Managing SAN Devices and Multipathing in Oracle® Solaris 11.3

Part No: E54792
November 2016

ORACLE

Managing SAN Devices and Multipathing in Oracle Solaris 11.3

Part No: E54792

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support 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.

Référence: E54792

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité à la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` ou le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` si vous êtes malentendant.

Contents

Using This Documentation

- **Overview** – Provides an overview of Oracle Solaris I/O multipathing features for the Oracle Solaris operating system and describes how to configure Oracle 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) SANs and SAS domains
- **Required knowledge** – Expertise in the management and maintenance of SANs and SAS domains

Product Documentation Library

Documentation and resources for this product and related products are available at http://www.oracle.com/pls/topic/lookup?ctx=E53394.

Feedback

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

♦ ♦ ♦ CHAPTER 1

Introduction to Oracle Solaris I/O Multipathing

This chapter provides an overview of the Oracle Solaris I/O multipathing features.

This chapter contains the following topics:

Oracle Solaris I/O Multipathing Overview

The Oracle Solaris I/O multipathing feature, also known as StorageTek Manager software, enables multiple access paths for systems that are running the Oracle Solaris operating system (OS). Oracle Solaris I/O multipathing is based on the open standards for communicating with devices and device management, ensuring interoperability with other standard-based devices and software. Multipathing provides higher availability for storage devices through the use of multipathed connections. Oracle Solaris I/O multipathing is enabled by default for x86 based platforms. Enabling or disabling multipathing is optional for Fibre Channel Devices on SPARC based systems that run the Oracle Solaris OS.

Oracle Solaris I/O Multipathing Features

The Oracle Solaris I/O multipathing has the following features:

- Path management – Oracle Solaris I/O multipathing dynamically manages the paths to any storage devices that the Oracle Solaris OS supports. The addition or removal of paths to a device is done automatically when a path is brought online or removed from service.
- Failover support – Implementing higher levels of reliability, availability, and serviceability (RAS) requires redundant host connectivity to storage devices. Oracle Solaris I/O multipathing features manage the failure of storage paths while maintaining host I/O connectivity through available secondary paths.

- Symmetric and asymmetric device support – Oracle Solaris I/O multipathing supports symmetric and asymmetric disk storage devices.
- I/O load balancing – In addition to providing simple failover support, the Oracle Solaris multipathing feature also provides I/O load balancing by routing I/O through multiple host connections.
- 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 Oracle 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.
- `stmsboot` command – The `stmsboot` command allows you to enable or disable the Oracle 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.

Note - For SPARC based system, you must use the `stmsboot` command as a post-installation step because SPARC based systems do not enable multipathing for FC devices by default.

For x86, use of the `stmsboot` command is not mandatory as a post-installation step because X86 based systems enable multipathing of FC devices by default.

- Device driver configuration – Driver customizations are made in the `/etc/driver/drv` directory rather than in the `/kernel/drv` directory so 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 the `fp.conf`, `mpt.conf`, or `scsi_vhci.conf` files should be made in the `/etc/driver/drv` directory.

Benefits of Oracle Solaris I/O Multipathing

- With Oracle Solaris I/O multipathing enabled, you can add more controllers to increase bandwidth and reliability, availability, and serviceability (RAS) without changing device names or modifying applications.
- Oracle storage products do not require you to manage configuration files or update databases. For storage from vendors other than Oracle, see the vendor documentation for methods to enable support and to ensure the product is qualified to run with Oracle Solaris I/O multipathing features.
- Multipathed devices are displayed as a single device instance instead of one device or device link per path. This feature reduces the cost of managing complex storage architectures with utilities because the Oracle Solaris I/O multipathing feature is completely integrated with Oracle Solaris OS. Commands such as the `format` command or a volume

management product sees one representation of a storage device instead of a separate device for each path.

- The failover support in Oracle Solaris I/O multipathing enables you to implement higher levels of RAS, which requires redundant host connectivity while maintaining host I/O connectivity to storage devices.
- Oracle Solaris I/O multipathing can use any active path to a storage device to send and receive I/O. With I/O routed through multiple host connections, you can increase bandwidth by the adding 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.
- The queue depth feature in Oracle Solaris I/O multipathing enables the disk driver to maintain and balance one queue to the LUN, effectively managing the queue depth. No other multipathing software available for the Oracle Solaris OS has this ability.
- Oracle Solaris I/O multipathing supports dynamic reconfiguration (DR) operations.

Storage Technologies Supported

The Oracle 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 Fibre Channel, iSCSI, and SAS storage devices.

Fibre Channel Features

Oracle Solaris I/O multipathing provides the following key features for FC devices:

- 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 Oracle 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 – This library was previously known as the Storage Networking Industry Association Fibre Channel host bus adapter (SNIA FC-HBA) library. The FC-HBA library

application programming interface (API) enables management of FC HBAs and provides a standard-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 the `libhbaapi(3LIB)` man page.

- 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 feature 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 available. The Oracle Solaris FCoE is a software implementation that is designed to work with normal Ethernet controllers.

For more information about FC specifications, go to `http://www.t11.org`.

iSCSI Features

iSCSI is an acronym for Internet SCSI (Small Computer System Interface), an Internet Protocol -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.

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

iSCSI based ZFS storage blocks can be integrated with Cinder OpenStack services. For more information, see Installing and Configuring OpenStack in Oracle Solaris 11.3.

For more information, see Chapter 4, “Configuring Oracle Solaris iSCSI Initiators”.

For information about how to configure targets and initiators in preparation for multipathing, see Chapter 8, “Configuring Storage Devices With COMSTAR” in *Managing Devices in Oracle Solaris 11.3*.

For more information about how to configure Oracle Solaris Zones Storage on an Oracle ZFS Storage Appliance, see How to Consolidate Zones Storage on an Oracle ZFS Storage Appliance.

SAS Software Features

Oracle Solaris I/O Multipathing provides the following features in a SAS domain:

- Dynamic storage discovery – The Oracle Solaris I/O multipathing feature automatically recognizes devices and any modifications made to device configurations. This capability 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 Oracle Solaris I/O multipathing feature maintain their device naming through reboots or reconfiguration.

Note - You must use the `stmsboot` command as a post-installation step to enable multipathing in a SAS domain because SAS multipathing is not enabled by default in the Oracle Solaris OS for either SPARC based or x86 based systems.

For more information about SAS domains, see Chapter 7, “Configuring SAS Devices”

Disk Storage Devices Supported

The Oracle Solaris I/O multipathing feature supports both symmetric and asymmetric disk 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. An asymmetric storage device is one in which paths to the storage device may have different access states. For example, active and standby paths, active or optimized, and active or non-optimized paths.

The following disk storage devices are supported by the Oracle Solaris I/O multipathing feature:

- All Oracle disk storage products, both symmetric and asymmetric.
- All T10 and T11 standards-compliant third-party symmetric disk devices.
- Many third-party asymmetric disk arrays.
- T10 Asymmetric Logical Unit Access (ALUA) supports asymmetricdevices that support this T10 standard. See your vendor documentation to find out whether the device is supported.

Device Standards Supported

Oracle Solaris I/O multipathing 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 Oracle Solaris I/O multipathing:

- 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, FC-LS, and FC-GS
- T11.5 storage management standards, including FC-HBA
- IETF standards, including RFC 2625
- Serial Attached SCSI-2 (SAS2)

♦ ♦ ♦ CHAPTER 2

Configuring Oracle Solaris I/O Multipathing Features

This chapter explains how to configure Oracle Solaris I/O multipathing features 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, tape drives, and third-party storage devices.

This chapter contains the following topics:

Multipathing Considerations

Before you change your multipathing configuration, consider the factors described in this section. You can then follow the instructions for your system architecture (SPARC or x86). 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 to indicate that they are under multipath control.

  The following example shows a device name with multipath disabled:

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

  The following example shows the device name with multipath enabled:

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

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 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 that 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 Oracle Solaris I/O Multipathing

The key component of Oracle Solaris I/O multipathing is a kernel module `scsi_vhci` and a few fops modules, which are delivered in the `/system/kernel` package. The `system/storage/multipath-utilities` package, which delivers `mpathadm` utility, is also installed by default.

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` command:

- You must reboot immediately after running the `stmsboot` command.
- Because the `stmsboot` command reboots the system to complete the operation, use the `eeprom` 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 Oracle Solaris 10 and previous releases, the `stmsboot` command was used to enable or disable multipathing on the boot device for SPARC based hosts only. Starting with Oracle Solaris 11, you can use the `stmsboot` command to enable or disable multipathing on serial-attached SCSI devices and FC devices.

▼ How to Enable Multipathing

This procedure describes how 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, see "Configuring Multipathing by Port Basis" on page 26.

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 19.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **If the multipathing software package is not installed, install it.**

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

   To check whether the package is installed, issue the following command:

```
# 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
```

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
```

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

4. **If necessary, configure your applications to use new device names as described in "Multipathing Considerations" on page 19.**

How to Disable Multipathing

This procedure describes how 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, see "Configuring Multipathing by Port Basis" on page 26.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **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
```

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

3. **If necessary, configure your applications to use new device names as described in "Multipathing Considerations" on page 19.**

How to Disable Multipathing for a Specific Device

Multipathing can be disabled for a specific device with a certain vendor ID and product ID combination. This exclusion is specified in the `scsi_vhci.conf` file.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 a comma (,) and the last vendor/product entry is terminated by 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";
```

For more information about tuning the target disk driver properties, see Appendix D, “Tuning Disk Target Driver Properties”.

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 Oracle Solaris I/O Multipathing” on page 20.**

Displaying Device Name Changes

You can display the mapping between non-multipathed and multipathed device names after changes are made to the multipath configuration through the `stmsboot` command. Both non-multipathed and the multipathed device names must exist in order to show the mapping.

Use the `-L` option to display the mapping of devices on all controllers. For example:

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

Use the `-l` option to display 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/rdsk/c3t8d0    /dev/rdsk/c10t500000E01046DEE0d0
/dev/rdsk/c3t0d0    /dev/rdsk/c10t500000E01046B070d0
```

```
/dev/rdsk/c3t3d0    /dev/rdsk/c10t20000020372A40AFd0
/dev/rdsk/c3t12d0    /dev/rdsk/c10t500000E01046DEF0d0
/dev/rdsk/c3t11d0    /dev/rdsk/c10t500000E01046E390d0
```

Determining Whether Multipathing Is Enabled

You can determine whether multipathing is enabled by issuing the following command as an administrator.

The following example shows that multipathing is disabled on this system because the command returns no multipathed device information.

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

The following example shows that multipathing is enabled on this system because the command returns multipathed device information.

```
# prtconf -vc /devices/scsi_vhci |grep dev_link.*s2
     dev_link=/dev/dsk/c0t5000C500335DC60Fd0s2
     dev_link=/dev/rdsk/c0t5000C500335DC60Fd0s2
     dev_link=/dev/dsk/c0t5000C500335E106Bd0s2
     dev_link=/dev/rdsk/c0t5000C500335E106Bd0s2
     dev_link=/dev/dsk/c0t5000C500335BA8C3d0s2
     dev_link=/dev/rdsk/c0t5000C500335BA8C3d0s2
     dev_link=/dev/dsk/c0t5000C500335FC3E7d0s2
     dev_link=/dev/rdsk/c0t5000C500335FC3E7d0s2
     dev_link=/dev/dsk/c0t50015179594B6F11d0s2
     dev_link=/dev/rdsk/c0t50015179594B6F11d0s2
     dev_link=/dev/dsk/c0t5000C500335F95E3d0s2
     dev_link=/dev/rdsk/c0t5000C500335F95E3d0s2
     dev_link=/dev/dsk/c0t5000C500335F907Fd0s2
     dev_link=/dev/rdsk/c0t5000C500335F907Fd0s2
     dev_link=/dev/dsk/c0t5000C500335BD117d0s2
     dev_link=/dev/rdsk/c0t5000C500335BD117d0s2
```

To confirm that MPxIO is enabled for a LUN, you can use the `format` command and check for the occurrence of the `/scsi_vhci` directory in the logical path of the LUN.

```
# format
c6t60060E800561CF00000061CF00000002d0
        /scsi_vhci/ssd@g60060e800561cf00000061cf00000002
```

You can also check for the status of `mpxio` in the configuration file to confirm whether MPxIO is disabled:

```
# egrep "^mpxio-disable=" /etc/driver/drv/*.conf
/etc/driver/drv/fp.conf:mpxio-disable="no";
```

Configuring Multipathing by Port Basis

You can enable or disable multipathing 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 factors:

- 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

This procedure describes how to configure multipathing by port. It 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 sample `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. **Enable or disable specific FC HBA ports.**

   - To enable an FC HBA port, add the following line to the `/etc/driver/drv/fp.conf` file:

     ```
     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 mpxio-disable="no";
     ```

   - To disable an FC HBA port, add the following line to the `/etc/driver/drv/fp.conf` file:

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

     For example:

     ```
     name="fp" parent="/pci@6,2000/SUNW,qlc@2" port=0 mpxio-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. **If necessary, configure your applications to use new device names as described in "Multipathing Considerations" on page 19.**

Configuring Multipathing on Tape Drives

Tape I/O multipathing is disabled by default on both SPARC and x86 based systems. Tape I/O multipathing intentionally does not support load-balancing because only one path is used for sending I/O.

When using Tape I/O multipathing, check if the tape device is supported by Oracle Solaris I/O multipathing. If one of the following conditions are true, a drive will be enumerated under `scsi_vhci` and uses the `sg_inq` command to retrieve drive's inquiry data:

- Drive's inquiry data has set any of the Target Port Group Support (TPGS) bits.
- Drive's inquiry data has set the Multi Port (MultiP) bit.
- Internal `VID` and `PID` table for a specific fops plug-in lists a drive's inquiry Vendor Identification (VID) and Product Identification (PID).

  You can use the following procedure to retrieve internal `VID` and `PID` table:

  1. Enable `scsi_vhci_f_tape` in the `scsi_vhci.conf` file. For more information, see "How to Configure Multipathing on Tape Drives" on page 29.
  2. Reboot the system.
  3. Check for the internal vendor and product information.

     ```
     # mpathadm show mpath-support libmpscsi_vhci.so | ggrep -B 1 SAMPLE
     Vendor:  EXAMPLE.Inc
     Product:  SAMPLE-4A3
     --
     Vendor:  EXAMPLE.Inc
     Product:  SAMPLE-4DT
     ```

- User adds the `VID` and `PID` pair to the `scsi_vhci.conf` file and specifies a fops plug-in known to the system. For example:

  ```
  scsi-vhci-failover-override =
      "VENDOR1 ProductA", "f_tape";
  ```

  For more information about how to set the property `scsi-vhci-failover-override`, see "How to Add Third-Party Devices" on page 30.

The following procedure describes how to enable or disable multipathing for tape drives. This procedure applies to both SPARC and x86 based systems starting with Oracle Solaris 11 OS.

▼ How to Configure Multipathing on Tape Drives

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Copy the file `/kernel/drv/scsi_vhci.conf` to `/etc/driver/drv/scsi_vhci.conf`.**

3. **Edit the `/etc/driver/drv/scsi_vhci.conf` file to configure multipathing.**

   - If the drive's inquiry data has set any of the TPGS bits, uncomment the following line:

     ```
     # "misc/scsi_vhci/scsi_vhci_f_tpgs_tape",
     ```

   - If the drive's inquiry data has set MultiP bit, uncomment the following line:

     ```
     # "misc/scsi_vhci/scsi_vhci_f_tape",
     ```

Note - Starting with Oracle Solaris 11, the configuration files are read from both `/kernel/drv` and `/etc/driver/drv` and then they are merged. For more information, see the `/kernel/drv/README.driver` file.

The configuration files in the `/etc/driver/drv/` directory must contain the customized configuration.

After enabling tape multipathing, existing device nodes under `/dev/rmt` is no longer valid because they will point to devices that is no longer existing after reboot.

The tape I/O multipathing utilizes only one path for I/O at a time even though there are multiple paths to the tape device. The remaining paths are used for automatic fail-over.

Configuring Third-Party Storage Devices

Before configuring any third-party device, ensure that it is 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, consider the following factors::

- The device must support the `REPORT_LUNS` SCSI command, and SCSI-3 `INQUIRY` command VPD Device Identification Page (0x83).
- You must know 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) man page.

When multipathing is enabled, 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 non-compliant devices, a failover implementation's probe will typically determine support based on a 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.

▼ How to Add Third-Party Devices

You can configure multipathing 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.

This procedure describes how to configure third-party devices if your system already has multipathing enabled. If your system has multipathing disabled, see "How to Enable Multipathing" on page 21.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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", "f_sym",
    "VendorID2ProductID2", "f_sym",
    ...
    "VendorIDnProductIDn", "f_sym";
```

Note - The entries are separated by a comma (,) and the last vendor/product entry is terminated by 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";
```

For more information about tuning the target disk driver properties, see Appendix D, “Tuning Disk Target Driver Properties”.

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 Oracle Solaris I/O Multipathing” on page 20.**

Configuring Automatic Failback

Some storage devices have controllers configured as primary and secondary as part of the array configuration. The secondary paths might 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 the `luxadm`(1M) man page.

▼ How to Configure Automatic Failback

1. **Become an administrator.**

   For more information, see “Using Your Assigned Administrative Rights” in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Copy the `/kernel/drv/scsi_vhci.conf` file to the `/etc/driver/drv/scsi_vhci.conf` file.**

3. **In the `/kerneldrv/scsi_vhci.conf` file, enable or disable automatic failback capability by changing the `auto-failback` entry.**

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

   or

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

4. **Save and exit the file.**

5. **Reboot the system.**

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

Configuring Automatic Failback for Multipathing Support

Symmetric 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 31.

How to Fail Over a Logical Unit

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 logical unit (LU) access

1. Determine the failover support.

```
# 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:
.
.
.
```

2. Display a list of multipathed LUs.

```
# mpathadm list lu
        /dev/rdsk/c0t600144F08069703400004E828EE10004d0s2
                Total Path Count: 8
                Operational Path Count: 8
        /dev/rdsk/c0t600144F08069703400004E8183DF0002d0s2
                Total Path Count: 8
```

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

3. Display a specific LU's configuration information.

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

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

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

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

                Initiator Port Name:  210000e08b841feb
                Target Port Name:  200900a0b826d63c
```

```
                Override Path:  NA
                Path State:  OK
                Disabled:  no

        Target Port Groups:
                ID:  5
                Explicit Failover:  yes
                Access State:  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
```

4. Manually force an LU failover.

```
# mpathadm failover lu /dev/rdsk/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.

5. Confirm the access state change.

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

        Paths:
                Initiator Port Name:  210000e08b841feb
                Target Port Name:  200800a0b826d63b
```

```
                Override Path:  NA
                Path State:  OK
                Disabled:  no

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

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

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

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

                        Name:  200800a0b826d63c
                        Relative ID:  0

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

                        Name:  200900a0b826d63c
                        Relative ID:  0
```

▼ How to Enable an LU Path

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

1. **Display a list of multipathed LUs.**

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

2. **Display the selected LU's configuration information.**

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

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

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

```
                Path State:  OK
                Disabled:  no

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

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

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

                        Name:  200800a0b826d63c
                        Relative ID:  0

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

                        Name:  200900a0b826d63c
                        Relative ID:  0
```

3. **Enable the LU path.**

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

▼ How to Disable an 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/rdsk/c0t600144F08069703400004E828EE10004d0s2
                Total Path Count: 8
                Operational Path Count: 8
        /dev/rdsk/c0t600144F08069703400004E8183DF0002d0s2
                Total Path Count: 8
                Operational Path Count: 8
        /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800029065C00007CF44E233FCFd0s2
                Total Path Count: 4
                Operational Path Count: 4
.
.
.
```

Note - After a system is installed with the Oracle Solaris OS and Oracle Solaris I/O multipathing is enabled, the multipathed device names begin with `c0`. For example:

```
# mpathadm list lu
        /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800029065C00007CF44E233FCFd0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800026D63A0000A4974E23424Ed0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800029065C00007CF34E233F89d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800026D63A0000A4964E234212d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800026D63A000099B94DE2DB34d0s2
```

2. Display the specific LU's configuration information.

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

        Paths:
                Initiator Port Name:  210000e08b841feb
                Target Port Name:  200800a0b826d63b
                Override Path:  NA
                Path State:  OK
```

```
                Disabled:  no

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

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

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

        Target Port Groups:
                ID:  8
                Explicit Failover:  yes
                Access State:  standby
                Target Ports:
                        Name:  200800a0b826d63b
                        Relative ID:  0

                        Name:  200800a0b826d63c
                        Relative ID:  0

                ID:  18
                Explicit Failover:  yes
                Access State:  active
                Target Ports:
                        Name:  200900a0b826d63b
                        Relative ID:  0

                        Name:  200900a0b826d63c
                        Relative ID:  0
```

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

4. **Disable the selected LU path.**

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

Displaying Multipathing Information

You determine and configure Oracle Solaris OS multipathing support by using the `mpathadm` command, which 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.

This section describes the following tasks to administer multipathing devices:

- "Displaying Multipathing Support Information" on page 43
- "Configuring Automatic Failback for Multipathing Support" on page 32

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 are 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 are supported by the multipathing support software. The multipathing support `libmpscsi_vhci.so` library file supports T10 target port group compliant devices by default.

4. (Optional) Display the path information for 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
.
.
.
```

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

1. Display a list of LUs.

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

2. Display specific LU information to determine the target ports.

```
# mpathadm show lu /dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
Logical Unit:  /dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
        mpath-support:  libmpscsi_vhci.so
        Vendor:  SUN
        Product:  CSM200_R
        Revision:  0660
        Name Type:  unknown type
        Name:  600a0b800029065c00007cf54e234013
        Asymmetric:  yes
        Current Load Balance:  round-robin
        Logical Unit Group ID:  NA
        Auto Failback:  on
        Auto Probing:  NA
```

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

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

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

                        Name:  200800a0b826d63c
                        Relative ID:  0

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

                        Name:  200900a0b826d63c
                        Relative ID:  0
```

3. Display the specific target port information.

```
# mpathadm list lu -t 20030003ba27d212
mpath-support:  libmpscsi_vhci.so
        /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
                Total Path Count: 4
                Operational Path Count: 4
mpath-support:  libmpscsi_vhci.so
        /dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
                Total Path Count: 4
                Operational Path Count: 4
mpath-support:  libmpscsi_vhci.so
        /dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
                Total Path Count: 4
```

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

▼ How to Display an LU With a Specific Name

You can 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.

You can 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/rdsk/c0t600144F08069703400004E828EE10004d0s2
                Total Path Count: 8
                Operational Path Count: 8
        /dev/rdsk/c0t600144F08069703400004E8183DF0002d0s2
                Total Path Count: 8
                Operational Path Count: 8
        /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
                Total Path Count: 4
                Operational Path Count: 4
        /dev/rdsk/c0t600A0B800029065C00007CF44E233FCFd0s2
                Total Path Count: 4
                Operational Path Count: 4
.
.
.
```

2. Display a specified LU's configuration information.

```
# mpathadm show lu /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
```

```
Logical Unit:  /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
        mpath-support:  libmpscsi_vhci.so
        Vendor:  SUN
        Product:  CSM200_R
        Revision:  0660
        Name Type:  unknown type
        Name:  600a0b800026d63a0000a4994e2342d4
        Asymmetric:  yes
        Current Load Balance:  round-robin
        Logical Unit Group ID:  NA
        Auto Failback:  on
        Auto Probing:  NA

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

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

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

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

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

                        Name:  200800a0b826d63c
                        Relative ID:  0

                ID:  13
                Explicit Failover:  yes
```

```
                Access State:  standby
                Target Ports:
                        Name:  200900a0b826d63b
                        Relative ID:  0

                        Name:  200900a0b826d63c
                        Relative ID:  0
```

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/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
                Total Path Count: 4
                Operational Path Count: 4
```

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

♦ ♦ ♦ CHAPTER 3

Configuring Fabric-Connected Devices

This chapter provides a high-level overview describing the configuration of fabric-connected (FC) devices. In the Oracle Solaris OS, FCAL, fabric, and point-to-point connected devices are made available to the system automatically. For more information about how to manually configure fabric-connected devices, see Appendix A, “Manual Configuration for Fabric-Connected Devices”.

This chapter covers the following topics:

Configuring FC Devices With Multipathing Considerations

Before you configure Oracle 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. For more information about LUN masking, see your vendor-specific storage documentation.
- Power management needs to be disabled for hosts and devices on a SAN. For more information about power management, see the `poweradm`(1M) man page.
- Connect arrays and other storage devices to the SAN with or without multipathing capability. Oracle Solaris multipathing is an associated application that is bundled with the product.
- The STMS boot utility is included with the Oracle 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` command. However, to make those devices permanently unavailable to the system, you can use 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”.

Adding FC Devices

The following commands are commonly used when adding and removing FC devices:

`cfgadm` and `cfgadm_fp`	Dynamically reconfigures devices and FC devices. These commands are used most frequently to configure storage devices on a SAN. For more information, see the `cfgadm`(1M) and `cfgadm_fp`(1M) man pages.
`fcadm`	Collects administrative information on Fibre Channel Host Bus Adapters (HBA) ports on a host. For more information, see the `fcadm`(1M) man page.
`fcinfo`	Collects administrative information on Fibre Channel Host Bus Adapters (HBA) ports on a host. For more information, see the `fcinfo`(1M) man page.
`format`	Identifies devices that are connected to the system and provides the ability to label a disk. For more information, see the `format`(1M) man page.
`luxadm`	Administers storage devices and FC_AL devices. For more information, see the `luxadm`(1M) man page.

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

The basic steps for adding an FC device are as follows:

1. Become an administrator. For more information, see “Using Your Assigned Administrative Rights” in *Securing Users and Processes in Oracle Solaris 11.3*.
2. Create the LUNs on the FC device.

3. If necessary, apply LUN masking for HBA control on the FC device. For more information about LUN masking, see your vendor-specific storage documentation.
4. Connect the storage device to the system.
5. If necessary, create port-based or WWN (World Wide Name) zones on the switch on the FC device.
6. Mount any existing file systems available on the storage device's LUNs or disk groups.

Configuring Fabric Boot Devices on SPARC Platform

You can set up a SPARC server that is running the Oracle Solaris OS to be booted from a fabric disk device. Fabric boot devices can be accessed through the Oracle Solaris installation methods just as internal boot devices have been accessed in the previous Oracle Solaris OS releases. For more information about Oracle Solaris installation, see *Installing Oracle Solaris 11.3 Systems*.

Consider the following points while enabling a fabric boot device:

- Minimize interference to boot devices through the following actions:
  - Ensure that the boot device is not an overly subscribed target or LUN
  - Avoid installing applications and software on a target or LUN
  - Reduce the 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 Oracle Solaris I/O Multipathing Features".

♦ ♦ ♦ CHAPTER 4

Configuring Oracle Solaris iSCSI Initiators

This chapter describes how to configure Oracle Solaris iSCSI initiators in Oracle Solaris OS. For information about the procedures associated with configuring iSCSI initiators, see "Configuring iSCSI Initiators" on page 58.

This chapter contains the following topics:

For information about configuring Oracle Solaris iSCSI targets with COMSTAR, see Chapter 8, "Configuring Storage Devices With COMSTAR" in *Managing Devices in Oracle Solaris 11.3*.

For more information about installing and booting an iSCSI disk, see Step 7 in "How to Perform a GUI Installation" in *Installing Oracle Solaris 11.3 Systems*.

About iSCSI Technology in Oracle Solaris

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

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

- iSCSI block devices or tapes – Translates SCSI commands and data from the block level into IP packets. You can have block-level access between one system and the target device, such as a tape device or a database, by using iSCSI in your network. 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.
- Network File System (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.

- iSCSI Extensions for RDMA (iSER) – Accelerates the iSCSI protocol by mapping the data transfer phases to Remote Direct Memory Access (RDMA). It allows data transfer directly between the iSCSI nodes without intermediate data copies.

iSCSI Software and Hardware Requirements in Oracle Solaris

The iSCSI software and hardware requirements in Oracle Solaris are as follows:

- Oracle Solaris storage software and devices
  - The `system/storage/iscsi/iscsi-initiator` software package for the iSCSI initiator management utilities
  - The `system/storage/iscsi/iscsi-target` software package for the iSCSI target management utilities.
- Any supported NIC

Benefits of Using iSCSI Devices in Oracle Solaris

Using iSCSI initiators and targets in Oracle Solaris has the following benefits:

- 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.
- Existing Fibre Channel devices can be connected to iSCSI 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.

Limitations of Using iSCSI Devices in Oracle Solaris

The limitations of using the iSCSI initiator software in Oracle Solaris are as follows:

- iSCSI devices that use Service Location Protocol (SLP) are not currently supported.

- iSCSI targets cannot be configured as dump devices.
- Transferring large amounts of data over your existing network can affect performance.
- The iSCSI initiator subsystem runs only on the global zone and not on 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.

iSCSI Concepts

Familiarity with the following concepts is helpful when configuring iSCSI targets and initiators.

Discovery	The process that presents the initiator with a list of available targets.
Discovery method	The way in which the iSCSI targets are found. The three discovery methods are: ■ 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 discovery address. ■ Static – Static target addressing is configured.
Initiator	The driver that initiates SCSI requests to the iSCSI target.
`iqn` or `eui` address format	An iSCSI device name can be in either iSCSI qualified name (`iqn`) format or extended unique identifier (`eui`) address format. An `iqn` format address is the unique identifier for a device in an iSCSI network using the form *iqn.date.authority:unique-ID*. An Oracle Solaris iSCSI initiator is assigned an IQN name automatically. An `eui` format address consists of 16 hexadecimal digits, and identifies a class of GUIDs that is used in both the SCSI and InfiniBand standards.
Logical Unit (LU)	A uniquely numbered component in a storage system. When an 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 portal	A target device that is identified by its IP address and its listening TCP port.
Target portal group	A group of target portals used to access the same set of iSCSI targets.

Best Practices for Configuring iSCSI

Consider the following iSCSI recommendations or considerations before configuring iSCSI devices in your network.

- Deploy iSCSI devices in fast and dedicated Ethernet network.
- Use jumbo frames to allow more data transfer in each Ethernet transaction and reduce the number of frames.
- Use multiple sessions per iSCSI target (MS/T) to increase performance. For more information about how to create multiple sessions for a target, see "How to Enable Multiple iSCSI Sessions for a Target" on page 71.

Note - MS/T provides better performance than multiple connections per session (MC/S) in Oracle Solaris.

- Spread iSCSI sessions across multiple physical links or subnets for higher availability of iSCSI devices.
- Use IP network multipathing (IPMP) for load spreading and better failover mechanism. For more information about failover mechanism in IPMP, see Chapter 1, "Administering TCP/IP Networks" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.3*.
- Use iSNS target discovery to simplify configurations and to enhance security in a large setup. For more information about administer iSNS discovery method, see "Administering Internet Storage Name Service Target Discovery Method" on page 60.
- Use CHAP authentication for additional security. For more information about configuring CHAP authentication, see "Configuring Authentication in an iSCSI-Based Storage Network" on page 63.

Configuring iSCSI Initiators

Tasks associated with configuring iSCSI initiators are described in the following sections:

For more information about how to configure and administer iSCSI initiator, see the `iscsiadm`(1M) man page.

You can also find related information in the following documents:

- *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.3*
- Chapter 8, "Configuring Storage Devices With COMSTAR" in *Managing Devices in Oracle Solaris 11.3*

Target Discovery Methods

Determine whether you want to configure one of the dynamic device discovery methods or to use static configuration.

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

- SendTargets – Target IP address, optionally TCP port
- iSNS – iSNS server address, optionally TCP port
- Static – Target IP address and target name, optionally TCP port and target portal group tag

Administering SendTargets Discovery Method

SendTargets is one of the dynamic target discovery method. 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 enable the iSCSI initiator to use the SendTargets features to perform the device discovery.

▼ How to Configure SendTargets Discovery Method

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Verify that the iSCSI initiator service is online.**

   ```
   initiator# svcs network/iscsi/initiator
   ```

3. **Configure the SendTargets discovery method by providing the discovery address.**

   ```
   initiator# iscsiadm add discovery-address 192.0.2.1
   ```

   If the default port 3260 is not used, obtain the target IP address and optionally TCP port.

4. **Enable the SendTargets discovery method.**

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

Administering Internet Storage Name Service Target Discovery Method

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 11.3*.

The Internet Storage Name Service (iSNS) enables 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 enable 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 at `http://www.ietf.org/rfc/rfc4171.txt`.

How to Configure Internet Storage Name Service Target Discovery Method

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Verify that the iSCSI initiator service is online.**

   ```
   initiator# svcs network/iscsi/initiator
   ```

3. **Obtain iSNS server IP address and optionally TCP port if the default port 3205 is not used.**

   For more information about how to retrieve this information, see the specific vendor documentation.

4. **Configure the iSNS discovery method, provide the IP address of the iSNS server.**

   ```
   initiator# iscsiadm add isns-server 192.0.2.2
   ```

5. **Enable the iSNS discovery method.**

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

Administering Static Configuration Discovery Method

If an iSCSI node has few targets, you can restrict the targets that the initiator attempts to access.

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 lead to slow performance, when the initiator is communicating with the iSCSI target device.

▼ How to Configure an iSCSI Target Statically

1. **Become an administrator.**
   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Verify that the iSCSI initiator service is online.**

   ```
   initiator# svcs network/iscsi/initiator
   ```

3. **Obtain the target IP address and target name from the iSCSI storage device management interface.**
   For more information about how to retrieve this information, see the specific vendor documentation.

4. **Configure the iSCSI target to be statically discovered.**

   ```
   initiator# iscsiadm add static-config iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-
   c3b36c123456,
   192.168.35.2
   ```

5. **Review the static configuration information.**

   ```
   initiator# iscsiadm list static-config
   Static Configuration Target: iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-
   c3b36c123456,
   192.168.35.2:3260
   ```

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

Accessing iSCSI Disks

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 becoming an administrator and by using the `format` utility or the `iscsiadm list target` command.

To display and review the iSCSI LU information by using the `iscsiadm list target` command:

```
initiator# iscsiadm list target -S | grep OS Device Name | sort -u
            OS Device Name: /dev/rdsk/c0t600144F019C1CA000000531F5C0B0001d0s2
```

```
            OS Device Name: /dev/rdsk/c0t600144F019C1CA000000531F5C130002d0s2
            OS Device Name: /dev/rdsk/c0t600144F0D1720B0000005400948600001d0s2
```

To display and review the iSCSI LU information by using the `format` command:

```
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 c0t600144F0B5418B0000004DDAC7C10001d0
[disk formatted]
```

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

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

The ZFS file system is automatically mounted when created and is remounted at boot time. For more information about ZFS file system, see *Managing ZFS File Systems in Oracle Solaris 11.3*.

Removing Discovered Target Devices

The associated targets are logged out after you perform any one 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, if they have mounted file systems, the logout of these devices will fail, and they will remain on the active target list.

To accomplish these sections, you must become an administrator and be logged in to the local system where access to a target device has already been configured.

Use the following commands to disable an iSCSI target discovery method.

- To disable the SendTargets discovery method:

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

- To disable the iSNS discovery method:

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

- To disable the static target discovery method:

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

Use the following commands to remove an iSCSI device discovery entry.

- To remove an iSCSI SendTargets discovery entry:

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

- To remove an iSCSI iSNS discovery entry:

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

- To remove a static iSCSI discovery entry:

```
initiator# iscsiadm remove static-config target-name
```

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 an iSCSI-Based Storage Network

In a secure environment, authentication for your iSCSI devices 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 from a given host. In this case, the iSCSI target can authenticate an initiator by using the Challenge-Handshake Authentication Protocol (CHAP).

CHAP authentication uses the notion of challenge and response, which means that the target challenges the initiator to prove its identity. For the challenge and response method to work, a target must know the initiator's secret key, and the initiator must be set up to respond to a challenge. See your array vendor 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 key must be defined.

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 an iSCSI Configuration" on page 66.

How to Configure CHAP Authentication for an iSCSI Initiator

You can set up either bidirectional or unidierctional CHAP authentication. This procedure assumes that you are logged in to the local system where you want to securely access the configured iSCSI target device.

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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Set the secret key on the initiator.**

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

   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.

3. **(Optional) Set a custom CHAP user name on the initiator.**

   In the Oracle Solaris OS, the CHAP user name is set to the initiator or target node name (the `iqn` name) by default. The CHAP user name can be set to any length of text that is less than 512 bytes.

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

4. **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.

This step completes the setup of unidirectional CHAP authentication. Continue with the remaining steps if you wan to set up bidirectional CHAP authentication.

5. **Enable bidirectional CHAP for connections with the target.**

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

6. **Set the authentication method to CHAP for the target.**

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

7. **Set the target device secret key that identifies the target.**

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

8. **If the target uses a custom CHAP user name, set the CHAP name that identifies the target.**

By default, the CHAP name of the target is set to the target name. If the target uses a custom name, set the CHAP name that identifies the target.

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

▼ How to Configure CHAP Authentication for an iSCSI Target

You can set up either bidirectional or unidirectional CHAP authentication. This procedure assumes that you are logged in to the local system that contains the iSCSI targets.

1. **Become an administrator.**

For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Configure the target to require that initiators identify themselves by using CHAP authentication.**

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

3. **Create an initiator context that describes the initiator with initiator's full node name and initiator's CHAP secret key.**

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

4. **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
```

This step completes the setup of unidirectional CHAP authentication. Continue with the remaining steps if you want to set up bidirectional CHAP authentication.

5. **Set the target device secret key that identifies this target.**

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

6. **If the target uses an custom CHAP user name, set the CHAP name that identifies the target.**

By default, the CHAP name of the target is set to the target node name (`iqn`). If the target uses a custom name, set the CHAP name that identifies the target.

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

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

You can use a third-party RADIUS (Remote Authentication Dial In User Service) 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. RADIUS can be independently configured on either the initiator or the target.

▼ How to Configure a RADIUS Server for an iSCSI Target

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

1. **Become an administrator.**

For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 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 information:**

   - 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, initiator's `iqn` name) and the secret key for each initiator that needs to be authenticated

▼ How to Configure a RADIUS Server for an iSCSI Initiator

This setup is useful only when the initiator is requesting bidirectional CHAP authentication.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 so that iSCSI can 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 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 information:**

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

Oracle Solaris iSCSI and RADIUS Server Troubleshooting

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

`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 an iSCSI Target" on page 66.

`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 an iSCSI Target" on page 66.

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.

  Note - You cannot use IPMP with an iSCSI boot device.

  For information about configuring multiple sessions per target, see "How to Enable Multiple iSCSI Sessions for a Target" on page 71. For information about configuring IPMP, see Chapter 3, "Administering IPMP" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.3*.
- **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 that match the MPxIO configuration that you want are supported. For example, in the `fp.conf` file, you can determine whether MPxIO is enabled globally on the HBA or on a per-port basis.

- **Third-party hardware considerations** – Find out whether your third-party HBA is qualified to work with Oracle Solaris iSCSI and MPxIO.

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

Enabling Multiple iSCSI Sessions per Target

The multiple session per target (MS/T) strategy provides better performance than using multiple connections per session. The use of MS/T and creating multiple TCP connections ensure the better usage of the networking stack. This feature also ensures better performance by sending and receiving multiple threads.

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.

- A typical MS/T configuration has two or more configured-sessions.

  However, if your storage supports multiple target portal group tag (TPGTs) and if you are using the SendTargets discovery method on your host system, then the number of configured sessions can be set to 1. SendTargets discovery automatically detects the existence of multiple paths, and multiple target sessions are created.
- Confirm that the `mxpio` 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 available and configured by using IPMP.

  ```
  # ipadm show-addr
  ```

Consider the following factors before configuring multiple sessions for an iSCSI target:

For more information about how iSCSI interacts with MPxIO paths, see “Setting Up iSCSI Multipathed Devices in Oracle Solaris” on page 69.

How to Enable Multiple iSCSI Sessions for a Target

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **List the current parameter values for the iSCSI initiator and the target.**

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

      For example:

      ```
      initiator# iscsiadm list initiator-node
      Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
      Initiator node alias: zzr1200
              .
              .
              .
              Configured Sessions: 1
      ```

   b. **List the current parameter values 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: -
              .
              .
              .
              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. **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.**

   - To apply the desired parameter value to the iSCSI initiator node:

     For example:

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

   - To apply the desired parameter value to the iSCSI target:

     For example:

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

4. **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 192.0.2.1,198.51.100.1
   ```

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

5. **Verify that the parameter was modified.**

   a. **Display the updated information for the initiator node.**

      For example:

      ```
      initiator# iscsiadm list initiator-node
      Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
      Initiator node alias: zzr1200
              .
              .
              .
              Configured Sessions: 2
      ```

   b. **Display the updated information for the target node.**

      For example:

      ```
      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
      ```

6. **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
   ```

Displaying iSCSI Configuration Information

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

The following example shows how to display information about the iSCSI initiator.

```
# 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
```

The following example shows how to display information about which discovery methods are in use.

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

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
                  Transport Type: socket
                  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, only the following parameters that can be modified on the iSCSI initiator:

- iSCSI initiator node 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 76.

- 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 an iSCSI Initiator" on page 64.
- Configured sessions – For more information about configuring multiple sessions, see "How to Enable Multiple iSCSI Sessions for a Target" on page 71.

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 vendor 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 when you issue the `iscsiadm modify` command.

Tuning iSCSI Parameters

You can tune iSCSI parameters to adjust various response or connection time values of the iSCSI initiator for all targets either which the initiator system is connected or to for a specific target.

To change a parameter value of a specific iSCSI target:

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

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
```

To adjust a parameter value for all targets:

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

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

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

The tunable parameters listed in the following table apply to the active connection, and change the behavior of the iSCSI initiator and the targets that connect to the initiator.

TABLE 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 Modify iSCSI Initiator and Target Parameters

This procedure illustrates how modified parameters of the iSCSI initiator are inherited by the iSCSI target device and 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **List the current parameter values for the iSCSI initiator and the target device.**

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

      For example:

```
initiator# 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
```

b. List the current parameter values 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: 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 "Displaying iSCSI Configuration Information" on page 72.

3. Modify the parameter of the iSCSI initiator.

For example, set the header digest to `CRC32`.

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

Note - 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. Also, if the new name does not belong to the same discovery domain as that of the targets, the targets might be logged out and removed from the initiator's target list. 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.

For example:

```
initiator# 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/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
          Transport Type: socket
          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.

   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...
   ```

   Tip - Use the `iscsiadm remove` command to reset all parameters to their default values.

   For information about the `iscsiadm remove target-param` command, see the `iscsiadm`(1M) man page.

Troubleshooting iSCSI Configuration Problems

You can use the following tools to troubleshoot general iSCSI configuration problems:

- `snoop` – This tool has been updated to support iSCSI packets.
- `wireshark` – This network monitoring is available from `http://www.wireshark.org/`.

Both tools can filter iSCSI packets on port `3260`.

This section describes how to troubleshoot various iSCSI issues and error messages.

No Connections to the iSCSI Target From the Local System

To identify possible causes for a connection problem, become an administrator and 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
```

- If no connections are listed in the output, check the `/var/adm/messages` file for possible reasons why the connection failed.

  You can verify whether the connection is accessible by using the `ping` command. You can also verify 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.

- If your target is not listed in the 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

To troubleshoot iSCSI device or disk unavailability, become an administrator and identify the LUNs that 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/rdsk/c3t34d0s2
```

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

- 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 the `/var/adm/messages` log file.
- Check the storage device's log files for errors.
- 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.

See your vendor specific storage documentation for more information about LUN masking.

General iSCSI Error Messages

This section describes the iSCSI messages that might appear 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#*)

TYPE	Either connection or session.
OID	The object ID of the connection or session. This ID is unique for an OS instance.
STRING	A description of the condition.
STATUS-CLASS#/ STATUS-DETAIL#	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 is 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: 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 your storage documentation or 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 your storage documentation or 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 your storage documentation or 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 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 your storage documentation or 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 your storage documentation or 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 configured to accept only 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 configured to accept only 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 the `devfsadm`(1M) man page.

`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 error number related to the connection failure, see the `/usr/include/sys/errno.h` file.

♦ ♦ ♦ CHAPTER 5

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?

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 virtualizing Fibre Channel ports are as follows:

- NPIV ports must not be used for booting on bare metal systems.
- 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 or point-to-point topology.

- NPIV ports cannot be created on FC single root I/O virtualization (SR-IOV) virtual functions.
- 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.**
   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Create an NPIV port.**

   ```
   # fcadm 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 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Delete an NPIV port.**

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

Example 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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

   ```
   # fcinfo hba-port
   ```

   The NPIV port list shows the lists of currently configured NPIV ports.

Example 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
```

♦ ♦ ♦ CHAPTER 6

Configuring FCoE Ports

This chapter describes how to configure FCoE ports hosted on normal Ethernet interfaces. This information does not apply to hardware FCoE ports on Converged Network Adapters (CNA).

This chapter covers the following topics:

What Is FCoE?

FCoE is a new T11 standard that transports encapsulated Fibre Channel frames over Enhanced Ethernet. FCoE enables network convergence and cost-effective SAN expansion in large deployments..

Limitations of Oracle Solaris FCoE

The Oracle Solaris FCoE initiator is a software implementation that works with normal Ethernet controllers. Note the following limitations of Oracle Solaris FCoE implementations:

- FCoE ports cannot be used for booting.
- FCoE ports cannot be configured in Oracle VM Server for SPARC platform 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.

Creating an FCoE Port

Before you create an FCoE port, 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 about 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 world wide name (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
  ```

As an administrator, issue the following command to create an FCoE port.

 # fcadm 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.

 # fcadm create-fcoe-port -i -f *Ethernet-interface*

For example:

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

Deleting an FCoE Port

As an administrator, delete an FCoE port by issuing the following command:

```
# fcadm delete-fcoe-port network-interface
```

For example:

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

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

Displaying FCoE Port Status

As an administrator, display the status of currently configured FCoE ports by issuing the following command:

```
# 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
```

To list FC specific information for all FCoE ports in the system:

```
# fcadm list-fcoe-ports
```

For example:

```
# fcadm 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
```

Forcing an FCoE Port Reinitialization

You might need to force an FCoE port reinitialization when new FC 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 the command issued on the target port side, the target port is reset. When the 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 does not cause any data loss.

As an administrator, force a link that is connected to a port to reinitialize by issuing the following command:

```
# fcadm force-lip port-number
```

For example:

```
# fcadm force-lip 200000144fc2d508
```

Configuring Oracle Solaris to Work With CEE-DCBX FCoE Switches

Oracle Solaris supports the IEEE 802.1qaz Data Center Bridging Exchange (DCBX) specification and also the pre-standard Converged Enhanced Ethernet (CEE) DCBX specification v1.01 to enable interoperation with a larger set of switches when using Data Center Bridging (DCB).

You can set the DCB mode of operation by changing the value of the `dcbx-version` property on Oracle Solaris. The `dcbx-version` set by the `lldpadm` command conforms the standards of FCoE CEE switches. You can use the following command to change the mode of operation:

```
# lldpadm set-agentprop -p dcbx-version=DCBX-mode net0
```

DCBX-mode can be one of the following values:

- `auto` – Transfers IEEE packets by default. However, when the host receives CEE packets from the peer, the mode switches to CEE automatically.
- `ieee` – Uses the IEEE protocol to exchange information.
- `cee` – Uses the CEE protocol to exchange information.

For more information about CEE-DCBX, see Chapter 7, “Managing Converged Networks by Using Data Center Bridging” in *Managing Network Datalinks in Oracle Solaris 11.3*.

In Oracle Solaris, you can use any of the following configuration options to work with FCoE switches:

- Using Priority-Based Flow Control Capable Ethernet Adapter
- Using Converged Network Adapters
- Using Non CEE-DCBX FCoE Switch

Using Priority-Based Flow Control Capable Ethernet Adapter

Traditional Ethernet adapters support only the IEEE 802.3x `PAUSE` function. The priority-based flow control (PFC) capable Ethernet adapters require a card which supports the IEEE 802.1 Qbb/802.3bd standard. For example, if you are using a 10G Ethernet, you must use the Intel 10G Ethernet adapter.

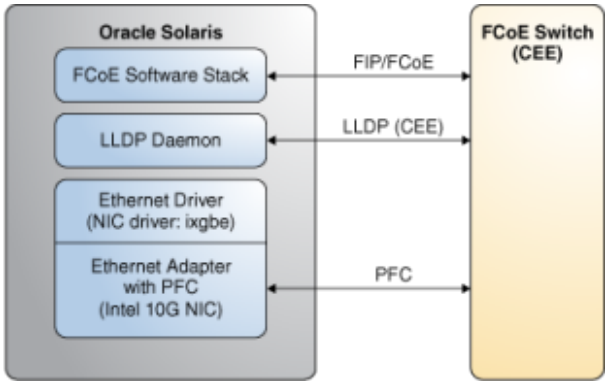
With the configuration of a PFC capable Ethernet adapter connected to the CEE enabled DCBX FCoE switch, you can transfer iSCSI signals between the peers.

You can use the following hardware components to deploy FCoE on Oracle Solaris:

- CEE-DCBX FCoE switch
- PFC capable Ethernet adapter

The following figure shows the connection with CEE-DCBX FCoE switch by using a PFC capable Ethernet adapter.

FIGURE 1 Connection With CEE-DCBX FCoE Switch by Using a PFC Capable Ethernet Adapter



▼ How to Connect to CEE-DCBX FCoE Switch by Using Priority-Based Flow Control

1. **Enable PFC on the NIC and set the MTU as required.**

   In the following example, the NIC MTU is set to 2500 to accommodate the FCoE payload.

   ```
   # dladm set-linkprop -p mtu=2500 net10
   ```

2. **Set priority-based Flow Control (PFC).**

   a. **Set the `dcb_mode` to `1` in the NIC driver file `/etc/driver/drv/ixgbe.conf`.**

      ```
      dcb_mode=1;
      ```

   b. **Run the `update_drv ixgbe` or `reboot` command.**

      ```
      # update_drv ixgbe
      ```

Or

```
# reboot
```

Tip - If the `update_drv` command fails to unload the `ixgbe` module, reboot the Oracle Solaris system.

3. **Set up the NIC flow control to `auto` or `pfc`.**

`auto` – Transfers IEEE packets by default. However, when the host receives CEE packets from the peer, the mode switches to CEE automatically.

- **To set the flow control to `auto`:**

```
# dladm set-linkprop -p flowctrl=auto net10
```

- **To set the flow control to `pfc`:**

```
# dladm set-linkprop -p flowctrl=pfc net10
```

Note - The Oracle Solaris FCoE software framework supports only the `ixgbe` driver. The LLDP and FCoE software stack does not work without enabling PFC.

4. **Check the effective value of the NIC settings.**

```
# dladm show-linkprop -p mtu,flowctrl,ntcs net10
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net10    mtu             rw   2500         2500         1500      576-15500
net10    flowctrl        rw   auto         pfc          no        no,tx,rx,bi,
                                                                  pfc,auto
net10    ntcs            r-   8            8            0         --
```

Using Converged Network Adapters

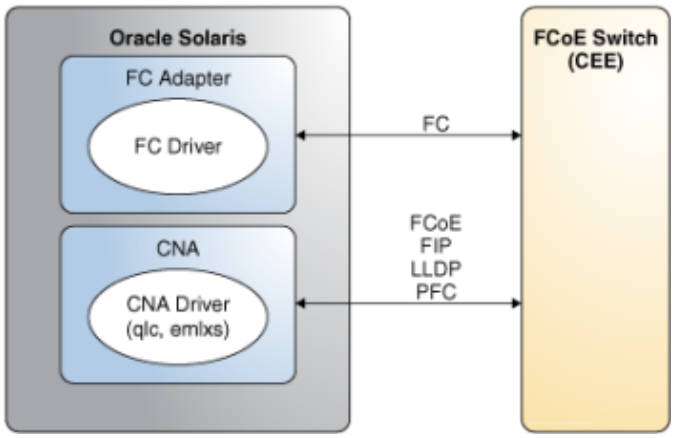
The FC kernel driver enables the Oracle Solaris OS to treat Converged Network Adapter (CNA) cards as FC cards.

You can use the following hardware components to work with CEE-DCBX FCoE switches by using CNA cards on Oracle Solaris:

- CEE-DCBX FCoE switch
- Converged Network Adapters

The following figure shows the connection with CEE-DCBX FCoE switch by using CNAs.

FIGURE 2 Connection With CEE-DCBX FCoE Switch by Using CNAs



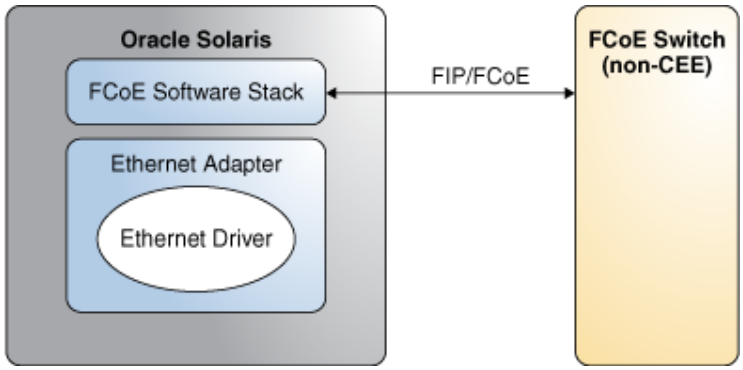
Using Non CEE-DCBX FCoE Switch

You can connect Oracle Solaris with a non-CEE FCoE switch by using the following hardware components:

- Non CEE-DCBX FCoE switch
- Any Ethernet adapter

The following figure shows the connection with a non CEE-DCBX FCoE switch.

FIGURE 3 Connection With Non CEE-DCBX FCoE Switch



How to Configure LLDP Daemon

1. **Install the Link Layer Discovery Protocol (LLDP) daemon.**

```
# pkg install lldp
```

2. **Configure the LLDP agent.**

```
# lldpadm set-ap -p mode=both net10
# lldpadm set-ap -p dcbx-version=cee net10
```

Note - For some switches, you must explicitly set the value to `cee` on Oracle Solaris to enable CEE.

3. **Check whether the value of the `EFFECTIVE` field is `cee`.**

```
# lldpadm show-ap -p mode,dcbx-version net10
AGENT      PROPERTY     PERM VALUE       EFFECTIVE   DEFAULT     POSSIBLE
net10      mode         rw   both        both        disable     txonly,rxonly,
                                                                 both,disable
net10      dcbx-version rw   cee         cee         auto        auto,ieee,cee
```

Once the FCoE configuration is complete, you can create the FCoE port.

If the FCoE port is not online, you can troubleshoot by verifying and configuring switch information. For more information, see "Verifying and Configuring Switch Information" on page 100.

For more information about LLDP, see Chapter 6, "Exchanging Network Connectivity Information With Link Layer Discovery Protocol" in *Managing Network Datalinks in Oracle Solaris 11.3*.

Verifying and Configuring Switch Information

You can verify the configuration on a switch. For example, on a Brocade switch, you can use the following commands to verify the configuration:

- Use the `show cee maps` command to check if DCBX is enabled.

```
switch1:root>cmsh
switch1# show cee maps
```

- Use the `show vlan brief` command to check if the `VLAN` is enabled on the port.

```
switch1# show vlan brief
```

- Use the `show lldp neighbors interface` command to check the DCBX information on the switch.

```
switch1# show lldp neighbors interface
```

▼ How to Change the Configured Port

This procedure provides an example of changing the configured port when the current port is not online:

1. Check for the configured port.

```
switch1# show running-config
...
!
interface TenGigabitEthernet 0/11
 mtu 9208
 switchport
 switchport mode converged
 switchport converged allowed vlan add 1002
 no shutdown
 cee default
!
```

2. Change the port.

```
switch1# configure terminal
switch1(config)# interface tengigabitethernet 0/11
switch1(conf-if-te-0/11)# switchport
switch1(conf-if-te-0/11)# vlan classifier activate group 1 vlan 10002
```

3. **Ensure the port is updated.**

```
switch1# show running-config
...
!
interface TenGigabitEthernet 0/11
 mtu 9208
 switchport
 switchport mode converged
 switchport converged allowed vlan add 1002
 vlan classifier activate group 1 vlan 1002
 no shutdown
 cee default
...
```

Note - The switch must have the line `vlan classifier activate group 1 vlan 1002`. This enables the VLAN ID `1002` in the VLAN group.

4. **Display the configuration information.**

```
switch1:root> fcoe --cfgshow
User Port Status     Port WWN               DeviceCount  Port  Type   MAC
 VF_ID
=============================================================================================================
...
19        ENABLED   20:13:00:05:1e:b0:1c:80     1        FCoE  VF-Port  00:05:1e:b0:1c:
8b  128
...
```

In this example, 19 corresponds to the 0/11 port.

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` file and modify the tunable values for your FCoE environment.

TABLE 2 Hardware Offload Tunable Parameters

Tunable Parameter	Description	Default Value
`fcoe_txcrc_enable`	Controls whether the `ixgbe` driver offloads FC CRC transactions for transmitted FCoE packets.	1

Tunable Parameter	Description	Default Value
`fcoe_lso_enable`	Controls whether the `ixgbe` driver offloads FC large send transactions for transmitted FCoE packets.	1
`fcoe_rxcrc_enable`	Controls whether the `ixgbe` driver offloads RC CRC transactions for received FCoE packets.	1
`fcoe_lro_enable`	Controls whether the `ixgbe` driver offloads FC large receive transactions for received FCoE packets.	0

For each parameter, a value of 0 disables the parameter and a value of 1 enables it.

Configuring Converged Network Adapters Based FCoE Ports

Converged Network Adapters (CNAs) combine the functionality of FC HBAs (Fibre Channel Host Bus Adapter) and Ethernet NICs (Ethernet Network Interface Card) to transfer Ethernet and FCoE traffic. CNAs convert the FCoE traffic into FC traffic, which is then sent to the connected SAN over the FC network.

Advantages of CNAs

Second-generation CNAs are generally used to set up the SAN. The advantages of using CNAs are:

- Using CNAs in storage networks reduces the number of adapters used in a network, which helps to reduce the number of switch ports, cables, and PCI Express slots.
- CNAs can offload FCoE protocol processing tasks, which helps to reduce the consumption of server CPU resources.
- The CNA connects to the server by using PCI Express expansion interface.
- CNAs can be used over FC networks, with FC switches and Fibre Channel Management utilities.
- CNAs can carry or transfer Ethernet traffic and FCoE traffic.
- CNAs can be used as a stand-alone 10 GE NIC if FCoE and FC SAN are not immediately available.

Installing and Configuring CNA-Based FCoE Ports

For more information about how to perform hardware installation and configuration of CNA based FCoE ports, refer to the following guides:

- For QLogic, see Sun Storage 16 Gb Fibre Channel PCIe Universal Host Bus Adapter, Qlogic
- For Emulex, see Sun Storage 16 Gb Fibre Channel PCIe Universal Host Bus Adapter, Emulex

For more information about the different types of cards supported by Oracle Solaris, see Oracle Storage Networking: Host Bus Adapters and Converged Network Adapters.

♦ ♦ ♦ CHAPTER 7

Configuring SAS Devices

Oracle Solaris contains several self-identifying drivers for SAS-1, SAS-2, and SAS-3 controllers. These SAS drivers enumerate all the supported targets automatically and do not require manual configuration.

Systems running the Oracle Solaris OS can boot from a SAS device or from a SATA device connected to a SAS controller.

Oracle Solaris also supports SAS expanders and SMP protocols. For more information about the hardware compatibility for SPARC and x86 platforms, see Oracle Hardware Compatibility List. For more information about supported disk controllers, see http://www.oracle.com/webfolder/technetwork/hcl/data/s11ga/components/views/disk_controller_all_results.techtype.page1.html.

This chapter provides information about the SAS devices discovery and displaying their configuration information.

It covers the following topics:

Dynamic Discovery of SAS Devices

Adding and removing SAS devices is performed dynamically in a SAN network. 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. You can verify the addition and removal of SAS device by using the `sasinfo` command. You can also use the `format` command to verify the addition and removal of disk devices. The `sasinfo` utility reports attributes of HBA ports and expander devices that might be connected to the HBA ports. For more information, see the `sasinfo`(1M) and `format`(1M) man pages.

Displaying SAS Configuration

This section provides information about a few subcommands of the `sasinfo` command. For more information about the other subcommands available for the `sasinfo` command, see the `sasinfo(1M)` man page.

You can display the list of the SAS HBAs on the host by using the `sasinfo hba -v` command.

For example:

```
# sasinfo hba -v
HBA Name: SUNW-mpt_sas-0
    Manufacturer: ABCETAS2304
    Model:  T5-2_D1
    Firmware Version: 14.0.0.0
    FCode/BIOS Version: not available
    Serial Number: ABCETAS2304ALLT5-2_D1
    Driver Name: mpt_sas
    Driver Version: MPTSAS HBA Driver 00.00.00.29
    Number of HBA Ports: 2
HBA Name: SUNW-mpt_sas-1
    Manufacturer: ABCETAS2304
    Model:  T5-2_D1
    Firmware Version: 14.0.0.0
    FCode/BIOS Version: not available
    Serial Number: ABCETAS2304LLT5-2_D1
    Driver Name: mpt_sas
    Driver Version: MPTSAS HBA Driver 00.00.00.29
    Number of HBA Ports: 3
```

You can display all the port information of the SAS HBAs connected to the host by using the `sasinfo hba-port -v` command.

For example:

```
# sasinfo hba-port -v
HBA Name: SUNW-mpt_sas-0
  HBA Port Name: /dev/cfg/c2
    Type: SAS Device
    State: online
    Local SAS Address: 508002000168bd40
    Attached SAS Address: 5000cca01615535d
    Number of Phys: 1
  HBA Port Name: /dev/cfg/c3
    Type: SAS Device
    State: online
    Local SAS Address: 508002000168bd40
```

```
    Attached SAS Address: 5000cca01628a9fd
    Number of Phys: 1
HBA Name: SUNW-mpt_sas-1
  HBA Port Name: /dev/cfg/c1
    Type: SAS Device
    State: online
    Local SAS Address: 508002000168bd41
    Attached SAS Address: 0000000000000003
    Number of Phys: 1
  HBA Port Name: /dev/cfg/c5
    Type: SAS Device
    State: online
    Local SAS Address: 508002000168bd41
    Attached SAS Address: 5000cca0561e9371
    Number of Phys: 1
  HBA Port Name: /dev/cfg/c6
    Type: SAS Device
    State: online
    Local SAS Address: 508002000168bd41
    Attached SAS Address: 5000cca056188879
    Number of Phys: 1
```

You can display all the SAS logical units by using the `sasinfo lu -v` command.

For example:

```
# sasinfo lu -v
OS Device Name: /dev/rdsk/c0t5000CCA056188878d0s2
    HBA Port Name: /dev/cfg/c6
        Target Port SAS Address: 5000cca056188879
            LUN: 0
    Vendor: EXAMPLE
    Product: A109060SAMPLE600G
    Device Type: Disk Device
OS Device Name: /dev/rdsk/c0t5000CCA0561E9370d0s2
    HBA Port Name: /dev/cfg/c5
        Target Port SAS Address: 5000cca0561e9371
            LUN: 0
    Vendor: EXAMPLE
    Product: A109060SAMPLE600G
    Device Type: Disk Device
```

♦ ♦ ♦ CHAPTER 8

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.

This chapter covers the following topics:

IPFC Considerations

Note the following considerations for IPFC (NFS/NAS and SNDR):

- 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.
- The only supported zone type is a Fabric zone with the HBA configured as an F-port point-to-point connection.
- Cascading is supported for fabric zones only.
- The maximum number of IPFC device ports per zone is 253.

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

The examples in this procedure assume that you have an array with an HBA card that is located in PCI adapter slot 5, and that the PCI adapter is in slot 1 of the I/O board.

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 following command:

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

2. **Determine the instance number.**

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

      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.**

   Note - The following method of deriving the device path of an HBA from its physical location in a server might not work for all of 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, you would multiply 5 by 2 to get 10.

      ii. **Add the PCI adapter I/O board slot number.**

         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 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,qlc@2/fp@0,0" 7 "fp"
         ```

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

c. Manually create each FP instance.

```
# ipadm create-ip interface-number
```

In this example, *interface-number* is `fcip7`.

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

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

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

▼ How to Create an IPFC Instance

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

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

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

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

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

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

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

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

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

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Configure the appropriate network interface.**

   Ask your network administrator for an appropriate IP address and netmask information. The following example enables an IPFC interface associated with `fp` instance `0` and an IP address of `192.168.201.10`.

   ```
   # ipadm create-ip fcip0
   # ipadm create-addr -T static -a 192.168.201.10 fcip0/ipv4
   ```

   For more information, see the `ipadm`(1M) man page.

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 in the examples, in this procedure, the hostname is `sys-A` and the IPFC network interface is `fcip0`.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 the `hosts`(4) man page.

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

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

```
127.0.0.1      localhost     loghost
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
```

♦ ♦ ♦ CHAPTER 9

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 Oracle 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 Oracle Solaris installation program.

This chapter covers the following topics:

Oracle Solaris OS Setup Requirements for Fibre Channel Devices

You must have the following items for installation:

- Installation DVD with Oracle Solaris 11 or later version. The following installation methods are available for configuring FC devices on an x86 based system:
  - Automatic installation – Install multiple clients on the network. You can boot from media for installing a single system but if you want to customize the installation for multiple clients, 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 systems:

- 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 for FC Devices 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 World Wide Name (WWN) and LUN. Record the HBA 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 a 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 you are using a custom disk layout, do not remove the overlap (`s2`) partition. The x86 boot installer has a dependency on this partition.
- By default, Oracle Solaris I/O multipathing features manage the FC boot devices with multiple paths to a single boot device.

Oracle Solaris OS Installation Procedure for FC Devices

The following procedures describe how to install the Oracle Solaris OS for FC devices.

How to Perform a DVD or Network-Based OS Installation for FC Devices

This procedure describes how to perform a DVD installation or a network-based installation of the Oracle Solaris OS on an x86 based system for FC devices.

For more information, refer to *Installing Oracle Solaris 11.3 Systems*.

Before You Begin Install the HBA hardware by following the instructions in the appropriate Oracle HBA installation guide found at `http://www.oracle.com/technetwork/documentation/oracle-storage-networking-190061.html`.

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.
   - 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 11.3 Text Installer and command line
     ```

   a. **Select a desired array and its associated LUN.**

   b. **Continue the installation by selecting the desired installation options from each installation screen.**

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

4. **After the installation is complete, prepare to configure your devices before the system reboots.**

   - 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 .

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

```
.
.
```

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

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

   For example:

```
# luxadm display /dev/rdsk/c0t600015D000202800000000000000001142d0s2
DEVICE PROPERTIES for disk: /dev/rdsk/c0t600015D000202800000000000000001142d0s2
  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/rdsk/c0t600015D000202800000000000000001142d0s2
  /devices/scsi_vhci/disk@g600015d000202800000000000000001142: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
```

You would use the sample output in the figure can be used to map the MPxIO based `c#t#d#` to the HBA WWN and array WWN as follows:

- MPxIO c#t#d# = `c0t600015d000202800000000000000001142d0`
- Array WWN = `213600015d207200, LUN 0`
- HBA WWNs = `210100e08b206812 and 210100e08b30a2f2`

6. During the reboot process, 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.

- **For the QLogic HBA BIOS:**

  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 selecting the array WWPN.

iv. **Save and then exit the HBA BIOS screen.**

- **For the Emulex HBA BIOS:**

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.**

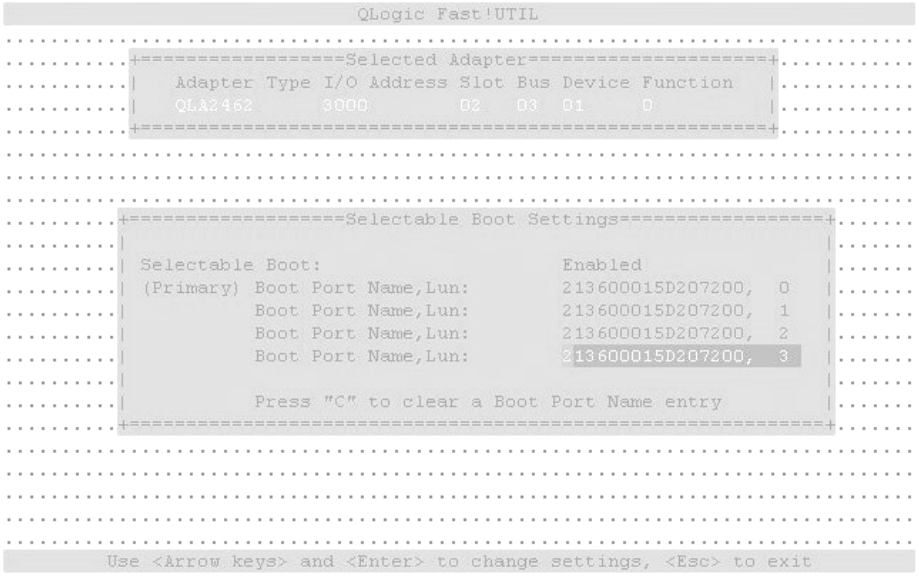
e. **Select the WWPN of the desired boot device.**

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.



The figure shows the following modifications:

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

7. **Repeat the appropriate modifications for all HBAs and all LUNs.**

8. **Provide 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.**

9. **Reboot to the newly installed Oracle Solaris OS by using the FC LUN specified in the system BIOS.**

♦ ♦ ♦ CHAPTER 10

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. It covers the following topics:

Persistent Binding Overview

To simplify management of servers in SAN-based data-centers, Oracle's StorageTek SAN Foundation software stack in the Oracle 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 prefer to be able 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 Oracle 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/`*N*`b`, and `/dev/rmt/`*N*`bn`, 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 the `devlinks(1M)` man page.

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 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` command.

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 in the following example.

```
# 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:
```

For example, if 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 would then add this line to the `devlink` file on every Oracle 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Create the entries in the `/etc/devlink.tab` file.**

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

   Note - Ensure to assign `/dev/rmt/N` numbers to avoid conflicts with any automatically configured devices.

   For more information about creating entreis in the `devlink.tab` file, see "Creating Tape Links" on page 124.

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

4. **Run the `devfsadm` command.**

   This command creates new links as per the entries in the `/etc/devlink.tab` file in addition to automatically creating links for any unspecified devices. For more information about the `devfsadm` command, see the `devfsadm`(1M) man page.

♦ ♦ ♦ APPENDIX A

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

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.**

For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 Oracle Solaris I/O multipathing features.**

   - "Configuring Fabric Device Nodes Without Multipathing Enabled" on page 130
   - "Configuring Fabric Device Nodes With Oracle Solaris Multipathing Enabled" on page 134

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 fabric devices, ports, and zones in your SAN, make sure that the system is aware of the connected fabric 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 fabric device availability and device-specific management. If fabric 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 fabric 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 fabric device configuration on fabric-connected host ports.

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

How to Manually Configure a Fabric 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.**
   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Identify the fabric 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 Fabric Devices Without Multipathing

Make sure you first identify the fabric devices visible to the system with the procedure "Ensuring That LUN Level Information Is Visible" on page 129. 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Identify the fabric 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 Fabric Device Nodes With Oracle 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 fabric 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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.

```
# cfgadm -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 two multipathed devices are created for the device represented by `c0::50020f2300006077` and `c2::50020f2300006107:`.

How to Configure Multiple Multipathed Fabric 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 129.

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

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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
```

   Fabric 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 fabric 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 fabric evice and unmount any file systems on the fabric device. See the Oracle Solaris administration documentation for unmounting instructions. If the fabric device is under any volume manager' control, see your volume manager documentation before unconfiguring the fabric 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.**
   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

2. **Identify the fabric 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.**
   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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

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

In this example, the `c0::50020f2300006077` and `c2::50020f2300006107` `Ap_Id`s 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 `cfgadm -c unconfigure` command.

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

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

---

5. Verify that the selected devices are unconfigured.

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

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

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

```
/dev/rdsk/c6t60020F20000061073AC8B52D000B74A3d0s2

/dev/rdsk/c6t60020F20000061073AC8B4C50004ED3Ad0s2
```

▼ How to Unconfigure One Path to a Multipathed FC Device

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

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 fabric device can be connected to multiple `Ap_Id`s and an `Ap_Id` can be connected to multiple devices.

```
/dev/rdsk/c6t60020F20000061073AC8B52D000B74A3d0s2
and
/dev/rdsk/c6t60020F20000061073AC8B4C50004ED3Ad0s2
```

▼ How to Unconfigure All Multipathed Fabric Devices

An `Ap_Id` on a fabric-connected host port is a path to a multipathed device.

When a multipathed device has multiple `Ap_Id`s 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.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.

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 “Oracle Solaris I/O Multipathing Overview” on page 13.

Supported Fibre Channel HBA API

Oracle 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

♦ ♦ ♦ APPENDIX C

Troubleshooting Multipathing-Related Problems

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

The following topics are covered:

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

     ```
     Oracle Solaris 11.3 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 have 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 have 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 have modified the device settings of your storage arrays, restore their original settings.

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

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

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

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

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

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

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

   b. **Update the boot archive.**

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

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

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

9. **Unmount the BE.**

   ```
   # beadm umount solaris
   ```

10. **Unmount the UFS root file system.**

    ```
    # umount /a
    ```

11. **Reboot the system.**

♦ ♦ ♦ APPENDIX D

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 152
- "Bit-Masking Format to Tune Disk Drivers" on page 154

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 3, "Supported Tunables and Their Flag Values on Different Platforms," on page 155
- `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:

`cache-nonvolatile`	`BOOLEAN`
`controller-type`	`UINT32`
`delay-busy`	`UINT32`
`disksort`	`BOOLEAN`
`emulation-rmw`	`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 "*VIDPID*", "*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 4, "Configuring Two Target Devices in an `sd.conf` File," on page 153.

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 4 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 5 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 "*VIDPID*", "*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.

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.

sd-ver1-conf-data Property array consisting of a version number which is `1`, a mask number, and tunable values to be set.

- *mask* — Value between `0x01` to `0x7FFFF`. The value of *mask* can sometimes be the OR result of multiple desired flags. Table 3, "Supported Tunables and Their Flag Values on Different Platforms," on page 155 shows the bit value of each tunable parameter.
- *value-sequence* — Sequence of valid property values and number of 0's corresponding to the value of *mask*. The length limit for *value-sequence* is 19 characters. See "Tunable Parameters for Disk Drivers" on page 151 for the data type of tunable parameters.

Note - The duplets with different `VID`s and `PID`s 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 3 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`
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`

Bit	Flag Value	sd Driver on SPARC	ssd Driver on SPARC	sd Driver on x86 or x64
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 6 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 = 1,0x801,300,0,0,0,0,0,0,0,0,0,5,0,0,0,0,0,0,0;
```

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 7 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 0x00004 and 0x00001 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*.

♦ ♦ ♦ APPENDIX E

Supported HBAs in FC or FCoE Mode

This appendix contains a list of a new class of supported HBAs that provides PCI selectable 16 GB FC or 10 GB Fibre Channel over Ethernet (FCoE) connectivity. These multifunctional cards can be used in FC or FCoE mode, depending on the optics inserted.

Supported HBAs in FC or FCoE Mode

Oracle Solaris supports the following 4 GB HBAs, which are listed along with their part numbers:

- Oracle's Sun StorageTek 4 GB Enterprise FC PCI-X HBA, SG-XPCI1FC-QF4 and SG-XPCI2FC-QF4
- Oracle's Sun StorageTek 4 GB Enterprise FC PCI-X 2.0 Single-Channel Network Adapter, SG-XPCI1FC-EM4-Z
- Oracle's Sun StorageTek 4 GB Enterprise FC PCI-X 2.0 Dual-Channel Network Adapter, SG-XPCI2FC-EM4-Z
- Oracle's Sun StorageTek Dual 8 GB FC DualGigabit Ethernet (GBE) ExpressModule HBA that includes:
  - SG-XPCIEFCGBE-E8-Z Emulex, Xoption
  - SG-PCIEFCGBE-E8-Z Emulex, Factory configured
  - SG-XPCIEFCGBE-Q8-Z Qlogic, Xoption
  - SG-PCIEFCGBE-Q8-Z Qlogic, Factory configured

  For more information about Oracle's Sun StorageTek Dual 8 GB FC DualGigabit Ethernet (GBE)ExpressModule HBA, see SUN STORAGETEK DUAL 8 GB FIBRE CHANNEL DUAL GBE EXPRESSMODULE HOST BUS ADAPTER.
- For Emulex, Oracle's StorageTek 8 GB FC PCI Express HBA includes:
  - SG-XPCIE1FC-EM8-N (Single Channel, only for selected servers)
  - SG-XPCIE2FC-EM8-N (Dual Channel)
- For Qlogic, Oracle's StorageTek 8 GB FC PCI Express HBA includes:
  - SG-XPCIE1FC-QF8-N (Single Channel, only for selected servers)

    - SG-XPCIE2FC-QF8-N (Dual Channel)
    - 7106958 (Dual channel, only for T5-4, T5-8, M5-32, and M6-32 servers)
- Oracle's Sun Storage 10 GbE FCoE ExpressModule Converged Network Adapter includes:
    - SG-XEMFCOE2-Q-SR FCoE ExpressModule (SR Optical, Xoption)
    - SG-XEMFCOE2-Q-TA FCoE ExpressModule (Copper, Xoption)
    - SG-EMFCOE2-Q-SR FCoE ExpressModule (SR Optical, for factory installation)
    - SG-EMFCOE2-Q-TA FCoE ExpressModule (Copper, for factory installation)
- Oracle's Sun Storage 10GbE FCoE PCIe Converged Network Adapter includes
    - SG-XPCIEFCOE2-Q-SR (Short Range)
    - SG-XPCIEFCOE2-Q-TA (Copper)
    - SG-PCIEFCOE2-Q-SR (Short Range, factory configured)
    - SG-PCIEFCOE2-Q-TA (Copper, factory configured)

    For more information about Oracle's Sun Storage 10GBE FCoE PCIe Converged Network Adapter see, Sun Storage 10 GbE FCoE PCIe Converged Network Adapter From QLogic.
- For Emulex, Oracle's StorageTek 16 GB FC PCIe Universal HBA includes:
    - 7101687 2 Sun Storage 10 GB FCoE SR optics (For factory installation)
    - 7101688 2 Sun Storage 10 GB FCoE SR optics
    - 7101683 Sun Storage Dual 16 GB FC PCIe Universal HBA (For factory installation)
    - 7101684 Sun Storage Dual 16 GB FC PCIe Universal HBA
    - 7101685 2 Sun Storage 16 GB FC SW optics (For factory installation)
    - 7101686 2 Sun Storage 16 GB FC SW optics
- For Qlogic, Oracle's Storage 16 GB FC PCIe Universal HBA includes:
    - 7101677 2 Sun Storage 10 GB FCoE SR optics (For factory installation)
    - 7101678 2 Sun Storage 10 GB FCoE SR optics
    - 7101673 Sun Storage Dual 16 GB FC PCIe Universal HBA (For factory installation)
    - 7101674 Sun Storage Dual 16 GB FC PCIe Universal HBA
    - 7101675 2 Sun Storage 16 GB FC SW optics (For factory installation)
    - 7101676 2 Sun Storage 16 GB FC SW optics
    - 7101680 2 Sun Storage 16 GB FC LW optics
- For Emulex, Oracle's Sun Storage 16 GB FC ExpressModule Universal HBA includes:
    - 7101689 Sun Storage Dual 16 GB FC ExpressModule Universal HBA (For factory installation)
    - 7101690 Sun Storage Dual 16 GB FC ExpressModule Universal HBA
    - 7101685 2 Sun Storage 16 GB FC SW optics (For factory installation)
    - 7101686 2 Sun Storage 16 GB FC SW optics
    - 7101687 2 Sun Storage 10 GB FCoE SR optics (For factory installation)
    - 7101688 2 Sun Storage 10 GB FCoE SR optics
- For Qlogic, Oracle's Storage 16 GB FC ExpressModule Universal HBA includes:

- 7101677 2 Sun Storage 10 GB FCoE SR optics (For factory installation)
- 7101678 2 Sun Storage 10 GB FCoE SR optics
- 7101681 Sun Storage Dual 16 GB FC ExpressModule Universal HBA (For factory installation)
- 7101682 Sun Storage Dual 16 GB FC ExpressModule Universal HBA
- 7101675 2 Sun Storage 16 GB FC SW optics (For factory installation)
- 7101676 2 Sun Storage 16 GB FC SW optics
- 7101680 2 Sun Storage 16 GB FC LW optics

Note - When you put the FC optics in and power on, the card changes its identity to FC. When you put in the FCoE optics and power on, the card naturally changes its identity to FCoE network. Depending on the optics inserted, the transceivers installed allows the card to multifunction in FC or FCoE mode. This feature is available only on 16GB and 10GB FC PCIe Universal and ExpressModule Universal HBAs.

For more information about FC and FCoE HBAs, see http://www.oracle.com/us/products/servers-storage/storage/storage-networking/hba-comparisons/index.html

Index

N

P

Q

R

S

T

W

X