for all devices by default.

- **`Buf`** is the number of buffer pointers (or buffer structures) pre-allocated for each logical device. For peak performance, `Buf` should be approximately equal to the maximum useful queue depth of the logical unit. The default value is `32`.

- `Balance` indicates whether load balancing is enabled for the device. `1` means load balancing enabled. `0` means load balancing not enabled.

- `RtrCount` is the number of times a failed I/O will be retried on the primary path—after it has tried unsuccessfully to use its alternate path(s) and returned again to the primary. When the specified number of retries have failed, the I/O fails. The default value is `20`.

- `RtrDelay` is the time interval, in milliseconds, between the retry attempts described in the preceding parameter. The default value is `3000` ms.

- `FailBack` indicates whether failback is enabled for the logical device. When failback is enabled (the default setting of `1`), SVA Path will keep testing a path that has failed and return it to service (as the primary path or in sequential load balancing) as soon as the path is restored.

If the default configuration is satisfactory, no configuration of SVA Path is necessary.

Changing the Default Configuration

The information displayed by `setsp -a` is stored in the configuration file `/kernel/drv/spd.conf`, which should never be edited directly. User-configurable parameters must be changed exclusively through the `setsp` command options provided for that purpose.

Specifying a Device for setsp

Any `setsp` command intended to act on a specific device, including those described below for changing devices' default SVA Path parameters, needs to specify the device to be acted upon. This is done with the `-l` option (note that this is a lowercase letter L, and not the number 1) followed by the numerical element of the device's `spd` number, as expressed in the syntax illustrations below as `-l`<*x*>. A device's `spd` number is listed in the first column of the `setsp -a` output (Figure 4-1, "Sample Output of `setsp -a Command`" on page 25).

To apply the command to all of the devices use `-l all`. Whether you are applying the configuration command to a single device or to all devices, only one parameter can be changed per command.

Assigning a New Primary Path

The syntax for changing an SVA LUN's primary data path is

```
setsp -l<x> -p<n>
```

where *<x>* is the numerical element of the `spd` number (see above) and *<n>* is the number of the new path. The path number can be obtained by simply counting down `setsp`'s list of physical paths (in the `Path/disk` column), starting from zero.

The command to change the primary path shown in Figure 4-1, on page 25, for the SVA device with the `spd` ID of `0` from its default path to controller 7 to the path to controller 8, then, would be:

```
# setsp -l0 -p1
```

If you run a `setsp` command with the verbose option (`-v`), the configuration change will be displayed on screen (below).

```
# setsp -v -l0 -p1
==============================================================================
 spd   Path/disk     Status   Pri    Exc    Buf Balance RtrCnt  RtrDly FailBack
==============================================================================
   0   c7t0d0/sd66    Good                   32    0       20     3000      1
       c8t0d0/sd88    Good     X
 spd0 = c3t0d0                                 ID = "STK V960 0000000010390000"
==============================================================================
```

Figure 4-2   Using setsp -p in Verbose Mode

Turning the Exclusion Setting Off and On

The syntax for changing the exclusion setting for a device is

```
setsp -l<x> -e{0|1}
```

where <x> is the spd number (or `all`) and the -e option takes one of two arguments:

- -e0 turns exclusion off (makes the device visible to the host);
- -e1 turns exclusion on (excludes the device).

In a single-host configuration all devices should be visible to the host. Devices can be included (or unexcluded) by using the command.

```
setsp -l all -e0
```

All devices may be included, because no other host can access them at the same time. However, in a multiple-host environment, where all spds are visible to SVA Path on all hosts, spds must be either excluded or included so that a host shares no spds. Use the ID number under the `spd` column on the `setsp -a` output to identify devices. Do not use the `spdX` number.

Redefining the Buffer Pointer Allocation

The syntax for changing the number of buffer pointers pre-allocated for a given device is

```
setsp -l<x> -n<n>
```

where *<x>* is the numerical element of the `spd` number (see "Specifying a Device for setsp" on page 27) and *<n>* is the new value. The number of buffer pointers recommended for a given device is approximately equal to the logical unit's maximum useful queue depth; the default value of `32` should suit most SVA devices.

Changing the Retry Count and Retry Delay

The syntax for changing the retry count for a device is

```
setsp -l<x> -r<n>
```

where *<x>* is the numerical element of the `spd` number (see "Specifying a Device for setsp" on page 27) and *<n>* is the number of times a failed I/O will be retried on the primary path (after its alternate paths have been tried unsuccessfully) before the path is marked as failed (with a `Status` of `Bad`).

The syntax for changing the retry delay for a device is

```
setsp -l<x> -d<n>
```

where *<x>* is the numerical element of the `spd` number and *<n>* is the interval in milliseconds between the retries specified by the retry count parameter.

Turning Failback Off and On

The syntax for changing a device's failback mode is

```
setsp -l<x> -f{0|1}
```

where <*x*> is the numerical element of the `spd` number (see "Specifying a Device for setsp" on page 27) and the `-f` option takes one of two arguments:

- `-f0` turns failback off;
- `-f1` turns failback on.

Configuring a New LUN

The new LUN must be created on the SVA before configuring it on SVA Path.

1. Run the following command to configure the `/devices` directory:

   `#` **`drvconfig -i sd`**

2. Run the following command to add `/dev` entries for the new LUN:

   `#` **`disks`**

3. Run the following to confirm that the new devices have been created:

   `#` **`format`**

4. Run the following command to update the `/etc/sppath.conf` configuration file:

   `#` **`sppath`**

5. Run the following command to update the `/etc/spd.conf` file:

   `#` **`setsp -g`**

6. Run the following command to stop the spd driver:

   `#` **`setsp -T`**

7. Start the spd driver for any new devices that have been created:

   `#` **`setsp -S`**

8. To turn off exclusion on the new logical device, run the following command:

```
# setsp -l<spd#> -e0 -l(LUN number)
```

9. Run the following:

```
# drvconfig
```

10. The run the following:

```
# disks
```

11. If you are running SVA Path with Veritas, run the following command, which will allow Veritas to scan for any new drives:

```
# vxdctl enable
```

Removing LUNs from SVA Path

When a LUN on the SVA is deleted, SVA Path will continue to reference it until the next re-boot. If desired, use the

```
# setsp -T -l <spd#>
```

command to remove the reference immediately.

Excluding a Device Using the -x Option

The -x option causes a device to be inaccessible from SVAPath. The device then will still be under SVA Path's control but can only be accessed from the system. This option is useful with LUN masking.

1. To exclude a device using the -x option, run the

```
# setsp -x -l<spd#>
```

command, where spd# is the number of the SPD to be excluded.

2. Next, you need to stop the spd using the following command:

```
setsp -T -l<spd#>
```

> **Note:** Notice that the SPD is still in sppath.conf, but not in spd.conf, indicating that this spd is still under SVAPath's control.

To check for devices that have been excluded using the -x option, type the following command:

```
setsp -u2
```

To remove the device from the excluded devices list (displayed with the u2 option), run the following commands:

```
setsp -g -l<spd#>
```

```
setsp -S -l<spd#>
```

Chapter 5. SVA Path Operation

This chapter describes SVA Path commands and options.

There are three basic commands in SVA Path

- `setsp` is used to examine and configure the system's operating parameters, and as such is the command most often invoked by the user.
- `spmon` monitors path states and implements load balancing.
- `sppath` identifies devices to be placed under SVA Path's control.

The setsp Command

The `setsp` command is used for most configuration tasks.

When specifying devices, only one device or all devices should be specified for any given command. Whether you are applying a configuration command to a single device or to all devices, only one parameter should be configured in a given command.

Table 5-1, "`setsp` Command Options", on the next page gives a brief explanation of `setsp` command options. Those used to configure SVA Path device parameters are covered in some detail in "Changing the Default Configuration" in the preceding chapter. In these cases, the page number for an option's more detailed explanation is noted parenthetically.

The output of `setsp` is also used in conjunction with the event listings in `/var/adm/messages` to determine the nature and physical location of failures. This is covered, in "Diagnosing Errors" starting on page 49.

Commands that take arguments (shown in brackets after the command) *require* an argument and should not be run without one.

Table 5-1 setsp Command Options

Option	**Effect**
-a	show current device configuration and state (default behavior if no other option is specified)
-b{0|1}	set load balancing for a logical drive: -b0 disables load balancing for the device; -b1 enables it (enabled by default)
-d<n>	set a retry delay of *n* for a logical drive, where *n* is the interval between retries in milliseconds (Page 29)
-e{0|1}	set exclusion for a logical drive: -e0 includes the device; -e1 excludes it; excluded devices are not accessible by user applications; devices are excluded by default
-f{0|1}	set failback for a logical drive: -f0 disables failback for the device; -f1 enables it (Page 29)
-g	generate configuration files after an sppath command (Page 46)
-i	show contents of driver configuration files

Table 5-1 setsp Command Options

Option	Effect
-L {parameter}	show current device configuration according to condition(s) defined by setsp command option parameter(s) (e.g., setsp -L -e1 lists all excluded devices; setsp -L -b0 -f1 lists all devices that have load balancing disabled and failback enabled); acceptable parameters are: -l, -e, -p, -b, -r, -d, and -f
-l{<n>|all}	specify a logical drive for the command, where; n is the drive's spd number ; all specifies all SVA Path devices (Page 27)
-N	runs a command to change the configuration files without affecting the running system (changes will take effect at the next boot)
-n<n>	allocate *n* buffer pointers for a logical drive, where *n* should be a number approximately equal to the device's maximum queue depth (Page 29)
-p<*n*>	select primary path *n*. Must be used with -l.
-r{n}	set a retry count of *n* for a logical drive, where *n* is the number of times a command will be retried (Page 29)
-S	start the spd driver
-T	terminate the spd driver
-x	ignores spd devices (changes will take effect at next reboot). Ignored devices behave like drives not under the control of the SVA Path driver (Microsoft Windows only).

Table 5-1 setsp Command Options

Option	Effect
`-u{0|1|2}`	show devices by their configuration status: `-u0` shows all available devices; `-u1` shows configured disks; `-u2` shows unconfigured disks
`-v`	runs a command in verbose mode (Page 28)

The spmon Command

The `spmon` command is primarily associated with load monitoring and balancing of SVA Path functional devices (FDevs). FDev is a logical disk as viewed by the host operating system, the applications and the users. An FDev can emulate one of a variety of SCSI and count-key-data (CKD) disk devices.

Note: For optimum performance when striping data across multiple LUNs, disable load balancing on the devices being striped using the `setsp -b0` option.

Configuring Load Balancing

Load balancing is enabled or disabled using `setsp -l all -b0` (disable) or `setsp -l all -b1` (enable) commands, with the default setting being load balancing is enabled. Parameters for load balancing are stored in the configuration file `/etc/spmon.conf`. This file is read automatically after each modification at the end of the expiration of the last measurement interval.

Following is a listing of available parameters in `/etc/spmon.conf`:

- **`balance-threshold`** ***`percentage`***

  This parameter is used to specify the maximum difference in load on the high and low paths that is permitted before the paths will be considered imbalanced. The load balancing equation must be less than the percentage value of `balance-threshold` to be considered balanced.

- **`reassignment-threshold`** ***`percentage`***

  This parameter determines whether or not a reassignment will be considered worthwhile.

- **`measurement-interval`** ***`time`***

  This parameter accepts an positive integer value with a suffix of "s" (seconds), "m" (minutes), or "h" (hours). A reasonable minimum value will be based on the CPU load presented to the system by the algorithm and the maximum value is based on the amount of time that can pass before the driver's internal counters overflow.

- **`reassignment-limit`** ***`number`***

  This parameter specifies the maximum number of devices that should be moved in one pass of the algorithm. The default value is equal to or one half of the FDevs. If the path group includes one or more multi-FDev LUNs, each FDev is considered a separate device.

- **`read-overhead`** ***µs-per-cmd µs-per-sector***

  This parameter is used to specify how bus connect time overhead is estimated for read commands. The first value specifies the number of microseconds estimated for read command overhead, while the second is an estimate of connect time required for each 512 bytes of data requested. For each read operation, the sum of the command overhead and the product of the transfer length and the per-sector overhead is added to a counter that is used to estimate overall bus utilization on a per-FDev and per-channel basis.

- **`write-overhead`** ***µs-per-cmd µs-per-sector***

  This parameter is analogous to the read-overhead statement. Sample values for these two statements are:

  ```
  read-overhead  1000 120
  write-overhead 1000 160
  ```

Tuning Path Balancing

The path balancing algorithm captures details about the I/O load going to each device, then enters an algorithm which performs several tests to determine whether a device movement will improve the balance across all possible physical paths.

For the purpose of this section, moving a device means re-directing I/O from one physical interface to another. This creates a certain amount of work to be done by the SVA. While tuning using the parameters below, you should consider how to achieve a satisfactory balance while moving the least amount of devices:

> Parameters should be changed one at a time and then monitored for a period of time. An indication of an incorrect parameter value can either be no device movements or too many device movements. Try to always err on the side of no device movements and adjust slowly until device movement is seen.

All device movements are logged to the system log file:

`/var/adm/messages`

Additionally, the `spmon` command can be a valuable asset to observe system load numbers either real-time, or over a period of time. The show option of `spmon` will display load information every time it is issued.

There is also a **log-data-directory** parameter in the `/etc/spmon.conf` file which identifies a file to place load information every time the algorithm runs, i.e., every measurement interval. This file can then be imported into a graphing utility, and load peaks and lows can be seen over time.

Read / Write µs-per-cmd & sector

The first 2 parameters which can be used to tune the algorithm are:

- Read-overhead µs-per-cmd µs-per-sector
- Write-overhead µs-per-cmd µs-per-sector

These parameters aid in determining the I/O load to the devices. Since writes require more subsystem resources than reads, these parameters set the skew. They must be altered to match the prevalent RFA blocksize being used by the operating system per the chart below:

Table 5-1Blocksize

	512 Byte	2K Byte	4K Byte	8K Byte	16K Byte
Read µs-per-sector	110	45	25	25	24
Write µs-per-sector	180	68	44	33	32

Note: When tuning, leave the Read / Write µs-per-command at 1000. It is also recommended to only change one of these parameters since they have a co-relationship.

Measurement Interval

The next parameter to consider tuning is the measurement interval time. This parameter determines how long the I/O load will be measured before entering the algorithm itself.

Raising the value too high could overrun the program buffers and cause errors.

Setting the value too low renders the sample unique, to the extreme where it is not representative of the actual load.

The value of this parameter should remain in the range of 20 - 60 seconds.

Balance Threshold Percentage

Balance Threshold Percentage is the first actual Load Balancing Algorithm parameter. It determines how far apart, in load, the paths can be allowed to get before the algorithm progresses with further tests. The actual formula is:

```
High_Load_Path - Low_Load_Path > High_Load_Path * Balance_Threshold_Percent
```

If the result of this formula is true, the algorithm will continue to the Re-assignment Threshold test. (See below.)

The default value for Balance Threshold Percentage is 10%. Try setting it lower if the load is mostly small I/O 's and higher if there are mostly large I/O's. As with all parameters, an indication of an incorrect parameter is either no movement or too many movements. Bear in mind that with a smaller number, more movement is likely; larger numbers restrict movement.

Re-assignment Threshold Percentage

Re-assignment Threshold Percentage is the second Load Balancing Algorithm parameter, and determines whether a device is worth moving. It forces the software to locate the best device to move and therefore achieve the tolerable balance set in the Balance Threshold Percent parameter. It asks the question, "Will an X% improvement be made to the balance?"

The default value is 50%. Setting this parameter too high can cause too many movements and impact device performance. Too low, and no devices will ever qualify to be moved.

For example, if there were a theoretical load on path A of 100 and a load on path B of 50:

> If Balance Threshold Percent = 10%, the maximum tolerable out-of-balance condition is within 65 - 85, with the ideal at 75.
>
> If the Re-Assignment Threshold Percentage is set to 50%, only a device with a load of 12.5 to 25 would be a candidate to be moved.

12.5 would change path A to 87.5 and path B to 62.5: a 50% improvement.

Re-Assignment Limit Number

The final tuning parameter is Re-Assignment Limit Number. This parameter controls how many devices can be moved after each Measurement Interval.

Set this parameter to 1/2 the number of paths under SVA Path control on that host, for example:

- If there are 2 paths, set it to 1
- If there are 4 paths, set it to 2
- If there are 3 paths, experiment with both 1 and 2, starting with 1.

How to Verify Load Balancing

In Figure 5-3, there are four SVA Path devices; spd0, spd1, spd2 and spd3.

```
# setsp -a
==============================================================================
  spd   Path/disk     Status   Pri    Exc    Buf Balance RtrCnt  RtrDly FailBack
==============================================================================
   0   c1t0d0/sd66    Good     X            32    1      20     3000      1
       c2t0d0/sd88    Good
       c4t0d0/sd145   Good
       c5t0d0/sd189   Good
 spd0 = c6t0d0                                   ID = "STK V960 0000000010850000"
==============================================================================
   1   c1t0d1/sd67    Good     X            32    1      20     3000      1
       c2t0d1/sd89    Good
       c4t0d1/sd146   Good
       c5t0d1/sd190   Good
 spd1 = c6t0d1                                   ID = "STK V960 0000000010850001"
==============================================================================
   2   c1t0d2/sd68    Good     X            32    1      20     3000      1
       c2t0d2/sd90    Good
       c4t0d2/sd147   Good
       c5t0d2/sd191   Good
 spd2 = c6t0d2                                   ID = "STK V960 0000000010850002"
==============================================================================
   3   c1t0d3/sd69    Good     X            32    1      20     3000      1
       c2t0d3/sd91    Good
       c4t0d3/sd148   Good
       c5t0d3/sd192   Good
 spd3 = c6t0d3                                   ID = "STK V960 0000000010850222"
==============================================================================
```

Figure 5-3 Example setsp -a Output in Determining Load Balancing

In this example, the following command will show that each device in Figure 5-3 is accessible via four paths, using HBAs "glm3", "glm4", “lpfc0” and “lpfc1”:

```
# spmon show
```

```
# spmon show
Total I/O load across monitored HBAs = 103.2s

    HBA Device     I/O Load  %load
    ------------------------------
    c1  glm3          40.1s  38.8%
    c2  glm4          20.8s  20.2%
    c4  lpfc0         21.2s  20.6%
    c5  lpfc1         21.1s  20.5%

Device c6t0d0 [STK 9200 0000000010850000]
    FDev      Path                  I/O Load   HBA Load
    ----------------------------------------------------------
    spd0:0    0/glm3/sd66               9.9s      40.1s  24.8%

Device c6t0d1 [STK 9200 0000000010850001]
    FDev      Path                  I/O Load   HBA Load
    ----------------------------------------------------------
    spd1:0    0/glm3/sd67              10.1s      40.1s  25.1%
    spd1:1    3/lpfc1/sd190            21.2s      21.1s 100.0%

Device c6t0d2 [STK 9200 0000000010850002]
    FDev      Path                  I/O Load   HBA Load
    ----------------------------------------------------------
    spd2:0    0/glm3/sd68               9.9s      40.1s  24.7%
    spd2:1    1/glm4/sd90              20.9s      20.8s 100.0%
    spd2:2    2/lpfc0/sd147            21.2s      21.2s 100.0%

Device c6t0d3 [STK 9200 0000000010850222]
    FDev      Path                  I/O Load   HBA Load
    ----------------------------------------------------------
    spd3:0    0/glm3/sd69              10.0s      40.1s  24.9%
```

Figure 5-4   Example spmon show Output

The column headings describe the various fields on the screen:

- **FDev** identifies the SVA Path device and the zero-based index of the FDev within the device, separated by a colon.
- **Path** identifies the current path assigned to the FDev and the name of the host bus adapter used by that path.
- **I/O Load** estimates channel utilization time during the current measurement interval. The I/O Load may be expressed as one of the following formats:

  `milliseconds` (with an "ms" suffix) if it is less than one second. For example, 350ms = 350 milliseconds

  `seconds` (with an "s" suffix) if it is less than 300 seconds. For example, 28.1s = 28.1 seconds `hours:minutes:seconds` (hh:mm:ss) if it is more than 59 minutes. For example, 0:12:42 = 0 hours, 12 minutes, and 42 seconds.

- **HBA Load** identifies the I/O load contributed by traffic to/from this FDev as a percentage of all I/O though the current HBA used to access this FDev. The figure in the right-hand column is a ratio `I/O Load` / `HBA Load`, expressed as a percentage.

In this example, all four paths have about the same load on them except for c1, which has nearly double the load of the other paths.

Shortly after this command was run, the following message appeared in the system log file `/var/adm/messages`:

```
spmon daemon: moving spd2:0 from glm3 to lpfc1
spmon daemon: moving spd3:0 from glm3 to glm4
```

Figure 5-5 Example Output of spmon Daemon

When the command, `spmon show`, is run again (Page 44), the output shows the load is redistributed among the four paths and the maximum load on c1 is significantly reduced from 38.8% to 32.5%:

```
# spmon show
Total I/O load across monitored HBAs = 105.1s

    HBA Device     I/O Load  %load
    ------------------------------
    c1  glm3          34.1s  32.5%
    c2  glm4          31.3s  29.8%
    c4  lpfc0         19.1s  18.1%
    c5  lpfc1         20.6s  19.6%

Device c6t0d0 [STK 9200 0000000010850000]
    FDev      Path                  I/O Load   HBA Load
    ---------------------------------------------------------
    spd0:0    0/glm3/sd66              16.9s      34.1s  49.5%

Device c6t0d1 [STK 9200 0000000010850001]
    FDev      Path                  I/O Load   HBA Load
    ---------------------------------------------------------
    spd1:0    0/glm3/sd67              17.3s      34.1s  50.5%
    spd1:1    3/lpfc1/sd190            10.2s      20.6s  49.6%

Device c6t0d2 [STK 9200 0000000010850002]
    FDev      Path                  I/O Load   HBA Load
    ---------------------------------------------------------
    spd2:0    3/lpfc1/sd191            10.4s      20.6s  50.4%
    spd2:1    1/glm4/sd90              15.4s      31.3s  49.3%
    spd2:2    2/lpfc0/sd147            19.1s      19.1s 100.0%

Device c6t0d3 [STK 9200 0000000010850222]
    FDev      Path                  I/O Load   HBA Load
    ---------------------------------------------------------
    spd3:0    1/glm4/sd91              15.9s      31.3s  50.7%
```

Figure 5-6   Example spmon show Output, Next Interval

Adding FDevs to an Existing Lun

After adding new FDevs to an existing lun via the “Add FDev” button on the SVA console, run the following command:

`# `**`spmon update-fdevs`**

This enables `spmon` to load balance the new FDevs separately from the rest of the FDevs that make up the LUN. If this command is not run, the FDevs will be treated as if they belonged to the last original FDev and will be prevented from being assigned to different paths. For example, if a LUN was originally composed of two FDevs `spd0:0` and `spd0:1` and the user adds three FDevs using the SVA console, the new FDevs are assigned to `spd0:1`.

The sppath Command

The `sppath` command is run automatically during reconfiguration reboots as part of a configuration process and should not normally be invoked by the user.

`sppath` examines disk devices attached to the system to determine the path(s) through which physical devices are accessible and whether those devices should be put under the control of the SVA Path driver. Qualifying devices are written to the `/etc/sppath.conf` file. This file should not normally be modified directly by the user; it contains data needed to ensure consistency across reconfiguration reboots.

To qualify, devices must have appropriate inquiry data.

The vendor ID must match one of:

- STK
- IBM
- RSBA

The product ID must match one of:

- 9200
- 9393
- 9500
- V960

`sppath`’s essential function is to create the configuration file that identifies devices to be put under SVA Path’s control. Its command options are used to display or modify this device set.

Table 5-2 gives a synopsis of `sppath` options. They are explained in greater detail below.

Table 5-2 sppath Command Options

Option	Effect
`-d`	display debug information
`-D`	clear the list of ignored devices. Should be followed by `setsp -g` (see note below)
`-I{cXtYdZ}`	ignore specified devices, where each device is preceded by the `-I` operand and identified by its device filename in the form `cX`, `cXtY`, or `cXtYdZ`. The argument to the sppath -I option should name the underlying device path, and <u>not</u> the spd device name. Run in conjunction with `setsp -t -l(n)`. Should be followed by `setsp -g` (see note below)
`-v`	display the contents of `/etc/sppath.conf` after writing the file

Note: After running `sppath` with options `-I` or `-D`, you must run `setsp -g` in order for the changes to be saved in SVA Path’s configuration files `/kernel/drv/spn.conf` and `/kernel/drv/spd.conf`. Changes will take effect at the next system reboot.

The `-v` option causes `sppath` to display the contents of `/etc/sppath.conf` after updating it (Figure 5-7).

```
# sppath -v
SPD=0 c5t0d0 dev=32,3840 type=2 SANID="STK V960 000000001038000E"
SPD=0 c6t0d0 dev=32,4800 type=2 SANID="STK V960 000000001038000E"
SPD=1 c5t0d1 dev=32,3848 type=2 SANID="STK V960 000000001038000F"
SPD=1 c6t0d1 dev=32,4808 type=2 SANID="STK V960 000000001038000F"
SPD=2 c1t0d0 dev=32,240 type=2 SANID="STK V960 000000001038000C"
SPD=3 c1t0d5 dev=32,2192 type=2 SANID="STK V960 000000001038021C"
SPD=4 c1t1d5 dev=32,2248 type=2 SANID="STK V960 000000001038021D"
SPD=5 c2t0d6 dev=32,3040 type=2 SANID="STK V960 00000000103802F4"
SPD=6 c2t1d6 dev=32,3096 type=2 SANID="STK V960 00000000103802F5"
```

Figure 5-7 Using sppath in Verbose Mode

The `-d` option displays inquiry data in raw format and is generally used only for debugging.

The `-I` option accepts symbolic device names corresponding to controllers or specific disks and omits them from `/etc/sppath.conf`. This prevents them from being put under SVA Path’s control. If a device is ignored, all of its paths should be specified with `-I` options. Multiple devices can be specified in a single `sppath` command, but each device specified must be preceded by `-I`. Once specified, the ignored device is remembered in `/etc/sppath.conf` and will be ignored until the list of ignored devices is cleared with the `-D` option.

The `-D` option clears the entire list of ignored devices, allowing any eligible device to be placed under SVA Path’s control upon the next reconfiguration reboot. To delete a specific ignored device (without deleting the entire list), remove the statement for only that device in `/etc/sppath.conf`.

Chapter 6. Diagnosing Errors

Path Failure Errors

SVA Path necessarily interfaces with system components at the I/O path level, which does not support sophisticated device error reporting. The application reports failed I/O to `/var/adm/messages` as well as when:

- a data path is found to have stopped functioning
- I/O is redirected to an alternate path
- the failed path resumes functioning.

Whether a path failure is intermittent or is caused by a hardware failure can usually be deduced from the events listed in `/var/adm/messages`, and a faulty device can be located physically by cross referencing the output of `setsp`.

Figure 6-1 shows the example configuration used in this section. Figure 6-2, on page 50 illustrates the lines from `/var/adm/messages` that will appear when a cable is pulled during active I/O. The `sd` or `ssd` driver performs a number of retries before reporting error status 5 (`i/o errno = 5` in `/var/adm/messages`) to the `spd` driver.

```
# setsp -a
==============================================================================
 spd   Path/disk     Status   Pri    Exc    Buf Balance RtrCnt  RtrDly FailBack
==============================================================================
   0   c7t0d0/sd66    Good     X             32    0      20     3000      1
       c8t0d0/sd88    Good
 spd0 = c3t0d0                                 ID = "STK V960 0000000010390000"
==============================================================================
   1   c7t0d1/sd67    Good     X             32    0      20     3000      1
       c8t0d1/sd89    Good
 spd1 = c3t0d1                                 ID = "STK V960 0000000010390001"
==============================================================================
   2   c7t0d2/sd68    Good     X             32    0      20     3000      1
       c8t0d2/sd90    Good
 spd2 = c3t0d2                                 ID = "STK V960 0000000010390002"
==============================================================================
```

Figure 6-1 Example setsp -a Output for a Fully Functioning System

Note: No value other than 5 is expected, but it may be possible in rare circumstances to see other values such as 6 (ENXIO--No such device or address) or 22 (EINVAL--Invalid argument) in the event of a configuration problem. (A reconfiguration reboot will normally clear these problems.)

```
unix: WARNING: /pci@1f,4000/scsi@2/sd@0,0 (sd66):
    SCSI transport failed: reason 'incomplete': retrying command

unix: WARNING: /pci@1f,4000/scsi@2/sd@0,0 (sd66):
    SCSI transport failed: reason 'incomplete': retrying command

unix: WARNING: /pci@1f,4000/scsi@2/sd@0,0 (sd66):
    SCSI transport failed: reason 'reset': retrying command

unix: WARNING: /pci@1f,4000/scsi@2/sd@0,0 (sd66):
    SCSI transport failed: reason 'incomplete': retrying command

unix: WARNING: /pci@1f,4000/scsi@2/sd@0,0 (sd66):
    SCSI transport failed: reason 'reset': retrying command

unix: WARNING: /pci@1f,4000/scsi@2/sd@0,0 (sd66):
    disk not responding to selection

unix: spd0: path 0 error (i/o errno=5)
```

Figure 6-2 Output of /var/adm/messages When a Cable Is Pulled

In the final message above, `spd0` can be located in the output from `setsp` to determine that the affected device is `c3t0d0`. The message also calls out `path 0` which corresponds to `c7t0d0/sd66` in the `setsp` output, which matches the device name (`sd66`) called out in the earlier messages. (The paths associated with a SVA Path device [`spd`] are numbered from 0 to *n*-1 and are listed in that order in the `setsp` output, so, for example, `spd2` path 1 is `c8t0d2` in the `setsp` output above).

Errors for other `spd` devices, representing the same physical I/O path that was interrupted by the pulled cable, will be reported, as shown in Figure 6-3.

```
unix: WARNING: /pci@1f,4000/scsi@2/sd@0,2 (sd68):
    disk not responding to selection

unix: spd2: path 0 error (probe failed; errno=5)
unix: WARNING: /pci@1f,4000/scsi@2/sd@0,1 (sd67):
    disk not responding to selection

unix: spd1: path 0 error (probe failed; errno=5)
```

Figure 6-3 Additional Messages for the Same Pulled Cable

Even when an `spd` device is idle, a SCSI Test Unit Ready command is sent periodically to the device to confirm that it is functioning. A failure of this test is reported as "`probe failed`" along with the `errno` value as described above.

Failover events are recorded in `/var/adm/messages` as well. The event in Figure 6-4 records that I/O for the device `spd0` will be redirected from path 0 (`c7t0d0`) to path 1 (`c8t0d0`).

```
unix: spd0: path failover from 0 to 1.
```

Figure 6-4 The Failover Event

The output from `setsp -a` (Figure 6-5, "`setsp` -a Output after the Failed Path Is Detected") now shows that every device's path 0 (those corresponding to device filenames beginning `c7`) have a `Status` designation of `Bad`.

```
# setsp -a
================================================================================
 spd   Path/disk     Status   Pri    Exc    Buf Balance RtrCnt  RtrDly FailBack
================================================================================
   0   c7t0d0/sd66    Bad       X            32    0      20     3000      1
       c8t0d0/sd88    Good
 spd0 = c3t0d0                                 ID = "STK V960 0000000010390000"
================================================================================
   1   c7t0d1/sd67    Bad       X            32    0      20     3000      1
       c8t0d1/sd89    Good
 spd1 = c3t0d1                                 ID = "STK V960 0000000010390001"
================================================================================
   2   c7t0d2/sd68    Bad       X            32    0      20     3000      1
       c8t0d2/sd90    Good
 spd2 = c3t0d2                                 ID = "STK V960 0000000010390002"
================================================================================
```

Figure 6-5 setsp -a Output after the Failed Path Is Detected

Path failures generate a lot of output, resulting primarily from the disk driver's attempts at error recovery (Figure 6-6, "Messages Generated by Disk Driver's Retry Attempts").

```
unix: WARNING: /pci@1f,4000/scsi@2/sd@0,0 (sd66):
    disk not responding to selection

unix: WARNING: /pci@1f,4000/scsi@2/sd@0,2 (sd68):
    disk not responding to selection

unix: WARNING: /pci@1f,4000/scsi@2 (glm3):
    Connected command timeout for Target 0.1
```

Figure 6-6 Messages Generated by Disk Driver's Retry Attempts

Finally, when the cable is plugged back in, SVA Path will detect that the primary I/O path for `spd0` is back to normal and redirect data to it (Figure 6-7, "The Failback Event").

```
unix: spd1: path 0 ok (probe errno=0)
unix: spd2: path 0 ok (probe errno=0)
```

Figure 6-7 The Failback Event

The other paths will be recovered, as well, and the output of `setsp -a` will return to normal (Figure 6-1, on page 49). Events were not reported for these recoveries in our example because I/O was not redirected for these devices (`spd1`'s primary path did not fail, and `spd2` was idle during the failure).

Load Balancing Messages

Load Balancing messages are logged with a severity code of "error" to ensure that those messages are placed in the system log file without having to edit the `/etc/syslog.conf`.

Note: Messages are logged differently for the Solaris 8 operating system. They are logged in `/var/adm/messages`, as follows:

```
[ID 702911 daemon.error]
moving spd2:1 from Ipfc1 to Ipfc0
```

Appendix A. SCSI Attach Information

The information provided in this section applies to SCSI attachment for SVA, which was previously available. It is provided only as a reference to users of older, SCSI-attached SVAs.

Table A-1 SVA Path Minimum System Requirements

Host hardware:	Sparc 2 with 128MB memory
Host software:	Solaris 2.6 (with Patch 105181-13 or later), or 7 (32, 64 bit), plus the applicable recommended patch cluster as of August 1, 1999 (the recommended cluster contains all patches needed for running the SVAA server).
HBA (SCSI)	SUN X1062A (DWIS) 2.6, 7 SBUS SUN X1065A (UDWIS) 2.6, 7 PCI X6541 A (Dual Channel HVD)
Host software	X Windows
Host disk space:	20MB free space in $TEMP and 20MB free space in the server installation directory.
SVA Subsystem Microcode:	K05.04.08 or greater
Extended SCSI Attach Feature:	One Extended SCSI Attach (microcode S358) per redundant SCSI path

SCSI Addressing Concepts

Host Bus Adapters/Initiators

The terms “host bus adapter” and “initiator” mean essentially the same thing. Typically, the HBA is a controller card installed within the host computer that assumes the role of initiator and issues commands on the SCSI bus. In the mainframe world, this is the rough equivalent of a channel.

Domains

StorageTek uses the concept of “domains” to allow open systems hosts access to blocks of logical devices (the domains) within an SVA. A domain is an additional layer of device addressing, but one that is manually set in the XSA feature and in the SVA. This layer of addressing divides the SVA into “domains of access.” There can be up to 16 (0–15) domains per SVA, with each domain having up to 120 devices (LUNs). There is a limit of 1024 total devices available within an SVA.

Each open systems host initiator is connected, through the XSA, to a domain, giving it access to the devices that have been configured within that domain. (An open systems host cannot see devices in domains other than the one to which it is attached.) There is no correlation between the XSA’s two SCSI ports to the host(s) and the two fibre cables connecting the XSA to the SVA. I/O requests from either SCSI path can be routed to either one of the two fibre paths.

Multiple initiators may be connected to the same domain through multiple XSAs. Full SVA Path functionality requires that redundant initiators can access the SVA over redundant data paths—which must include redundant XSAs.

. Using all the combinations of allowed domains, targets, and LUNs, there are more than 1024 potential addresses, but the SVA has a limit of 1024 logical devices.

Figure A-1, “Redundant Paths from the Host to the SVA” shows two data paths connecting the open systems platform to domains 0 and 1 of the attached SVA.

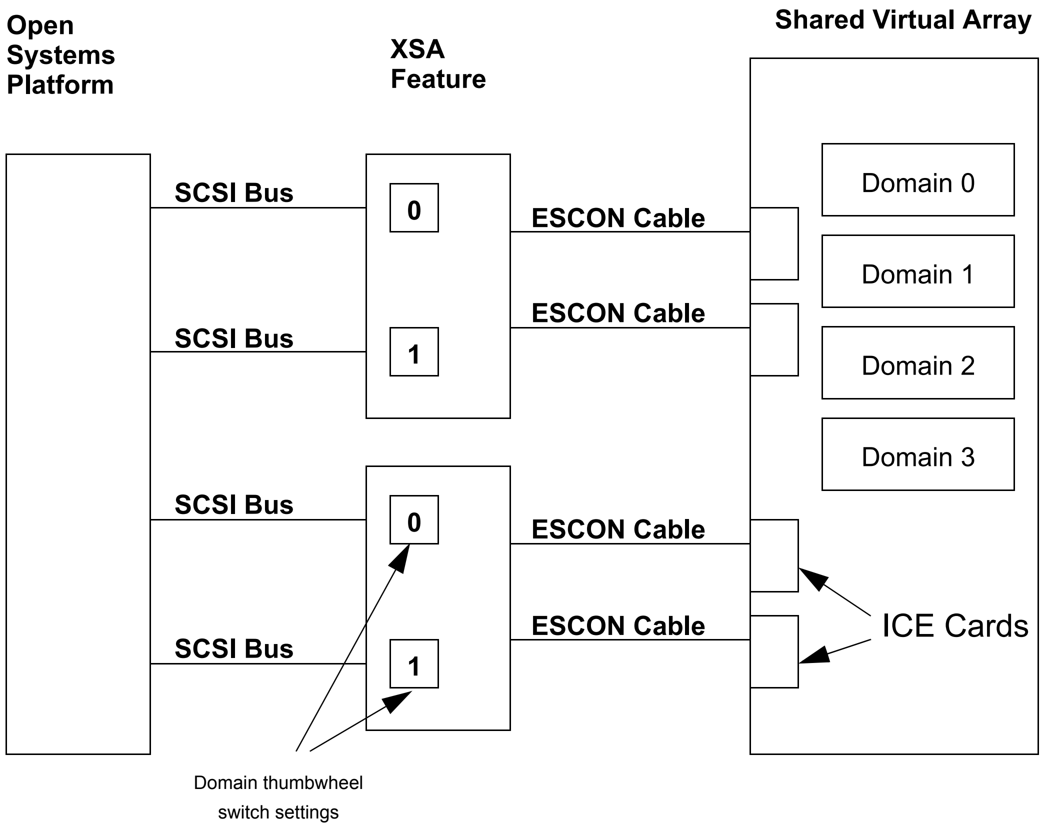

Figure A-1 Redundant Paths from the Host to the SVA

Note: In the above figure, a domain can represent up to 120 logical devices (15 Targets × 8 LUNs = 120 Logical Devices). An SVA is limited to 1024 logical devices.

Domain Numbering

The domain number is configured in two places: in the SVA and in the XSAs. Both of these are transparent to the host platform.

The domain number is appended by the XSA to the I/O request and sent to the SVA (along with the XSA’s serial number) to indicate which domain within the SVA the I/O request is addressing. The XSA serial number is used to make sure the information is returned to the correct XSA, since more than one XSA with the same domain numbers can be attached to an SVA. (see “XSA Domain Numbering with SVA Path” below). An I/O request is returned to the XSA back through the fibre channel over which it was received, with the domain number included, so the XSA will know to which host initiator to return the information.

Again, the domain number is never seen by the open systems host; from its point of view, just a target and logical unit number are involved in an I/O operation.

XSA Domain Numbering with SVA Path

While it is not normally advisable to attach more than one XSA configured with the same domain numbers to a given SVA (in case two hosts attempt to share the same LUN and so corrupt the data stored on it), SVA Path requires just this configuration. One XSA is required for each instance of data path redundancy: dual-path configurations require two XSAs, etc.

Once more, SVA Path manages multiple paths from a single host, using identical domain numbers. This is the only case where multiple XSAs will have the same domain number/thumbwheel settings.

Configuration in a Mixed SCSI/Fibre Environment

Failover can only be accomplished from one SCSI interface to another, or from one Fibre interface to another.

Figure A-2, "Mixed SCSI and Fibre Paths from the Host to the SVA" shows SCSI and Fibre data paths connecting the open systems platform to the attached SVA.

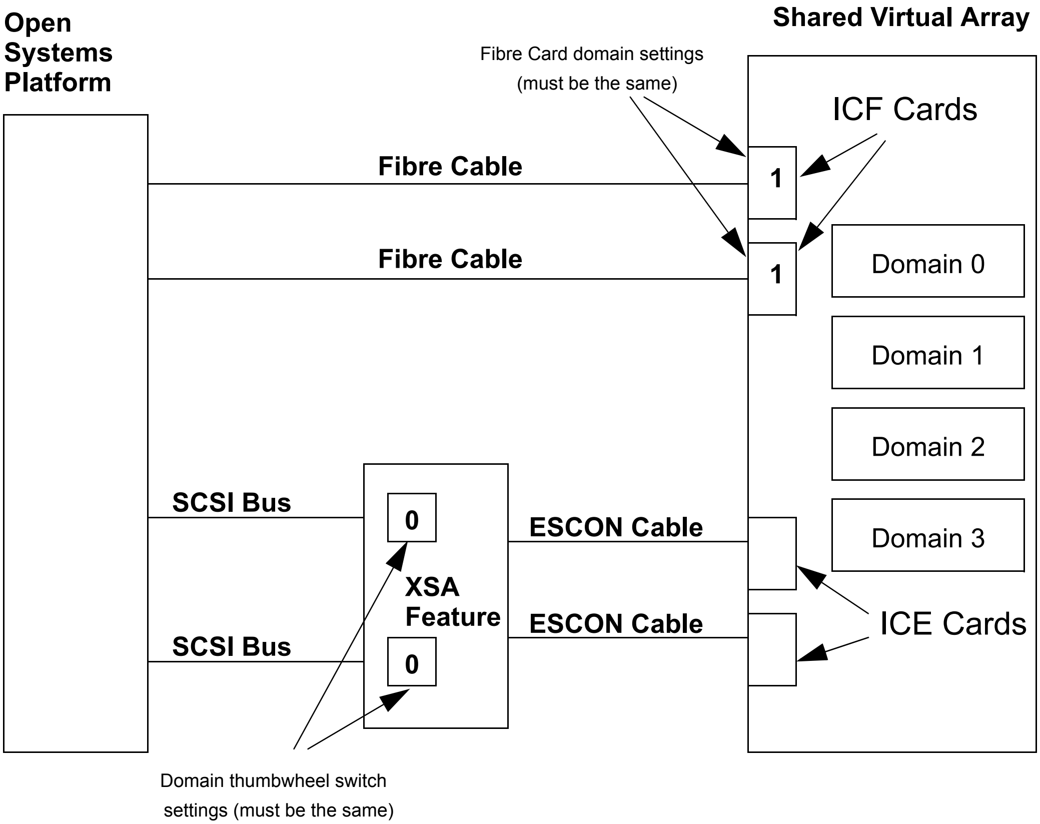

Figure A-2 Mixed SCSI and Fibre Paths from the Host to the SVA

READER'S COMMENT FORM

Manual Name: ______________________________ Manual PN: __________

Please check or fill in the items; adding explanations/comments in the space provided.

Which of the following terms best describes your job?

- ☐ Field Engineer
- ☐ Engineer
- ☐ Instructor
- ☐ Manager
- ☐ Mathematician
- ☐ Operator
- ☐ Programmer
- ☐ Sales Representative
- ☐ Student/Trainee
- ☐ Systems Analyst
- ☐ Systems Engineer
- ☐ Other (explain below)

How did you use this publication?

- ☐ Introductory text
- ☐ Reference manual
- ☐ Student/Trainee
- ☐ Instructor text
- ☐ Other (explain) ______________________________

Did you find the material easy to read and understand? ☐ Yes ☐ No (explain below)

Did you find the material organized for convenient use? ☐ Yes ☐ No (explain below)

Specific criticisms (explain below):

Clarifications on pages ______________________________

Additions on pages ______________________________

Deletions of pages ______________________________

Errors on pages ______________________________

Explanations and other comments:

Note: *Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form. If you would like a reply, please supply your name and address on the reverse side of this form.*

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A.

TO COMPLY WITH POSTAL REGULATIONS, FOLD EXACTLY ON DOTTED LINES AND TAPE (DO NOT STAPLE)

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 2 LOUISVILLE, CO U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

STORAGE TECHNOLOGY CORPORATION
ENTERPRISE DISK INFORMATION DEVELOPMENT
ONE STORAGETEK DRIVE
LOUISVILLE CO 80028-2201

FOLD HERE AND TAPE DO NOT STAPLE FOLD HERE AND TAPE

If you would like a reply, please print:

Your Name: ______________________________

Company Name: ____________________ Department: ____________________

Street Address: ______________________________

Developed in U.S.A