

Managing SAN Devices and Multipathing in Oracle® Solaris 11.2

ORACLE

Part No: E36836-03
December 2014

Copyright © 2009, 2014, 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 END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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.

Copyright © 2009, 2014, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Contents

Using This Documentation	7
1 Solaris I/O Multipathing Overview	9
Overview of Solaris I/O Multipathing	9
Fibre Channel Software Features	9
iSCSI Software Features	10
SAS Software Features	11
Solaris I/O Multipathing Features	11
Supported Device Standards	15
FC Device Configuration Overview	15
Configuring FC Devices With Multipathing Considerations	16
2 Configuring Solaris I/O Multipathing Features	17
Configuring Multipathing I/O Features	17
Multipathing Considerations	17
Enabling and Disabling Multipathing	18
▼ How to Enable Multipathing	19
▼ How to Disable Multipathing	20
▼ How to Determine if Multipathing is Enabled or Disabled	21
Enabling or Disabling Multipathing on a Per-Port Basis	22
Port Configuration Considerations	22
Configuring Third-Party Storage Devices	24
Third-Party Device Configuration Considerations	24
Configuring Third-Party Storage Devices: New Devices	25
Configuring Third-Party Storage Devices: Disabling Devices	26
Displaying Device Name Changes	27
Configuring Automatic Failback	28
▼ How to Configure Automatic Failback	28
Administering Solaris I/O Multipathing	29
Displaying Multipathing Support Information	29

Configuring Automatic Failback for Multipathing Support	36
3 Configuring Fabric-Connected Devices	47
FC Device Considerations	47
Adding FC Devices	48
▼ How to Add a FC Device	48
Configuring Fabric Boot Devices on SPARC	49
Fabric Boot Device Considerations	49
4 Configuring Solaris iSCSI Initiators	51
Oracle Solaris iSCSI Technology (Overview)	51
Identifying Oracle Solaris iSCSI Software and Hardware Requirements	52
Configuring iSCSI Initiators Tasks	53
iSCSI Terminology	53
Recommended iSCSI Configuration Practices	54
Configuring Dynamic or Static Target Discovery	55
▼ How to Configure an iSCSI Initiator	55
▼ How to Access iSCSI Disks	57
▼ How to Remove Discovered Target Devices	58
Configuring Authentication in Your iSCSI-Based Storage Network	59
▼ How to Configure CHAP Authentication for Your iSCSI Initiator	60
▼ How to Configure CHAP Authentication for Your iSCSI Target	62
Using a Third-Party RADIUS Server to Simplify CHAP Management in Your iSCSI Configuration	63
Setting Up iSCSI Multipathed Devices in Oracle Solaris	65
▼ How to Enable Multiple iSCSI Sessions for a Target	66
Monitoring Your iSCSI Configuration	69
▼ How to Display iSCSI Configuration Information	69
Modifying iSCSI Initiator and Target Parameters	71
Tuning iSCSI Parameters	71
▼ How to Modify iSCSI Initiator and Target Parameters	74
Troubleshooting iSCSI Configuration Problems	76
No Connections to the iSCSI Target From the Local System	77
iSCSI Device or Disk Is Not Available on the Local System	78
Use LUN Masking When Using the iSNS Discovery Method	79
General iSCSI Error Messages	79
5 Configuring Virtual Fibre Channel Ports	85
What Is NPIV?	85

Limitations of NPIV	85
Working With NPIV Ports	86
▼ How to Create an NPIV Port	86
▼ How to Delete an NPIV Port	87
▼ How to Display NPIV Port Status	87
6 Configuring FCoE Ports	89
What Is FCoE?	89
Limitations of FCoE	89
Configuring FCoE Ports	90
▼ How to Create an FCoE Port	90
▼ How to Delete an FCoE Port	91
▼ How to Display FCoE Port Status	91
▼ How to Force an FCoE Port Reinitialization	92
Configuring FCoE Hardware Offload	93
7 Configuring SAS Domains	95
SAS Multipathing Considerations	95
Dynamic Discovery of SAS Devices	95
Configuring SAS Boot Devices	96
8 Configuring IPFC SAN Devices	97
IPFC Considerations	97
Determining Fibre Channel Adapter Port Instances	98
Invoking and Configuring IPFC	100
▼ How to Start a Network Interface Manually	101
▼ How to Configure the Host for Automatic Network Configuration	101
9 Booting the Oracle Solaris OS From Fibre Channel Devices on x86 Based Systems	103
Oracle Solaris OS Setup Requirements	103
Oracle Solaris OS Installation Overview	104
Oracle Solaris OS Installation Procedure	105
▼ How to Install the Oracle Solaris OS	105
▼ How to Perform a DVD or Network-Based OS Installation	105
10 Persistent Binding for Tape Devices	111
Persistent Binding Overview	111

Creating Tape Links	112
▼ How to Create Tape Device Links	113
A Manual Configuration for Fabric-Connected Devices	115
Manually Configuring FC Devices	115
▼ How to Manually Configure a FC Device	115
Configuring Fabric Device Nodes	116
Ensuring That LUN Level Information Is Visible	117
Configuring Device Nodes Without Multipathing Enabled	118
▼ How to Manually Configure an FC Device Without Multipathing	119
▼ How to Configure Multiple FC Devices Without Multipathing	120
Configuring Device Nodes With Solaris Multipathing Enabled	121
▼ How to Configure Individual Multipathed FC Devices	122
▼ How to Configure Multiple Multipathed FC Devices	123
Unconfiguring Fabric Devices	125
Unconfiguring a Fabric Device	125
B Supported FC-HBA API	131
Supported Fibre Channel HBA API	131
C Troubleshooting Multipathing-Related Problems	133
System Fails to Boot or Crashes During <code>stmsboot</code>	133
▼ How to Recover Boot Failure in Single User Mode	133
▼ How to Recover from a System Crash	133
D Tuning Disk Target Driver Properties	137
Tunable Parameters for Disk Drivers	137
<i>Name:Value</i> Pair Format to Tune Disk Drivers	138
Bit-Masking Format to Tune Disk Drivers	140
Index	145

Using This Documentation

- **Overview** – Provides an overview of Solaris I/O multipathing features for the Oracle Solaris operating system and describes how to configure Solaris iSCSI Initiators, Fibre Channel over Ethernet (FCoE) ports, storage area network (SAN) devices, and serial-attached SCSI (SAS) Domains
- **Audience** – System, storage and network administrators who create and maintain Fibre Channel (FC) storage area networks (SANs) and serial-attached SCSI (SAS) Domains
- **Required knowledge** – Expertise in the management and maintenance of SANS and SAS Domains

Product Documentation Library

Late-breaking information and known issues for this product are included in the documentation library at <http://www.oracle.com/pls/topic/lookup?ctx=E36784>.

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.

Feedback

Provide feedback about this documentation at <http://www.oracle.com/goto/docfeedback>.

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:

- [“Overview of Solaris I/O Multipathing” on page 9](#)
- [“Supported Device Standards” on page 15](#)

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. This section provides the following information:

- [“Fibre Channel Software Features” on page 9](#)
- [“SAS Software Features” on page 11](#)
- [“Solaris I/O Multipathing Features” on page 11](#)

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, iSCSI, and SAS storage devices.

For information on how to configure targets and initiators in preparation for multipathing, see [Chapter 8, “Configuring Storage Devices With COMSTAR,”](#) in [“Managing Devices in Oracle Solaris 11.2”](#).

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.

For more information about common FC-HBA APIs, see [libhbaapi\(3LIB\)](#). For additional information about FC specifications, go 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 based systems.
- 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.

iSCSI Software Features

iSCSI is an acronym for Internet SCSI (Small Computer System Interface), an Internet Protocol (IP)-based storage networking standard for linking data storage subsystems. By carrying SCSI commands over IP networks, the iSCSI protocol enables you to access block devices from across the network as if they were connected to the local system.

This feature means that a Solaris system can act as either an iSCSI server (*target*) or a client (*initiator*). The advantage of setting up Solaris iSCSI targets is you might have existing fibre-channel devices that can be connected to clients without additional FC HBAs. In addition, systems with dedicated arrays can now share replicated storage with ZFS or UFS file systems.

For more information, see [Chapter 4, “Configuring Solaris iSCSI Initiators”](#).

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 that the OS 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 from vendors other than Oracle, contact the 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 enables 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 based and x86 based systems, and provides support for SAS multipathing.

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

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

X86 based 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.
- **Multipathed device names** – After a system is installed with the Oracle Solaris OS 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
.
```

- Displaying MPxIO path information – The `prtconf` and `fmdump` commands have both 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 `diskinfo` 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:

```
# diskinfo -O occupant-compdev
c0t50015179594B6F11d0
c0t5000C500335F95E3d0
c0t5000C500335F907Fd0
c0t5000C500335BD117d0
c0t5000C500335DC60Fd0
c0t5000C500335E106Bd0
c0t5000C500335BA8C3d0
c0t5000C500335FC3E7d0
```

- 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.

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)

FC Device Configuration 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, a description of each task, and the reference information where you can find related procedures.

TABLE 1-1 Tasks for Configuring Multipathed FC Devices

Configuration Task	Task Description	Reference Information
Enable multipathing features	Multipathing is enabled as follows: <ul style="list-style-type: none"> ■ By default, for FC devices on x86/x64 platforms ■ By manual configuration on SPARC platforms ■ By manual configuration for SAS devices 	Chapter 2, “Configuring Solaris I/O Multipathing Features” .
Configure FC devices	In the Oracle Solaris OS, FCAL, fabric, and point-to-point connected devices are made available to the host.	Chapter 3, “Configuring Fabric-Connected Devices”
Configure virtual FC ports	You can configure N Port ID Virtualization (NPIV) ports, which is especially useful for virtualization environments.	Chapter 5, “Configuring Virtual Fibre Channel Ports”
Set up FC boot device	Solaris I/O multipathing features enable a Solaris server to boot from a fabric device.	“FC Device Considerations” on page 47
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 8, “Configuring IPFC SAN Devices”

Configuration Task	Task Description	Reference Information
Configure SAS devices	In the Oracle Solaris OS, multipathing of SAS devices is supported.	Chapter 7, “Configuring SAS Domains”

Configuring FC Devices With Multipathing 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 automatically updates the `/etc/vfstab` file 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 devices on SPARC based systems and enabled by default for devices on x86 based systems.
- 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, you have to manually unconfigure those devices by using `cfgadm -c unconfigure` commands. However, to make those devices permanently unavailable to the system, you might 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, 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 covered:

- [“Configuring Multipathing I/O Features” on page 17](#)
- [“Multipathing Considerations” on page 17](#)
- [“Enabling and Disabling Multipathing” on page 18](#)
- [“Enabling or Disabling Multipathing on a Per-Port Basis” on page 22](#)
- [“Configuring Third-Party Storage Devices” on page 24](#)
- [“Configuring Automatic Failback” on page 28](#)

Configuring Multipathing I/O Features

You can configure the multipathing features for FC devices 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 based and x86 based systems, the `stmsboot` command 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.
- Because the `stmsboot` command reboots the machine to complete the operation, use the `eprom` to ensure the system boots from the current boot device.

- 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 Multipathing-Related 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 22.

The multipathing software automatically recognizes Oracle-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 17.

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 mpxio status for driver fp
Checking mpxio 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 17.

▼ 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 22.

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 mpxio status for driver fp
Checking mpxio 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 17.

▼ How to Determine if Multipathing is Enabled or Disabled

1. Become an administrator.
2. Determine if multipathed device information is available on the system.

Multipathing is disabled on this system because the following command returns no multipathed device information.

```
# prtconf -vc /devices/scsi_vhci |grep dev_link.*s2
```

Multipathing is enabled on this system because the following command returns multipathed device information.

```

# prtconf -vc /devices/scsi_vhci |grep dev_link.*s2
dev_link=/dev/dsk/c0t5000C500335DC60Fd0s2
dev_link=/dev/rdisk/c0t5000C500335DC60Fd0s2
dev_link=/dev/dsk/c0t5000C500335E106Bd0s2
dev_link=/dev/rdisk/c0t5000C500335E106Bd0s2
dev_link=/dev/dsk/c0t5000C500335BA8C3d0s2
dev_link=/dev/rdisk/c0t5000C500335BA8C3d0s2

```

```
dev_link=/dev/dsk/c0t5000C500335FC3E7d0s2
dev_link=/dev/rdisk/c0t5000C500335FC3E7d0s2
dev_link=/dev/dsk/c0t50015179594B6F11d0s2
dev_link=/dev/rdisk/c0t50015179594B6F11d0s2
dev_link=/dev/dsk/c0t5000C500335F95E3d0s2
dev_link=/dev/rdisk/c0t5000C500335F95E3d0s2
dev_link=/dev/dsk/c0t5000C500335F907Fd0s2
dev_link=/dev/rdisk/c0t5000C500335F907Fd0s2
dev_link=/dev/dsk/c0t5000C500335BD117d0s2
dev_link=/dev/rdisk/c0t5000C500335BD117d0s2
```

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.

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 `/kernel/drv/fp.conf` file. Per-port multipath settings have priority over the global setting. Therefore, 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 the `/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

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

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.**

To list the available devices, perform an `ls -l` command on the `/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.

Determine the port or ports for which you want to explicitly enable or disable multipathing.

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:**
 - For each FC HBA port that is to be enabled in the `/etc/driver/drv/fp.conf`, add the following line:

```
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@6,2000/SUNW,qlc@2" port=0 mpxio-disable="no";
```

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

- For each FC HBA port that is to be disabled, add the following line:

```
name="fp" parent="parent-name" port=port-number mpzio-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 17](#).**

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 is for the `scsi_vhci` code to automatically 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 one 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 (a comma) and the last vendor/product entry is terminated by the ';' character (a semicolon).

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

```
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 18](#).**

Configuring Third-Party Storage Devices: Disabling Devices

Multipathing can be disabled for all devices of a certain vendor ID/product ID combination. 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. **Add the vendor ID and product ID entries to the `/etc/driver/drv/scsi_vhci.conf` file.**

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 (a comma) and the last vendor/product entry is terminated by the ';' character (a semicolon). 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 18](#).**

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 to keep 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. **In the `/kernel/drv/scsi_vhci.conf` file, enable or disable automatic failback capability by changing the `auto-failback` entry.**

```
auto-failback="enable";
```

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

4. **Save and exit the file.**
5. **Reboot the system.**

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

Administering Solaris I/O Multipathing

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 LU are consistent with the T10 specification.

Perform the following tasks to administer multipathing devices.

- [“Displaying Multipathing Support Information” on page 29](#)
- [“Configuring Automatic Failback for Multipathing Support” on page 36](#)

Displaying Multipathing Support Information

You can use the `mpathadm` command to display multipathing support information and also manage multipathing discovery. Multipathing support and property information is identified with the Multipath Management API plug-in library name, which is displayed by using the `mpathadm` command.

▼ How to Display Multipathing Support Information

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` library file 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. List the initiator ports.

```
# 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
```

If you do not specify the initiator port with the `mpathadm show initiator-port` command, the properties of all the discovered initiator ports are displayed.

▼ How to Display Specific LU Information

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

1. Display a list of multipathed LUs.

```
# 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 LU using the LU name from the list.

```
# 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
```

If you do not specify the LU name with the `mpathadm show lu` command, the information about all the discovered logical units are displayed.

▼ How to Display All LUs Associated With a Specific Target Port

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

1. Display a list of LUs.

```
# 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 LU 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/rdisk/c0t600A0B800026D63A0000A4994E2342D4d0s2
                 Total Path Count: 4
                 Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
                 /dev/rdisk/c0t600A0B800029065C00007CF54E234013d0s2
                 Total Path Count: 4
                 Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
                 /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
                 Total Path Count: 4
                 Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
                 /dev/rdisk/c0t600A0B800029065C00007CF44E233FCFd0s2
                 Total Path Count: 4
                 Operational Path Count: 4
.
.
.
```

▼ How to Display a LU With a Specific Name

Display detailed information on multipathed LUs, including path and target port group information. Note that the name property in the information represents the identifier for this LU, 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 LUs.

```
# 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
.
.
.

```

2. Display a selected LU's configuration information.

```

# mpathadm show lu /dev/rdisk/c0t600A0B800026D63A0000A4994E2342D4d0s2
Logical Unit: /dev/rdisk/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
```

You can specify the `-e` or `--path-missing` option with the `mpathadm show lu` command to display the logical units only for missing paths.

3. Display the selected LU information.

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

Configuring Automatic Failback for Multipathing Support

Symmetrical devices can provide 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 automatic failback is enabled, the initial path comes back online and failover to the initial path automatically occurs.

▼ How to Configure Automatic Failback for Specific Multipathing Support

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 automatic 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/drv/scsi_vhci.conf` file. Refer to [“Configuring Automatic Failback” on page 28](#).

▼ How to Fail Over a LU

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 LU access

1. Display a list of multipathed LUs.

```
# 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 a specific LU'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 LU failover.

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

If this operation is successful, the access state of the device's 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 LU Path

If the path to a LU 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 LU. To verify the change, run the `show` command for the logical unit.

1. Display a list of multipathed LUs.

```

# 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
.
.
.

```

2. Display the selected LU's configuration information.

```

# mpathadm show lu
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
```

3. Enable the LU path.

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

▼ How to Disable a LU Path

This operation disables the LU path, 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 LUs.

```
# 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
.
.
.
```

2. Display the specific LU's configuration information.

```
# mpathadm show lu /dev/rdisk/c0t600A0B800029065C00007CF34E233F89d0s2
Logical Unit: /dev/rdisk/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 LU path.**

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


Configuring Fabric-Connected Devices

This chapter provides a high level overview of the steps used to configure fabric-connected (FC) devices. In the Oracle Solaris OS, FCAL, fabric, and point-to-point connected devices are made available to the system automatically. If you want to manually configure fabric-connected devices, see [Appendix A, “Manual Configuration for Fabric-Connected Devices”](#).

The following topics are covered:

- [“FC Device Considerations” on page 47](#)
- [“Adding FC Devices” on page 48](#)
- [“Configuring Fabric Boot Devices on SPARC” on page 49](#)

FC 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 are required to manually configure devices.
- LUN masking enables specific LUNs to be seen by specific systems. See the 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 activates 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 devices on SPARC based systems and enabled by default for devices on x86 based systems.

Adding FC Devices

The following table lists the common commands used when adding and removing FC devices.

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.	cfgadm(1M) and cfgadm_fp(1M)
<code>fcinfo</code>	Collects administrative information on fibre channel Host Bus Adapters (HBA) ports on a host.	fcinfo(1M)
<code>format</code>	Identifies devices that are connected to the system and provides the ability to label a disk.	format(1M)
<code>luxadm</code>	Administers storage devices and FC_AL devices.	luxadm(1M)

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 FC Device

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

Configuring Fabric Boot Devices on SPARC

You can set up 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 a target or LUN
 - Reducing physical distance between the 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 2, “Configuring Solaris I/O Multipathing Features”](#).

Configuring Solaris iSCSI Initiators

This chapter describes how to configure Solaris iSCSI initiators in the Oracle Solaris release. For information about the procedures associated with configuring iSCSI initiators, see [“Configuring iSCSI Initiators Tasks”](#) on page 53.

The following topics are covered:

- [“Oracle Solaris iSCSI Technology \(Overview\)”](#) on page 51
- [“Recommended iSCSI Configuration Practices”](#) on page 54
- [“Configuring iSCSI Initiators Tasks”](#) on page 53
- [“Configuring Authentication in Your iSCSI-Based Storage Network”](#) on page 59
- [“Setting Up iSCSI Multipathed Devices in Oracle Solaris”](#) on page 65
- [“Monitoring Your iSCSI Configuration”](#) on page 69
- [“Modifying iSCSI Initiator and Target Parameters”](#) on page 71
- [“Troubleshooting iSCSI Configuration Problems”](#) on page 76

For information about configuring Oracle Solaris iSCSI targets with COMSTAR, see [Chapter 8](#), [“Configuring Storage Devices With COMSTAR,”](#) in [“Managing Devices in Oracle Solaris 11.2”](#).

For more information about installing and booting an iSCSI disk, see Step 7 in [“How to Perform a GUI Installation”](#) in [“Installing Oracle Solaris 11.2 Systems”](#).

Oracle Solaris iSCSI Technology (Overview)

iSCSI is an acronym for Internet SCSI (Small Computer System Interface), an Internet Protocol (IP)-based storage networking standard for linking data storage subsystems.

By carrying SCSI commands over IP networks, the iSCSI protocol enables you to access block devices from across the network as if they were connected to the local system.

The following solutions are available to use storage devices in your existing TCP/IP network:

- iSCSI block devices or tape – Translates SCSI commands and data from the block level into IP packets. Using iSCSI in your network is advantageous when you need to have

block-level access between one system and the target device, such as a tape device or a database. Access to a block-level device is not locked so that you can have multiple users or systems accessing a block-level device such as an iSCSI target device.

- NFS – Transfers file data over IP. The advantage of using NFS in your network is that you can share file data across many systems. Access to file data is locked appropriately when many users are accessing data that is available in an NFS environment.

Here are the benefits of using iSCSI targets and initiators in Oracle Solaris:

- The iSCSI protocol runs across existing Ethernet networks.
 - You can use any supported network interface card (NIC), Ethernet hub, or Ethernet switch.
 - One IP port can handle multiple iSCSI target devices.
 - You can use existing infrastructure and management tools for IP networks.
- You might have existing Fibre-Channel devices that can be connected to clients without the cost of Fibre-Channel HBAs. In addition, systems with dedicated arrays can now export replicated storage with Oracle Solaris ZFS or UFS file systems.
- The protocol can be used to connect to Fibre Channel or iSCSI Storage Area Network (SAN) environments with the appropriate hardware.

Here are the current limitations or restrictions of using the iSCSI initiator software in Oracle Solaris:

- Support for iSCSI devices that use SLP is not currently available.
- iSCSI targets cannot be configured as dump devices.
- Transferring large amounts of data over your existing network can have an impact on performance.
- The iSCSI initiator subsystem runs only on the global zone and not on the non-global zones. The workaround to access the iSCSI devices from within a non-global zone is to configure an iSCSI initiator in a global zone and add the device to a non-global zone by using the `add device` command during zone configuration.

Identifying Oracle Solaris iSCSI Software and Hardware Requirements

- Oracle Solaris storage software and devices
 - The `group/feature/storage-server` software package for the system that provides the storage devices
 - The `system/storage/iscsi/iscsi-initiator` software package for the iSCSI management utilities
- Any supported NIC

Configuring iSCSI Initiators Tasks

This is a general list of tasks associated with configuring iSCSI initiators. Some of the tasks are optional depending on your network configuration needs. Some of the links below will take you to separate documents that describe network configuration and target configuration.

- [“Identifying Oracle Solaris iSCSI Software and Hardware Requirements” on page 52](#)
- [“Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.2 ”](#)
- [Chapter 8, “Configuring Storage Devices With COMSTAR,” in “Managing Devices in Oracle Solaris 11.2 ”](#)
- [“Configuring Dynamic or Static Target Discovery” on page 55](#)
- [“How to Configure an iSCSI Initiator” on page 55](#)
- [“How to Access iSCSI Disks” on page 57](#)
- [“Configuring Authentication in Your iSCSI-Based Storage Network” on page 59](#)
- [“Setting Up iSCSI Multipathed Devices in Oracle Solaris” on page 65](#)
- [“Monitoring Your iSCSI Configuration” on page 69](#)

iSCSI Terminology

Review the following terminology before configuring iSCSI targets and initiators.

Term	Description
Discovery	The process that presents the initiator with a list of available targets.
Discovery method	The way in which the iSCSI targets can be found. Three methods are currently available: <ul style="list-style-type: none"> ■ Internet Storage Name Service (iSNS) – Potential targets are discovered by interacting with one or more iSNS servers. ■ SendTargets – Potential targets are discovered by using a <i>discovery-address</i>. ■ Static – Static target addressing is configured.
Initiator	The driver that initiates SCSI requests to the iSCSI target.
Initiator group	A set of initiators. When an initiator group is associated with a LU, only initiators from that group may access the LU.
iqn or eui address format	An <i>iqn</i> (iSCSI qualified name) address is the unique identifier for a device in an iSCSI network using the form <i>iqn.date.authority:uniqueid</i> . An iSCSI initiator or target is assigned an IQN name automatically when the iSCSI initiator or target is initialized. An <i>eui</i> (extended unique identifier) address consists of 16 hexadecimal digits, and identifies a class of GUIDs that is used in both the SCSI and InfiniBand standards. SRP devices use the <i>eui</i> address format.

Term	Description
Logical unit (LU)	A uniquely numbered component in a storage system. When a LU is associated with one or more SCSI targets, the target can be accessed by one or more SCSI initiators. A logical unit is identified by a logical unit number (LUN).
Target device	The iSCSI storage component.
Target group	A set of targets. A LU can be made available to all targets in one target group.
Target portal group	A list of IP addresses that determines which interfaces a specific iSCSI target will listen to. A TPG contains IP addresses and TCP port numbers.

Recommended iSCSI Configuration Practices

Review the following iSCSI recommendations before configuring iSCSI devices in your network.

- **Devices**
 - Consider using multipathed device paths for increased availability.
 - Multiple connections per session (MCS) support allow multiple TCP/IP connections from the initiator to the target for the same iSCSI session.
- **Network**
 - Deploy iSCSI devices in a fast (gigE or better), dedicated network.
 - Use jumbo frames, if possible to allow more data to be transferred in each Ethernet transaction to reduce the number of frames.
 - Use CAT6 rated cables for Gigabit network infrastructures.
 - Segregate iSCSI storage networks from your local area network traffic.
 - Configure multiple sessions or connections to utilize multiple threads in TCP/IP stack.
 - Consider TCP tuning, such as disabling Nagle algorithm.
- **Security**
 - In addition to physical security, use CHAP authentication, which ensures that each host has its own password.
 - Consider using iSNS target discovery domains, which enhance security by providing access control to targets that are not enabled with their own access controls, while limiting the logon process of each initiator to a relevant subset of the available targets in the network.

Configuring Dynamic or Static Target Discovery

Determine whether you want to configure one of the dynamic device discovery methods or use static iSCSI initiator targets to perform device discovery.

- **Dynamic device discovery** – Two dynamic device discovery methods are available:
 - **SendTargets** – If an iSCSI node exposes a large number of targets, such as an iSCSI to Fibre-Channel bridge, you can supply the iSCSI node IP address/port combination and allow the iSCSI initiator to use the SendTargets features to perform the device discovery.
 - **iSNS** – The Internet Storage Name Service (iSNS) allows the iSCSI initiator to discover the targets to which it has access using as little configuration information as possible. It also provides state change notification to notify the iSCSI initiator when changes in the operational state of storage nodes occur. To use the iSNS discovery method, you can supply the iSNS server address/port combination and allow the iSCSI initiator to query the iSNS servers that you specified to perform the device discovery. The default port for the iSNS server is 3205. For more information about iSNS, see RFC 4171:

<http://www.ietf.org/rfc/rfc4171.txt>

The iSNS discovery service provides an administrative model to discover all targets on a network.

For more information about setting up iSNS support in Oracle Solaris, see [Chapter 9, “Configuring and Managing the Oracle Solaris Internet Storage Name Service \(iSNS\),”](#) in “Managing Devices in Oracle Solaris 11.2”.

- **Static device discovery** – If an iSCSI node has few targets or if you want to restrict the targets that the initiator attempts to access, you can statically configure the *target-name* by using the following static target address naming convention:

target,target-address[:port-number]

You can determine the static target address from the array's management tool.

Note - Do not configure an iSCSI target to be discovered by both static and dynamic device discovery methods. The consequence of using redundant discovery methods might be slow performance when the initiator is communicating with the iSCSI target device.

▼ How to Configure an iSCSI Initiator

Part of the initiator configuration process is to identify the iSCSI target discovery method, which presents an initiator with a list of available targets. You can configure iSCSI targets for static, SendTargets, or iSNS dynamic discovery. Dynamic discovery using the SendTargets option is the optimum configuration for an iSCSI initiator that accesses a large number of

targets, such over an iSCSI to Fibre Channel bridge. SendTargets dynamic discovery requires the IP address and port combination of the iSCSI target for the iSCSI initiator to perform the target discovery. The most common discovery method is SendTargets.

When configuring the target discovery method, you must provide the following information, depending on which method you choose:

- SendTargets – Target IP address
- iSNS – iSNS server address
- Static – Target IP address and target name

For more information about configuring target discovery methods, see [“Configuring Dynamic or Static Target Discovery” on page 55.](#)

1. Enable the iSCSI initiator service.

```
initiator# svcadm enable network/iscsi/initiator
```

2. Verify the target's name and IP address while logged in to the server that is providing the target.

```
target# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4       static    ok         127.0.0.1/8
e1000g0/_b   dhcp     ok         1.2.3.4/24
lo0/v6       static    ok         ::1/128
e1000g0/_a   addrconf ok         fe80::123:1234:fe27:360c/10
target# itadm list-target -v
TARGET NAME                                STATE  SESSIONS
iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-c3b36c123456  online  0
  alias:                                     -
  auth:                                       none (defaults)
  targetchapuser:                            -
  targetchapsecret:                          unset
  tpg-tags:                                   default
```

3. Configure the target to be statically discovered.

```
initiator# iscsiadm add static-config iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-c3b36c123456,
1.2.3.4
```

4. Review the static configuration information.

```
initiator# iscsiadm list static-config
Static Configuration Target: iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-c3b36c123456,
1.2.3.4:3260
```

The iSCSI connection is not initiated until the discovery method is enabled. See the next step.

5. Configure one of the following target discovery methods:

- If you have configured a dynamically discovered (SendTargets) target, configure the SendTargets discovery method.

```
initiator# iscsiadm add discovery-address 1.2.3.4
```

- If you have configured a dynamically discovered (iSNS) target, configure the iSNS discovery method.

```
initiator# iscsiadm add isns-server 1.2.3.4
```

6. Enable one of the following the target discovery methods:

- If you have configured a dynamically discovered (SendTargets) target, enable the SendTargets discovery method.

```
initiator# iscsiadm modify discovery --sendtargets enable
```

- If you have configured a dynamically discovered (iSNS) target, enable the iSNS discovery method.

```
initiator# iscsiadm modify discovery --iSNS enable
```

- If you have configured static targets, enable the static target discovery method.

```
initiator# iscsiadm modify discovery --static enable
```

7. Reconfigure the /dev namespace to recognize the iSCSI disk, if necessary.

```
initiator# devfsadm -i iscsi
```

▼ How to Access iSCSI Disks

After the devices have been discovered by the Oracle Solaris iSCSI initiator, the login negotiation occurs automatically. The Oracle Solaris iSCSI driver determines the number of available LUs and creates the device nodes. Then, the iSCSI devices can be treated as any other SCSI device.

You can create a ZFS storage pool on the LU and then create a ZFS file system.

You can view the iSCSI disks on the local system by using the `format` utility.

1. Review the iSCSI LU information in the `format` output.

```
initiator# format
0. c0t600144F0B5418B0000004DDAC7C10001d0 <SUN-COMSTAR-1.0 cyl 1022 alt 2 hd 128 sec 32>
   /scsi_vhci/disk@g600144f0b5418b0000004ddac7c10001
1. c8t0d0 <Sun-STK RAID INT-V1.0 cyl 17830 alt 2 hd 255 sec 63>
```

```
    /pci@0,0/pci10de,375@f/pci108e,286@0/disk@0,0
2. c8t1d0 <Sun-STK RAID INT-V1.0-136.61GB>
    /pci@0,0/pci10de,375@f/pci108e,286@0/disk@1,0
3. c8t2d0 <Sun-STK RAID INT-V1.0-136.61GB>
    /pci@0,0/pci10de,375@f/pci108e,286@0/disk@2,0
4. c8t3d0 <Sun-STK RAID INT-V1.0 cyl 17830 alt 2 hd 255 sec 63>
    /pci@0,0/pci10de,375@f/pci108e,286@0/disk@3,0
Specify disk (enter its number): 0
selecting c0t600144F0B5418B000004DDAC7C10001d0
[disk formatted]
```

In the above output, disk 0 is an iSCSI LU under MPxIO control. Disks 1-4 are local disks.

2. You can create a ZFS storage pool and ZFS file systems on the iSCSI LU.

```
initiator# zpool create pool-name c0t600144F0B5418B000004DDAC7C10001d0
initiator# zfs create pool-name/fs-name
```

The ZFS file system is automatically mounted when created and is remounted at boot time.

▼ How to Remove Discovered Target Devices

The associated targets are logged out after you perform any of the following actions:

- Remove a discovery address
- Remove an iSNS server
- Remove a static configuration
- Disable a discovery method

If these associated targets are still in use, for example, they have mounted file systems, the logout of these devices will fail, and they will remain on the active target list.

This optional procedure assumes that you are logged in to the local system where access to an target device has already been configured.

1. **Become an administrator.**
2. **(Optional) Disable an iSCSI target discovery method by using one of the following:**
 - If you need to disable the SendTargets discovery method, use the following command:

```
initiator# iscsiadm modify discovery --sendtargets disable
```
 - If you need to disable the iSNS discovery method, use the following command:

```
initiator# iscsiadm modify discovery --iSNS disable
```
 - If you need to disable the static target discovery method, use the following command:

```
initiator# iscsiadm modify discovery --static disable
```

3. Remove an iSCSI device discovery entry by using one of the following:

- Remove an iSCSI SendTargets discovery entry.

For example:

```
initiator# iscsiadm remove discovery-address 1.2.3.4:3260
```

- Remove an iSCSI iSNS discovery entry.

For example:

```
# iscsiadm remove isns-server 1.2.3.4:3205
```

- Remove a static iSCSI discovery entry.

For example:

```
initiator# iscsiadm remove static-config eui.5000ABCD78945E2B,1.2.3.4
```

Note - If you attempt to disable or remove a discovery entry that has an associated logical unit (LU) in use, the disable or remove operation fails with the following message:

```
logical unit in use
```

If this error occurs, stop all associated I/O on the LU, unmount the file systems. Then, repeat the disable or remove operation.

Configuring Authentication in Your iSCSI-Based Storage Network

Setting up authentication for your iSCSI devices is optional.

In a secure environment, authentication is not required because only trusted initiators can access the targets.

In a less secure environment, the target cannot determine if a connection request is truly from a given host. In that case, the target can authenticate an initiator by using the Challenge-Handshake Authentication Protocol (CHAP).

CHAP authentication uses the notion of a challenge and response, which means that the target challenges the initiator to prove its identity. For the challenge/response method to work, the target must know the initiator's secret key, and the initiator must be set up to respond to a challenge. Refer to the array vendor's documentation for instructions on setting up the secret key on the array.

iSCSI supports unidirectional and bidirectional authentication as follows:

- *Unidirectional authentication* enables the target to authenticate the identity of the initiator. Unidirectional authentication is done on behalf of the target to authenticate the initiator.
- *Bidirectional authentication* adds a second level of security by enabling the initiator to authenticate the identity of the target. Bidirectional authentication is driven from the initiator, which controls whether bidirectional authentication is performed. The only setup required for the target is that the chap user and chap secret must be correctly defined.

▼ How to Configure CHAP Authentication for Your iSCSI Initiator

This procedure assumes that you are logged in to the local system where you want to securely access the configured iSCSI target device.

- The length of the CHAP secret key for the COMSTAR iSCSI target must be a minimum of 12 characters and a maximum of 255 characters. Some initiators support only a shorter maximum length for the secret key.
- Each node identifying itself using CHAP must have both a user name and a password. In the Oracle Solaris OS, the CHAP user name is set to the initiator or target node name (that is, the `iqn` name) by default. The CHAP user name can be set to any length of text that is less than 512 bytes. The 512-byte length limit is an Oracle Solaris limitation. However, if you do not set the CHAP user name, it is set to the node name upon initialization.

You can simplify CHAP secret key management by using a third-party RADIUS server, which acts as a centralized authentication service. When you use RADIUS, the RADIUS server stores the set of node names and matching CHAP secret keys. The system performing the authentication forwards the node name of the requester and the supplied secret of the requester to the RADIUS server. The RADIUS server confirms whether the secret key is the appropriate key to authenticate the given node name. Both iSCSI and iSER support the use of a RADIUS server.

For more information about using a third-party RADIUS server, see [“Using a Third-Party RADIUS Server to Simplify CHAP Management in Your iSCSI Configuration”](#) on page 63.

1. Become an administrator.

For more information, see [“Using Your Assigned Administrative Rights”](#) in [“Securing Users and Processes in Oracle Solaris 11.2”](#).

2. Determine whether you want to configure unidirectional or bidirectional CHAP.

- Unidirectional authentication, the default method, enables the target to validate the initiator. Complete steps 3–5 only.

- Bidirectional authentication adds a second level of security by enabling the initiator to authenticate the target. Complete steps 3–9.

3. Unidirectional CHAP: Set the secret key on the initiator.

The following command initiates a dialogue to define the CHAP secret key:

```
initiator# iscsiadm modify initiator-node --CHAP-secret
Enter CHAP secret: *****
Re-enter secret: *****
```

4. (Optional) Unidirectional CHAP: Set the CHAP user name on the initiator.

By default, the initiator's CHAP user name is set to the initiator node name.

Use the following command to use your own initiator CHAP user name:

```
initiator# iscsiadm modify initiator-node --CHAP-name new-CHAP-name
```

5. Unidirectional CHAP – Enable CHAP authentication on the initiator.

```
initiator# iscsiadm modify initiator-node --authentication CHAP
```

CHAP requires that the initiator node have both a user name and a password. The user name is typically used by the target to look up the secret key for the given user name.

6. Select one of the following to enable or disable bidirectional CHAP.

- Enable bidirectional CHAP for connections with the target.

```
initiator# iscsiadm modify target-param -B enable target-iqn
```

- Disable bidirectional CHAP.

```
initiator# iscsiadm modify target-param -B disable target-iqn
```

7. Bidirectional CHAP: Set the authentication method to CHAP for the target.

```
initiator# iscsiadm modify target-param --authentication CHAP target-iqn
```

8. Bidirectional CHAP: Set the target device secret key that identifies the target.

The following command initiates a dialogue to define the CHAP secret key:

```
initiator# iscsiadm modify target-param --CHAP-secret target-iqn
```

9. Bidirectional CHAP: If the target uses an alternate CHAP user name, set the CHAP name that identifies the target.

By default, the target's CHAP name is set to the target name.

You can use the following command to change the target's CHAP name:

```
initiator# iscsiadm modify target-param --CHAP-name target-CHAP-name
```

▼ How to Configure CHAP Authentication for Your iSCSI Target

This procedure assumes that you are logged in to the local system that contains the iSCSI targets.

1. **Become an administrator.**
2. **Determine whether you want to configure unidirectional or bidirectional CHAP.**
 - Unidirectional authentication is the default method. Complete steps 3–5 only.
 - For bidirectional authentication. Complete steps 3–7.
3. **Unidirectional/Bidirectional CHAP: Configure the target to require that initiators identify themselves using CHAP.**

```
target# itadm modify-target -a chap target-iqn
```

4. **Unidirectional/Bidirectional CHAP: Create an initiator context that describes the initiator.**

Create the initiator context with the initiator's full node name and with the initiator's CHAP secret key.

```
target# itadm create-initiator -s initiator-iqn
Enter CHAP secret: *****
Re-enter secret: *****
```

5. **Unidirectional/Bidirectional CHAP: If the initiator uses an alternate CHAP name, then configure the initiator-context with the alternate name.**

```
target# itadm modify-initiator -u initiator-CHAP-name initiator-iqn
```

6. **Bidirectional CHAP: Set the target device secret key that identifies this target.**

```
target# itadm modify-target -s target-iqn
Enter CHAP secret: *****
Re-enter secret: *****
```

7. **(Optional) Bidirectional CHAP: If the target uses an alternate CHAP user name other than the target node name (iqn), modify the target.**

```
target# itadm modify-target -u target-CHAP-name target-iqn
```

Using a Third-Party RADIUS Server to Simplify CHAP Management in Your iSCSI Configuration

You can use a third-party RADIUS server that acts as a centralized authentication service to simplify CHAP key secret management. With this method, the recommended practice is to use the default CHAP name for each initiator node. In the common case when all initiators are using the default CHAP name, you do not have to create initiator contexts on the target.

▼ How to Configure a RADIUS Server for Your iSCSI Target

You can use a third-party RADIUS server that acts as a centralized authentication service to simplify CHAP key secret management. With this method, the recommended practice is to use the default CHAP name for each initiator node. In the common case when all initiators are using the default CHAP name, you do not have to create initiator contexts on the target.

This procedure assumes that you are logged in to the local system where you want to securely access the configured iSCSI target device.

1. **Become an administrator.**
2. **Configure the initiator node with the IP address and the port of the RADIUS server.**

The default port is 1812. This configuration is completed once for all iSCSI targets on the target system.

```
initiator# itadm modify-defaults -r RADIUS-server-IP-address
Enter RADIUS secret: *****
Re-enter secret: *****
```

3. **Configure the shared secret key that is used for communication between the target system and the RADIUS server.**

```
initiator# itadm modify-defaults -d
Enter RADIUS secret: *****
Re-enter secret: *****
```

4. **Configure the target system to require RADIUS authentication.**

This configuration can be performed for an individual target or as a default for all targets.

```
initiator# itadm modify-target -a radius target-iqn
```

5. **Configure the RADIUS server with the following components:**

- The identity of the target node (for example, its IP address)

- The shared secret key that the target node uses to communicate with the RADIUS server
- The initiator's CHAP name (for example, it's iqn name) and the secret key for each initiator that needs to be authenticated

▼ How to Configure a RADIUS Server for Your iSCSI Initiator

You can use a third-party RADIUS server that acts as a centralized authentication service to simplify CHAP secret key management. This setup is only useful when the initiator is requesting bidirectional CHAP authentication. You must still specify the initiator's CHAP secret key, but you are not required to specify the CHAP secret key for each target on an initiator when using bidirectional authentication with a RADIUS server. RADIUS can be independently configured on either the initiator or the target. The initiator and the target do not have to use RADIUS.

1. **Become an administrator.**
2. **Configure the initiator node with the IP address and the port of the RADIUS server.**

The default port is 1812.

```
# iscsiadm modify initiator-node --radius-server ip-address:1812
```

3. **Configure the initiator node with the shared secret key of the RADIUS server.**

The RADIUS server must be configured with a shared secret for iSCSI to interact with the server.

```
# iscsiadm modify initiator-node --radius-shared-secret
Enter secret:
Re-enter secret
```

4. **Enable the use of the RADIUS server.**

```
# iscsiadm modify initiator-node --radius-access enable
```

5. **Set up the other aspects of CHAP bidirectional authentication.**

```
# iscsiadm modify initiator-node --authentication CHAP
# iscsiadm modify target-param --bi-directional-authentication enable target-iqn
# iscsiadm modify target-param --authentication CHAP target-iqn
```

6. **Configure the RADIUS server with the following components:**

- The identity of this node (for example, its IP address)
- The shared secret key that this node uses to communicate with the RADIUS server

- The target's CHAP name (for example, its `iqn` name) and the secret key for each target that needs to be authenticated

Oracle Solaris iSCSI and RADIUS Server Error Messages

This section describes the error messages that are related to an Oracle Solaris iSCSI and RADIUS server configuration. Potential solutions for recovery are also provided.

`empty RADIUS shared secret`

Cause: The RADIUS server is enabled on the initiator, but the RADIUS shared secret key is not set.

Solution: Configure the initiator with the RADIUS shared secret key. For more information, see [“How to Configure a RADIUS Server for Your iSCSI Target” on page 63](#).

`WARNING: RADIUS packet authentication failed`

Cause: The initiator failed to authenticate the RADIUS data packet. This error can occur if the shared secret key that is configured on the initiator node is different from the shared secret key on the RADIUS server.

Solution: Reconfigure the initiator with the correct RADIUS shared secret. For more information, see [“How to Configure a RADIUS Server for Your iSCSI Target” on page 63](#).

Setting Up iSCSI Multipathed Devices in Oracle Solaris

Multipathed I/O (MPxIO) enables I/O devices to be accessed through multiple host controller interfaces from a single instance of the I/O device.

Consider the following guidelines when using iSCSI multipathed (MPxIO) devices in Oracle Solaris:

- **Oracle Solaris iSCSI and MPxIO** – MPxIO supports target port aggregation and availability in Oracle Solaris iSCSI configurations that configure multiple sessions per target (MS/T) on the iSCSI initiator.
 - Use IP network multipathing (IPMP) for aggregation and failover of two or more NICs.
 - A basic configuration for an iSCSI host is a server with two NICs that are dedicated to iSCSI traffic. The NICs are configured by using IPMP. Additional NICs are provided for non-iSCSI traffic to optimize performance.

- Active multipathing can only be achieved by using the iSCSI MS/T feature in Oracle Solaris, and the failover and redundancy of an IPMP configuration.
 - If one NIC fails in an IPMP configuration, IPMP handles the failover. The MPxIO driver does not detect the failure. In a non-IPMP configuration, the MPxIO driver fails and offlines the path.
 - If one target port fails in an IPMP configuration, the MPxIO driver detects the failure and provides the failover. In a non-IPMP configuration, the MPxIO driver detects the failure and provides the failover.

For information about configuring multiple sessions per target, see [“How to Enable Multiple iSCSI Sessions for a Target” on page 66](#). For information about configuring IPMP, see [Chapter 3, “Administering IPMP,” in “Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.2”](#).

- **Oracle Solaris iSCSI, Fibre Channel (FC), and MPxIO** – The MPxIO driver provides the following behavior in more complex iSCSI/FC configurations:
 - If you have dual iSCSI to FC bridges in an FC SAN, iSCSI presents target paths to MPxIO. MPxIO matches the unique SCSI per LU identifier, and if they are identical, presents one path to the iSCSI driver.
 - If you have a configuration that connects a target by using both iSCSI and FC, the MPxIO driver can provide different transports to the same device. In this configuration, MPxIO utilizes both paths.
 - If you are using iSCSI and FC with MPxIO, make sure that the MPxIO parameters in the `/etc/driver/drv/fp.conf` and the `/driver/drv/iscsi.conf` files match the MPxIO configuration that you want supported. For example, in `fp.conf`, you can determine whether MPxIO is enabled globally on the HBA or on a per-port basis.
- **Third-party hardware considerations** – Find out if your third-party HBA is qualified to work with Oracle Solaris iSCSI and MPxIO.

If you are using a third-party HBA, you might need to ask the vendor for the symmetric-option information for the `/driver/drv/scsi_vhci.conf` file.

▼ How to Enable Multiple iSCSI Sessions for a Target

You can use this procedure to create multiple iSCSI sessions that connect to a single target. This scenario is useful with iSCSI target devices that support login redirection or have multiple target portals in the same target portal group. Use iSCSI multiple sessions per target with the SCSI Multipathing (MPxIO) feature of Oracle Solaris. You can also achieve higher bandwidth if you utilize multiple NICs on the host side to connect to multiple portals on the same target.

The MS/T feature creates two or more sessions on the target by varying the initiator's session ID (ISID). Enabling this feature creates two SCSI layer paths on the network so that multiple

targets are exposed through the iSCSI layer to the Oracle Solaris I/O layer. The MPxIO driver handles the reservations across these paths.

For more information about how iSCSI interacts with MPxIO paths, see [“Setting Up iSCSI Multipathed Devices in Oracle Solaris” on page 65](#).

Review the following items before configuring multiple sessions for an iSCSI target:

- A typical MS/T configuration has two or more configured-sessions. However, if your storage supports multiple TPGTs and if you are using the SendTarget discovery method on your host system, then the number of configured sessions can be set to 1. SendTarget discovery automatically detects the existence of multiple paths, and multiple target sessions are created.
- Confirm that the `mpxio` configuration parameter is enabled in the `/etc/driver/drv/iscsi.conf` file.

```
# cd /etc/driver/drv
# grep mpxio iscsi.conf iscsi.conf
iscsi.conf:mpxio-disable="no";
```

- Confirm that the multiple network connections are configured by using IPMP.
- Confirm that the multiple network connections are available.

```
# ipadm show-addr
```

1. **Become an administrator.**
2. **List the current parameter values for the iSCSI initiator and the target.**

- a. **List the current parameter values for the iSCSI initiator.**

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: z zr1200
.
.
.
Configured Sessions: 1
```

- b. **List the current parameter values for the iSCSI target device.**

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
Alias: -
.
.
.
Configured Sessions: 1
```

The configured sessions value is the number of configured iSCSI sessions that will be created for each target name in a target portal group.

3. Select one of the following to modify the number of configured sessions either at the initiator node to apply to all targets or at a target level to apply to a specific target:

- Apply the desired parameter value to the iSCSI initiator node.

```
initiator# iscsiadm modify initiator-node -c 2
```

- Apply the desired parameter value to the iSCSI target.

```
initiator# iscsiadm modify target-param -c 2 iqn.1992-08.com.abcstorage:sn.84186266
```

- Bind configured sessions to one or more local IP addresses.

Configured sessions can also be bound to a local IP address. Using this method, one or more local IP addresses are supplied in a comma-separated list. Each IP address represents an iSCSI session. This method can also be used at the `initiator-node` or `target-param` level. For example:

```
initiator# iscsiadm modify initiator-node -c 10.0.0.1,10.0.0.2
```

Note - If the specified IP address is not routable, the address is ignored and the default Oracle Solaris route and IP address are used for this session.

4. Verify that the parameter was modified.

a. Display the updated information for the initiator node.

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: z zr1200
.
.
.
Configured Sessions: 2
```

b. Display the updated information for the target node.

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
Alias: -
.
.
.
Configured Sessions: 2
```

5. List the multiple paths to confirm that the OS device name matches the `iscsiadm list` output, and that the path count is 2 or more.

```
initiator# mpathadm list lu
```

Monitoring Your iSCSI Configuration

You can display information about the iSCSI initiator and target devices by using the `iscsiadm list` command.

▼ How to Display iSCSI Configuration Information

1. Become an administrator.
2. Display information about the iSCSI initiator.

For example:

```
# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
  Login Parameters (Default/Configured):
    Header Digest: NONE/-
    Data Digest: NONE/-
  Authentication Type: NONE
  RADIUS Server: NONE
  RADIUS access: unknown
  Configured Sessions: 1
```

3. Display information about which discovery methods are in use.

For example:

```
# iscsiadm list discovery
Discovery:
  Static: enabled
  Send Targets: enabled
  iSNS: enabled
```

Example 4-1 Displaying iSCSI Target Information

The following example shows how to display the parameter values for a specific iSCSI target:

```
# iscsiadm list target-param iqn.1992-08.com.abcstorage:sn.33592219
Target: iqn.1992-08.com.abcstorage:sn.33592219
```

The `iscsiadm list target-param -v` command displays the following information:

- The authentication values for the target
- The default values for the target login parameters
- The configured value for each login parameter

The `iscsiadm list target-param -v` command displays the *default* parameter value before the `/` designator and the *configured* parameter value after the `/` designator. If you have not configured a parameter, its value displays as a hyphen (-). For more information, see the following examples.

```
# iscsiadm list target-param -v eui.50060e8004275511 Target: eui.50060e8004275511
Alias: -
Bi-directional Authentication: disabled
Authentication Type: NONE
Login Parameters (Default/Configured):
  Data Sequence In Order: yes/-
  Data PDU In Order: yes/-
  Default Time To Retain: 20/-
  Default Time To Wait: 2/-
  Error Recovery Level: 0/-
  First Burst Length: 65536/-
  Immediate Data: yes/-
  Initial Ready To Transfer (R2T): yes/-
  Max Burst Length: 262144/-
  Max Outstanding R2T: 1/-
  Max Receive Data Segment Length: 65536/-
  Max Connections: 1/-
  Header Digest: NONE/-
  Data Digest: NONE/-
Configured Sessions: 1
```

The following example output displays the parameters that were negotiated between the target and the initiator:

```
# iscsiadm list target -v eui.50060e8004275511
Target: eui.50060e8004275511
  TPGT: 1
  ISID: 4000002a0000
  Connections: 1
    CID: 0
      IP address (Local): 172.20.101.71:32813
      IP address (Peer): 172.20.101.40:3260
      Discovery Method: Static
      Login Parameters (Negotiated):
        Data Sequence In Order: yes
        Data PDU In Order: yes
        Default Time To Retain: 0
        Default Time To Wait: 3
        Error Recovery Level: 0
        First Burst Length: 65536
        Immediate Data: yes
        Initial Ready To Transfer (R2T): yes
        Max Burst Length: 262144
        Max Outstanding R2T: 1
```

Max Receive Data Segment Length: 65536
Max Connections: 1
Header Digest: NONE
Data Digest: NONE

Modifying iSCSI Initiator and Target Parameters

You can modify parameters on both the iSCSI initiator and the iSCSI target device. However, the only parameters that can be modified on the iSCSI initiator are the following:

- iSCSI initiator node name – You can change the initiator node name to a different name. If you change the initiator node name, the targets that were discovered by iSNS might be removed from the initiator's target list, depending on the discovery domain configuration on the iSNS server at the time when the name was changed. For more information, see [“How to Modify iSCSI Initiator and Target Parameters” on page 74](#).
- Header digest – NONE, the default value or CRC32.
- Data digest – NONE, the default value or CRC32.
- Authentication and CHAP secret key – For more information about setting up authentication, see [“How to Configure CHAP Authentication for Your iSCSI Initiator” on page 60](#).
- Configured sessions – For more information about configuring multiple sessions, see [“How to Enable Multiple iSCSI Sessions for a Target” on page 66](#).

The iSCSI driver provides default values for the iSCSI initiator and iSCSI target device parameters. If you modify the parameters of the iSCSI initiator, the modified parameters are inherited by the iSCSI target device, unless the iSCSI target device already has different values.



Caution - Ensure that the target software supports the parameter to be modified. Otherwise, you might be unable to log in to the iSCSI target device. See your array documentation for a list of supported parameters.

You can modify the iSCSI parameters only after I/O between the initiator and the target is complete. The iSCSI driver reconnects the session after the changes are made by using the `iscsiadm modify` command.

Tuning iSCSI Parameters

iSCSI parameters can be tuned to adjust various response or connection time values of the iSCSI initiator. You can tune the iSCSI parameters depending on whether you want to adjust a parameter on the initiator for all targets for which the initiator system is connected or if you want to adjust parameters for a specific target.

Use the following command to change a parameter value of a specific iSCSI target.

```
iscsiadm modify target-param -T tunable-prop=value target-name
```

Use the following command to adjust a parameter value for all targets:

```
iscsiadm modify initiator-node -T tunable-prop=value
```

The tunable parameters below apply to the active connection, and change the behavior of the iSCSI initiator and the targets that connect to the initiator. The capability to dynamically tune parameters provides flexibility when configuring your iSCSI initiators.

TABLE 4-1 iSCSI Tunable Parameters

Parameter Name	Description	Valid Values (seconds)	Default Value (seconds)
recv-login-rsp-timeout	Session login response time – Specifies how long an iSCSI initiator waits for the response of an iSCSI session login request from a given iSCSI target.	0 – 3600	60
conn-login-max	Maximum connection retry time – Determines the maximum number of times the iSCSI initiator tries to connect to the target, after the iSCSI initiator to target I/O times out, or the connection fails.	0 – 3600	180
polling-login-delay	Login retry time interval – Determines the time interval between each iSCSI session login retry, after the iSCSI initiator to target I/O times out or the connection fails.	0 – 3600	60

▼ How to Tune iSCSI Parameters

1. Display all tunable iSCSI parameters.

Display the iSCSI parameter information for all targets.

```
# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:e00000000000.4e36d278
Initiator node alias: unknown
  Login Parameters (Default/Configured):
    Header Digest: NONE/-
    Data Digest: NONE/-
    Max Connections: 65535/-
  Authentication Type: NONE
  RADIUS Server: NONE
  RADIUS Access: disabled
  Tunable Parameters (Default/Configured):
    Session Login Response Time: 60/-
    Maximum Connection Retry Time: 180/-
    Login Retry Time Interval: 60/-
```


Configured Sessions: 1

Display the iSCSI parameter information for a specific target.

```
# iscsiadm list target-param [target-name]
```

For example:

```
# iscsiadm list target-param -v iqn.1986-03.com.sun:02:47ac0
506-cd48-67f5-fc0d-ab7544d37538
Target: iqn.1986-03.com.sun:02:47ac0506-cd48-67f5-fc0d-ab7544d37538
Alias: -
Bi-directional Authentication: disabled
Authentication Type: NONE
Login Parameters (Default/Configured):
  Data Sequence In Order: yes/-
  Data PDU In Order: yes/-
  Default Time To Retain: 20/-
  Default Time To Wait: 2/-
  Error Recovery Level: 0/-
  First Burst Length: 65536/-
  Immediate Data: yes/-
  Initial Ready To Transfer (R2T): yes/-
  Max Burst Length: 262144/-
  Max Outstanding R2T: 1/-
  Max Receive Data Segment Length: 8192/-
  Max Connections: 65535/-
  Header Digest: NONE/-
  Data Digest: NONE/-
Tunable Parameters (Default/Configured):
  Session Login Response Time: 60/-
  Maximum Connection Retry Time: 180/-
  Login Retry Time Interval: 60/-
Configured Sessions: 1
```

2. Tune an iSCSI parameter.

For example, to set the maximum connection retry time to 90 seconds for one target:

```
# iscsiadm modify target-param -T conn-login-max=90 iqn.1986-03.com.sun:
02:47ac0506-cd48-67f5-fc0d-ab7544d37538
```

For example, to set the maximum connection retry time to 90 seconds for all targets:

```
# iscsiadm modify initiator-node -T conn-login-max=90
```

▼ How to Modify iSCSI Initiator and Target Parameters

The first part of this procedure illustrates how modified parameters of the iSCSI initiator are inherited by the iSCSI target device. The second part of this procedure shows how to actually modify parameters on the iSCSI target device.

This optional procedure assumes that you are logged in to the local system where access to an iSCSI target device has already been configured.

1. **Become an administrator.**
2. **List the current parameter values for the iSCSI initiator and the target device.**
 - a. **List the current parameter values for the iSCSI initiator.**

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: z zr1200
  Login Parameters (Default/Configured):
    Header Digest: NONE/-
    Data Digest: NONE/-
  Authentication Type: NONE
  RADIUS Server: NONE
  RADIUS access: unknown
  Configured Sessions: 1
```

- b. **List the current parameter values for the iSCSI target device.**

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
Alias: -
Bi-directional Authentication: disabled
Authentication Type: NONE
  Login Parameters (Default/Configured):
    Data Sequence In Order: yes/-
    Data PDU In Order: yes/-
    Default Time To Retain: 20/-
    Default Time To Wait: 2/-
    Error Recovery Level: 0/-
    First Burst Length: 65536/-
    Immediate Data: yes/-
    Initial Ready To Transfer (R2T): yes/-
    Max Burst Length: 262144/-
    Max Outstanding R2T: 1/-
    Max Receive Data Segment Length: 65536/-
    Max Connections: 1/-
    Header Digest: NONE/-
    Data Digest: NONE/-
  Configured Sessions: 1
```

Note that both header digest and data digest parameters are currently set to NONE for both the iSCSI initiator and the iSCSI target device.

To review the default parameter values for the iSCSI target device, see the `iscsiadm list target-param` output in [Example 4-1](#).

3. Modify the parameter of the iSCSI initiator.

For example, set the header digest to CRC32.

```
initiator# iscsiadm modify initiator-node -h CRC32
```

If you change the initiator node name, the targets that were discovered by iSNS might be logged out and removed from the initiator's target list, if the new name does not belong to the same discovery domain as that of the targets. However, if the targets are in use, they are not removed. For example, if a file is open or a file system is mounted on these targets, the targets will not be removed.

You might also see new targets after the name change if these targets and the new initiator node name belong to the same discovery domain.

4. Verify that the parameter was modified.

a. Display the updated parameter information for the iSCSI initiator.

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: z zr1200
  Login Parameters (Default/Configured):
    Header Digest: NONE/CRC32
    Data Digest: NONE/-
  Authentication Type: NONE
  RADIUS Server: NONE
  RADIUS access: unknown
  Configured Sessions: 1
```

Note that the header digest is now set to CRC32.

b. Display the updated parameter information for the iSCSI target device. For example:

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
Alias: -
Bi-directional Authentication: disabled
Authentication Type: NONE
  Login Parameters (Default/Configured):
    Data Sequence In Order: yes/-
    Data PDU In Order: yes/-
    Default Time To Retain: 20/-
    Default Time To Wait: 2/-
```

```

Error Recovery Level: 0/-
First Burst Length: 65536/-
Immediate Data: yes/-
Initial Ready To Transfer (R2T): yes/-
Max Burst Length: 262144/-
Max Outstanding R2T: 1/-
Max Receive Data Segment Length: 65536/-
Max Connections: 1/-
Header Digest: CRC32/-
Data Digest: NONE/-
Configured Sessions: 1
    
```

Note that the header digest is now set to CRC32.

5. Verify that the iSCSI initiator has reconnected to the iSCSI target.

```

initiator# iscsiadm list target -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
  TPGT: 2
  ISID: 4000002a0000
  Connections: 1
    CID: 0
      IP address (Local): nnn.nn.nn.nnn:64369
      IP address (Peer): nnn.nn.nn.nnn:3260
      Discovery Method: SendTargets
      Login Parameters (Negotiated):
        .
        .
        .
      Header Digest: CRC32
      Data Digest: NONE
    
```

6. (Optional) Reset an iSCSI initiator parameter or an iSCSI target device parameter.

You can reset a parameter to its default value by using the `iscsiadm modify` command. Or, you can use the `iscsiadm remove` command to reset all parameters to their default values.

The `iscsiadm modify target-param` command changes only the parameters that are specified on the command line.

The following example shows how to reset the header digest to NONE:

```
initiator# iscsiadm modify target-param -h none iqn.1992-08.com.abcstorage:sn...
```

For information about the `iscsiadm remove target-param` command, see [iscsiadm\(1M\)](#).

Troubleshooting iSCSI Configuration Problems

The following tools are available to troubleshoot general iSCSI configuration problems:

- snoop – This tool has been updated to support iSCSI packets.
- Wireshark – This product is available from <http://www.wireshark.org/>.

Both tools can filter iSCSI packets on port 3260.

The following sections describe how to troubleshoot various iSCSI issues and error messages.

No Connections to the iSCSI Target From the Local System

▼ How to Troubleshoot iSCSI Connection Problems

1. **Become an administrator.**
2. **List your iSCSI target information.**

For example:

```
initiator# iscsiadm list target
Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zzr1200-01
      TPGT: default
      ISID: 4000002a0000
      Connections: 0
```

3. **If no connections are listed in the `iscsiadm list target` output, check the `/var/adm/messages` file for possible reasons why the connection failed.**

You can also verify whether the connection is accessible by using the `ping` command or by connecting to the storage device's iSCSI port by using the `telnet` command to ensure that the iSCSI service is available. The default port is 3260.

In addition, check the storage device's log file for errors.

4. **If your target is not listed in the `iscsiadm list target` output, check the `/var/adm/messages` file for possible causes.**

If you are using `SendTargets` as the discovery method, try listing the *discovery-address* by using the `-v` option to ensure that the expected targets are visible to the host. For example:

```
initiator# iscsiadm list discovery-address -v 10.0.0.1
Discovery Address: 10.0.0.1:3260
  Target name: eui.210000203787dfc0
  Target address: 10.0.0.1:11824
  Target name: eui.210000203787e07b
  Target address: 10.0.0.1:11824
```

If you are using iSNS as the discovery method, try enabling the iSNS discovery method and listing the *isns-server* using the `-v` option to ensure that the expected targets are visible to the host. For example:

```
initiator# iscsiadm list isns-server -v
iSNS Server IP Address: 10.20.56.56:3205
  Target name: iqn.1992-08.com.xyz:sn.1234566
    Target address: 10.20.57.161:3260, 1
  Target name: iqn.2003-10.com.abc:group-0:154:abc-65-01
    Target address: 10.20.56.206:3260, 1
  Target name: iqn.2003-10.com.abc:group-0:154:abc-65-02
    Target address: 10.20.56.206:3260, 1
.
.
.
```

iSCSI Device or Disk Is Not Available on the Local System

▼ How to Troubleshoot iSCSI Device or Disk Unavailability

1. **Become an administrator.**
2. **Identify the LUNs that were discovered on this target during enumeration.**

The `-S` option shows which LUNs were discovered on this target during enumeration.

For example:

```
# iscsiadm list target -S
Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zzr1200-01
  TPGT: default
  ISID: 4000002a0000
  Connections: 1
  LUN: 0
    Vendor: ABCSTOR
    Product: 0010
    OS Device Name: /dev/rdisk/c3t34d0s2
```

3. **Review the `/var/adm/messages` file to see if an error was reported.**

If you think a LUN should be listed but it is not, then check this log file.

4. **Check the storage device's log files for errors.**
5. **Ensure that any storage device LUN masking is properly configured.**

Use LUN Masking When Using the iSNS Discovery Method

Avoid using the iSNS discovery domain as the means to control storage authorization to specific initiators. Use *LUN masking* instead to ensure that only authorized initiators can access a LUN.

If you remove a target from a discovery domain while the target is in use, the iSCSI initiator does not log out from this target. If you do not want this initiator to access this target (and the associated LUNs), you must use LUN masking. Removing the target from the discovery domain is not sufficient.

General iSCSI Error Messages

This section describes the iSCSI messages that might be found in the `/var/adm/messages` file and potential solutions for recovery.

The message format is as follows:

```
iscsi TYPE (OID) STRING (STATUS-CLASS#/STATUS-DETAIL#)
```

<i>TYPE</i>	Is either connection or session.
<i>OID</i>	Is the object ID of the connection or session. This ID is unique for an OS instance.
<i>STRING</i>	Is a description of the condition.
<i>STATUS-CLASS#/ STATUS-DETAIL#</i>	These values are returned in an iSCSI login response as defined by RFC 3720.

```
iscsi connection(OID) login failed - Miscellaneous iSCSI initiator errors.
```

Cause: The device login failed due to some form of initiator error.

```
iscsi connection(OID) login failed - Initiator could not be successfully authenticated.
```

Cause: The device could not successfully authenticate the initiator.

Solution: If applicable, verify that the settings for CHAP names, CHAP passwords, or the RADIUS server are correct.

```
iscsi connection(OID) login failed - Initiator is not allowed access to the given target.
```

Cause: The device cannot allow the initiator access to the iSCSI target device.

Solution: Verify your initiator name and confirm that it is properly masked or provisioned by the storage device.

iscsi connection(*OID*) login failed - Requested ITN does not exist at this address.

Cause: The device does not provide access to the iSCSI target name (ITN) that you are requesting.

Solution: Verify that the initiator discovery information is specified properly and that the storage device is configured properly.

iscsi connection(*OID*) login failed - Requested ITN has been removed and no forwarding address is provided.

Cause: The device can no longer provide access to the iSCSI target name (ITN) that you are requesting.

Solution: Verify that the initiator discovery information has been specified properly and that the storage device has been configured properly.

iscsi connection(*OID*) login failed - Requested iSCSI version range is not supported by the target.

Cause: The initiator's iSCSI version is not supported by the storage device.

iscsi connection(*OID*) login failed - No more connections can be accepted on this Session ID (SSID).

Cause: The storage device cannot accept another connection for this initiator node to the iSCSI target device.

iscsi connection(*OID*) login failed - Missing parameters (e.g., iSCSI initiator and/or target name).

Cause: The storage device is reporting that the initiator or target name has not been properly specified.

Solution: Properly specify the iSCSI initiator or target name.

iscsi connection(*OID*) login failed - Target hardware or software error.

Cause: The storage device encountered a hardware or software error.

Solution: Consult the storage documentation, or contact the storage vendor for further assistance.

iscsi connection(*OID*) login failed - iSCSI service or target is not currently operational.

Cause: The storage device is currently not operational.

Solution: Consult the storage documentation, or contact the storage vendor for further assistance.

iscsi connection(*OID*) login failed - Target has insufficient session, connection or other resources.

Cause: The storage device has insufficient resources.

Solution: Consult the storage documentation, or contact the storage vendor for further assistance.

iscsi connection(*OID*) login failed - unable to initialize authentication

iscsi connection(*OID*) login failed - unable to set authentication

iscsi connection(*OID*) login failed - unable to set username

iscsi connection(*OID*) login failed - unable to set password

iscsi connection(*OID*) login failed - unable to set ipsec

iscsi connection(*OID*) login failed - unable to set remote authentication

Cause: The initiator was unable to initialize or set authentication properly.

Solution: Verify that your initiator settings for authentication are properly configured.

iscsi connection(*OID*) login failed - unable to make login pdu

Cause: The initiator was unable to make a login payload data unit (PDU) based on the initiator or storage device settings.

Solution: Try resetting any target login parameters or other nondefault settings.

iscsi connection(*OID*) login failed - failed to transfer login

iscsi connection(*OID*) login failed - failed to receive login response

Cause: The initiator failed to transfer or receive a login payload data unit (PDU) across the network connection.

Solution: Verify that the network connection is reachable.

iscsi connection(*OID*) login failed - received invalid login response (*OP CODE*)

Cause: The storage device has responded to a login with an unexpected response.

iscsi connection(*OID*) login failed - login failed to authenticate with target

Cause: The initiator was unable to authenticate the storage device.

Solution: Verify that your initiator settings for authentication are properly configured.

iscsi connection(*OID*) login failed - initiator name is required

Cause: An initiator name must be configured to perform all actions.

Solution: Verify that the initiator name is configured.

iscsi connection(*OID*) login failed - authentication receive failed

iscsi connection(*OID*) login failed - authentication transmit failed

Cause: The initiator was unable to transmit or receive authentication information.

Solution: Verify network connectivity with the storage device or the RADIUS server, as applicable.

iscsi connection(*OID*) login failed - login redirection invalid

Cause: The storage device attempted to redirect the initiator to an invalid destination.

Solution: Consult the storage documentation, or contact the storage vendor for further assistance.

iscsi connection(*OID*) login failed - target protocol group tag mismatch, expected <TPGT>, received <TPGT>

Cause: The initiator and target had a TPGT (target portal group tag) mismatch.

Solution: Verify your TPGT discovery settings on the initiator or the storage device.

iscsi connection(*OID*) login failed - can't accept *PARAMETER* in security stage

Cause: The device responded with an unsupported login parameter during the security phase of login.

Solution: The parameter name is noted for reference. Consult the storage documentation, or contact the storage vendor for further assistance.

iscsi connection(*OID*) login failed - HeaderDigest=CRC32 is required, can't accept *VALUE*

iscsi connection(*OID*) login failed - DataDigest=CRC32 is required, can't accept *VALUE*

Cause: The initiator is only configured to accept a HeaderDigest or DataDigest that is set to CRC32 for this target. The device returned the value of *VALUE*.

Solution: Verify that the initiator and device digest settings are compatible.

iscsi connection(*OID*) login failed - HeaderDigest=None is required, can't accept *VALUE*

iscsi connection(*OID*) login failed - DataDigest=None is required, can't accept *VALUE*

Cause: The initiator is only configured to accept a HeaderDigest or DataDigest that is set to NONE for this target. The device returned the value of *VALUE*.

Solution: Verify that the initiator and device digest settings are compatible.

iscsi connection(*OID*) login failed - can't accept *PARAMETER*

Cause: The initiator does not support this parameter.

iscsi connection(*OID*) login failed - can't accept MaxOutstandingR2T *VALUE*

Cause: The initiator does not accept MaxOutstandingR2T of the noted *VALUE*.

iscsi connection(*OID*) login failed - can't accept MaxConnections *VALUE*

Cause: The initiator does not accept the maximum connections of the noted *VALUE*.

iscsi connection(*OID*) login failed - can't accept ErrorRecoveryLevel *VALUE*

Cause: The initiator does not accept an error recovery level of the noted *VALUE*.

iscsi session(*OID*) *NAME* offline

Cause: All connections for this target *NAME* have been removed or have failed.

iscsi connection(*OID*) failure - unable to schedule enumeration

Cause: The initiator was unable to enumerate the LUNs on this target.

Solution: You can force LUN enumeration by running the `devfsadm -i iscsi` command. For more information, see [devfsadm\(1M\)](#).

`iscsi connection(OID) unable to connect to target NAME (errno:ERRNO)`

Cause: The initiator failed to establish a network connection.

Solution: For information about the specific *ERRNO* on the connection failure, see the `/usr/include/sys/errno.h` file.

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.

The following topics are covered:

- [“What Is NPIV?” on page 85](#)
- [“Limitations of NPIV” on page 85](#)
- [“Working With NPIV Ports” on page 86](#)

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

NPIV limitations when used to virtualize Fibre Channel ports are as follows:

- 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 might 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 a large number of ports are 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, and not 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 capability is defined by the resources on the switch. Switches might have to be updated to the latest firmware levels for NPIV support.

Working With 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, regardless of 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 no distinction is 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(1M)` 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 5-1 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 5-2 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 5-3 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 covered:

- [“What Is FCoE?” on page 89](#)
- [“Limitations of FCoE” on page 89](#)
- [“Configuring FCoE Ports” on page 90](#)

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 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. Use the `fcinfo` and `fcadm` commands to determine the status of FCoE ports. These 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 no distinction is 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 KB) 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:

```
# fcadm create-fcoe-port -i net0
```

▼ 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.**

```
# fcadm delete-fcoe-port network_interface
```

For example:

```
# fcadm delete-fcoe-port net0
```

▼ 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
```

List FC specific information for all FCoE ports in the system.

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

For example:

```
# fcdm list-fcoe-ports
HBA Port WWN: 2000001b2165a630
  Port Mode: Initiator
  Port ID: e00033
  VLAN ID: 7
  Link Name: net2
  MTU Size: 2500
  Primary MAC Address: 00:1b:21:65:a6:30
  FCoE MAC Address: 0e:fc:00:e0:00:33
  Promiscuous Mode: Off
  State: Online
  FIP Mode: On
  FCoE Hardware Offload: Supported
HBA Port WWN: 2000001b2165a631
  Port Mode: Target
  Port ID: e00034
  VLAN ID: 7
  Link Name: net3
  MTU Size: 2500
  Primary MAC Address: 00:1b:21:65:a6:31
  FCoE MAC Address: 0e:fc:00:e0:00:34
  Promiscuous Mode: Off
  State: Online
  FIP Mode: On
  FCoE Hardware Offload: Supported
```

▼ How to Force an FCoE Port Reinitialization

Use the steps below when an FCoE port needs to be reinitialized. You might need to force an 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-lip 200000144fc2d508
```

Configuring FCoE Hardware Offload

You can use the tunables listed in the following table 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 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 6-1 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
<code>fcoe_lso_enable</code>	Controls whether the <code>ixgbe</code> driver offloads FC large send transactions for transmitted FCoE packets.	0 – Disable FC LSO transaction offload 1 – Enable FC LSO transaction offload	1
<code>fcoe_rxcrc_enable</code>	Controls whether the <code>ixgbe</code> driver offloads RC CRC transactions for received FCoE packets.	0 – Disable FC RX CRC transaction offload 1 – Enable FC RX CRC transaction offload	1
<code>fcoe_lro_enable</code>	Controls whether the <code>ixgbe</code> 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 SAS domain considerations, SAS device discovery, and SAS boot device configuration.

The following topics are covered:

- “SAS Multipathing Considerations” on page 95
- “Dynamic Discovery of SAS Devices” on page 95
- “Configuring SAS Boot Devices” on page 96

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 its presence or removal. If a device has been added, it is visible to and available through the `format` command.

Configuring SAS Boot Devices

Systems running the Oracle Solaris OS can 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

This chapter provides Internet Protocol over Fibre Channel (IPFC) configuration information for a host system to describe 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.

The following topics are covered:

- [“IPFC Considerations” on page 97](#)
- [“Invoking and Configuring IPFC” on page 100](#)

IPFC Considerations

The following table shows the supported features available for IPFC.

TABLE 8-1 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 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 OS, 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 the slot information to calculate the Fibre Channel (FC) adapter port instances. You can also determine the slot in which the card exists and the number that is associated with the card by using the command:

```
prtdiag | grep -i pci
```

The examples in this procedure assume that you have an array with an HBA card that is 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.

- 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,qlc@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,qlc@1/fp@0,0 /etc/path_to_inst
"/pci@8,600000/SUNW,qlc@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,qlc@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 `192.168.201.10`, type:

```
# ipadm create-ip fcip0
# ipadm create-addr -T static -a 192.168.201.10 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 = "neo-1"
```

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. For additional information, see [hosts\(4\)](#).

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    localhost
192.168.200.70  neo-1      #This is the local host name
192.168.201.10 fcip0     #Interface to network 192.168.201.10
```

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 Oracle Solaris OS From Fibre Channel Devices on x86 Based Systems

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

The following topics are covered:

- [“Oracle Solaris OS Setup Requirements” on page 103](#)
- [“Oracle Solaris OS Installation Overview” on page 104](#)
- [“Oracle Solaris OS Installation Procedure” on page 105](#)

Oracle Solaris OS Setup Requirements

You must have the following items for installation:

- Any one of Oracle Solaris 11, Oracle Solaris 11.1, or Oracle Solaris 11.2 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'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 based systems. Most 2 Gb HBAs do support booting over a SAN on x86 based 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
-

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. Record the HBA World Wide Name (WWN) and array WWN port information.

During rebooting, use the WWN and LUN 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

The following procedures describe how to install Oracle Solaris OS.

▼ 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 11.2 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 11.2 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 based 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 11.2 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 9-1](#).

```
# luxadm display /dev/rdisk/c0t600015D0002028000000000000001142d0s2
DEVICE PROPERTIES for disk: /dev/rdisk/c0t600015D000202800000000000001142d0s2
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
/dev/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 9-1 luxadm display Command and Output

The screenshot shows the QLogic FastUTIL utility interface. It displays the selected adapter and its settings. The selected adapter is a QLA2462 with I/O Address 3400, Slot 02, Bus 03, Device 01, and Function 1. The adapter settings include 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), and Data Rate (2).

```

QLogic FastUTIL
=====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# = c0t600015d0002028000000000000001142d0
- 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 9-2](#) and [Figure 9-3](#).

- **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 9-2 HBA BIOS Screen for an HBA WWN

```

QLogic FastUTIL
=====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 9-3 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
(PPrimary) 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

This chapter describes how to create persistent bindings for tape devices to ensure the `/dev` entries are identical across multiple servers on a SAN.

The following topics are covered:

- [“Persistent Binding Overview” on page 111](#)
- [“Creating Tape Links” on page 112](#)

Persistent Binding Overview

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.

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,qlc@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 112](#).**

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.

The following topics are covered:

- [“Manually Configuring FC Devices” on page 115](#)
- [“Configuring Fabric Device Nodes” on page 116](#)
- [“Configuring Device Nodes Without Multipathing Enabled” on page 118](#)
- [“Configuring Device Nodes With Solaris Multipathing Enabled” on page 121](#)
- [“Unconfiguring Fabric Devices” on page 125](#)

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;
```

For more information about this setting, see [`cfgadm_fp\(1M\)`](#) and [`fp\(7d\)`](#).

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 118](#)
 - [“Configuring Device Nodes With Solaris Multipathing Enabled” on page 121](#)

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;
```

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 117](#). 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 117](#).

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.**

```
# 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   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 your volume manager documentation 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 your volume manager documentation 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.**

```

# 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
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 your volume manager documentation 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 `cfadm -c unconfigure` command.

```
# cfmadm -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/rdisk/c6t60020F20000061073AC8B52D000B74A3d0s2
/dev/rdisk/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 your volume manager documentation 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.

◆◆◆ APPENDIX B

Supported FC-HBA API

This appendix contains the list of supported FC-HBA Interfaces. For further information regarding the API, refer to [“Overview of Solaris I/O Multipathing” on page 9](#).

Supported Fibre Channel HBA API

Solaris I/O multipathing supports the following FC-HBA interfaces:

- HBA_GetVersion
- HBA_LoadLibrary
- HBA_FreeLibrary
- HBA_GetNumberOfAdapters
- HBA_GetAdapterName
- HBA_OpenAdapter
- HBA_CloseAdapter
- HBA_GetAdapterAttributes
- HBA_GetAdapterPortAttributes
- HBA_GetDiscoveredPortAttributes
- HBA_GetPortAttributesbyWWN
- HBA_SendCTPassThru
- HBA_SendCTPassThruV2
- HBA_RefreshInformation
- HBA_GetFcpTargetMapping
- HBA_SendScsiInquiry
- HBA_SendReportLuns
- HBA_SendReadCapacity
- HBA_SetRNIDMgmtInfo
- HBA_GetRNIDMgmtInfo

- HBA_SendRNID
- HBA_SendRNIDV2
- HBA_ScsiInquiryV2
- HBA_ScsiReportLUNsV2
- HBA_ScsiReadCapacityV2
- HBA_OpenAdapterByWWN
- HBA_RefreshAdapterConfiguration
- HBA_GetVendorLibraryAttributes
- HBA_GetFcpTargetMappingV2
- HBA_SendRLS
- HBA_RegisterForAdapterEvents
- HBA_RegisterForAdapterAddEvents
- HBA_RegisterForAdapterPortEvents
- HBA_RegisterForTargetEvents
- HBA_RegisterForAdapterTargetEvents

Troubleshooting Multipathing-Related Problems

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

The following topics are covered:

- [“How to Recover Boot Failure in Single User Mode” on page 133](#)
- [“How to Recover from a System Crash” on page 133](#)

System Fails to Boot or Crashes 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.

▼ How to Recover from a System Crash

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

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 11.1 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.

For example, if you modified the `scsi_vhci.conf` file, undo the changes you made to this file by editing the `/a/kernel/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 unmount solaris
```

10. Unmount the UFS root file system.

```
# umount /a
```

11. Reboot the system.

Tuning Disk Target Driver Properties

This appendix explains how to tune the Oracle Solaris disk target driver (sd or ssd driver) properties in the `.conf` file by using the `sd-config-list` or `ssd-config-list` global property.

The appendix describes the following formats for tuning disk target driver properties:

- [“Name:Value Pair Format to Tune Disk Drivers” on page 138](#)
- [“Bit-Masking Format to Tune Disk Drivers” on page 140](#)

Tunable Parameters for Disk Drivers

The tunable parameters to tune a disk driver uses the following prefix categories:

- `BCD` – Binary-Coded-Decimal as shown in [Table D-1](#)
- `delay` – Delay time in issuing a retry
- `timeout` – Maximum time allowed by a process
- `reset` – Reset control
- `retries` – Number of retries before failure
- `throttle` – Activity control

The following tunable names and their data types are supported by Oracle Solaris 10 and Oracle Solaris 11:

<code>cache-nonvolatile</code>	BOOLEAN
<code>controller-type</code>	UINT32
<code>delay-busy</code>	UINT32
<code>disksort</code>	BOOLEAN
<code>emulation-rmw</code>	UINT32

physical-block-size	UINT32
reset-lun	BOOLEAN
retries-busy	UINT32
retries-timeout	UINT32
retries-notready	UINT32
retries-reset	UINT32
rmw-type	UINT32
timeout-releasereservation	UINT32
throttle-max	UINT32
throttle-min	UINT32

The tunable parameters that are supported only in Oracle Solaris 11 are all BOOLEAN data types. The parameters are:

- cdb-suppress-dpofua
- mmc-gesn-polling
- power-condition

Note - The BOOLEAN value must be either TRUE or FALSE.

Name:Value Pair Format to Tune Disk Drivers

The `sd` and `ssd` drivers support the JSON-text *name:value* format, which enables you to set specific tunable property values.

Use the following syntax for an `sd` driver:

```
sd-config-list = duplet [, duplet]*;
```

Note - Be sure to end the entry with a semicolon or the configuration will be invalid and the properties will retain their default values.

Use the following syntax for an `ssd` driver:

```
ssd-config-list = duplet [, duplet]*;
```

where, *duplet* is "VID PID", "*tunable* [, *tunable*]*"

VID Content of the vendor identification (VID) field of the device response to a SCSI INQUIRY command. The VID field must be eight characters in length. If the VID field is less than eight characters, you must add whitespaces to make the length of the VID to eight characters. See [Example D-1](#).

PID Content of the product identification (PID) field of the device response to a SCSI INQUIRY command. The PID can contain a maximum of 16 left-adjusted characters. If you provide less than 16 characters, then the comparison is limited to the length of the PID that you have provided.

tunable [, *tunable*]* *tunable* is the *name:value* pair.

Note - If you misspell a tunable or do not end the entry with a semicolon, the configuration is invalid and the device properties will retain their default values.

The PID value is considered a match when the prefix value returned by the SCSI INQUIRY command and the PID in the `sd-config-list` or `ssd-config-list` are same. For example, if CMS200 is the PID in the `sd-config-list` or `ssd-config-list` entry and if the PID returned by the SCSI INQUIRY command is CMS200-R, CMS200-T, or CMS200-UV10 would all be considered a match.

On a SPARC platform, the target devices can be bound to `sd` or `ssd` driver depending on whether the device is an FC device and whether MPxIO is enabled on the device. You can use the `prtconf` command to check the device configuration information. See the [prtconf\(1M\)](#) man page.

EXAMPLE D-1 Configuring Two Target Devices in an `sd.conf` File

The following example shows how to configure two target devices SAMPLE and SUM in an `sd.conf` file.

```
sd-config-list = "SAMPLE STTU1234566AB", "delay-busy:6000000000",  
                "SUM ABC200_R", "retries-busy:5, throttle-max:300";
```

In this example, for the device SAMPLE, the VID is SAMPLE and the PID is STTU1234566AB. The delay time before retrying is set to 6 seconds.

For the device SUM, the VID is SUM and the PID is ABC200_R. The number of retries on an I/O busy status is set to 5. The maximum throttle value is set to 300.

EXAMPLE D-2 Configuring Two Target Devices in an `ssd.conf` File

The following example shows how to configure two target devices GATES and SINE in an `ssd.conf` file.

```
ssd-config-list = "GATES  AB568536611CD46G", "reset-lun:TRUE",  
                 "SINE   XYZ200_R", "retries-notready:6, throttle-min:200";
```

In this example, for the device GATES, the VID is GATES and the PID is AB568536611CD46G. The value TRUE for the `reset-lun` tunable parameter indicates that the LUN is reset.

For the device SINE, the VID is SINE and the PID is XYZ200_R. The number of retries when the I/O is not ready is set to 6. The minimum throttle value is set to 200.

Bit-Masking Format to Tune Disk Drivers

You can tune parameters by using the bit-masking format, also known as the Version1 format. The bit-masking format includes the `sd-config-list` property, which contains entries for the property array.

Use the following syntax for an `sd` driver:

```
sd-config-list = duplet [, duplet ]*;  
sd-ver1-conf-data = 1, mask, value-sequence;
```

Note - Be sure to end the entry with a semicolon or the configuration will be invalid and the properties will retain their default values.

Use the following syntax for an `ssd` driver:

```
ssd-config-list = duplet [, duplet ]*;  
sd-ver1-conf-data = 1, mask, value-sequence;
```

where *duplet* is "VID PID", "sd-ver1-conf-data"

VID Content of the vendor identification (VID) field of the device response to a SCSI INQUIRY command. The VID field must be eight characters in length. If the vendor identification field is less than eight characters, you must add whitespaces to make the length of the VID eight characters.

<i>PID</i>	Content of the product identification (PID) field of the device response to a SCSI INQUIRY command. The PID can contain a maximum of 16 left-adjusted characters. If you provide less than 16 characters, then the comparison is limited to the length of the PID that you have provided.	
<i>sd-ver1-conf-data</i>	Property array consisting of a version number which is 1, a mask number, and tunable values to be set.	
	<i>mask</i>	Value between 0x01 to 0x7FFFF. The value of <i>mask</i> can sometimes be the OR result of multiple desired flags. Table D-1 shows the bit value of each tunable parameter.
	<i>value-sequence</i>	Sequence of valid property values and number of 0's corresponding to the value of <i>mask</i> . The length limit for <i>value-sequence</i> is 19 characters. See “Tunable Parameters for Disk Drivers” on page 137 for the data type of tunable parameters.

Note - The duplets with different VIDs and PIDs either share the same *sd-ver1-conf-data* property array or define their own property array. You can customize the property array name.

The PID value is considered a match when the prefix value returned by the SCSI INQUIRY command and the PID in the *sd-config-list* or *ssd-config-list* are same. For example, if CMS200 is the PID in the *sd-config-list* or *ssd-config-list* entry and if the PID returned by the SCSI INQUIRY command is CMS200-R, CMS200-T, or CMS200-UV10 would all be considered a match.

The definition of each bit's position depends on the platform. A tunable might correspond to a different flag value on a different platform. Both Oracle Solaris 11 and Oracle Solaris 10 support the same set of 19 tunable parameters in the bit-masking format.

The following table lists the tunable parameters and their flag values on different platforms.

TABLE D-1 Supported Tunables and Their Flag Values on Different Platforms

Bit	Flag Value	sd Driver on SPARC	ssd Driver on SPARC	sd Driver on x86 or x64
1	0x00001	throttle-max	throttle-max	throttle-max
2	0x00002	controller-type	retries-notready	controller-type
3	0x00004	retries-notready	retries-busy	fab-devid

Bit	Flag Value	sd Driver on SPARC	ssd Driver on SPARC	sd Driver on x86 or x64
4	0x00008	fab-devid	fab-devid	disable_caching
5	0x00010	disable_caching	disable_caching	BCD-play
6	0x00020	retries-busy	controller-type	BCD-read-subchannel
7	0x00040	BCD-play	BCD-play	BCD-read-TOC-TRK
8	0x00080	BCD-read-subchannel	BCD-read-subchannel	BCD-read-TOC-ADDR
9	0x00100	BCD-read-TOC-TRK	BCD-read-TOC-TRK	no-READ-HDR
10	0x00200	BCD-read-TOC-ADDR	BCD-read-TOC-ADDR	CD-read-XD4
11	0x00400	no-READ-HDR	no-READ-HDR	retries-notready
12	0x00800	CD-read-XD4	CD-read-XD4	retries-busy
13	0x01000	retries-reset	retries-reset	retries-reset
14	0x02000	timeout-releasereservation	timeout-releasereservation	timeout-releasereservation
15	0x04000	tur-check	tur-check	tur-check
16	0x08000	throttle-min	throttle-min	throttle-min
17	0x10000	disksort	disksort	disksort
18	0x20000	reset-lun	reset-lun	reset-lun
19	0x40000	cache-nonvolatile	cache-nonvolatile	cache-nonvolatile

EXAMPLE D-3 Configuring the Tunable Parameters for an sd Driver on an x86 Platform

This example shows how to tune the parameters in an `sd.conf` file on an x86 platform.

```
sd-config-list = "SUM      ABC200_R", "sd-ver1-x86-example";
sd-ver1-x86-example = 1,0x801,300,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,0,0,0,0,0,0,0;
```

In this example, for the device SUM, the VID is SUM and the PID is ABC200_R.

The mask value 0x801 is the bitwise OR value of 0x00800 and 0x00001 flags. These values are `retries-busy` and `throttle-max` tunable parameters on an x86 platform.

EXAMPLE D-4 Configuring the Tunable Parameters for an ssd Driver on a SPARC Platform

This example shows how to tune the parameters in an `ssd.conf` file on a SPARC platform.

```
ssd-config-list = "SUM      XYZ200_R", "sd-ver1-ssd-example";
```

```
sd-ver1-ssd-example = 1,0x5,300,0,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
```

In this example, for the device SUM, the VID is SUM and the PID is XYZ200_R.

The mask value 0x5 is the bitwise OR value of 0x000004 and 0x000001 flags. These values are `retries-busy` and `throttle-max` tunable parameters on a SPARC platform.

If a `.conf` file contains multiple `sd-config-list` or `ssd-config-list` entries tune the same target device, only the first entry takes effect. All subsequent entries with the same VID and PID are ignored. This behavior is the same for both the bit-masking format and the `name:value` pair format.

If an `sd-config-list` property contains more than one duplet with the same with the same VID and PID, the succeeding specifications in the order of entries take precedence and replace the values that appeared in earlier duplets. This behavior is the same for both the bit-masking format and the `name:value` pair format.

In the bit-masking format, if the length of the `value-sequence` parameter is more than 19 characters, values after the 19th character are ignored. If the length of the `value-sequence` is less than 19 characters, the configuration result depends on its preceding `mask` value. If the corresponding value position of the flagged bit in the `mask` value is empty, then it results in a random value is assigned for that property in the target driver.

The target driver does not provide a syntax check for the `.conf` file so no warning messages for errors like misspelling of tunable names and incorrect entries for `value-sequence`.

Index

Numbers and Symbols

24-bit FC addressing devices, 116

A

accessing

 iSCSI disks (how to), 57

ANSI standard Multipath Management API, 29

Ap_Id, 117

automatic failback

 configuration, 28

B

Bit-Masking Format

 Version1 format, 140

boot disk, 49

booting, 10

broadcasting, 97

C

cfgadm

 -c, 119

 -l, 117, 118

cfgadm -al -o show_SCSI_LUN, 117

cfgadm -c unconfigure, 127

cfgadm configure, 118

cfgadm(1M), 118

cfgadm_fp(1M), 121

configuration

 device, 117

 dump, 18

 fabric devices, 118

 FCoE, 90

 manual, 115

 multipathing, 17

 SAS devices, 96

 tasks, 15

configure

 automatic failback, 28

 fabric-connected devices, 47

 FC devices, 47

 multipathing feature, 17, 47

 NPIV ports, 85

 SAS devices, 96

 third-party devices, 24, 25

 virtual Fibre Channel ports, 85

configuring

 unidirectional or bidirectional CHAP authentication
 for iSCSI (how to), 60

considerations

 device name changes, 18

 device specific, 18

 fabric boot devices, 49

 IPFC, 97

 multipathing, 17

 per-port configuration, 22

 SAN devices, 47

 SAS devices, 95

 stmsboot command, 16

 StorageTek Traffic Manager, 16

 third-party device configuration, 24

D

device

 asymmetrical, 12

 configuration, 118

 storage, 12

device management, 15

devices, persistent naming

tape, 10

disabling

multipathing, 18

discovery

dynamic, 95

SAS devices, 95

E

enabling

multipathing, 18

error messages

iSCSI, 79

F

fabric device configuration, 118

fabric-connected host ports, 118, 122

fc-fabric, 122

FCAL, 10

FCoE

configuring ports, 90

creating ports, 90

deleting ports, 91

limitations, 89

reporting status on ports, 91

Fibre Channel

arbitrated loop, 10

booting x86 based systems, 103

configuring IPFC SAN devices, 97

determining adapter port instances, 98

enabling and disabling multipathing, 18

HBA, 10

IPFC considerations, 97

listing LUN level information, 117

multipathing configuration, 17

supported HBA interfaces, 131

Fibre Channel ports

configuring, 85

I

initiator ports

displaying properties, 30

Internet Protocol over Fibre Channel *See* IPFC

IPFC

considerations, 97

iSCSI

accessing iSCSI disks (how to), 57

configuring unidirectional or bidirectional CHAP

authentication for (how to), 60

general iSCSI error messages, 79

modifying iSCSI initiator and target parameters

(how to), 74

monitoring your iSCSI configuration (how to), 69

removing discovered iSCSI targets (how to), 58

static and dynamic target discovery, 55

troubleshooting iSCSI configuration problems (how to), 76

`iscsiadm list`

displaying iSCSI configuration information

(example of), 69

`iscsiadm modify` command

enabling CHAP (example of), 61

enabling or disabling static or dynamic targets

(example of), 58

`iscsiadm remove` command

removing static or dynamic targets (example of), 59

L

limitations

FCoE, 89

LUN

masking, 47

recognition, 48

SCSI devices, 124

`luxadm(1M)`, 28

M

modifying

iSCSI initiator and target parameters (how to), 74

`modinfo`, 117

monitoring

your iSCSI configuration (how to), 69

`mpathadm` commands, 29

automatic failback, configuring, 36

- configure automatic failback, 36
- disable a path, 43
- enable a path, 41
- initiator ports
 - properties, 30
- logical unit with a particular name, 34
- logical units
 - associated with target port, 32
 - fail over, 38
 - properties and associated path, 31
- mpt driver, 95
- Multipath Management API plug-in
 - library, 29
 - properties, 29
- multipathing
 - boot considerations, 49
 - configuring automatic failback, 28
 - configuring device nodes, 121
 - configuring third-party storage devices, 24
 - disabled, device node configuration, 118
 - disabled, multiple device configuration, 120
 - enabling and disabling, 18
 - enabling and disabling on a per-port basis, 22
 - fabric device node configuration, 116
 - features, 11
 - manual configuration, 115
 - mpathadm command, 29
 - SAS boot device configuration, 96
 - SAS considerations, 95
 - support, 29
 - troubleshooting, 133
- multipathing, enabling and disabling, 18

N

- Name:Value Format
 - JSON-text format, 138
- NPIV ports
 - configuring, 85

P

- per-port configuration
 - considerations, 22
 - enabling and disabling multipathing, 22

- persistent binding of tape devices, 111
- physical device, 119
- Promiscuous mode, 97

R

- removing
 - discovered iSCSI targets (how to), 58
- REPORT LUNS command, 24

S

- SAS devices
 - configuration, 96
 - dynamic discovery, 95
 - multipathing considerations, 95
- sd driver, 95
- show_SCSI_LUN, 119, 124
- snoop(1M), 97
- ssd driver, 117
- st driver, 117
- stmsboot command, 18
 - boot considerations, 49
 - considerations, 16
 - determining device paths, 18
 - enabling and disabling multipathing, 13
 - FC device considerations, 47
 - system crashes, 133
- storage device, 28

T

- T10 standard, 29
- T11 standard, 10
- tape device persistent binding, 111
- troubleshooting
 - iSCSI configuration problems (how to), 76
 - multipathing, 133

X

- x86 based system OS booting procedure
 - Fibre Channel, 103

