

# **Oracle® Solaris Administration: SAN Configuration and Multipathing**

Copyright © 2009, 2012, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

# Contents

---

<b>Preface</b> .....	7
<b>1 Solaris I/O Multipathing Overview</b> .....	11
What's New in Solaris I/O Multipathing? .....	11
Overview of Solaris I/O Multipathing .....	13
Fibre Channel Software Features .....	13
SAS Software Features .....	14
Solaris I/O Multipathing Features .....	15
Supported Device Standards .....	17
<b>2 Fibre Channel Multipathing Configuration Overview</b> .....	19
FC Device Configuration Task Overview .....	19
Solaris I/O Multipathing Configuration Considerations .....	20
<b>3 Configuring Solaris I/O Multipathing Features</b> .....	21
Configuring Multipathing I/O Features .....	21
Multipathing Considerations .....	21
Enabling and Disabling Multipathing .....	22
▼ How to Enable Multipathing .....	23
▼ How to Disable Multipathing .....	24
Enabling or Disabling Multipathing on a Per-Port Basis .....	25
Port Configuration Considerations .....	25
Configuring Third-Party Storage Devices .....	27
Third-Party Device Configuration Considerations .....	27
Configuring Third-Party Storage Devices: New Devices .....	28
Configuring Third-Party Storage Devices: Disabling Devices .....	29
Displaying Device Name Changes .....	30

Configuring Automatic Failback .....	31
▼ How to Configure Automatic Failback .....	31
<b>4 Administering Multipathing Devices (mpathadm) .....</b>	<b>33</b>
Administering Multipathing Devices .....	33
▼ How to Display Multipathing Support Information .....	33
▼ How to Display Properties of a Specific Initiator Port .....	34
▼ How to Display Specific LUN Information .....	34
▼ How to Display All LUNs Associated With a Specific Target Port .....	36
▼ How to Display a LUN With a Specific Name .....	38
▼ How to Configure Automatic Failback for Specific Multipathing Support .....	40
▼ How to Fail Over a LUN .....	41
▼ How to Enable a LUN Path .....	44
▼ How to Disable a LUN Path .....	45
<b>5 Configuring SAN Devices .....</b>	<b>49</b>
SAN Device Considerations .....	49
Adding SAN Devices .....	50
▼ How to Add a SAN Device .....	50
Configuring Fabric Boot Devices on SPARC .....	51
Fabric Boot Device Considerations .....	51
<b>6 Configuring Virtual Fibre Channel Ports .....</b>	<b>53</b>
What is NPIV? .....	53
Limitations of NPIV .....	53
Creating NPIV Ports .....	54
▼ How to Create an NPIV Port .....	54
▼ How to Delete an NPIV Port .....	55
▼ How to Display NPIV Port Status .....	55
<b>7 Configuring FCoE Ports .....</b>	<b>57</b>
What is FCoE? .....	57
Limitations of FCoE .....	57
Configuring FCoE Ports .....	58

---

▼ How to Create an FCoE Port .....	58
▼ How to Delete an FCoE Port .....	59
▼ How to Display FCoE Port Status .....	59
▼ How to Force a FCoE Port Reinitialization .....	60
Configuring FCoE Hardware Offload .....	60
<b>8 Configuring SAS Domains .....</b>	<b>63</b>
SAS Multipathing Considerations .....	63
Dynamic Discovery of SAS Devices .....	63
Configuring SAS Boot Devices .....	64
<b>9 Configuring IPFC SAN Devices .....</b>	<b>65</b>
IPFC Considerations .....	65
Determining Fibre Channel Adapter Port Instances .....	66
Invoking and Configuring IPFC .....	68
▼ How to Start a Network Interface Manually .....	68
▼ How to Configure the Host for Automatic Network Configuration .....	69
<b>10 Booting the Solaris OS From Fibre Channel Devices on x86 Systems .....</b>	<b>71</b>
Oracle Solaris OS Setup Requirements .....	72
Oracle Solaris OS Installation Overview .....	72
Oracle Solaris OS Installation Procedure .....	73
▼ How to Install the Oracle Solaris OS .....	73
▼ How to Perform a DVD or Network-Based OS Installation .....	73
<b>11 Persistent Binding for Tape Devices .....</b>	<b>79</b>
Persistent Binding Overview .....	79
Creating Tape Links .....	80
▼ How to Create Tape Device Links .....	81
<b>A Manual Configuration for Fabric-Connected Devices .....</b>	<b>83</b>
Manually Configuring FC Devices .....	83
▼ How to Manually Configure a FC Device .....	83
Configuring Fabric Device Nodes .....	84

Ensuring That LUN Level Information Is Visible .....	85
Configuring Device Nodes Without Multipathing Enabled .....	86
▼ How to Manually Configure an FC Device Without Multipathing .....	86
▼ How to Configure Multiple FC Devices Without Multipathing .....	88
Configuring Device Nodes With Solaris Multipathing Enabled .....	89
▼ How to Configure Individual Multipathed FC Devices .....	89
▼ How to Configure Multiple Multipathed FC Devices .....	91
Unconfiguring Fabric Devices .....	92
Unconfiguring a Fabric Device .....	92
<b>B Supported FC-HBA API .....</b>	<b>99</b>
Supported Fibre Channel HBA API .....	99
<b>C Troubleshooting Multipathed Device Problems .....</b>	<b>103</b>
System Fails to Boot During stmsboot .....	103
▼ How to Recover Boot Failure in Single User Mode .....	103
System Crashes During stmsboot .....	104
▼ How to Recover from a System Crash .....	104
<b>Index .....</b>	<b>107</b>

# Preface

---

The *Oracle Solaris Administration: SAN Configuration and Multipathing* provides an overview of the Solaris I/O multipathing features, previously known as the SunStorageTek Traffic Manager software, as an integrated part of the Oracle Solaris OS. This guide includes instructions about how to install and configure the software and devices.

This guide is intended for system, storage and network administrators who create and maintain Fibre Channel (FC) storage area networks (SANs) and serial attached SCSI (SAS) Domains. A high level of expertise in the management and maintenance of SANS and SAS Domains is assumed.

## Before You Read This Document

Before you read this book, review the latest Oracle Solaris 11 Operating System (OS) release notes.

## How This Document Is Organized

Chapter	Description
<a href="#">Chapter 1, “Solaris I/O Multipathing Overview”</a>	Provides an overview of the Solaris I/O multipathing features.
<a href="#">Chapter 2, “Fibre Channel Multipathing Configuration Overview”</a>	Provides an overview of configuring Solaris I/O multipathing features.
<a href="#">Chapter 3, “Configuring Solaris I/O Multipathing Features”</a>	Describes how to configure the multipathing feature for FC devices.
<a href="#">Chapter 4, “Administering Multipathing Devices (mpathadm)”</a>	Describes how to use the <code>mpathadm</code> command to enable multipathing administration through the ANSI standard Multipath Management API.
<a href="#">Chapter 5, “Configuring SAN Devices”</a>	Provides high level overviews of the steps used to configure SAN devices.

Chapter	Description
Chapter 6, “Configuring Virtual Fibre Channel Ports”	Provides the steps used to configure N Port ID Virtualization (NPIV) ports, also known as virtual Fibre Channel ports.
Chapter 7, “Configuring FCoE Ports”	Provides the steps used to configure FCoE ports hosted on normal Ethernet interfaces.
Chapter 8, “Configuring SAS Domains”	Provides high-level overviews of the steps used to configure SAS domains.
Chapter 9, “Configuring IPFC SAN Devices”	Describes the considerations when configuring IPFC SAN Devices.
Chapter 10, “Bootting the Solaris OS From Fibre Channel Devices on x86 Systems”	Describes how to manually install the Oracle Solaris OS on x86 based systems.
Chapter 11, “Persistent Binding for Tape Devices”	Describes how you can specify tape binding in the Oracle Solaris OS while retaining the benefits of automatic discovery for disk-based devices.
Appendix A, “Manual Configuration for Fabric-Connected Devices”	Describes the manual configuration for Fabric Connected devices.
Appendix C, “Troubleshooting Multipath Device Problems”	Provides trouble shooting information for potential problems that might occur while you use Solaris I/O multipathing features.

## Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P-1 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>



TABLE P-1 Typographic Conventions (Continued)

Typeface	Meaning	Example
<b>AaBbCc123</b>	What you type, contrasted with onscreen computer output	machine_name% <b>su</b> Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <i>rm filename</i> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> save the file. <b>Note:</b> Some emphasized items appear bold online.

## Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P-2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	\$
Bash shell, Korn shell, and Bourne shell for superuser	#
C shell	machine_name%
C shell for superuser	machine_name#



# Solaris I/O Multipathing Overview

---

This chapter provides an overview of the Solaris I/O multipathing features, formerly known as StorageTek Traffic Manager software. The following topics are covered:

- “What's New in Solaris I/O Multipathing?” on page 11
- “Overview of Solaris I/O Multipathing” on page 13
- “Supported Device Standards” on page 17

## What's New in Solaris I/O Multipathing?

- **Multipathing package change** – The Oracle Solaris 11 package name is `system/storage/multipath-utilities`. For information about installing this package, see “How to Enable Multipathing” on page 23.
- **Multipathed device names** – After a system is installed with the Oracle Solaris 11 release and Solaris I/O multipathing is enabled, the multipathed device names begin with `c0`. For example:

```
# mpathadm list lu
/dev/rdisk/c0t600A0B800026D63A0000A4994E2342D4d0s2
  Total Path Count: 4
  Operational Path Count: 4
/dev/rdisk/c0t600A0B800029065C00007CF54E234013d0s2
  Total Path Count: 4
  Operational Path Count: 4
/dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
  Total Path Count: 4
  Operational Path Count: 4
/dev/rdisk/c0t600A0B800029065C00007CF44E233FCFd0s2
  Total Path Count: 4
  Operational Path Count: 4
/dev/rdisk/c0t600A0B800026D63A0000A4974E23424Ed0s2
  Total Path Count: 4
  Operational Path Count: 4
/dev/rdisk/c0t600A0B800029065C00007CF34E233F89d0s2
  Total Path Count: 4
  Operational Path Count: 4
```

```

/dev/rdisk/c0t600A0B800026D63A0000A4964E234212d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdisk/c0t600A0B800026D63A000099B94DE2DB34d0s2

```

```

.
.
.

```

- **N\_Port ID virtualization (NPIV)** – Is a Fibre Channel facility that enables one Fibre Channel adapter to have many *N* port IDs. Each *N* port has a unique identity (port WWN and node WWN) on the SAN and can be used for zoning and LUN masking. Soft zoning, which you can use to group ports together by port WWN, is the preferred method of zoning. For more information, see [Chapter 6, “Configuring Virtual Fibre Channel Ports.”](#)
- **Fibre Channel over Ethernet (FCoE)** – Is a proposed standard that is being developed by INCITS T11. The FCoE protocol specification maps Fibre Channel natively over Ethernet and is independent of the Ethernet forwarding scheme. The protocol provides I/O consolidation by preserving all Fibre Channel constructs, maintaining the same latency, security, and traffic management attributes of FC, while preserving investments in FC tools, training, and SANs. For more information, see [Chapter 7, “Configuring FCoE Ports.”](#)
- **FCoE port reinitialization** – You can use the `fcadm force -lip` command to force a port link to reinitialize. Be aware that a FCoE port reinitialization will likely cause a registered state change notification (RSCN) from the switch to all zoned initiators. For more information, see [“How to Force a FCoE Port Reinitialization” on page 60.](#)
- **Displaying MPxIO path information** – Both the `prtconf` command and the `fmdump` have been updated to provide MPxIO path information.

For example, the following output is from a system with multipathed devices.

```

# prtconf -v | grep path | more
Paths from multipath bus adapters:
    name='path-class' type=string items=1
    name='path-class' type=string items=1
    name='path-class' type=string items=1
    name='path-class' type=string items=1
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:a
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:a, raw
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:b
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:b, raw
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:c
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:c, raw
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:d
.
.
.

```

The `croinfo` command, which displays chassis, receptacle, and occupant information for disks or disk occupants of bay receptacles, can also be used to display multipathed device names. For example:

```

# croinfo -O occupant-compdev
c0t5000C50010420FEBd0
c0t5000C5000940F54Fd0

```

```
c0t5000C50007DD498Fd0
c0t5000C50002FB622Fd0
c0t5000C500103C9033d0
c0t5000C5000940F733d0
```

- **Device driver configuration** – Driver customizations are made in the `/etc/driver/drv` directory rather than in the `/kernel/drv` directory as in previous releases. This improvement means that your driver customizations are not overwritten when the system is upgraded. The files in the `/etc/driver/drv` directory are preserved during the upgrade. Any modifications to `fp.conf`, `mpt.conf`, or `scsi_vhci.conf` should be made in the `/etc/driver/drv` directory.

## Overview of Solaris I/O Multipathing

The Solaris I/O multipathing features enable multiple access paths for systems that are running the Oracle Solaris OS. Multipathing provides higher availability for storage devices through the use of multipathed connections.

- [“Fibre Channel Software Features” on page 13](#)
- [“SAS Software Features” on page 14](#)
- [“Solaris I/O Multipathing Features” on page 15](#)

The Solaris I/O multipathing features identify the storage devices on your SAN or SAS Domain. The software enables you to attach Fibre Channel storage devices in either loop, fabric, or point-to-point mode. The software provides a common interface for managing both Fibre Channel and SAS storage.

For information on how to configure targets and initiators in preparation for multipathing, see [Chapter 14, “Configuring Storage Devices With COMSTAR,”](#) in *Oracle Solaris Administration: Devices and File Systems*.

## Fibre Channel Software Features

Solaris I/O multipathing provides the following key features:

- **Dynamic storage discovery** – The software automatically recognizes devices and any modifications made to device configurations. This feature makes devices available to the system without requiring you to reboot or manually change information in configuration files.
- **Persistent device naming** – Devices that are configured within the software maintain their device naming through reboots or reconfiguration. The only exception to this policy are tape devices found in `/dev/rmt` that will not change unless they are removed, and then regenerated at a later date.
- **Fibre Channel Arbitrated Loop (FCAL) support** – OpenBoot PROM (OBP) commands that are used on servers can access FCAL attached storage for scanning the FC loop.

- Fabric booting – The Solaris OS supports booting from fabric devices as well as non-fabric Fibre Channel devices. Fabric topologies with Fibre Channel switches provide higher speed, more connections, and port isolation.
- FC-HBA library – What was previously known as the Storage Networking Industry Association Fibre Channel host bus adapter (SNIA FC-HBA) library is now known as the FC-HBA library. The FC-HBA library application programming interface (API) enables management of FC HBAs and provides a standards-based interface for other applications (such as Oracle's StorEdge Enterprise Storage Manager) that can be used to gather information about FC HBAs.  

Man pages for common FC-HBA APIs are included in the `libhbaapi(3LIB)` man page section. For additional information about FC specifications, refer to <http://www.t11.org>.
- Fibre Channel virtualization – N Port ID Virtualization (NPIV) is an extension to the Fibre Channel standard, which enables one Fibre Channel port to simulate many ports on the SAN. This is useful for virtualization environments such as Oracle VM Server for SPARC or Oracle VM Server 3.0 for x86.
- Fibre Channel over Ethernet (FCoE) – A new T11 standard to transport encapsulated Fibre Channel frames over Enhanced Ethernet is now available. Solaris FCoE is a software implementation that is designed to work with normal Ethernet controllers.

## SAS Software Features

- Dynamic storage discovery – The Oracle Solaris OS multipathing software automatically recognizes devices and any modifications made to device configurations. This makes devices available to the system without requiring you to reboot or manually change information in configuration files.
- Persistent device naming – Dynamic Storage Discovery Devices that are configured within the Solaris OS multipathing software maintain their device naming through reboots or reconfiguration.

## Solaris I/O Multipathing Features

Solaris I/O multipathing is enabled by default for x86-based platforms and optional for SPARC-based systems that run the Oracle Solaris OS. The software contains the following features:

- Path management – Solaris I/O multipathing features dynamically manage the paths to any storage devices it supports. The addition or removal of paths to a device is done automatically when a path is brought online or removed from a service. Even with multipathing enabled, you can add more controllers to increase bandwidth and RAS, without changing device names or modifying applications. Oracle storage products do not require configuration files to manage or databases to keep current. For storage other than Oracle's, contact your storage vendor for methods to enable support and to ensure it is qualified with Solaris I/O multipathing features.
- Single device instances – Solaris I/O multipathing features are fully integrated with the Oracle Solaris OS. Multipathed devices are displayed as single device instances instead of as one device or device link per path. This feature reduces the cost of managing complex storage architectures with utilities such as the `format` command or a volume management product to *see* one representation of a storage device instead of a separate device for each path.
- Failover support – Implementing higher levels of RAS requires redundant host connectivity to storage devices. Solaris I/O multipathing features manage the failure of storage paths while maintaining host I/O connectivity through available secondary paths.

You can determine the failover support for your devices by using the following command:

```
# mpathadm show mpath-support libmpscsi_vhci.so
mpath-support: libmpscsi_vhci.so
                Vendor: Sun Microsystems
                Driver Name: scsi_vhci
                Default Load Balance: round-robin
                Supported Load Balance Types:
                    round-robin
                    logical-block
                Allows To Activate Target Port Group Access: yes
                Allows Path Override: no
                Supported Auto Failback Config: 1
                Auto Failback: on
                Failback Polling Rate (current/max): 0/0
                Supported Auto Probing Config: 0
                Auto Probing: NA
                Probing Polling Rate (current/max): NA/NA
                Supported Devices:
                .
                .
                .
```

- Symmetrical/Asymmetrical device support – The following disk storage devices are supported:
  - All Oracle disk storage products, both symmetric and asymmetric
  - All T10/T11 standards-compliant third-party symmetrical disk devices

- Many third-party asymmetrical disk arrays
- T10 Asymmetric Logical Unit Access (ALUA) support has been added for asymmetrical devices that support this T10 standard. Contact your storage vendor to see whether your device is supported.

If your disk storage array uses the `f_asym_lsi` failover module, you will need to manually change the sequence of `ddi-forceload` section by moving the `f_asym_lsi` to end of the `scsi_vhci.conf` file, similar to the following:

```
ddi-forceload =
    "misc/scsi_vhci/scsi_vhci_f_asym_sun",
    "misc/scsi_vhci/scsi_vhci_f_asym_emc",
    "misc/scsi_vhci/scsi_vhci_f_sym_emc",
    "misc/scsi_vhci/scsi_vhci_f_sym_hds",
    "misc/scsi_vhci/scsi_vhci_f_sym",
#    "misc/scsi_vhci/scsi_vhci_f_tpgs_tape",
#    "misc/scsi_vhci/scsi_vhci_f_tape",
    "misc/scsi_vhci/scsi_vhci_f_tpgs",
    "misc/scsi_vhci/scsi_vhci_f_asym_lsi";
```

- I/O load balancing – In addition to providing simple failover support, Solaris multipathing features can use any active paths to a storage device to send and receive I/O. With I/O routed through multiple host connections, bandwidth can be increased by the addition of host controllers. The software uses a round-robin load-balancing algorithm by which individual I/O requests are routed to active host controllers in a series one after the other.
- Queue depth – SCSI storage arrays present storage to a system in the form of a logical unit number (LUN). LUNs have a finite set of resources available, such as the amount of data that can be stored, as well as the number of active commands that a device or LUN can process at one time. The number of active commands that can be issued before a device blocks further I/O is known as queue depth. When Solaris I/O multipathing is enabled, a single queue is created for each LUN regardless of the number of distinct or separate paths it may have to the host. This feature enables the disk driver to maintain and balance one queue to the LUN, effectively managing queue depth. No other multipathing software available for the Oracle Solaris OS has this ability.
- `stmsboot` command – The Oracle Solaris OS includes the `stmsboot` command, which allows you to enable or disable the Solaris multipathing features on your boot device after the OS installation has been completed. This command is available for both SPARC and x86 systems, and provides support for SAS multipathing.

SAS multipathing is not enabled by default in the Oracle Solaris OS for either SPARC or x86 systems, so use of the `stmsboot` command is required as a post-installation step if you wish to enable multipathing.

SPARC systems do not enable multipathing for FC devices by default, so use of the `stmsboot` command is required as a post-installation step.

X86 systems do enable multipathing of FC devices by default, so the use of the `stmsboot` command is not mandatory as a post-installation step.

- Dynamic reconfiguration – Solaris I/O multipathing features support dynamic reconfiguration (DR) operations.



## Supported Device Standards

Solaris I/O multipathing features is based on open standards for communicating with devices and device management, ensuring interoperability with other standards-based devices and software. The following standards are supported by Solaris I/O multipathing features:

- ANSI Standard: Information Technology – SNIA Multipath Management API Specification (ANSI INCITS 412-2006)
- T10 standards, including SCSI-3, SAM, FCP, SPC, and SBC
- T11.3 FC standards, including FC-PH, FC-AL, FC-LS, and FC-GS
- T11.5 storage management standards, including FC-HBA
- IETF standards, including RFC 2625
- Serial Attached SCSI-2 (SAS2)



# Fibre Channel Multipathing Configuration Overview

---

This chapter provides information about configuring Fibre Channel (FC) devices and Solaris I/O multipathing features. The following topics are discussed:

- “FC Device Configuration Task Overview” on page 19
- “Solaris I/O Multipathing Configuration Considerations” on page 20

## FC Device Configuration Task Overview

This section provides a high level overview of the tasks for configuring Solaris I/O multipathing features for FC devices.

The following table lists the configuration tasks, description about each task and the reference information where you can find the procedures to perform each task.

**TABLE 2-1** Tasks for Configuring Multipathed FC Devices

Configuration Task	Task Description	Reference Information
Enable multipathing features	Enabling multipathing is provided as follows: <ul style="list-style-type: none"> <li>▪ By default, for FC devices on x86/x64 platforms</li> <li>▪ By manual configuration on SPARC platforms</li> <li>▪ By manual configuration for SAS devices</li> </ul>	The steps for enabling multipathing are outlined in <a href="#">Chapter 3, “Configuring Solaris I/O Multipathing Features.”</a>
Configure FC devices	In the Oracle Solaris OS, FCAL, fabric, and point-to-point connected devices are made available to the host.	<a href="#">Chapter 5, “Configuring SAN Devices”</a>
Configure virtual FC ports	You can configure N Port ID Virtualization (NPIV) ports, which is especially useful for virtualization environments.	<a href="#">Chapter 6, “Configuring Virtual Fibre Channel Ports”</a>

TABLE 2-1 Tasks for Configuring Multipathed FC Devices (Continued)

Configuration Task	Task Description	Reference Information
Set up FC boot device	Solaris I/O multipathing features enable a Solaris server to boot from a fabric device.	“SAN Device Considerations” on page 49
Configure IPFC SAN devices	You can configure host recognition of IPFC devices and implementation of IP over FC in a SAN. Configuration of IPFC depends on the instance of the Fabric Port (FP) driver for the FC adapter ports.	Chapter 9, “Configuring IPFC SAN Devices”
Configure SAS devices	In the Oracle Solaris OS, multipathing of SAS devices is supported.	

## Solaris I/O Multipathing Configuration Considerations

Before you configure Solaris I/O multipathing features, consider the following:

- Configure ports and zones according to the vendor-specific documentation for storage and switches.
- LUN masking enables specific LUNs to be seen by specific hosts. See your vendor-specific storage documentation that describes masking.
- Power management needs to be disabled for hosts and devices on a SAN. For more information about power management, see [poweradm\(1M\)](#).
- The STMS boot utility is included with Solaris I/O multipathing features that manage the SAN booting process. Issuing the `stmsboot` command allows the automatic update of `/etc/vfstab` and the dump configuration to reflect device name changes when enabling or disabling the multipathing software. Note that the software is disabled by default for SPARC devices and enabled by default with x86 devices.
- Fabric-connected devices are configured and made available to the host automatically during installation and boot time.

---

**Note** – If you are performing an upgrade and want to make any FC devices unavailable after upgrade, those devices have to be manually unconfigured using `cfgadm -c unconfigure` commands. However, to make those devices permanently unavailable to the system, you may want to consider using switch zoning or LUN masking. The changes made by `cfgadm -c unconfigure` do not persist after a reboot unless manual configuration for FC devices has been enabled. To find out how to disable the FC devices discovery during boot or install, please refer to [Appendix A, “Manual Configuration for Fabric-Connected Devices.”](#)

---

# Configuring Solaris I/O Multipathing Features

---

This chapter explains how to configure Solaris I/O multipathing features for FC devices in the Oracle Solaris OS. It also provides the considerations while enabling or disabling the multipathing feature on SPARC based systems, x86 based systems, on a per-port basis, and third-party storage devices.

The following topics are discussed:

- “Configuring Multipathing I/O Features” on page 21
- “Multipathing Considerations” on page 21
- “Enabling and Disabling Multipathing” on page 22
- “Enabling or Disabling Multipathing on a Per-Port Basis” on page 25
- “Configuring Third-Party Storage Devices” on page 27
- “Configuring Automatic Failback” on page 31

## Configuring Multipathing I/O Features

The multipathing features for FC devices can be configured to control all supported FC HBAs. Multipathing is disabled by default for FC devices on SPARC based systems, but is enabled by default on x86 based systems. Configuration of the multipathing features depends on how you intend to use your system.

---

**Note** – The multipathing feature is not available for parallel SCSI devices, but is available for FC, SAS, and iSCSI devices. Multipathing is also supported for tape drives and libraries.

---

## Multipathing Considerations

Before you change multipathing configuration note the following considerations. Then follow the instructions for your machine architecture (SPARC or x86) described in the subsequent

sections. Some devices need to be properly configured to work with the multipathing software. Refer to your storage array documentation for details on the device specific configuration for your device.

- Device specific and device name change considerations

In the `/dev` and `/devices` trees, multipathed devices receive new names that indicate that they are under multipath control. A device therefore will have a different name from its original name when it is under multipath control.

Device name with multipath disabled:

```
/dev/dsk/c1t1d0s0
```

Device name with multipath enabled:

```
/dev/dsk/c0t60003BA27D5170003E5D2A7A0007F3D2d0s0
```

Therefore, applications that use device names directly must be configured to use the new names whenever you change a multipath configuration from disabled to enabled or vice-versa.

- Updates to `/etc/vfstab` entries and dump configuration

The system's `/etc/vfstab` file and the dump configuration also contain references to device names. On both SPARC and x86-based systems the `stmsboot` command described in the following sections automatically updates the `/etc/vfstab` file dump configuration with the new device names. If you have application-dependent file systems which are not listed in the file `/etc/vfstab`, you can use the `stmsboot` command to determine the mapping between the old and new device paths.



**Caution** – If you have run the `devfsadm -C` or performed a reconfiguration boot, the old device paths will not exist and the `stmsboot -L` command will fail to provide this information.

---

## Enabling and Disabling Multipathing

You can use the `stmsboot` command to enable or disable multipathing for Fibre Channel (FC) and SAS devices. The `stmsboot` command updates the `/etc/vfstab` file and dump configuration to reflect device name changes during the next reboot. You do not need to manually edit the `fp.conf` or `mpt.conf` files.

The following considerations apply to the `stmsboot -e`, `-d`, and `-u` options:

- You should reboot immediately after running the `stmsboot` command.
- Ensure that `eeprom boot device` is set to boot from your current boot device. This is required since the `stmsboot` command reboots the machine to complete the operation.

- The `stmsboot` command saves a copy of the original `/kernel/drv/fp.conf`, `/kernel/drv/mpt.conf` and `/etc/vfstab` files before modifying them. See [Appendix C, “Troubleshooting Multipathed Device Problems,”](#) if you encounter unexpected problems while using the `stmsboot` command.

---

**Note** – In previous Oracle Solaris releases, the `stmsboot` command was used to enable or disable multipathing on the boot device only for SPARC based hosts only. In current Oracle Solaris releases, the command is used to enable or disable multipathing on Serial-Attached SCSI devices as well as FC devices.

---

## ▼ How to Enable Multipathing

Perform the following steps to enable multipathing on all multipath-capable devices on SPARC or x86 based systems. If you want to enable multipathing only on specific FC or SAS HBA ports, refer to [“Enabling or Disabling Multipathing on a Per-Port Basis”](#) on page 25.

The multipathing software automatically recognizes Oracle's supported devices. If you want to enable multipathing on third-party devices, copy the `/kernel/drv/scsi_vhci.conf` file to `/etc/driver/drv` and add entries as described by *Device Name Change Considerations* in [“Multipathing Considerations”](#) on page 21.

- 1 **Become an administrator.**
- 2 **Confirm that the multipathing software package is installed.**

```
# pkg info system/storage/multipath-utilities
Name: system/storage/multipath-utilities
Summary: Solaris Multipathing CLI
Description: Path Administration Utility for a Solaris Multipathing device
Category: Applications/System Utilities
State: Installed
Publisher: solaris
Version: 0.5.11
Build Release: 5.11
Branch: 0.175.0.0.0.0.0
Packaging Date: Tue Sep 27 01:40:01 2011
Size: 77.29 kB
FMRI: pkg://solaris/system/storage/multipath-utilities@
0.5.11,5.11-0.175.0.0.0.0.0:20110927T014001Z
```

If not, install it.

```
# pkg install system/storage/multipath-utilities
```

- 3 **Enable device multipathing.**

```
# stmsboot -e
WARNING: stmsboot operates on each supported multipath-capable controller
detected in a host. In your system, these controllers are
```

```

/devices/pci@780/pci@0/pci@8/SUNW,qlc@0/fp@0,0
/devices/pci@780/pci@0/pci@8/SUNW,qlc@0,1/fp@0,0
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@1
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@1
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2
/devices/pci@7c0/pci@0/pci@9/LSILogic,sas@0
/devices/pci@7c0/pci@0/pci@9/LSILogic,sas@0

```

If you do NOT wish to operate on these controllers, please quit `stmsboot` and re-invoke with `-D { fp | mpt }` to specify which controllers you wish to modify your multipathing configuration for.

```

Do you wish to continue? [y/n] (default: y) y
Checking mpzio status for driver fp
Checking mpzio status for driver mpt
WARNING: This operation will require a reboot.
Do you want to continue ? [y/n] (default: y) y
The changes will come into effect after rebooting the system.
Reboot the system now ? [y/n] (default: y) y

```

---

**Note** – During the reboot, `/etc/vfstab` and the dump configuration are updated to reflect the device name changes.

---

- 4 (Optional) After the reboot, if necessary, configure your applications to use new device names as described by *Device Name Change Considerations* in [“Multipathing Considerations” on page 21](#).

## ▼ How to Disable Multipathing

Perform the following steps to disable multipathing on all multipath-capable devices on SPARC or x86 based systems. If you want to disable multipathing only on specific FC or SAS HBA ports, refer to [“Enabling or Disabling Multipathing on a Per-Port Basis” on page 25](#).

### 1 Disable device multipathing.

```
# stmsboot -d
```

WARNING: `stmsboot` operates on each supported multipath-capable controller detected in a host. In your system, these controllers are

```

/devices/pci@780/pci@0/pci@8/SUNW,qlc@0/fp@0,0
/devices/pci@780/pci@0/pci@8/SUNW,qlc@0,1/fp@0,0
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@1
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@1
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2
/devices/pci@7c0/pci@0/pci@9/LSILogic,sas@0
/devices/pci@7c0/pci@0/pci@9/LSILogic,sas@0

```

If you do NOT wish to operate on these controllers, please quit `stmsboot` and re-invoke with `-D { fp | mpt }` to specify which controllers you wish to modify your multipathing configuration for.



```

Do you wish to continue? [y/n] (default: y) y
Checking mpzio status for driver fp
Checking mpzio status for driver mpt
WARNING: This operation will require a reboot.
Do you want to continue ? [y/n] (default: y) y
The changes will come into effect after rebooting the system.
Reboot the system now ? [y/n] (default: y) y

```

---

**Note** – During the reboot, `/etc/vfstab` and the dump configuration are updated to reflect the device name changes.

---

- 2 (Optional) After the reboot, if necessary, configure your applications to use new device names as described by *Device Name Change Considerations* in [“Multipathing Considerations” on page 21](#).

## Enabling or Disabling Multipathing on a Per-Port Basis

Multipathing can be enabled or disabled on specific Fibre Channel Host Bus Adapter (HBA) controller ports. If you enable multipathing on a specific HBA port controller port, all supported devices connected to that controller port will be enabled for multipath operation.

The following procedure applies to both SPARC and x86 based machines.

### Port Configuration Considerations

Before you start configuring the software by port, consider the following:

- FC global and per-port multipath settings are specified in the file `/kernel/drv/fp.conf`. Per-port multipath settings have priority over the global setting. This means that if global multipathing is enabled but a specific port has been disabled for multipathing, the port will not be available in the multipathing configuration. Conversely, even if global multipathing has been disabled, specific ports may be enabled for multipathing if they are listed in the appropriate `driver.conf` file.
- Load balancing is controlled by the global `load-balance` property in `/kernel/drv/scsi_vhci.conf` file and is not controlled on a per-port basis.
- If a device has more than one path to the host, all paths to the device must be configured with multipathing enabled or disabled.
- Configuring multipathing by port enables the multipathing software to coexist with other multipathing solutions like Symantec (VERITAS) Dynamic Multipathing (DMP), or EMC PowerPath. However, devices and paths should not be shared between the multipathing software and other multipathing solutions.

## ▼ How to Configure Multipathing by Port

Depending on how many ports you want the multipathing software to control, you can enable or disable multipathing globally or for specified ports.

### 1 Become an administrator.

### 2 Determine the HBA controller ports that you want the multipathing software to control.

For example, to select the desired device, perform an `ls -l` command on `/dev/cfg` directory. The following example shows the `ls -l` command output.

```
# ls -l
lrwxrwxrwx 1 root  root  50 Jan 29 21:33 c0 ->
    ../../devices/pci@7c0/pci@0/pci@1/pci@0/ide@8:scsi
lrwxrwxrwx 1 root  root  61 Jan 29 21:33 c1 ->
    ../../devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@1:scsi
lrwxrwxrwx 1 root  root  61 Jan 29 21:33 c2 ->
    ../../devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2:scsi
lrwxrwxrwx 1 root  root  53 Jan 29 21:33 c3 ->
    ../../devices/pci@7c0/pci@0/pci@9/LSILogic,sas@0:scsi
lrwxrwxrwx 1 root  root  54 Apr 16 20:28 c5 ->
    ../../devices/pci@780/pci@0/pci@8/SUNW,qlc@0/fp@0,0:fc
lrwxrwxrwx 1 root  root  56 Apr 16 20:28 c6 ->
    ../../devices/pci@780/pci@0/pci@8/SUNW,qlc@0,1/fp@0,0:fc
```

---

**Note** – Controllers c5 and c6 are ports A and B on a dual-port FC HBA. Controllers c1 and c3 are single port SAS HBA ports. Controller c2 is the internal SAS controller in Oracle's Sun Fire T2000 server.

---

Once you have determined the port, or ports, you want to explicitly enable or disable multipathing for, go to the next step.

### 3 Copy the `/kernel/drv/fp.conf` file to the `/etc/driver/drv/fp.conf` file.

### 4 Select one of the following to enable or disable specific FC HBA ports:

- Add the following line for each FC HBA port that is to be enabled in the `/etc/driver/drv/fp.conf`:

```
name="fp" parent="parent name" port=port-number mpxio-disable="no";
```

where *parent name* is the port device name, and *port-number* is the FC HBA port number.

For example, the following entries disable multipathing on all FC HBA controller ports except for the two specified ports:

```
mpxio-disable="yes";
name="fp" parent="/pci@06,2000/SUNW,qlc@2" port=0 mpxio-disable="no";
name="fp" parent="/pci@13,2000/pci@2/SUNW,qlc@5" port=0 mpxio-disable="no";
```

- Add the following line for each FC HBA port that is to be disabled:

```
name="fp" parent="parent name" port=port-number mpxio-disable="yes";
```

For example:

```
name="fp" parent="/pci@6,2000/SUNW,qlc@2" port=0 mpzio-disable="yes";
```

##### 5 Start the reboot and configuration process.

```
# stmsboot -u
```

You are prompted to reboot. During the reboot, the `/etc/vfstab` file and your dump device configuration are updated to reflect any device name changes.

##### 6 (Optional) After the reboot, if necessary, configure your applications to use new device names as described in [“Multipathing Considerations”](#) on page 21.

## Configuring Third-Party Storage Devices

---

**Note** – Before configuring any third-party device, ensure that they are supported. Refer to your third-party user documentation, or third-party vendor for information on proper vendor and product IDs, modes and various settings required for the device to work with multipathing software.

---

### Third-Party Device Configuration Considerations

Before you configure third-party devices for multipathing, be aware of the following:

- The device must support the `REPORT_LUNS` SCSI command, and SCSI-3 `INQUIRY` command VPD Device Identification Page (0x83).
- You will need the vendor ID (VID) and product ID (PID) of the device. You can obtain them by using the `format` command followed by the `inquiry` option on your system. For more information, see [format\(1M\)](#).

When multipathing is enabled, the multipath access still depends on a device specific `scsi_vhci` failover implementation accepting the device. The default way this occurs, automatically, is for the `scsi_vhci` code to call a *probe* function in each failover implementation, looking for the first *probe* result that indicates the device is supported.

A *probe* implementation determines support based on some combination of `scsi_inquiry(9S)` data. A device with `INQUIRY` data indicating T10 Target-Port-Group-Support (TPGS) compliance will use the standards-based TPGS failover implementation. For noncompliant devices, a failover implementation's *probe* will typically determine support based on VID/PID match against a private compiled-in table.

To override the *probe* process, the `scsi_vhci.conf` file supports a `scsi-vhci-failover-override` property. The value of `scsi-vhci-failover-override` can be used to establish support for a device not currently accepted by *probe*, override *probe* support, or disable multipath support for a device.

## Configuring Third-Party Storage Devices: New Devices

Multipathing can be configured on third-party symmetric storage devices. A symmetric storage device is defined as a storage device in which all paths to the storage device are active and I/O commands can be issued through any path.

### ▼ How to Configure Third-Party Devices

Perform the following steps to configure third-party devices if your system already has multipathing enabled. If your system has multipathing disabled, you can configure third-party devices while enabling multipathing as described earlier in this chapter.

- 1 **Become an administrator.**
- 2 **Copy the `/kernel/drv/scsi_vhci.conf` file to the `/etc/driver/drv/scsi_vhci.conf` file.**
- 3 **Edit the `/etc/driver/drv/scsi_vhci.conf` file to add the vendor ID and product ID entries.**

The vendor ID and product ID are the vendor and product identification strings that the device returns in SCSI INQUIRY data. The vendor ID must be eight characters long. You must specify all eight characters even if the trailing characters are spaces.

The product ID can be up to 16 characters long.

```
scsi-vhci-failover-override =
"VendorID1ProductID1", "f_sym",
"VendorID2ProductID2", "f_sym",
...
"VendorIDnProductIDn", "f_sym";
```

Note that the entries are separated by the `;` character and the last vendor/product entry is terminated by the `;` character.

For example, to add a device from a “ACME” vendor with a product ID of “MSU”, and to add a device from vendor “XYZ” vendor with a product ID of “ABC”, you would add the following lines to the `/etc/driver/drv/scsi_vhci.conf` file:

To add a device from a “XYZ” vendor with a product ID of “ABC”, you would add:

```
scsi-vhci-failover-override =
"ACME    MSU", "f_sym",
"XYZ     ABC", "f_sym";
```

- 4 **Save and exit the `/etc/driver/drv/scsi_vhci.conf` file.**

**5 Start the reboot and configuration process.**

```
# stmsboot -u
```

You are prompted to reboot. During the reboot, the `/etc/vfstab` file and the dump configuration are updated to reflect the device name changes.

**6 If necessary, perform device name updates as described in “Enabling and Disabling Multipathing” on page 22.**

## Configuring Third-Party Storage Devices: Disabling Devices

Multipathing can be disabled for all devices of a certain vendor ID/product ID combination from Inquiry data. This exclusion is specified in the `scsi_vhci.conf` file.

### ▼ How to Disable Third-Party Devices

**1 Become an administrator.****2 Copy the `/kernel/drv/scsi_vhci.conf` file to the `/etc/driver/drv/scsi_vhci.conf` file.****3 Edit the `/etc/driver/drv/scsi_vhci.conf` file to add the vendor ID and product ID entries.**

The vendor ID and product ID are the vendor and product identification strings that the device returns in SCSI INQUIRY data. The vendor ID must be eight characters long. You must specify all eight characters even if the trailing characters are spaces. The product ID can be up to 16 characters long.

```
scsi-vhci-failover-override =
"VendorID1ProductID1", "NONE",
"VendorID2ProductID2", "NONE",
...
"VendorIDnProductIDn", "NONE";
```

The entries in the preceding example are separated by the `,` character and the last vendor/product entry is terminated by the `;` character. For example, to add a device from vendor “ACME” with a product ID of “MSU,” and a vendor device from vendor “XYZ” with product ID “ABC,” you would add the following lines to the file `/etc/driver/drv/scsi_vhci.conf`:

```
scsi-vhci-failover-override =
"ACME    MSU", "NONE",
"XYZ     ABC", "NONE";
```

**4 Save and exit the `scsi_vhci.conf` file.**

**5 Start the reboot and configuration process.**

```
# stmsboot -u
```

You are prompted to reboot. During the reboot, the `/etc/vfstab` file and the dump configuration are updated to reflect the device name changes.

**6 If necessary, perform the device name updates as described in “Enabling and Disabling Multipathing” on page 22.**

## Displaying Device Name Changes

You can display the mapping between non-multipathed and multipathed device names, after changes are made to the multipath configuration by invoking the `stmsboot` command with the `-e`, `-d`, or `-u` option. Both non-multipathed and the multipathed device names must exist in order to show the mapping.

Display the mapping of devices on all controllers. For example:

```
# stmsboot -L
non-STMS device name          STMS device name
-----
/dev/rdisk/c2t8d0             /dev/rdisk/c10t500000E01046DEE0d0
/dev/rdisk/c2t0d0             /dev/rdisk/c10t500000E01046B070d0
/dev/rdisk/c2t3d0             /dev/rdisk/c10t20000020372A40AFd0
/dev/rdisk/c2t12d0            /dev/rdisk/c10t500000E01046DEF0d0
/dev/rdisk/c2t11d0            /dev/rdisk/c10t500000E01046E390d0
/dev/rdisk/c3t8d0             /dev/rdisk/c10t500000E01046DEE0d0
/dev/rdisk/c3t0d0             /dev/rdisk/c10t500000E01046B070d0
/dev/rdisk/c3t3d0             /dev/rdisk/c10t20000020372A40AFd0
/dev/rdisk/c3t12d0            /dev/rdisk/c10t500000E01046DEF0d0
/dev/rdisk/c3t11d0            /dev/rdisk/c10t500000E01046E390d0
```

The `-l` option displays the mapping of devices on only the specified controller. The following example displays the mapping of controller 3.

```
# stmsboot -l3
non-STMS device name          STMS device name
-----
/dev/rdisk/c3t8d0             /dev/rdisk/c10t500000E01046DEE0d0
/dev/rdisk/c3t0d0             /dev/rdisk/c10t500000E01046B070d0
/dev/rdisk/c3t3d0             /dev/rdisk/c10t20000020372A40AFd0
/dev/rdisk/c3t12d0            /dev/rdisk/c10t500000E01046DEF0d0
/dev/rdisk/c3t11d0            /dev/rdisk/c10t500000E01046E390d0
```

# Configuring Automatic Failback

Some storage devices have controllers configured as PRIMARY and SECONDARY as part of the array configuration. The secondary paths may operate at a lower performance level than the primary paths. The multipathing software uses the primary path to talk to the storage device and keeps the secondary path on standby. In the event of a primary path failure, the multipathing software automatically directs all I/O traffic over the secondary path, with the primary path taken offline. This process is called a “failover” operation. When the failure associated with the primary path has been repaired, the multipathing software automatically directs all I/O traffic over the primary path and keeps the secondary path standby as before. This process is called a *failback* operation.

You can disable the automatic failback operation so the multipathing software does not automatically failback to the primary path. Later, after the failure associated with the primary path has been repaired, you can do a manual failback operation using the `luxadm` command. For more information, see [luxadm\(1M\)](#).

## ▼ How to Configure Automatic Failback

- 1 **Become an administrator.**
- 2 **Copy the `/kernel/drv/scsi_vhci.conf` file to the `/etc/driver/drv/scsi_vhci.conf` file.**
- 3 **Edit the `/etc/driver/drv/scsi_vhci.conf` file.**
- 4 **Disable automatic failback capability by changing the `auto-failback` entry to:**  

```
auto-failback="disable";
```

Enable automatic failback capability (if you disabled it before), by changing the `auto-failback` entry to:  

```
auto-failback="enable";
```
- 5 **Save and exit the file.**
- 6 **Reboot the system.**  

```
# shutdown -g0 -y -i6
```





# Administering Multipathing Devices (mpathadm)

---

You determine and configure Solaris OS multipathing support by using `mpathadm` commands. The `mpathadm` command enables multipathing administration through the ANSI standard Multipath Management API. The terms used in this chapter to denote a path, initiator port, target port, and LUN are consistent with the T10 specification.

## Administering Multipathing Devices

Perform the following tasks to administer any multipathing devices.

### ▼ How to Display Multipathing Support Information

Multipathing support and property information is identified with the Multipath Management API plug-in library name.

1 **Become an administrator.**

2 **Identify the multipathing support on your system.**

```
# mpathadm list mpath-support
mpath-support: libmpscsi_vhci.so
```

3 **Display the multipathing support properties for a specified mpath support name.**

```
# mpathadm show mpath-support libmpscsi_vhci.so
mpath-support: libmpscsi_vhci.so
  Vendor: Sun Microsystems
  Driver Name: scsi_vhci
  Default Load Balance: round-robin
  Supported Load Balance Types:
    round-robin
    logical-block
  Allows To Activate Target Port Group Access: yes
  Allows Path Override: no
```

```

Supported Auto Failback Config: yes
Auto Failback: on
Failback Polling Rate (current/max): 0/0
Supported Auto Probing Config: 0
Auto Probing: NA
Probing Polling Rate (current/max): NA/NA
Supported Devices:
  Vendor: SUN
  Product: T300
  Revision:
  Supported Load Balance Types:
    round-robin

  Vendor: SUN
  Product: T4
  Revision:
  Supported Load Balance Types:
    round-robin
.
.
.

```

The command output also shows a list of device products that is supported by the multipathing support software. The multipathing support `libmpscsi_vhci` . so supports T10 target port group compliant devices by default.

## ▼ How to Display Properties of a Specific Initiator Port

The following steps describe how to display the properties of an initiator port.

### 1 Identify the initiator port.

```

# mpathadm list initiator-port
Initiator Port: iqn.1986-03.com.sun:01:ffffffffffff.4e94f9bd,4000002a00ff
Initiator Port: 210100e08ba41feb
Initiator Port: 210000e08b841feb

```

### 2 Display the specific initiator port's properties.

```

# mpathadm show initiator-port 2000000173018713
Initiator Port: 210100e08ba41feb
  Transport Type: Fibre Channel
  OS Device File: /devices/pci@1,0/pci1022,7450@1/pci1077,141@2,1/fp@0,0

```

## ▼ How to Display Specific LUN Information

Display the list of LUNs, along with the properties for each LUN by using the `mpathadm` command. The displayed list of LUNs contain names that in turn can be used to display the properties of a particular LUN.

## 1 Display a list of multipathed LUNs.

```
# mpathadm list lu
/dev/rdisk/c0t600144F08069703400004E828EE10004d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdisk/c0t600144F08069703400004E8183DF0002d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdisk/c0t600A0B800026D63A0000A4994E2342D4d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdisk/c0t600A0B800029065C00007CF54E234013d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdisk/c0t600A0B800029065C00007CF44E233FCFd0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdisk/c0t600A0B800026D63A0000A4974E23424Ed0s2
    Total Path Count: 4
    Operational Path Count: 4
.
.
.
```

## 2 Display configuration information about a specific LUN.

```
# mpathadm show lu /dev/rdisk/c4t60003BA27D2120004204AC2B000DAB00d0s2
mpath-support: libmpscsi_vhci.so
Vendor: SUN
Product: Sun Storage NAS
Revision: 1.0
Name Type: unknown type
Name: 600144f08069703400004e828ee10004
Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:
    Initiator Port Name: 210100e08ba41feb
    Target Port Name: 2100001b329b6c3f
    Override Path: NA
    Path State: OK
    Disabled: no

    Initiator Port Name: 210100e08ba41feb
    Target Port Name: 2101001b32bb6c3f
    Override Path: NA
    Path State: OK
    Disabled: no

    Initiator Port Name: 210100e08ba41feb
    Target Port Name: 2100001b329b793c
    Override Path: NA
```

```

Path State: OK
Disabled: no
.
.
.

Target Port Groups:
ID: 0
Explicit Failover: no
Access State: active optimized
Target Ports:
    Name: 2100001b329b6c3f
    Relative ID: 1

    Name: 2101001b32bb6c3f
    Relative ID: 2

ID: 1
Explicit Failover: no
Access State: standby
Target Ports:
    Name: 2100001b329b793c
    Relative ID: 257

    Name: 2101001b32bb793c
    Relative ID: 256

```

## ▼ How to Display All LUNs Associated With a Specific Target Port

Use the steps below to display paths through a target port.

### 1 Display a list of LUNs.

```

# mpathadm list lu
/dev/rdisk/c0t600144F08069703400004E828EE10004d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdisk/c0t600144F08069703400004E8183DF0002d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdisk/c0t600A0B800026D63A0000A4994E2342D4d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdisk/c0t600A0B800029065C00007CF54E234013d0s2
    Total Path Count: 4
    Operational Path Count: 4
.
.
.

```

### 2 Display specific LUN information to determine the target ports.

```

# mpathadm show lu /dev/rdisk/c0t600A0B800029065C00007CF54E234013d0s2
Logical Unit: /dev/rdisk/c0t600A0B800029065C00007CF54E234013d0s2
mpath-support: libmpscsi_vhci.so

```

```

Vendor: SUN
Product: CSM200_R
Revision: 0660
Name Type: unknown type
Name: 600a0b800029065c00007cf54e234013
Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:
  Initiator Port Name: 210000e08b841feb
  Target Port Name: 200800a0b826d63b
  Override Path: NA
  Path State: OK
  Disabled: no

  Initiator Port Name: 210000e08b841feb
  Target Port Name: 200900a0b826d63b
  Override Path: NA
  Path State: OK
  Disabled: no

```

```

.
.
.

```

```

Target Port Groups:
  ID: 4
  Explicit Failover: yes
  Access State: standby
  Target Ports:
    Name: 200800a0b826d63b
    Relative ID: 0

    Name: 200800a0b826d63c
    Relative ID: 0

  ID: 14
  Explicit Failover: yes
  Access State: active
  Target Ports:
    Name: 200900a0b826d63b
    Relative ID: 0

    Name: 200900a0b826d63c
    Relative ID: 0

```

### 3 Display the specific target port information.

```

# mpathadm list lu -t 20030003ba27d212
mpath-support: libmpscsi_vhci.so
/dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
  Total Path Count: 4
  Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
/dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
  Total Path Count: 4
  Operational Path Count: 4

```

```

mpath-support: libmpscsi_vhci.so
/dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
    Total Path Count: 4
    Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
/dev/rdsk/c0t600A0B800029065C00007CF44E233FCFd0s2
    Total Path Count: 4
    Operational Path Count: 4
.
.
.

```

## ▼ How to Display a LUN With a Specific Name

Display detailed information on multipathed LUNs, including path and target port group information. Note that the name property in the information represents the identifier for this LUN, derived from the hardware, and used by this system. If the name is derived from SCSI Inquiry Vital Product Data (VPD) page 83h, the name type property represents an associated identifier type defined by the SCSI standards.

### 1 Display a list of multipathed LUNs.

```

# mpathadm list lu
/dev/rdsk/c0t600144F08069703400004E828EE10004d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdsk/c0t600144F08069703400004E8183DF0002d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF44E233FCFd0s2
    Total Path Count: 4
    Operational Path Count: 4
.
.
.

```

### 2 Display a selected LUN's configuration information.

```

# mpathadm show lu /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
Logical Unit: /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
mpath-support: libmpscsi_vhci.so
Vendor: SUN
Product: CSM200_R
Revision: 0660
Name Type: unknown type
Name: 600a0b800026d63a0000a4994e2342d4

```

```

Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

```

Paths:

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

```

Target Port Groups:

```

ID: 3
Explicit Failover: yes
Access State: active
Target Ports:
    Name: 200800a0b826d63b
    Relative ID: 0

    Name: 200800a0b826d63c
    Relative ID: 0

```

```

ID: 13
Explicit Failover: yes
Access State: standby
Target Ports:
    Name: 200900a0b826d63b
    Relative ID: 0

    Name: 200900a0b826d63c
    Relative ID: 0

```

### 3 Display the selected LUN information.

```

# mpathadm list lu -n 600a0b800026d63a0000a4994e2342d4
mpath-support: libmpscsi_vhci.so
/dev/rdisk/c0t600A0B800026D63A0000A4994E2342D4d0s2
Total Path Count: 4
Operational Path Count: 4

```

## ▼ How to Configure Automatic Failback for Specific Multipathing Support

Symmetrical devices allow for automatic failback to a possible optimal path. If a failover occurs on the initial path, the standby path becomes the new online path. Usually the standby path is a suboptimal path. When auto-failback is enabled, the initial path comes back online, failover to the initial path automatically occurs.

1 **Become an administrator.**

2 **Display the supported multipath driver information.**

```
# mpathadm list mpath-support
mpath-support: libmpscsi_vhci.so
```

3 **Enable automatic failback support for the supported multipath driver.**

```
# mpathadm modify mpath-support -a on libmpscsi_vhci.so
```

4 **Confirm the configuration change.**

```
# mpathadm show mpath-support libmpscsi_vhci.so
mpath-support: libmpscsi_vhci.so
  Vendor: Sun Microsystems
  Driver Name: scsi_vhci
  Default Load Balance: round-robin
  Supported Load Balance Types:
    round-robin
    logical-block
  Allows To Activate Target Port Group Access: yes
  Allows Path Override: no
  Supported Auto Failback Config: 1
  Auto Failback: on
  Failback Polling Rate (current/max): 0/0
  Supported Auto Probing Config: 0
  Auto Probing: NA
  Probing Polling Rate (current/max): NA/NA
  Supported Devices:
    Vendor: SUN
    Product: T300
    Revision:
    Supported Load Balance Types:
      round-robin

    Vendor: SUN
    Product: T4
    Revision:
    Supported Load Balance Types:
      round-robin

.
.
.
```



---

**Note** – The auto-failback display initiated by the `mpathadm modify` command setting is effective while the system is running. However, to keep the changed setting persistent, you must update the `/etc/driver/driv/scsi_vhci.conf` file. Refer to [“Configuring Automatic Failback” on page 31](#).

---

## ▼ How to Fail Over a LUN

This operation is applicable only to devices in the following two categories:

- Asymmetric devices with a proprietary failover mechanism recognized and supported by multipathing support
- Devices conforming to the T10 Standard Target Port Group Support `libmpscsi_vhci.so` and providing *explicit* mode asymmetric LUN access

### 1 Display a list of multipathed LUNs.

```
# mpathadm list lu
/dev/rdisk/c0t600144F08069703400004E828EE10004d0s2
  Total Path Count: 8
  Operational Path Count: 8
/dev/rdisk/c0t600144F08069703400004E8183DF0002d0s2
  Total Path Count: 8
  Operational Path Count: 8
/dev/rdisk/c0t600A0B800026D63A0000A4994E2342D4d0s2
  Total Path Count: 4
  Operational Path Count: 4
/dev/rdisk/c0t600A0B800029065C00007CF54E234013d0s2
  Total Path Count: 4
  Operational Path Count: 4
/dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
  Total Path Count: 4
  Operational Path Count: 4
.
.
.
```

### 2 Display the selected LUN's configuration information.

```
# mpathadm show lu /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
Logical Unit: /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
  mpath-support: libmpscsi_vhci.so
  Vendor: SUN
  Product: CSM200_R
  Revision: 0660
  Name Type: unknown type
  Name: 600a0b800026d63a0000a4984e234298
  Asymmetric: yes
  Current Load Balance: round-robin
  Logical Unit Group ID: NA
  Auto Failback: on
  Auto Probing: NA
```

## Paths:

```
Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63b
Override Path: NA
Path State: OK
Disabled: no
```

```
Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63b
Override Path: NA
Path State: OK
Disabled: no
```

```
Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63c
Override Path: NA
Path State: OK
Disabled: no
```

```
Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63c
Override Path: NA
Path State: OK
Disabled: no
```

## Target Port Groups:

```
ID: 5
Explicit Failover: yes
Access State: active
Target Ports:
  Name: 200800a0b826d63b
  Relative ID: 0
```

```
  Name: 200800a0b826d63c
  Relative ID: 0
```

```
ID: 15
Explicit Failover: yes
Access State: standby
Target Ports:
  Name: 200900a0b826d63b
  Relative ID: 0
```

```
  Name: 200900a0b826d63c
  Relative ID: 0
```

### 3 Manually force a LUN failover.

```
# mpathadm failover lu /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
```

If this operation is successful, the access state of the devices' target port groups changes as a result of the logical unit failover.

### 4 Confirm the access state change.

```
# mpathadm show lu /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
Logical Unit: /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
mpath-support: libmpscsi_vhci.so
```

Vendor: SUN  
Product: CSM200\_R  
Revision: 0660  
Name Type: unknown type  
Name: 600a0b800026d63a0000a4984e234298  
Asymmetric: yes  
Current Load Balance: round-robin  
Logical Unit Group ID: NA  
Auto Failback: on  
Auto Probing: NA

## Paths:

Initiator Port Name: 210000e08b841feb  
Target Port Name: 200800a0b826d63b  
Override Path: NA  
Path State: OK  
Disabled: no

Initiator Port Name: 210000e08b841feb  
Target Port Name: 200900a0b826d63b  
Override Path: NA  
Path State: OK  
Disabled: no

Initiator Port Name: 210000e08b841feb  
Target Port Name: 200800a0b826d63c  
Override Path: NA  
Path State: OK  
Disabled: no

Initiator Port Name: 210000e08b841feb  
Target Port Name: 200900a0b826d63c  
Override Path: NA  
Path State: OK  
Disabled: no

## Target Port Groups:

ID: 5  
Explicit Failover: yes  
Access State: standby  
Target Ports:  
Name: 200800a0b826d63b  
Relative ID: 0  
  
Name: 200800a0b826d63c  
Relative ID: 0

ID: 15  
Explicit Failover: yes  
Access State: active  
Target Ports:  
Name: 200900a0b826d63b  
Relative ID: 0  
  
Name: 200900a0b826d63c  
Relative ID: 0

## ▼ How to Enable a LUN Path

If the path to a LUN is disabled, the `enable` command changes the path back to enabled. The full path must be specified using the initiator port name, target port name, and the LUN. To verify the change, run the `show` command for the logical unit.

### 1 Display a list of multipathed LUNs.

```
# mpathadm list lu
/dev/rdsk/c0t600144F08069703400004E828EE10004d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdsk/c0t600144F08069703400004E8183DF0002d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF44E233FCFd0s2
    Total Path Count: 4
    Operational Path Count: 4
.
.
.
```

### 2 Display the selected LUN's configuration information.

```
# mpathadm show lu
Logical Unit: /dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
mpath-support: libmpscsi_vhci.so
Vendor: SUN
Product: CSM200_R
Revision: 0660
Name Type: unknown type
Name: 600a0b800026d63a0000a4984e234298
Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:
    Initiator Port Name: 210000e08b841feb
    Target Port Name: 200800a0b826d63b
    Override Path: NA
    Path State: OK
    Disabled: no

    Initiator Port Name: 210000e08b841feb
    Target Port Name: 200900a0b826d63b
    Override Path: NA
```

```

Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

```

```

Target Port Groups:
ID: 5
Explicit Failover: yes
Access State: standby
Target Ports:
    Name: 200800a0b826d63b
    Relative ID: 0

    Name: 200800a0b826d63c
    Relative ID: 0

ID: 15
Explicit Failover: yes
Access State: active
Target Ports:
    Name: 200900a0b826d63b
    Relative ID: 0

    Name: 200900a0b826d63c
    Relative ID: 0

```

### 3 Enable the selected LUN path.

```
# mpathadm enable path -i 210000e08b841feb -t 200900a0b826d63b \
-l /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
```

## ▼ How to Disable a LUN Path

This operation makes the path unavailable for the LUN, regardless of its operational state.

---

**Note** – The disabled state is not persistent across rebooting. If the path is operational before the next boot sequence, it is enabled by default. This operation is not allowed when the given path is the last operational path remaining.

---

### 1 Display a list of multipathed LUNs.

```
# mpathadm list lu
/dev/rdisk/c0t600144F08069703400004E828EE10004d0s2
Total Path Count: 8
```

```

        Operational Path Count: 8
/dev/rdsk/c0t600144F08069703400004E8183DF0002d0s2
        Total Path Count: 8
        Operational Path Count: 8
/dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
        Total Path Count: 4
        Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
        Total Path Count: 4
        Operational Path Count: 4
/dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
        Total Path Count: 4
        Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF44E233FCFd0s2
        Total Path Count: 4
        Operational Path Count: 4
.
.
.

```

## 2 Display the selected LUN's configuration information.

```
# mpathadm show lu /dev/rdsk/c0t600A0B800029065C00007CF34E233F89d0s2
```

```

Logical Unit: /dev/rdsk/c0t600A0B800029065C00007CF34E233F89d0s2
mpath-support: libmpscsi_vhci.so
Vendor: SUN
Product: CSM200_R
Revision: 0660
Name Type: unknown type
Name: 600a0b800029065c00007cf34e233f89
Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

```

### Paths:

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63c
Override Path: NA
Path State: OK

```

Disabled: no

Target Port Groups:

ID: 8

Explicit Failover: yes

Access State: standby

Target Ports:

Name: 200800a0b826d63b

Relative ID: 0

Name: 200800a0b826d63c

Relative ID: 0

ID: 18

Explicit Failover: yes

Access State: active

Target Ports:

Name: 200900a0b826d63b

Relative ID: 0

Name: 200900a0b826d63c

Relative ID: 0

**3 Select an initiator port and a target port name.**

**4 Disable the selected LUN path.**

```
# mpathadm disable path -i 210000e08b841feb -t 200900a0b826d63b \  
-l /dev/rdisk/c0t600A0B800029065C00007CF34E233F89d0s2
```





# Configuring SAN Devices

---

This chapter provides a high level overview of the steps used to configure SAN devices. In the Oracle Solaris OS, FCAL, fabric, and point-to-point connected devices are made available to the system automatically. This feature differs from previous versions of the SAN Foundation software running in previous Solaris releases. In those versions, manual configuration steps were required to achieve fabric-connected device availability on the host. To find out how to manually configure fabric-connected devices, refer to [Appendix A, “Manual Configuration for Fabric-Connected Devices.”](#)

The following topics are discussed:

- “SAN Device Considerations” on page 49
- “Adding SAN Devices” on page 50
- “Configuring Fabric Boot Devices on SPARC” on page 51

## SAN Device Considerations

Review the following considerations before configuring Solaris I/O multipathing features.

- Configure ports and zones according to the vendor-specific documentation for storage and switches.
- You no longer have to manually configure devices unless you choose manual device configuration.
- LUN masking enables specific LUNs to be seen by specific systems. See your vendor-specific storage documentation that describes masking.
- Connect arrays and other storage devices to the SAN with or without multipathing capability. Solaris multipathing is an associated application that is bundled with the product name.
- The STMS boot utility is included with Solaris I/O multipathing features that manage the SAN booting process. Issuing the `stmsboot` command allows the automatic update of `/etc/vfstab` and the dump configuration to reflect device name changes when enabling or

disabling the multipathing software. Note that the software is disabled by default with SPARC devices and enabled by default with x86 devices.

## Adding SAN Devices

Adding and removing SAN devices requires knowledge of the following commands:

Command	Description	For More Information
<code>cfgadm</code> and <code>cfgadm_fp</code>	Dynamically reconfigures devices and FC devices. These commands are used most frequently to configure storage devices on a SAN.	<a href="#">cfgadm(1M)</a> and <a href="#">cfgadm_fp(1M)</a>
<code>format</code>	Identifies devices that are connected to the system and provides the ability to label a disk.	<a href="#">format(1M)</a>
<code>luxadm</code>	Administers storage devices and FC_AL devices.	<a href="#">luxadm(1M)</a>

---

**Note** – If you use the `format` command when the multipathing features are enabled, you will see only one instance of a device identifier for each LUN. Without the multipathing features enabled, you will see one identifier for each path.

---

### ▼ How to Add a SAN Device

- 1 Create the LUNs desired on the SAN device.
- 2 If necessary, apply LUN masking for HBA control on the SAN device.
- 3 Connect the storage device to the system.
- 4 If necessary, create port-based or WWN zones on the switch on the SAN device.
- 5 Mount any existing file systems available on the storage device's LUNs or disk groups.

# Configuring Fabric Boot Devices on SPARC

Solaris I/O multipathing features allow a SPARC server that is running the Oracle Solaris OS to be booted from a fabric disk device.

## Fabric Boot Device Considerations

Fabric boot devices can be accessed through the Oracle Solaris installation methods just as internal boot devices have been accessed in the previous Solaris OS releases. Refer to the Oracle Solaris installation guide for details. Consider the following points while enabling a fabric boot device.

- Minimize interference to boot devices through the following actions:
  - Ensuring the boot device is not an overly subscribed target or LUN
  - Avoiding installation of applications and software on target or LUN
  - Reducing physical distance between host and fabric device, as well as the number of hops
- Remove the boot disk from volume manager control prior to beginning the fabric boot procedure.
- Ensure that the latest HBA fcode and drivers are loaded for the HBAs on the system.
- If multipathing is desired on the boot device, use the `stmsboot` command as described in [Chapter 3, “Configuring Solaris I/O Multipathing Features.”](#)



# Configuring Virtual Fibre Channel Ports

---

This chapter provides the steps used to configure *N* Port ID Virtualization (NPIV) ports, also known as virtual Fibre Channel ports.

- “What is NPIV?” on page 53
- “Limitations of NPIV” on page 53
- “Creating NPIV Ports” on page 54

## What is NPIV?

NPIV is a Fibre Channel facility that enables one Fibre Channel adapter to have many *N* Port IDs. Each *N* Port has a unique identity (port WWN and node WWN) on the SAN and can be used for zoning and LUN masking. Soft zoning, which you can use to group ports together by port WWN, is the preferred method of zoning.

## Limitations of NPIV

Here is the list of NPIV limitations when used to virtualize Fibre Channel ports:

- NPIV ports may not be used for booting.
- NPIV ports are best used in SANs with a relatively small number of ports, either virtual or physical. Also, some targets may not have enough resources to process the large number of ports that NPIV can create. This limitation exists because processing state change notifications (SCN) on the SAN takes significant time if there are a large number of ports on the SAN. You can work around this limitation on a large SAN by using zoning, which can limit the number of visible ports.
- MPxIO can be used with NPIV, although you should ensure that different paths are physically redundant.
- NPIV is supported only in a Fabric topology. It is not supported in an FC-AL or point-to-point topology.

- Not all hardware supports NPIV. Both switches and HBAs (although not targets) must support NPIV in a SAN. By specification, HBAs should support up to 255 virtual ports, although this is defined by the resources on the switch. Switches may have to be updated to the latest firmware levels for NPIV support.

## Creating NPIV Ports

You can configure NPIV for non-virtualized environments by using the `fcadm` command.

The `fcinfo` and `fcadm` status commands are available to determine the status of NPIV ports, whether the ports are created from `fcadm`. The commands also report the relationship between the physical port and the virtual ports hosted on that port.

Other Fibre Channel commands, such as `luxadm` and `cfgadm`, report NPIV information, although there is no distinction made between virtual and physical ports.

### ▼ How to Create an NPIV Port

**Before You Begin** Each virtual port must have a port and node name. The port name must be unique on the SAN. You can assign names manually or use the built-in random WWN generator. If you attempt to register duplicate names, most switches will report an error status on the newly registered WWN, and the switch will not register the new WWN.

For more information on acceptable name formats, refer to the T11 standard: Fibre Channel Framing and Signaling (FC-FS 2).

If you try to create an NPIV port on an HBA that does not support NPIV, an error will occur. If you try to create an NPIV port on an HBA that supports NPIV, but it is attached to a switch which does not support NPIV, the port will be created with an offline status. The status will be reported in the `fcinfo(IM)` output.

**1 Become an administrator.**

**2 Create an NPIV port.**

```
# fcdm create-npiv-port -p Virtual_Port_WWN -n Virtual_Node_WWN
PhysicalPort_port_WWN
```

Without the `-p` and `-n` options, a random WWN will be assigned for the virtual port and virtual node, respectively.

#### Example 6-1 Creating an NPIV Port

The following example creates an NPIV port on a physical HBA port with a WWN of 210000e08b170f1c, a virtual port WWN set to 2000000000000001, and a virtual node WWN set to 2100000000000001.

```
# fcadm create-npiv-port -p 2000000000000001 -n 2100000000000001 210000e08b170f1c
```

## ▼ How to Delete an NPIV Port

**Before You Begin** You can use the `fcinfo hba-port` command to display the current WWN values for the NPIV ports.

1 **Become an administrator.**

2 **Delete an NPIV port.**

```
# fcadm delete-npiv-port -p Virtual_Port_WWN -n Virtual_Node_WWN
PhysicalPort_port_WWN
```

### Example 6-2 Deleting an NPIV Port

The following example deletes an NPIV port on a physical HBA port with a WWN of 210000e08b170f1c.

```
# fcadm delete-npiv-port -p 2000000000000001 -n 2100000000000001 210000e08b170f1c
```

## ▼ How to Display NPIV Port Status

1 **Become an administrator.**

2 **Display the currently configured NPIV ports.**

```
# fcinfo hba-port
```

The NPIV Port List shows the lists of currently configured NPIV ports.

### Example 6-3 Displaying Status on NPIV Ports

The following example shows that HBA port 210000e08b84f7eb has one virtual port.

```
# fcinfo hba-port
HBA Port WWN: 210000e08b84f7eb
  Port Mode: Initiator
  Port ID: 10100
  OS Device Name: /dev/cfg/c7
  Manufacturer: QLogic Corp.
  Model: 375-3294-01
  Firmware Version: 04.04.00
  FCode/BIOS Version: BIOS: 1.4; fcode: 1.11; EFI: 1.0;
  Serial Number: 0402F00-0549112895
  Driver Name: qlc
  Driver Version: 20080430-0.00
```

```
Type: N-port
State: online
Supported Speeds: 1Gb 2Gb 4Gb
Current Speed: 4Gb
Node WWN: 200000e08b84f7eb
Max NPIV Ports: 63
NPIV port list:
  Virtual Port1:
    Node WWN: 1110000000000000
    Port WWN: 1210000000000000
```



# Configuring FCoE Ports

---

This chapter provides the steps used to configure FCoE ports hosted on normal Ethernet interfaces. This chapter does not apply to hardware FCoE ports on Converged Network Adapters (CNA).

The following topics are discussed:

- [“What is FCoE?” on page 57](#)
- [“Limitations of FCoE” on page 57](#)
- [“Configuring FCoE Ports” on page 58](#)

## What is FCoE?

FCoE is a new T11 standard that transports encapsulated Fibre Channel frames over Enhanced Ethernet. FCoE is designed to enable network convergence and cost-effective SAN expansion in data centers.

## Limitations of FCoE

The Solaris FCoE initiator is a pure software implementation that is designed to work with normal Ethernet controllers. However, you must comply with the following limitations of Solaris FCoE implementations:

- FCoE ports cannot be used for booting.
- FCoE ports cannot be configured in Oracle VM Server for SPARC or Oracle VM Server 3.0 for x86 guest operating systems.
- FCoE is supported in fabric and point-to-point topologies.

FCoE is not supported on all hardware. FCoE works with Ethernet controllers that support 802.3x PAUSE and jumbo frames, and have a GLDv3 driver.

## Configuring FCoE Ports

You can configure FCoE ports by using the `fcadm` command. The `fcinfo` and `fcadm` commands are available to determine the status of FCoE ports. The commands also report the relationship between the Ethernet interface and the FCoE port hosted on that interface.

Other Fibre Channel commands, such as `luxadm` and `cfgadm`, report FCoE information, although there is no distinction made between FCoE and native FC ports.

### ▼ How to Create an FCoE Port

**Before You Begin** Before you begin this procedure, you must perform the following tasks:

- Enable the 802.3x (also called PAUSE) setting on the Ethernet interface. This setting ensures a lossless Ethernet transport.
- Enable jumbo frames (greater than 2.5 Kbytes) on the Ethernet interface. A Fibre Channel data frame can be as large as 2136 bytes.

These settings can vary for different Ethernet hardware and drivers. In most cases, you must modify the `driver.conf` file of the Ethernet interface and then reboot. See the `driver.conf` file for your Ethernet interface for details on how to enable these features.

Each virtual port must have a port and node name. The port name must be unique on the SAN. You can assign names manually or use the built-in WWN generator. If you attempt to register duplicate names, the switch will report an error status on the newly registered WWN, and the switch will not register the new WWN. For more information on acceptable name formats, refer to the T11 standard: Fibre Channel Framing and Signaling (FC-FS 2).

If you try to create an FCoE port on a network interface that does not support FCoE, an error occurs and the FCoE port is not created.

- Enable the following services:

```
# svcadm enable svc:/system/fcoe_target:default
# svcadm enable svc:/system/stmf:default
```

#### 1 Become an administrator.

#### 2 Create an FCoE port.

```
# fcdm create-fcoe-port -i -p Port_WWN -n Node_WWN Ethernet_Interface
```

If the selected Ethernet interface does not support Multiple Unicast Address, you are prompted to explicitly enable promiscuous mode on that interface.

```
# fcdm create-fcoe-port -i -f Ethernet_Interface
```

For example:

```
# fcdm create-fcoe-port -i nxge0
```

## ▼ How to Delete an FCoE Port

**Before You Begin** You can use the `fcadm list-fcoe-ports` command to display the Ethernet interfaces hosting the FCoE ports.

1 **Become an administrator.**

2 **Delete an FCoE port.**

```
# fcdm delete-fcoe-port network_interface
```

For example:

```
# fcdm delete-fcoe-port nxge0
```

## ▼ How to Display FCoE Port Status

1 **Become an administrator.**

2 **Display the status of currently configured FCoE ports.**

```
# fcinfo hba-port -e
```

For example:

```
# fcinfo hba-port -e
HBA Port WWN: 200000144fc1f5c8
  Port Mode: Initiator
  Port ID: 9a0042
  OS Device Name: /dev/cfg/c6
  Manufacturer: Sun Microsystems, Inc.
  Model: FCoE Virtual FC HBA
  Firmware Version: N/A
  FCode/BIOS Version: N/A
  Serial Number: N/A
  Driver Name: SunFC FCoEI v20090422-1.00
  Driver Version: v20090422-1.00
  Type: N-port
  State: online
  Supported Speeds: 1Gb 10Gb
  Current Speed: 10 Gb
  Node WWN: 100000144fc1f5c8
```

This command retrieves a list of FC specific information for all FCoE ports in the system.

```
# fcdm list-fcoe-ports
```

For example:

```
# fcdm list-fcoe-ports
HBA Port WWN: 200000144fc1f5c8
  Port Type: Initiator
```

```

MAC Name: nxge0
MTU Size: 9194
Primary MAC Address: 00144fc1f5c8
Current MAC Address: 0efc009a0042
Promiscuous Mode: Off

```

This command retrieves a list of FCoE specific information for all FCoE ports in the system.

## ▼ How to Force a FCoE Port Reinitialization

Use the steps below when a FCoE port needs to be reinitialized. You might need to force a FCoE port reinitialization when new devices are added to an FC SAN or because of some misbehaving device on the SAN. In many cases, this operation can resolve problems in an FC-SAN.

When this command issued on the target port side, the target port is reset. When this command is issued from the host port side, the host port is reset.

When an FC switch is connected, other FC ports in the SAN get a remote state change notification (RSCN). Furthermore, other initiators will always rediscover the port after this operation, and the FC login session will be established or reused. This command is disruptive to I/Os, but I/Os continue. This command is not destructive because it does not cause any data loss.

- 1 **Become an administrator.**
- 2 **Force a link that is connected to a port to reinitialize.**

For example:

```
# fcadm force-link 200000144fc2d508
```

## Configuring FCoE Hardware Offload

The following four tunables can be used to reduce CPU utilization and improve performance on a system with FCoE ports. These tunables are supported by the Intel 10 Gb Ethernet devices and can set in the `ixgbe.conf` file.

Copy the `/kernel/drv/ixgbe.conf` to `/etc/driver/drv/ixgbe.conf` and modify the tunable values for your FCoE environment.

TABLE 7-1 FCoE Hardware Offload Tunable Parameters

Tunable Parameter	Description	Range	Default Value
<code>fcoe_txcrc_enable</code>	Controls whether the <code>ixgbe</code> driver offloads FC CRC transactions for transmitted FCoE packets.	0 – Disable FC CRC transaction offload 1 – Enable FC CRC transaction offload	1

TABLE 7-1 FCoE Hardware Offload Tunable Parameters *(Continued)*

Tunable Parameter	Description	Range	Default Value
fcoe_lso_enable	Controls whether the ixgbe driver offloads FC large send transactions for transmitted FCoE packets.	0 – Disable FC LSO transaction offload 1 – Enable FC LSO transaction offload	1
fcoe_rxrcrc_enable	Controls whether the ixgbe driver offloads RC CRC transactions for received FCoE packets.	0 – Disable FC RX CRC transaction offload 1 – Enable FC RX CRC transaction offload	1
fcoe_lro_enable	Controls whether the ixgbe driver offloads FC large receive transactions for received FCoE packets.	0 – Disable FC LRO transaction offload 1 – Enable FC LRO transaction offload	0



# Configuring SAS Domains

---

This chapter provides overviews of SAS Domain considerations, SAS device discovery, and SAS boot device configuration.

## SAS Multipathing Considerations

- SAS multipathing is supported in the Oracle Solaris release when using the bundled `mpt` driver.
- SAS expanders are not supported in the Oracle Solaris release.
- LUN masking is not supported with SAS-attached devices in the Oracle Solaris release.
- Disable power management on system connected to the SAS Domain to prevent unexpected results as one server attempts to power down a device while another attempts to gain access. For information about power management, see [poweradm\(1M\)](#).

## Dynamic Discovery of SAS Devices

Adding and removing SAS devices is performed dynamically when using the `mpt` driver. You are no longer required to edit the `sd.conf` file to enable your system to detect specific targets and LUNs attached with the `mpt` driver. For more information see [mpt\(7D\)](#) and [mpt\\_sas\(7D\)](#).

If you add or remove a device in your SAS domain, messages are written to the `/var/adm/messages` file indicating presence or removal. If a device has been added, it is visible to and available by the `format` command.

## Configuring SAS Boot Devices

The Solaris multipathing I/O features allow systems running the Oracle Solaris OS to boot from a multipathed SAS device, or from a SATA device connected to a SAS controller. Only the bundled `mpt` driver is supported for SAS multipathing in the Oracle Solaris release.

Some SAS and SATA devices might not support multipathing but will still function as non-multipathed devices. For more information, go to the My Oracle support site.



# Configuring IPFC SAN Devices

---

Configuring Internet Protocol over Fibre Channel (IPFC) on the host system describes host recognition of IPFC devices and implementation of IP over FC in a SAN. The IPFC driver is based on RFC 2625 and allows IP traffic to run over FC.

This chapter contains the following topics:

- [“IPFC Considerations” on page 65](#)
- [“Invoking and Configuring IPFC” on page 68](#)

## IPFC Considerations

The following table shows the supported features available for IPFC.

TABLE 9-1 IPFC (NFS/NAS and SNDR)

Feature	Supported
Cascading	Yes, with fabric zones only
Zone type	Fabric zone with the HBA configured as an F-port point-to-point connection
Maximum number of device ports per zone	253

The following restrictions apply:

- IPFC is not supported on Oracle's 1 Gbit switches.
- Promiscuous mode is not supported. The snoop utility cannot be used.
- Multicasting is supported through broadcasting only.
- Network cards using IPFC cannot be used as routers. In the Oracle Solaris 11 release, IP forwarding is disabled by default.

- Any standard network commands can be used after IPFC is attached. These commands (telnet, ping, or ftp) are used in this environment in the same way as in an Ethernet setup.

## Determining Fibre Channel Adapter Port Instances

This section explains how to configure the desired host system for IPFC. It includes the procedures to determine the port instance and to plumb an IPFC instance.

### ▼ How to Determine Port Instances

#### 1 Determine the HBA PCI adapter slot and the I/O board PCI slot.

You need this information to perform the calculation in [“Determining Fibre Channel Adapter Port Instances” on page 66](#).

For example, assume you have an array with an HBA card located in PCI adapter slot 5, and the PCI adapter is in slot 1 of the I/O board.

#### 2 Determine the instance number.

##### a. Search for the fp driver binding name in the `/etc/path_to_inst` file.

---

**Note** – Determine the correct entry by finding the hardware path described in your server hardware manual.

---

##### b. Narrow the search by using the I/O board and slot information from Step 1.

---

**Note** – The following method of deriving the device path of an HBA from its physical location in server might not work for all Oracle's Sun server hardware.

---

##### i. Multiply the PCI adapter slot number by the number of adapter ports.

For example, if the HBA has two ports, multiply by 2. Using the array with an HBA in the PCI adapter slot 5, multiply 5 by 2 to get 10.

##### ii. Add the PCI adapter I/O board slot number to the number derived in Step i.

Using an HBA in PCI adapter slot 5 and PCI slot 1 of the I/O board, add 1 to 10 for a sum of 11.

##### iii. Convert the number derived in Step ii to hexadecimal.

The number 11 converts to “b” in hexadecimal.

iv. Search for the `fp` entry with `pci@hex` where `hex` is the number you derived in Step iii.

The following table shows the elements of the device path for a PCI single FC network adapter device that has the following path:

```
"/pci@b,2000/SUNW,q1c@2/fp@0,0" 7 "fp"
```

Device Name	Value
Physical Name	/pci@b,2000/SUNW,q1c@2/fp@0,0
Instance Number	7
Driver Binding Name	fp

c. Manually create each FP instance.

In this example, the value of `interface-number` is `fcip7`.

```
# ipadm create-ip fcip7
```

If the command is successful, a message appears on both the console and in the messages file. For example:

```
Sep 13 15:52:30 bytownite ip: ip: joining multicasts failed (7) on fcip0 -
will use link layer brocasts for multicast
```

## ▼ How to Create an IPFC Instance

Each FP instance on the system has an entry in `/dev/fc`. If HBAs have been removed, some stale links might exist. Use this procedure to load and create IPFC.

1 For each entry in `/dev/fc` file, display all the devices that are visible through that HBA port:

```
# luxadm -e dump_map /dev/fc/fp0
Pos  Port_ID Hard_Addr Port WWN          Node WWN          Type
0    610100 0           210000e08b049f53 200000e08b049f53 0x1f (Unknown Type)
1    620d02 0           210000e08b02c32a 200000e08b02c32a 0x1f (Unknown Type)
2    620f00 0           210000e08b03eb4b 200000e08b03eb4b 0x1f (Unknown Type)
3    620e00 0           210100e08b220713 200100e08b220713 0x1f (Unknown Type,Host Bus Adapter)
# luxadm -e dump_map /dev/fc/fp1
No FC devices found. - /dev/fc/fp1
```

2 Based on the list of devices, determine which destination HBAs are visible to the remote host with which you want to establish IPFC communications.

In the example for this procedure, the destination HBAs have port IDs `610100` and `620d02`. The originating HBA's port ID is `620e00`.

3 List the physical path of the originating HBA port from which you can see the destination HBA port, where `originating-hba-link` is a variable for the link determined in Step 2.

```
# ls -l /dev/fc/fp originating-hba-link
```

In the following example, 0 is the number for the *originating-hba-link*:

```
# ls -l /dev/fc/fp 0
lrwxrwxrwx  1 root      root          51 Sep  4 08:23 /dev/fc/fp0 ->
../devices/pci@8,600000/SUNW,q1c@1/fp@0,0:devctl
```

#### 4 Search the physical path identified in Step 3.

You must remove the leading `../devices` from the path name output. For example:

```
# grep pci@8,600000/SUNW,q1c@1/fp@0,0 /etc/path_to_inst
"/pci@8,600000/SUNW,q1c@1/fp@0,0" 0 "fp"
```

#### 5 Determine the fp instance for the originating HBA port from the output of the command in Step 4.

The instance number precedes “fp” in the output. In the following example output, the instance number is 0.

```
"/pci@8,600000/SUNW,q1c@1/fp@0,0" 0 "fp"
```

#### 6 Use the instance number from Step 5 to load IPFC and create the IPFC interface.

In this example, the instance is 0.

```
# ipadm create-ip fcip 0
```

## Invoking and Configuring IPFC

Immediately upon installation, start IPFC manually with the `ipadm` command. You can configure the host so that on subsequent reboot, the IPFC network interface starts automatically. This section describes the procedures to start a network interface manually and to configure the host for automatic plumbing upon reboot.

### ▼ How to Start a Network Interface Manually

Use this procedure when you want to plumb IPFC with specific netmask values and get the IPFC interface up and running.

#### 1 Become an administrator.

#### 2 Configure the appropriate network interface.

Ask your network administrator for an appropriate IP address and netmask information. For example, to enable an IPFC interface associated with fp instance 0 and an IP address of 192.9.201.10, type:

```
# ipadm create-ip fcip0
# ipadm create-addr -T static -a 192.9.201.10 fcip0/ipv4
```

For more information, see [ipadm\(1M\)](#).

### 3 Confirm that the network is operational.

```
# ipadm show-if
```

## ▼ How to Configure the Host for Automatic Network Configuration

A system's hostname is set in the `svc:/system/identity:node` service. For example, the hostname is `sys-A` and the IPFC network interface is `fcip0`.

### 1 Become an administrator.

### 2 Confirm that the hostname is set.

```
# svccfg -s identity:node
svc:/system/identity:node> listprop config/nodename
config/nodename astring sys-A
```

If you need to set the hostname, use syntax similar to the following:

```
# svccfg -s identity:node setprop config/nodename = "sys-A"
```

### 3 Make any additional entries to the `/etc/inet/hosts` file.

The installation program creates the `/etc/inet/hosts` file with minimum entries. You must manually make additional entries with a text editor. See the `hosts(4)` man page for additional information.

The `/etc/inet/hosts` file contains the hosts database. This file contains the host names and the primary network interface IP addresses, as well as the IP addresses of other network interfaces attached to the system and of any other network interfaces that the machine must know about.

The following example shows an `etc/inet/host` file.

```
127.0.0.1    localhost    loghost
192.9.200.70  neol        #This is the local host name
192.9.201.10 fcip0       #Interface to network 192.9.201.10
```

### 4 Confirm that the name service SMF service is configured with files `nis` for hosts.

```
# svccfg
svc:> select network/nis/client:default
svc:/network/nis/client:default> select name-service/switch
svc:/system/name-service/switch> listprop config/host
config/host astring "files nis"
svc:/system/name-service/switch> quit
```



# Booting the Solaris OS From Fibre Channel Devices on x86 Systems

---

You can select Fibre Channel (FC) devices to boot from in the disk selection portion of Solaris installation program.

This chapter describes how to manually install the Oracle Solaris OS that includes the Solaris multipathing I/O features with 2-Gbit and 4-Gbit Fibre Channel (FC) HBA drivers on x86-based systems.

---

## Note –

Oracle's 1 Gb HBAs do *not* support booting over a SAN on x86-based systems. Oracle's 4 Gb HBAs do support booting over a SAN on x86 systems. Most 2 Gb HBAs do support booting over a SAN on x86 systems, except for the following:

- Oracle's StorageTek 2 Gb Enterprise Class Dual-Port Fibre HBA, SG-XPCI2FC-QF2-Z
- Oracle's StorageTek 2 Gb PCI Dual-Port Fibre HBA, SG-XPCI2FC-QF2
- Oracle's StorageTek 2 Gb FC PCI Single-Channel Network Adapter, X6767A
- Oracle's StorageTek 2 Gb FC PCI Dual-Channel Network Adapter, X6768A
- Oracle's StorageTek 2 Gb FC PCI-X Enterprise Single-Port HBA, SG-XPCI1FC-QL2
- Oracle's StorageTek 2 Gb FC PCI Single-Port Host Adapter, SG-XPCI1FC-QF2

---

The Oracle Solaris OS can be installed on an x86-based system from a DVD-ROM or the network.

This chapter contains the following topics:

- [“Oracle Solaris OS Setup Requirements” on page 72](#)
- [“Oracle Solaris OS Installation Overview” on page 72](#)
- [“Oracle Solaris OS Installation Procedure” on page 73](#)

## Oracle Solaris OS Setup Requirements

You must have the following items for installation.

- Oracle Solaris OS 11 installation DVD – The following installation methods are available for configuring FC devices on an x86 based system:
  - Automatic installation – Install multiple client systems on the network. You can boot from media for installing a single system, but if you want to customize the installation for multiple client systems, you will need an installation server.
  - Text installation – Installs a single system from media or from an installation server.
- FC HBA that is connected to an x86 based system
- 10/100/1000 Mbit/sec Ethernet network for network-based OS installation

## Oracle Solaris OS Installation Overview

In order to successfully complete the installation, use an FC-based device during the disk selection portion of the installation. At the end of the interactive installation, you must change the x86 BIOS and FC HBA BIOS to identify the FC initiator that will be used to boot Oracle Solaris from a remote multipathed disk.

After OS installation and before rebooting, gather the configuration information by issuing the `luxadm` command on the newly installed logical unit number (LUN). The `luxadm` command output provides a map from `c#t#d#` to the array WWN and LUN number. Record the HBA World Wide Name (WWN) and array WWN port information.

During rebooting, use the WWN and LUN number to set the HBA BIOS for each HBA to be used to boot from each LUN on the same array. Change the system BIOS to perform booting from CD-ROM or Network to disk.

---

**Note** – Console access is required for HBA and system BIOS changes.

---

In addition, note the following during an Oracle Solaris OS installation.

- If using a custom disk layout, do not remove the overlap (`s2`) partition. The x86 boot installer has a dependency on this partition.
- By default, Solaris I/O multipathing features manage the FC boot devices with multiple paths to a single boot device.



# Oracle Solaris OS Installation Procedure

## ▼ How to Install the Oracle Solaris OS

### 1 Install the HBA hardware.

Follow the instructions in the appropriate Oracle HBA installation guide found at (<http://www.oracle.com/technetwork/documentation/oracle-storage-networking-190061.html>).

### 2 Install the Oracle Solaris OS.

If you select an automatic installation and you want to select specific devices to be installed during the installation, see *Installing Oracle Solaris 11 Systems*.

## ▼ How to Perform a DVD or Network-Based OS Installation

After installing the HBA, perform the following steps for a DVD installation or a network-based installation of the Oracle Solaris OS on an x86 based system.

For more information, refer to *Installing Oracle Solaris 11 Systems*.

- 1 If you are installing from a DVD-ROM rather than from the network, insert the Oracle Solaris Software DVD into the DVD-ROM drive.
- 2 Upon initial power up, provide the system BIOS and configure it to boot from either the network or DVD-ROM as applicable.
- 3 Install the Oracle Solaris OS by selecting one of the following methods.
  - Automatic installation – You can begin the automated installation by selecting a network boot from the x86 system's BIOS. If you select an automatic installation, skip to step 7.
  - Text installation – You can begin a text installation by selecting the following option from the GRUB menu when booted from media or when booted from an installation server.  
  
Oracle Solaris 11 11/11 Text Installer and command line  
  
After the installation is complete, you can exit the installer and configure the devices.
- 4 Text installation – Select a desired array and its associated LUN.
- 5 Text installation – Continue the installation by selecting the desired installation options from each installation menu.

**6 Text installation – At the end of the installation screens, verify your selections to start the Oracle Solaris OS installation.**

**7 After the installation is complete, select one of the following based on whether you performed an automatic or text installation.**

- Automatic installation – By default, a system is not rebooted after the installation because of the following manifest keywords in the `/usr/share/auto_install/default.xml` file. This means you can configure your devices before the system reboots.

```
<auto_install>
  <ai_instance name="default">
  .
  .
  .
```

If a previous installation set the following keyword value to `true`, change this value to `false` so that you can configure your devices before the system reboots.

```
<auto_install>
  <ai_instance name="default" auto_reboot="true">
  .
  .
  .
```

- Text installation – When the installation is complete, select the `Quit` option to exit the installer to configure your devices.

**8 Before rebooting after the installation completes, issue the `luxadm display` command on the LUN that was selected during installation.**

See [Figure 10–1](#).

```
# luxadm display /dev/rdisk/c0t600015D0002028000000000000001142d0s2
DEVICE PROPERTIES for disk: /dev/rdisk/c0t600015D0002028000000000000001142d0s2
Vendor:      SUN
Product ID:  SE6920
Revision:    0202
Serial Num:  00500057
Unformatted capacity: 10240.000 MBytes
Read Cache:  Enabled
  Minimum prefetch: 0x0
  Maximum prefetch: 0xffff
Device Type:  Disk device
Path(s):

/dev/rdisk/c0t600015D0002028000000000000001142d0s2
/devices/scsi_vhci/disk@g600015d0002028000000000000001142:c,raw
Controller   /dev/cfg/c4
  Device Address      213600015d207200,0
  Host controller port WWN      210100e08b206812
  Class               primary
  State               ONLINE
Controller     /dev/cfg/c11
  Device Address      213600015d207200,0
  Host controller port WWN      210100e08b30a2f2
```

```

Class      primary
State     ONLINE

```

FIGURE 10-1 luxadm display Command and Output

```

QLogic Fast!UTIL
=====Selected Adapter=====
Adapter Type I/O Address Slot Bus Device Function
QLA2462      3400          02  03  01    1
=====
=====Adapter Settings=====
BIOS Address:                CF000
BIOS Revision:               1.05
Adapter Serial Number:      A04712
Interrupt Level:            5
Adapter Port Name:          210100E08B206812
Host Adapter BIOS:          Enabled
Frame Size:                 2048
Loop Reset Delay:           5
Adapter Hard Loop ID:       Disabled
Hard Loop ID:               0
Spinup Delay:               Disabled
Connection Options:         2
Fibre Channel Tape Support:Enabled
Data Rate:                  2
=====
Use <Arrow keys> and <Enter> to change settings, <Esc> to exit

```

The following luxadm command output from the example in the figure can be used to map the MPxIO based c#t#d# to the HBA WWN and array WWN:

- MPxIO c#t#d# = c0t600015d00020280000000000000001142d0
- Array WWN = 213600015d207200, LUN 0
- HBA WWNs = 210100e08b206812 and 210100e08b30a2f2

- 9 **During the reboot process, watch the monitor so you can enter the HBA #1 BIOS screen and specify the boot device to be the FC LUN onto which you just installed the Oracle Solaris OS.**

Follow this step for each HBA to be used for multipathing, and specify the boot device to be the FC LUN on which you installed the Oracle Solaris OS. See [Figure 10–2](#) and [Figure 10–3](#).

  - **For the QLogic HBA BIOS, perform the following steps.**
    - a. **During host rebooting, press Control-Q to display the HBA BIOS screen.**
    - b. **Select the HBA that you want to enable as the boot device and enable boot.**
    - c. **Configure the boot device:**
      - i. **Select Configuration Settings.**
      - ii. **Select Selectable Boot Settings.**
      - iii. **Make sure that Selectable Boot is set to enable.**

In this menu, you can select the boot device/LUN by the array WWPN.
      - iv. **Save and then exit the HBA BIOS screen.**
  - **For the Emulex HBA BIOS, perform the following steps.**
    - a. **During host rebooting, press Alt-E to display the HBA BIOS screen.**
    - b. **Select the HBA that you want to enable as the boot device and enable boot.**
    - c. **Select Configure Boot Devices.**
    - d. **Select a boot entry.**
    - e. **Select the WWPN of the desired boot device.**
    - f. **Type the LUN number.**
    - g. **Select the boot LUN.**
    - h. **Select Boot Device with the Array WWPN.**
    - i. **Save and exit the HBA BIOS screen.**

FIGURE 10-2 HBA BIOS Screen for an HBA WWN

```

QLogic Fast!UTIL
=====Selected Adapter=====
Adapter Type I/O Address Slot Bus Device Function
QLA2462      3400          02  03  01    1
=====

=====Adapter Settings=====

BIOS Address:                CF000
BIOS Revision:               1.05
Adapter Serial Number:      A04712
Interrupt Level:            5
Adapter Port Name:          210100E08B206812
Host Adapter BIOS:         Enabled
Frame Size:                 2048
Loop Reset Delay:           5
Adapter Hard Loop ID:       Disabled
Hard Loop ID:               0
Spinup Delay:               Disabled
Connection Options:         2
Fibre Channel Tape Support:Enabled
Data Rate:                  2
=====

Use <Arrow keys> and <Enter> to change settings, <Esc> to exit

```

FIGURE 10-3 HBA BIOS Screen for Enabling Boot From the FC LUN

```

QLogic FastUTIL
=====Selected Adapter=====
| Adapter Type I/O Address Slot Bus Device Function |
|  QLA2462      3000          02  03  01  0         |
=====
=====Selectable Boot Settings=====
| Selectable Boot: Enabled |
| (Primary) Boot Port Name, Lun: 213600015D207200, 0 |
| Boot Port Name, Lun: 213600015D207200, 1 |
| Boot Port Name, Lun: 213600015D207200, 2 |
| Boot Port Name, Lun: 213600015D207200, 3 |
|
| Press "C" to clear a Boot Port Name entry |
=====
Use <Arrow keys> and <Enter> to change settings, <Esc> to exit

```

**Note –**

The figure shows the following modifications:

- Selectable boot = Enabled
- ARRAY WWN = 213600015d207200
- ARRAY LUN = 0
- HBA WWN = 210100e08b206812

- 10 Repeat the appropriate modifications for all HBAs and all LUNs.
- 11 Type the system BIOS per the vendor's access method and specify the boot device to be the FC LUN on which you installed the Oracle Solaris OS.
- 12 Reboot to the newly installed Oracle Solaris OS by using the FC LUN specified in the system BIOS.

# Persistent Binding for Tape Devices

---

To simplify management of servers in SAN-based data-centers, Oracle's StorageTek SAN Foundation software stack in the Solaris OS dynamically detects devices in a SAN and builds associated `/dev` tree entries without requiring you to edit configuration files.

In most cases, this process greatly simplifies SAN management. However, for tape devices, you might like the ability to explicitly specify how that `/dev` entries are created and to ensure the `/dev` entries are identical across multiple servers on a SAN. This chapter describes how you can specify this tape binding in the Solaris OS while retaining the benefits of automatic discovery for disk-based devices.

## Persistent Binding Overview

The `/dev/rmt` directory contains links to physical devices under `/devices` for tape devices. Each tape LUN seen by the system is represented by 24 minor nodes in the form of `/dev/rmt/N`, `/dev/rmt/Nb`, and `/dev/rmt/Nbn`, where  $N$  is an integer counter starting from 0. This number is picked by `devfsadm` during enumeration of new devices. Every new tape logical unit number (LUN) found by `devfsadm` gets the next available number in `/dev/rmt`.

Because the `/dev/rmt` name depends on the order in which devices appear in the device tree, it changes from system to system. For a given tape drive that is seen by two or more different systems, the `/dev/rmt` link can be different on each of these systems. This difference can cause problems for the most common usage of Symantec (VERITAS) NetBackup (SSO option). Also, if the drive is replaced, the links change unless the vendor provides a way to retain the port World-Wide Name (*PWWN*) of the drive.

## Creating Tape Links

The `/etc/devlink.tab` file is called the default device table file. It specifies rules that `devfsadm` uses to create links in the `/dev` directory. This file does not contain any entries for tapes because `devfsadm` is already able to create links for tape drives, but rules can be added that will modify the default behavior for creating tape links. For more information, see [devlinks\(1M\)](#).

For any tape drive visible to Oracle Solaris OS but not specified in the `devlink` file, `devfsadm` automatically assigns a minor node number starting from 0. These minor node numbers will conflict with any lower numbers assigned manually in `/etc/devlink.tab`, so be sure to assign numbers there that are high enough to avoid conflicts.

This approach can easily lead to duplicate links in `/dev/rmt`. Any tapes discovered before entries were specified in `/etc/devlink.tab` have automatically created links. When entries are added and `devfsadm` is run, the original links remain in `/dev/rmt`, resulting in duplicate links. To remove the original links in `/dev/rmt`, run the `rm /dev/rmt/*` command before running `devfsadm`.

This approach cannot be used with multiple-port tape drives that are attached to multiple HBA ports. If multiple HBA ports are attached to the same tape LUN, the system detects two tape drives instead of one. The one that appears last in the `prtconf` output gets the link generated by the `/etc/devlink.tab`.

The following example shows a sample entry for tape in the `devlink.tab` file.

```
type=ddi_byte:tape;addr=PWWN,LUN-number; rmt/rmt-number\M0
```

Change the `rmt #` to whatever `/dev/rmt/N` is required. Then change the PWWN and LUN to match the desired tape device. You can obtain this value by running the `ls -l` command on the existing `/dev/rmt/ link` as shown below.

```
# ls -l /dev/rmt/4
lrwxrwxrwx 1 root root 69 Oct 6 14:57 /dev/rmt/4 ->
../../../../devices/pci@1f,700000/SUNW,q1c@2/fp@0,0/st@w5005076300617717,0:
```

If, for example, you wanted the `/dev/rmt/` number to be 40, you would create an entry in `/etc/devlink.tab` like the following example:

```
# type=ddi_byte:tape;addr=w5005076300617717,0; rmt/40\M0
```

You can then add this line to the `devlink` file on every Solaris server on the SAN that uses this drive so that it always appears as minor node 40.



## ▼ How to Create Tape Device Links

**1 Become an administrator.**

**2 Create the entries in `/etc/devlink.tab` as described in “Creating Tape Links” on page 80.**

If `devfsadm` has previously discovered the devices, you must determine the device address by running the `ls -l` command on the existing link.

---

**Note** – Be sure to assign `/dev/rmt/ N` numbers to avoid conflicts with any automatically configured devices, as described above.

---

**3 Remove existing links from `/dev/rmt` by running the `rm /dev/rmt/*` command.**

**4 Run `devfsadm`.**

This command creates new links as per the entries in `/etc/devlink.tab` in addition to automatically creating links for any unspecified devices.



# Manual Configuration for Fabric-Connected Devices

---

This appendix explains how to configure and unconfigure the fabric devices in the Oracle Solaris OS. It explains how the visible fabric devices on a host are detected and configured with and without enabling the multipathing software.

This appendix contains the following sections:

- “Manually Configuring FC Devices” on page 83
- “Configuring Fabric Device Nodes” on page 84
- “Configuring Device Nodes Without Multipathing Enabled” on page 86
- “Configuring Device Nodes With Solaris Multipathing Enabled” on page 89
- “Unconfiguring Fabric Devices” on page 92

## Manually Configuring FC Devices

In the Oracle Solaris release, fabric-connected devices are available automatically to the Oracle Solaris system.

If you want to manually configure the fabric-connected devices, use the following steps to change the default behavior.

---

**Note** – Changing the default behavior makes all of your fabric-connected devices unavailable, which can cause problems for fabric-connected devices that are required to be available at boot time.

---

### ▼ How to Manually Configure a FC Device

- 1 Become an administrator.
- 2 Copy the `/kernel/drv/fp.conf` file to the `/etc/driver/drv/fp.conf` file.

- 3 Enable manual configuration by making sure that the following line in the `/etc/driver/drv/fp.conf` file is uncommented.**

```
manual_configuration_only=1;
```

Refer to the `fp(7D)` man page and the `cfgadm_fp(1M)` man page for further information on this setting.

- 4 Reboot the system.**
- 5 For each fabric-connected device to be made available, select one of the following tasks, depending on whether you are using the Solaris I/O multipathing features.**
  - [“Configuring Device Nodes Without Multipathing Enabled” on page 86](#)
  - [“Configuring Device Nodes With Solaris Multipathing Enabled” on page 89](#)

If the original default behavior for fabric-connected devices is desired, see the next step.

- 6 Disable manual configuration by making sure that the following line in the `/etc/driver/drv/fp.conf` file is commented:**

```
# manual_configuration_only=1;
```

For more information about this setting, see `cfgadm_fp(1M)` and `fp(7D)`.

- 7 Reboot the system.**

```
# init 6
```

## Configuring Fabric Device Nodes

After you configure the hardware in your direct-attach system or SAN, you must ensure that the systems recognize the devices. This section explains host recognition of fabric devices, also known as 24-bit FC addressing devices on the SAN. After configuring the devices, ports, and zones in your SAN, make sure that the system is aware of the devices. You can have up to 16 million fabric devices connected together on a SAN with FC support.

This section is limited to the operations required from the perspective of the Oracle Solaris OS. It does *not* cover other aspects, such as device availability and device-specific management. If devices are managed by other software, such as a volume manager, refer to the volume manager product documentation for additional instructions.

## Ensuring That LUN Level Information Is Visible

### ▼ How to Ensure LUN Level Information is Visible

1 Become an administrator.

2 Identify the LUN level information.

```
# cfgadm -al -o show_SCSI_LUN
```

If you issue the `cfgadm -al -o show_SCSI_LUN controller-ID` command immediately after a system boots, the output might not show the Fibre Channel Protocol (FCP) SCSI LUN level information. The information does not appear because the storage device drivers, such as the `ssd` and `st` driver, are not loaded yet on the running system.

3 Determine whether the drivers are loaded.

For example:

```
# modinfo | grep ssd
```

After the drivers are loaded, the LUN level information is visible in the `cfgadm` output.

### ▼ How to Detect Visible Fabric Devices on a System

This section provides an example of the procedure for detecting fabric devices using FC host ports `c0` and `c1`. This procedure also shows the device configuration information that is displayed with the `cfgadm` command.

---

**Note** – In the following examples, only failover path attachment point IDs (`Ap_Ids`) are listed. The `Ap_Ids` displayed on your system depend on your system configuration.

---

1 Become an administrator.

2 Display the information about the attachment points on the system.

```
# cfgadm -l
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	fc-fabric	connected	unconfigured	unknown
c1	fc-private	connected	configured	unknown

In this example, `c0` represents a fabric-connected host port, and `c1` represents a private, loop-connected host port. Use the `cfgadm` command to manage the device configuration on fabric-connected host ports.

By default, the device configuration on private, loop-connected host ports is managed by a system running the Oracle Solaris OS.

### 3 Display information about the host ports and their attached devices.

```
# cfgadm -al
Ap_Id                Type      Receptacle  Occupant  Condition
c0                   fc-fabric connected  unconfigured unknown
c0::50020f2300006077 disk      connected  unconfigured unknown
c0::50020f23000063a9 disk      connected  unconfigured unknown
c0::50020f2300005f24 disk      connected  unconfigured unknown
c0::50020f2300006107 disk      connected  unconfigured unknown
c1                   fc-private connected  configured  unknown
c1::220203708b69c32b disk      connected  configured  unknown
c1::220203708ba7d832 disk      connected  configured  unknown
c1::220203708b8d45f2 disk      connected  configured  unknown
c1::220203708b9b20b2 disk      connected  configured  unknown
```

---

**Note** – The `cfgadm -l` command displays information about FC host ports. You can also use the `cfgadm -al` command to display information about FC devices. The lines that include a port world wide name (WWN) in the `Ap_Id` field associated with `c0` represent a fabric device. Use the `cfgadm configure` and `cfgadm unconfigure` commands to manage those devices and make them available to systems using the Oracle Solaris OS. The `Ap_Id` devices with port WWNs under `c1` represent private-loop devices that are configured through the `c1` host port.

---

## Configuring Device Nodes Without Multipathing Enabled

This section describes fabric device configuration tasks on a system that does not have multipathing enabled.

The procedures in this section show how to detect fabric devices that are visible on a system and to configure and make them available to a system running the Oracle Solaris OS. The procedures in this section use specific devices as examples to illustrate how to use the `cfgadm` command to detect and configure fabric devices.

The device information that you supply and that is displayed by the `cfgadm` command depends on your system configuration.

### ▼ How to Manually Configure an FC Device Without Multipathing

This sample procedure describes how to configure a fabric device that is attached to the fabric-connected host port `c0`.

- 1 Become an administrator.
- 2 Identify the device to be configured.

```
# cfgadm -al
Ap_Id                Type      Receptacle  Occupant  Condition
```

```

c0                fc-fabric  connected  unconfigured  unknown
c0::50020f2300006077 disk    connected  unconfigured  unknown
c0::50020f23000063a9 disk    connected  unconfigured  unknown
c0::50020f2300005f24 disk    connected  unconfigured  unknown
c0::50020f2300006107 disk    connected  unconfigured  unknown
c1                fc-private connected  configured    unknown
c1::220203708b69c32b disk    connected  configured    unknown
c1::220203708ba7d832 disk    connected  configured    unknown
c1::220203708b8d45f2 disk    connected  configured    unknown
c1::220203708b9b20b2 disk    connected  configured    unknown

```

### 3 Configure the fabric device.

```
# cfgadm -c configure c0::50020f2300006077
```

### 4 Verify that the selected fabric device is configured.

```

# cfgadm -al
Ap_Id            Type      Receptacle  Occupant    Condition
c0                fc-fabric  connected   configured  unknown
c0::50020f2300006077 disk    connected   configured  unknown
c0::50020f23000063a9 disk    connected   unconfigured  unknown
c0::50020f2300005f24 disk    connected   unconfigured  unknown
c0::50020f2300006107 disk    connected   unconfigured  unknown
c1                fc-private connected   configured  unknown
c1::220203708b69c32b disk    connected   configured  unknown
c1::220203708ba7d832 disk    connected   configured  unknown
c1::220203708b8d45f2 disk    connected   configured  unknown
c1::220203708b9b20b2 disk    connected   configured  unknown

```

Notice that the Occupant column for both `c0` and `c0::50020f2300006077` displays as configured, indicating that the `c0` port has a configured occupant and that the `c0::50020f2300006077` device is configured.

### 5 Display FCP SCSI LUN information for multi-LUN SCSI devices.

The following code example shows that the physical devices connected through `Ap_Id` `c0::50020f2300006077` have four LUNs configured.

```

# cfgadm -al -o show_SCSI_LUN c0
Ap_Id            Type      Receptacle  Occupant    Condition
c0                fc-fabric  connected   configured  unknown
c0::50020f2300006077,0 disk    connected   configured  unknown
c0::50020f2300006077,1 disk    connected   configured  unknown
c0::50020f2300006077,2 disk    connected   configured  unknown
c0::50020f2300006077,3 disk    connected   configured  unknown

```

The device is now available on the system running the Oracle Solaris OS. The paths represent each SCSI LUN in the physical device represented by `c0::50020f2300006077`.

## ▼ How to Configure Multiple FC Devices Without Multipathing

Make sure you first identify the devices visible to the system with the procedure “[Ensuring That LUN Level Information Is Visible](#)” on page 85. This procedure describes how to configure all unconfigured fabric devices that are attached to a fabric-connected host port. The port used as an example is `c0`.

- 1 Become an administrator.
- 2 Identify the devices to be configured.

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric    connected   unconfigured unknown
c0::50020f2300006077 disk         connected   unconfigured unknown
c0::50020f23000063a9 disk         connected   unconfigured unknown
c0::50020f2300005f24 disk         connected   unconfigured unknown
c0::50020f2300006107 disk         connected   unconfigured unknown
c1             fc-private   connected   configured   unknown
c1::220203708b69c32b disk         connected   configured   unknown
c1::220203708ba7d832 disk         connected   configured   unknown
c1::220203708b8d45f2 disk         connected   configured   unknown
c1::220203708b9b20b2 disk         connected   configured   unknown
```

- 3 Configure all of the unconfigured devices on the selected port.

```
# cfgadm -c configure c0
```

---

**Note** – This operation repeats the `configure` operation of an individual device for all the devices on `c0`. This can be time consuming if the number of devices on `c0` is large.

---

- 4 Verify that all devices on `c0` are configured.

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric    connected   configured   unknown
c0::50020f2300006077 disk         connected   configured   unknown
c0::50020f23000063a9 disk         connected   configured   unknown
c0::50020f2300005f24 disk         connected   configured   unknown
c0::50020f2300006107 disk         connected   configured   unknown
c1             fc-private   connected   configured   unknown
c1::220203708b69c32b disk         connected   configured   unknown
c1::220203708ba7d832 disk         connected   configured   unknown
c1::220203708b8d45f2 disk         connected   configured   unknown
c1::220203708b9b20b2 disk         connected   configured   unknown
```



## 5 Display FCP SCSI LUN information for multi-LUN SCSI devices.

The following code example shows that the physical devices represented by `c0::50020f2300006077` and `c0::50020f2300006107` each have four LUNs configured. The physical devices represented by `c0::50020f23000063a9` and `c0::50020f2300005f24` each have two LUNs configured.

```
# cfgadm -al -o show_SCSI_LUN c0
Ap_Id          Type      Receptacle  Occupant    Condition
c0             fc-fabric connected   configured  unknown
c0::50020f2300006077,0 disk      connected   configured  unknown
c0::50020f2300006077,1 disk      connected   configured  unknown
c0::50020f2300006077,2 disk      connected   configured  unknown
c0::50020f2300006077,3 disk      connected   configured  unknown
c0::50020f23000063a9,0 disk      connected   configured  unknown
c0::50020f23000063a9,1 disk      connected   configured  unknown
c0::50020f2300005f24,0 disk      connected   configured  unknown
c0::50020f2300005f24,1 disk      connected   configured  unknown
c0::50020f2300006107,0 disk      connected   configured  unknown
c0::50020f2300006107,1 disk      connected   configured  unknown
c0::50020f2300006107,2 disk      connected   configured  unknown
c0::50020f2300006107,3 disk      connected   configured  unknown
```

# Configuring Device Nodes With Solaris Multipathing Enabled

This section describes how to perform fabric device configuration steps on a system that has the multipathing features enabled.

The devices attached to the fabric-connected host port are not configured by default and so are not available to the system. Use the `cfgadm configure` and `cfgadm unconfigure` commands to manage device node creation for fabric devices. For more information, see [cfgadm\\_fp\(1M\)](#). The procedures in this section illustrate steps to detect fabric devices that are visible on a system and to configure them as multipathing devices to make them available to the system.

The device information that you supply, and that is displayed by the `cfgadm` command, depends on your system configuration.

## ▼ How to Configure Individual Multipathed FC Devices

This sample procedure uses fabric-connected host ports `c0` and `c2` to configure fabric devices as multipathed devices on a system that has the multipathing software enabled.

The `cfgadm -c configure` command for fabric devices is the same regardless of whether multipathing is enabled.

### 1 Become an administrator.

## 2 Identify the port WWN of the device to be configured as a multipathed device.

Look for devices on a fabric-connected host port marked as `fc-fabric`. These devices are the devices you can configure with the `cfgadm -c` configure command.

```
# cfgadm -al
Ap_Id                Type      Receptacle Occupant  Condition
c0                   fc-fabric connected  unconfigured unknown
c0::50020f2300006077 disk      connected  unconfigured unknown
c0::50020f23000063a9 disk      connected  unconfigured unknown
c1                   fc-private connected  configured  unknown
c1::220203708b69c32b disk      connected  configured  unknown
c1::220203708ba7d832 disk      connected  configured  unknown
c1::220203708b8d45f2 disk      connected  configured  unknown
c1::220203708b9b20b2 disk      connected  configured  unknown
c2                   fc-fabric connected  unconfigured unknown
c2::50020f2300005f24 disk      connected  unconfigured unknown
c2::50020f2300006107 disk      connected  unconfigured unknown
```

In the above example, the `c0::50020f2300006077` and `c2::50020f2300006107` `Ap_Ids` represent the same storage device with different port WWNs for the storage device controllers. The `c0` and `c2` host ports are enabled for multipathing.

## 3 Configure the fabric device and make the devices available to the system.

```
# cfgadm -c configure c0::50020f2300006077 c2::50020f2300006107
```

## 4 Verify that the selected devices are configured.

```
# cfgadm -al
Ap_Id                Type      Receptacle Occupant  Condition
c0                   fc-fabric connected  configured unknown
c0::50020f2300006077 disk      connected  configured unknown
c0::50020f23000063a9 disk      connected  unconfigured unknown
c1                   fc-private connected  configured  unknown
c1::220203708b69c32b disk      connected  configured  unknown
c1::220203708ba7d832 disk      connected  configured  unknown
c1::220203708b8d45f2 disk      connected  configured  unknown
c1::220203708b9b20b2 disk      connected  configured  unknown
c2                   fc-fabric connected  configured  unknown
c2::50020f2300005f24 disk      connected  unconfigured unknown
c2::50020f2300006107 disk      connected  configured  unknown
```

Notice that the `Occupant` column of `c0` and `c0::50020f2300006077` specifies `configured`, which indicates that the `c0` port has at least one configured occupant and that the `c0::50020f2300006077` device is configured. The same change has been made in `c2` and `c2::50020f2300006107`.

When the configure operation has been completed without an error, multipathed devices are created on the system. If the physical device represented by `c0::50020f2300006077` and `c2::50020f2300006107` has multiple SCSI LUNs configured, each LUN is configured as a

multipathed device. The example below shows that two LUNs are configured through `c0::50020f2300006077` and `c2::50020f2300006107`. Each `Ap_Id` is associated with a path to multipathed devices.

```
# cfmadm -al -o show_SCSI_LUN c0::50020f2300006077\ c2::50020f2300006107
Ap_Id      Type      Receptacle  Occupant  Condition
c0::50020f2300006077,0 disk      connected   configured unknown
c0::50020f2300006077,1 disk      connected   configured unknown
c2::50020f2300006107,0 disk      connected   configured unknown
c2::50020f2300006107,1 disk      connected   configured unknown
```

The example above shows that the following two multipathed devices are created for the device represented by `c0::50020f2300006077` and `c2::50020f2300006107`:

## ▼ How to Configure Multiple Multipathed FC Devices

Before you configure or remove device nodes, be sure to first identify the fabric devices by using the procedure [“Ensuring That LUN Level Information Is Visible”](#) on page 85.

In this example, an `Ap_Id` on a fabric-connected host port is a path to a multipathed device. For example, all devices with a path through `c2` are to be configured, but none through `c0` are to be configured. `c2` is an attachment point from the system to the fabric, whereas `c2::50020f2300006107` is an attachment point from the storage to the fabric. A system detects all the storage devices in a fabric for which it is configured.

Configuring an `Ap_Id` on the device that has already been configured through another `Ap_Id` results in an additional path to the previously configured device. A new device node is not created in this case. The device node is created only the first time an `Ap_Id` to the corresponding device is configured.

- 1 Become an administrator.
- 2 Identify the fabric-connected host port to be configured.

```
# cfmadm -al
Ap_Id      Type      Receptacle  Occupant  Condition
c0         fc-fabric connected   configured unknown
c0::50020f2300006077 disk      connected   configured unknown
c0::50020f23000063a9 disk      connected   configured unknown
c1         fc-private connected   configured unknown
c1::220203708b69c32b disk      connected   configured unknown
c1::220203708ba7d832 disk      connected   configured unknown
c1::220203708b8d45f2 disk      connected   configured unknown
c1::220203708b9b20b2 disk      connected   configured unknown
c2         fc-fabric connected   unconfigured unknown
c2::50020f2300005f24 disk      connected   unconfigured unknown
c2::50020f2300006107 disk      connected   unconfigured unknown
```

Devices represented by `Ap_Ids` `c0::50020f2300006077` and `c2::50020f2300006107` are two paths to the same physical device, with `c0::50020f2300006077` already configured. Configure

the unconfigured devices on the selected port. This operation repeats the `configure` command of an individual device for all the devices on `c2`. This can be time-consuming if the number of devices on `c2` is large.

```
# cfgadm -c configure c2
```

### 3 Verify that all devices on `c2` are configured.

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant  Condition
c0             fc-fabric    connected   configured unknown
c0::50020f2300006077 disk         connected   configured unknown
c0::50020f23000063a9 disk         connected   configured unknown
c1             fc-private   connected   configured unknown
c1::220203708b69c32b disk         connected   configured unknown
c1::220203708ba7d832 disk         connected   configured unknown
c1::220203708b8d45f2 disk         connected   configured unknown
c1::220203708b9b20b2 disk         connected   configured unknown
c2             fc-fabric    connected   configured unknown
c2::50020f2300005f24 disk         connected   configured unknown
c2::50020f2300006107 disk         connected   configured unknown
```

Notice that the `Occupant` column of `c2` and all of the devices under `c2` is marked as `configured`.

The `show_SCSI_LUN` command displays FCP SCSI LUN information for multiple LUN SCSI devices. The following code example shows that the physical devices connected through by `c2::50020f2300006107` and `c2::50020f2300005f24` each have two LUNs configured.

```
# cfgadm -al -o show_SCSI_LUN c2
Ap_Id          Type          Receptacle  Occupant  Condition
c2             fc-fabric    connected   configured unknown
c2::50020f2300005f24,0 disk         connected   configured unknown
c2::50020f2300005f24,1 disk         connected   configured unknown
c2::50020f2300006107,0 disk         connected   configured unknown
c2::50020f2300006107,1 disk         connected   configured unknown
```

## Unconfiguring Fabric Devices

This section provides information about unconfiguring the multipathing features for fabric devices.

### Unconfiguring a Fabric Device

Before you unconfigure a fabric device, stop all activity to the device and unmount any file systems on the fabric device. See the Oracle Solaris administration documentation for unmounting instructions. If the device is under any volume manager's control, see the documentation for your volume manager before unconfiguring the device.

## ▼ How to Manually Unconfigure a FC Device

This procedure describes how to unconfigure a fabric device that is attached to the fabric-connected host port c0.

### 1 Become an administrator.

### 2 Identify the device to be unconfigured.

Only devices on a fabric-connected host port can be unconfigured.

```
# cfgadm -al
Ap_Id                Type          Receptacle  Occupant  Condition
c0                   fc-fabric    connected   configured unknown
c0::50020f2300006077 disk          connected   configured unknown
c0::50020f23000063a9 disk          connected   configured unknown
c1                   fc-private   connected   configured unknown
c1::220203708b69c32b disk          connected   configured unknown
c1::220203708ba7d832 disk          connected   configured unknown
```

### 3 Unconfigure the fabric device.

```
# cfgadm -c unconfigure c0::50020f2300006077
```

### 4 Verify that the selected fabric device is unconfigured.

```
# cfgadm -al
Ap_Id                Type          Receptacle  Occupant  Condition
c0                   fc-fabric    connected   configured unknown
c0::50020f2300006077 disk          connected   unconfigured unknown
c0::50020f23000063a9 disk          connected   configured unknown
c1                   fc-private   connected   configured unknown
c1::220203708b69c32b disk          connected   configured unknown
c1::220203708ba7d832 disk          connected   configured unknown
```

## ▼ How to Unconfigure All FC Devices on a FC Host Port

This procedure describes how to unconfigure all configured fabric devices that are attached to a fabric-connected host port.

### 1 Become an administrator.

### 2 Identify the fabric devices to be unconfigured.

Only devices on a fabric-connected host port can be unconfigured.

```
# cfgadm -al
Ap_Id                Type          Receptacle  Occupant  Condition
c0                   fc-fabric    connected   configured unknown
c0::50020f2300006077 disk          connected   configured unknown
c0::50020f23000063a9 disk          connected   configured unknown
c1                   fc-private   connected   configured unknown
c1::220203708b69c32b disk          connected   configured unknown
c1::220203708ba7d832 disk          connected   configured unknown
```

### 3 Stop all activity to each fabric device on the selected port and unmount any file systems on each fabric device.

If the device is under any volume manager's control, see the documentation for your volume manager before unconfiguring the device.

```
# cfgadm -c unconfigure c0
```

### 4 Unconfigure all of the configured fabric devices on a selected port.

---

**Note** – This operation repeats the unconfigure operation of an individual device for all the devices on c0. This process can be time-consuming if the number of devices on c0 is large.

---

### 5 Verify that all the devices on c0 are unconfigured.

```
# cfgadm -al
Ap_Id                Type           Receptacle  Occupant    Condition
c0                   fc-fabric     connected   unconfigured unknown
c0::50020f2300006077 disk          connected   unconfigured unknown
c0::50020f23000063a9 disk          connected   unconfigured unknown
c1                   fc-private    connected   configured  unknown
c1::220203708b69c32b disk          connected   configured  unknown
c1::220203708ba7d832 disk          connected   configured  unknown
```

Notice that the Occupant column of c0 and all the fabric devices attached to it are displayed as unconfigured.

## ▼ How to Unconfigure a Multipathed FC Device

This procedure shows fabric-connected host ports c0 and c2 to illustrate how to unconfigure fabric devices associated with multipathed devices.

### 1 Become an administrator.

### 2 Identify the port WWN of the fabric device to be unconfigured.

```
# cfgadm -al
Ap_Id                Type           Receptacle  Occupant    Condition
c0                   fc-fabric     connected   configured  unknown
c0::50020f2300006077 disk          connected   configured  unknown
c0::50020f23000063a9 disk          connected   configured  unknown
c1                   fc-private    connected   configured  unknown
c1::220203708b69c32b disk          connected   configured  unknown
c1::220203708ba7d832 disk          connected   configured  unknown
c2                   fc-fabric     connected   configured  unknown
c2::50020f2300005f24 disk          connected   configured  unknown
c2::50020f2300006107 disk          connected   configured  unknown
```

In this example, the c0::50020f2300006077 and c2::50020f2300006107 Ap\_Ids represent different port WWNs for the same device associated with a multipathed device. The c0 and c2 host ports are enabled for use.

### 3 Stop all device activity to each fabric device on the selected port and unmount any file systems on each fabric device.

If the device is under any volume manager's control, see the documentation for your volume manager for maintaining the fabric device.

### 4 Unconfigure fabric devices associated with the device.

Only devices on a fabric-connected host port can be unconfigured through the `cfgadm -c unconfigure` command.

```
# cfgadm -c unconfigure c0::50020f2300006077 c2::50020f2300006107
```

---

**Note** – You can remove a device from up to eight paths individually, as in the example command `cfgadm -c unconfigure c0::1111, c1::2222, c3::3333`, and so on. As an alternative, you can remove an entire set of paths from the host, as in the example `cfgadm -c unconfigure c0`.

---

### 5 Verify that the selected devices are unconfigured.

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant  Condition
c0             fc-fabric    connected   configured unknown
c0::50020f2300006077 disk         connected   unconfigured unknown
c0::50020f23000063a9 disk         connected   configured  unknown
c1             fc-private   connected   configured  unknown
c1::220203708b69c32b disk         connected   configured  unknown
c1::220203708ba7d832 disk         connected   configured  unknown
c2             fc-fabric    connected   configured  unknown
c2::50020f2300005f24 disk         connected   configured  unknown
c2::50020f2300006107 disk         connected   unconfigured unknown
```

Notice that the Ap\_Ids `c0::50020f2300006077` and `c2::50020f2300006107` are unconfigured. The Occupant column of `c0` and `c2` still displays those ports as configured because they have other configured occupants.

Multipathed devices associated with the Ap\_Ids `c0::50020f2300006077` and `c2::50020f2300006107` are no longer available to the system. The following two devices are removed from the system:

```
/dev/rdsk/c6t60020F20000061073AC8B52D000B74A3d0s2
```

```
/dev/rdsk/c6t60020F20000061073AC8B4C50004ED3Ad0s2
```

## ▼ How to Unconfigure One Path to a Multipathed FC Device

In contrast to the procedure in the preceding section, this procedure shows how to unconfigure one device associated with `c2::50020f2300006107` and leave the other device, `50020f2300006077`, configured. Only devices on a fabric-connected host port can be unconfigured through the `cfgadm unconfigure` command.

### 1 Become an administrator.

## 2 Identify the Ap\_Id of the multipathed device to be unconfigured.

```
# cfgadm -al
Ap_Id                Type      Receptacle  Occupant  Condition
c0                   fc-fabric connected  configured unknown
c0::50020f2300006077 disk      connected  configured unknown
c0::50020f23000063a9 disk      connected  configured unknown
c1                   fc-private connected  configured unknown
c1::220203708b69c32b disk      connected  configured unknown
c1::220203708ba7d832 disk      connected  configured unknown
c2                   fc-fabric connected  configured unknown
c2::50020f2300005f24 disk      connected  configured unknown
c2::50020f2300006107 disk      connected  configured unknown
```

In this example, `c0::50020f2300006077` and `c2::50020f2300006107` Ap\_Ids represent different port WWNs for the same device.

## 3 Unconfigure the Ap\_Id associated with the device.

---

**Note** – If the Ap\_Id represents the last configured path to the device, stop all activity to the path and unmount any file systems on it. If the multipathing device is under any volume manager's control, see the documentation for your volume manager for maintaining the fabric device.

---

In the example that follows, the path represented as `c2::50020f2300006107` is unconfigured, and `c0::50020f2300006077` remains configured to show how you can unconfigure just one of multiple paths for a multipathing device.

```
# cfgadm -c unconfigure c2::50020f2300006107
```

## 4 Verify that the selected path `c2::50020f2300006107` is unconfigured.

```
# cfgadm -al
Ap_Id                Type      Receptacle  Occupant  Condition
c0                   fc-fabric connected  configured unknown
c0::50020f2300006077 disk      connected  configured unknown
c0::50020f23000063a9 disk      connected  configured unknown
c1                   fc-private connected  configured unknown
c1::220203708b69c32b disk      connected  configured unknown
c1::220203708ba7d832 disk      connected  configured unknown
c2                   fc-fabric connected  configured unknown
c2::50020f2300005f24 disk      connected  configured unknown
c2::50020f2300006107 disk      connected  unconfigured unknown
```

The devices associated with that Ap\_Id are still available to a system through the other path, represented by `c0::50020f2300006077`. A device can be connected to multiple Ap\_Ids and an Ap\_Id can be connected to multiple devices.

```
/dev/rdisk/c6t60020F20000061073AC8B52D000B74A3d0s2
and
/dev/rdisk/c6t60020F20000061073AC8B4C50004ED3Ad0s2
```



## ▼ How to Unconfigure All Multipathed FC Devices

An Ap\_Id on a fabric-connected host port is a path to a multipathed device.

When a multipathed device has multiple Ap\_Ids connected to it, the device is still available to the system after you unconfigure an Ap\_Id. After you unconfigure the last Ap\_Id, no additional paths remain and the device is unavailable to the system. Only devices on a fabric-connected host port can be unconfigured.

- 1 Become an administrator
- 2 Identify the devices to be unconfigured.

```
# cfgadm -al
Ap_Id                Type      Receptacle  Occupant  Condition
c0                   fc-fabric connected   configured unknown
c0::50020f2300006077 disk      connected   configured unknown
c0::50020f23000063a9 disk      connected   configured unknown
c1                   fc-private connected   configured unknown
c1::220203708b69c32b disk      connected   configured unknown
c1::220203708b9b20b2 disk      connected   configured unknown
c2                   fc-fabric connected   configured unknown
c2::50020f2300005f24 disk      connected   configured unknown
```

- 3 Unconfigure all of the configured devices on the selected port.

```
# cfgadm -c unconfigure c2
```

---

**Note** – This operation repeats the unconfigure command of an individual device for all devices on c2. This process can be time-consuming if the number of devices on c2 is large.

---

- 4 Verify that all devices on c2 are unconfigured.

```
# cfgadm -al
Ap_Id                Type      Receptacle  Occupant  Condition
c0                   fc-fabric connected   configured unknown
c0::50020f2300006077 disk      connected   configured unknown
c1                   fc-private connected   configured unknown
c1::220203708b69c32b disk      connected   configured unknown
c1::220203708ba7d832 disk      connected   configured unknown
c2                   fc-fabric connected   unconfigured unknown
c2::50020f2300005f24 disk      connected   unconfigured unknown
c2::50020f2300006107 disk      connected   unconfigured unknown
```

Notice that the Occupant column lists c2 and all the devices attached to c2 as unconfigured.



# Supported FC-HBA API

---

This appendix contains the list of supported and unsupported FC-HBA Interfaces. For further information regarding the API, refer to [“Overview of Solaris I/O Multipathing” on page 13](#).

## Supported Fibre Channel HBA API

TABLE B-1 Supported and Unsupported FC-HBA Interfaces

FC - HBA Interfaces	Solaris I/O Multipathing Support
HBA_GetVersion	Supported
HBA_LoadLibrary	Supported
HBA_FreeLibrary	Supported
HBA_GetNumberOfAdapters	Supported
HBA_GetAdapterName	Supported
HBA_OpenAdapter	Supported
HBA_CloseAdapter	Supported
HBA_GetAdapterAttributes	Supported
HBA_GetAdapterPortAttributes	Supported
HBA_GetDiscoveredPortAttributes	Supported
HBA_GetPortAttributesbyWWN	Supported
HBA_SendCTPassThru	Supported
HBA_SendCTPassThruV2	Supported
HBA_RefreshInformation	Supported

TABLE B-1 Supported and Unsupported FC-HBA Interfaces (Continued)

FC-HBA Interfaces	Solaris I/O Multipathing Support
HBA_GetFcpTargetMapping	Supported
HBA_SendScsiInquiry	Supported
HBA_SendReportLuns	Supported
HBA_SendReadCapacity	Supported
HBA_GetPortStatistics	Supported
HBA_ResetStatistics	Not supported
HBA_GetFcpPersistentBinding	Not supported
HBA_GetEventBuffer	Not supported
HBA_SetRNIDMgmtInfo	Supported
HBA_GetRNIDMgmtInfo	Supported
HBA_SendRNID	Supported
HBA_SendRNIDV2	Supported
HBA_ScsiInquiryV2	Supported
HBA_ScsiReportLUNsV2	Supported
HBA_ScsiReadCapacityV2	Supported
HBA_OpenAdapterByWWN	Supported
HBA_RefreshAdapterConfiguration	Supported
HBA_GetVendorLibraryAttributes	Supported
HBA_GetFcpTargetMappingV2	Supported
HBA_SendRPL	Not supported
HBA_SendRPS	Not supported
HBA_SendSRL	Not supported
HBA_SendLIRR	Not supported
HBA_SendRLS	Supported
HBA_RemoveCallback	Supported
HBA_RegisterForAdapterEvents	Supported
HBA_RegisterForAdapterAddEvents	Supported
HBA_RegisterForAdapterPortEvents	Supported

TABLE B-1 Supported and Unsupported FC-HBA Interfaces (Continued)

FC - HBA Interfaces	Solaris I/O Multipathing Support
HBA_RegisterForAdapterPortStatEvents	Not supported
HBA_RegisterForTargetEvents	Supported
HBA_RegisterForAdapterLinkEvents	Not supported
HBA_RegisterForAdapterTargetEvents	Supported
HBA_GetFC4Statistics	Not supported
HBA_GetFCPStatistics	Not supported
HBA_GetBindingCapability	Not supported
HBA_GetBindingSupport	Not supported
HBA_SetBindingSupport	Not supported
HBA_SetPersistentBindingV2	Not supported
HBA_GetPersistentBindingV2	Not supported
HBA_RemovePersistentBinding	Not supported
HBA_RemoveAllPersistentBindings	Not supported



# Troubleshooting Multipath Device Problems

---

This appendix provides solutions to potential problems that might occur while running the multipathing features.

This appendix contains the following sections:

- [“System Fails to Boot During stmsboot” on page 103](#)
- [“System Crashes During stmsboot” on page 104](#)

## System Fails to Boot During stmsboot

Perform the following steps to recover in single-user mode if the system fails to boot after a stmsboot enable (-e), disable (-d), or update (-u) operation.

### ▼ How to Recover Boot Failure in Single User Mode

- 1 **Type the root password and enter single user mode.**
- 2 **Restart the mpxio-upgrade service.**

```
# svcadm restart svc:/system/device/mpxio-upgrade:default
```

If this command is not successful, follow the instructions in the next section to recover your original configuration.

# System Crashes During stmsboot

Perform the following steps to recover your original multipathed configuration if your system hangs, panics, or fails to boot after a `stmsboot enable (-e)`, `stmsboot disable (-d)`, or `stmsboot update (-u)` operation.

## ▼ How to Recover from a System Crash

### 1 Boot the system from another bootable disk, installation DVD, or over the network.

- SPARC: If you boot from installation media or an installation server on the network, select the text installation. If you boot from an installation server, use the following command:

```
ok boot net:dhcp
```

- x86: If you boot from installation media or an installation server on the network, select this text installation option from the GRUB menu:

```
Oracle Solaris 11 11/11 Text Installer and command line
```

- Select option 3 Shell from the following menu:

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently sun-color)
5 Reboot
```

```
Please enter a number [1]: 3
To return to the main menu, exit the shell
```

### 2 Import your ZFS root pool.

```
# zpool import -f rpool
```

### 3 Mount the root BE.

```
# mkdir /a
# beadm mount solaris /a
```

### 4 Restore your original `fp.conf` file (for FC multipathing) or `mpt.conf` (for SAS multipathing) as follows.

- If you ran the `stmsboot -e` command or `stmsboot -d` command:

- For SAS multipathing:

```
# cp /a/etc/mpxio/mpt.conf /a/etc/driver/drv/mpt.conf
```

- For FC multipathing:

```
# cp /a/etc/mpxio/fp.conf /a/etc/driver/drv/fp.conf
```



- If you ran the `stmsboot -u` command, and you modified either the `fp.conf` file or the `mpt.conf` undo the changes you made to this file by editing either the `/a/etc/driver/drv/fp.conf` or the `/a/etc/driver/drv/mpt.conf` files.

**5 Undo any other changes you made to the multipathing configuration prior to running the `stmsboot` command.**

For example, if you modified the `scsi_vhci.conf` file, undo the changes you made to this file by editing the `/a/etc/driver/drv/scsi_vhci.conf` file.

If you modified the device settings of your storage arrays, restore their original settings.

**6 Restore your original `/etc/vfstab` file:**

```
# cp /a/etc/mpxio/vfstab /a/etc/vfstab
```

The `/a/etc/mpxio/vfstab` file is a copy your original `/etc/vfstab` file that the `stmsboot` command saved prior to updating your `vfstab` file. A `/a/etc/mpxio/vfstab` file will not exist if the `stmsboot` command has not modified your `vfstab` file.

**7 If the system is running on the Oracle Solaris OS on an x86 based system, perform the following steps:**

**a. Restore your original `/boot/solaris/bootenv.rc` file.**

```
# cp /a/etc/mpxio/bootenv.rc /a/boot/solaris/bootenv.rc
```

The `/a/etc/mpxio/bootenv.rc` file is a copy your original `/boot/solaris/bootenv.rc` file that the `stmsboot` command saved prior to updating your `bootenv.rc` file. A `/a/etc/mpxio/bootenv.rc` file will not exist if the `stmsboot` command has not modified your `bootenv.rc` file.

**b. Update the boot archive.**

```
# bootadm update-archive -R /a
```

**8 Disable the `mpxio-upgrade` service:**

```
# /usr/sbin/svccfg -f /a/etc/mpxio/svccfg_recover
```

**9 Unmount the BE.**

```
# beadm umount solaris
```

**10 Reboot the system.**



# Index

---

## Numbers and Symbols

24-bit FC addressing devices, 84

## A

ANSI standard Multipath Management API, 33–47

Ap\_Id, 85

automatic failback, configuration, 31

## B

boot disk, 51

booting, 14

broadcasting, 65

## C

cfgadm

-c, 87

-l, 85, 86

cfgadm -al -o show\_SCSI\_LUN, 85

cfgadm -c unconfigure, 95

cfgadm(1M), 85

cfgadm configure, 86

cfgadm\_fp(1M), 89

configuration

device, 85

dump, 22

fabric devices, 86

FCoE, 58–60

configuration (*Continued*)

Fibre Channel overview, 19–20

manual, 83–84

multipathing, 21–22

SAS devices, 64

tasks, 19

configure

automatic failback, 31

fabric-connected devices, 49–51

multipathing feature, 21, 49–51

NPIV ports, 53–56

SAN devices, 49–51

SAS devices, 64

third-party devices, 27–28, 28

virtual Fibre Channel ports, 53–56

considerations

device name changes, 22

device specific, 22

fabric boot devices, 51

IPFC, 65–68

multipathing, 21–22

per-port configuration, 25–27

SAN devices, 49–50

SAS devices, 63

stmsboot command, 20

StorageTek Traffic Manager, 20

third-party device configuration, 27–28

**D**

## device

- asymmetrical, 16
- configuration, 86
- storage, 16

## device management, 17

## devices, persistent naming, tape, 13

## disabling, multipathing, 22–25

## discovery

- dynamic, 63
- SAS devices, 63

**E**

## enabling, multipathing, 22–25

**F**

## fabric-connected host ports, 85, 89

## fabric device configuration, 86

## fc-fabric, 90

## FCAL, 13

## FCoE

- configuring ports, 58–60
- creating ports, 58
- deleting ports, 59
- limitations, 57
- reporting status on ports, 59–60

## Fibre Channel

- arbitrated loop, 13
- booting Oracle Solaris x86-based systems, 71–78
- configuration overview, 19–20
- configuring IPFC SAN devices, 65–69
- determining adapter port instances, 66–68
- enabling and disabling multipathing, 22–25
- HBA, 14
- IPFC considerations, 65–68
- listing LUN level information, 85–86
- multipathing configuration, 21–22
- supported HBA interfaces, 99–101

## Fibre Channel ports, configuring, 53–56

**I**

## initiator ports, displaying properties, 34

Internet Protocol over Fibre Channel, *See* IPFC

## IPFC, considerations, 65–68

**L**

## limitations, FCoE, 57

## LUN

- masking, 49
- recognition, 50
- SCSI devices, 92

## luxadm(1M), 31

**M**

## modinfo, 85

## mpathadm commands, 33–47

- automatic failback, configuring, 40
- configure automatic failback, 40
- disable a path, 45–47
- enable a path, 44–45
- initiator ports
  - properties, 34
- logical unit with a particular name, 38
- logical units
  - associated with target port, 36
  - fail over, 41–43
  - properties and associated path, 34

## mpt driver, 63

## Multipath Management API plug-in

- library, 33
- properties, 33

## multipathing

- boot considerations, 51
- configuring automatic failback, 31
- configuring device nodes, 89–92
- configuring third-party storage devices, 27–30
- disabled, device node configuration, 86
- disabled, multiple device configuration, 88–89
- enabling and disabling, 22–25
- enabling and disabling on a per-port basis, 25–27
- fabric device node configuration, 84

---

**multipathing** (*Continued*)

- features, 15
  - manual configuration, 83–84
  - mpathadm command, 33–47
  - SAS boot device configuration, 64
  - SAS considerations, 63
  - support, 33–47
  - troubleshooting, 103–105
- multipathing, enabling and disabling, 22–25

**N**

- NPIV ports, configuring, 53–56

**O**

- Oracle Solaris x86-based system FC OS booting procedure, 71–78

**P**

- per-port configuration
- considerations, 25–27
  - enabling and disabling multipathing, 25–27
- persistent binding of tape devices, 79–81
- physical device, 87
- Promiscuous mode, 65

**R**

- REPORT LUNS command, 27

**S**

- SAS devices
- configuration, 64
  - dynamic discovery, 63
  - multipathing considerations, 63
- sd driver, 63
- Serial Attached SCSI, *See* SAS

- show\_SCSI\_LUN, 87, 92
- snoop(1M), 65
- ssd driver, 85
- st driver, 85
- stmsboot command, 22
- boot considerations, 51
  - considerations, 20
  - determining device paths, 22
  - enabling and disabling multipathing, 16
  - SAN device considerations, 50
  - system crashes, 104
- storage device, 31

**T**

- T10 standard, 33–47
- T11 standard, 14
- tape device persistent binding, 79–81
- troubleshooting, multipathing, 103–105

