Managing SAN Devices and I/O Multipathing in Oracle® Solaris 11.4

ORACLE®

Managing SAN Devices and I/O Multipathing in Oracle Solaris 11.4

Part No: E61018

Copyright © 2009, 2020, Oracle and/or its affiliates.

License Restrictions Warranty/Consequential Damages Disclaimer

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.

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité de 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 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 `https://docs.oracle.com/cd/E37838_01/`.

Feedback

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

1

About Oracle Solaris Storage Stack

Oracle Solaris storage stack is a multilayered modular stack, which consists of multiple layers with well-defined interfaces between them. The storage stack works together with other storage-related software to access or store data by using different transport protocols, which helps provide better performance, availability, and security. Oracle Solaris supports Fibre Channel (FC), iSCSI, and Serial Attached SCSI (SAS) transport protocols. These transport protocols enable you to connect to different types of storage devices such as a single disk, a tape device, or a large enterprise system. These devices have management command line options, which manage input and output (I/O) communication. The Oracle Solaris I/O multipathing feature enables these storage devices to have multiple paths to a host.

This chapter provides information about the I/O communication between the different components in a storage stack.

I/O Communication Between Oracle Solaris Storage Stack Components

Oracle Solaris I/O multipathing is an integral part of the Oracle Solaris storage stack, which supports most of the storage devices. Oracle Solaris I/O multipathing feature provides high availability, reliability, and persistent names for the Oracle Solaris operating system (OS). For high availability, the storage stack devices use the I/O multipathing feature to ensure that the secondary path is online when the primary path of the device goes offline.

FIGURE 1 I/O Communication Between Oracle Solaris Storage Stack Components

The storage I/O communicates with processes, applications, and other components in the storage stack such as disks, tape devices, and target device drivers, by using a multiplexer

through the Signal Computing System Architecture (SCSA) layer. Each device in the storage stack has its driver, which helps in I/O communication. The SCSI target driver includes drivers such as `sd`, `ssd`, `st`, `ses`, and `sgen`. These device drivers include specific device codes to communicate with the specific storage device available in the network. Target drivers can either be character or block device drivers, depending on the device. For example, tape devices have character device drivers, and disks have block device drivers.

The multiplexer manages multiple I/O operations between the software and the hardware interfaces in a storage setup. In Oracle Solaris, the I/O multipathing feature is implemented by the `scsi_vhci` driver. This driver is SCSA compliant and supports FC, iSCSI, and SAS transport protocols in Oracle Solaris. These transport protocols enable you to access data from the storage devices for read or write operations. The Oracle Solaris storage stack uses file systems or applications, which can directly access block or collector devices. Oracle Solaris also supports FC switched, point-to-point, and FC-AL topologies.

In Oracle Solaris, the FC transport protocol supports different types of Fibre Channel Adapters (FCAs) and Fibre Channel over Ethernet (FCoE) cards. The FCA cards have `qlc` or `emulex` drivers are integrated on the adapter, which enables automatic configuration. You can use an FCoE card either as an Ethernet NIC, FCoE card, or FC card. Oracle Solaris also provides software implementation for these FCoE cards. For more information about Fibre Channel devices in Oracle Solaris, see Chapter 4, "Configuring Fibre Channel Devices".

The iSCSI transport protocol supports the regular TCP/IP protocol and the iSCSI Extension for RDMA (iSER). iSER also supports InfiniBand remote direct memory access (RDMA). You can map multiple targets to a single initiator and access targets by using Oracle Solaris I/O multipathing. For more information about administering iSCSI devices, see Chapter 5, "Configuring an Oracle Solaris iSCSI Initiator".

Oracle Solaris supports SAS devices, expanders, and controllers. These SAS devices have drivers that enable you to access data from storage devices that are connected to an Oracle Solaris host that supports automatic configuration. For more information, see Chapter 6, "Configuring SAS Devices".

2 ♦♦♦ C H A P T E R   2

About Oracle Solaris I/O Multipathing

The Oracle Solaris I/O multipathing feature, previously known as MPxIO, enables an Oracle Solaris system to access storage systems in multiple paths. Oracle Solaris I/O multipathing is based on the open standards for communicating with devices and device management interfaces, ensuring interoperability with other standard-based devices and software. I/O multipathing provides high availability for storage devices by using multiple connections. The Oracle Solaris I/O multipathing feature enumerates the storage devices discovered by the transport layer on your storage area network (SAN) or SAS domain.

Oracle Solaris I/O multipathing is enabled by default for SPARC based and x86 based systems. Because Oracle Solaris I/O multipathing is enabled, the host probes all the connected devices automatically in a storage setup. For third-party devices, which do not configure automatically, you must configure the device to work with the Oracle Solaris I/O multipathing feature. For more information about device-specific configuration, see your storage array documentation. For more information about how to configure symmetric third-party devices that do not configure automatically, see "Configuring Third-Party Storage Devices" on page 33.

> ⚠️ **Caution -** Do not disable I/O multipathing unless you are using a third-party multipathing solution or your third-party device documentation specifies to disable I/O multipathing.

Oracle Solaris I/O multipathing is disabled for tape devices only. For more information, see "Configuring Multipathing on Tape Drives" on page 35.

This chapter provides an overview of the Oracle Solaris I/O multipathing feature and covers the following topics:

- "Oracle Solaris I/O Multipathing Features and Benefits" on page 17
- "Disk Storage Devices and Device Standards" on page 19

Oracle Solaris I/O Multipathing Features and Benefits

The following sections describe the features and benefits of Oracle Solaris I/O multipathing.

Oracle Solaris I/O Multipathing Features

The Oracle Solaris I/O multipathing is enabled by default on both SPARC based and x86 based systems and has the following features:

- Supports dynamic reconfiguration (DR) operations.
- Path management – Manages the path to any storage device dynamically that Oracle Solaris supports. The addition and removal of paths to a device is done automatically when there is a change in path configuration.
- Failover support – Manages the failure of storage paths while maintaining host I/O connectivity through the available secondary path. This feature implements higher levels of reliability, availability, and serviceability (RAS), which requires redundant host connectivity to storage devices.
- Symmetric and asymmetric device support – Supports symmetric and asymmetric disk storage devices.
- I/O load balancing – Provides I/O load balancing by routing I/O through multiple host connections along with simple failover support.
- Queue depth – Used for flow control. SCSI storage arrays present storage to a system in the form of a logical unit. A logical unit has a finite set of available resources, such as the amount of data that can be stored and the number of active commands that a device can process at a time. The number of active commands that can be issued before a device blocks further I/O is known as queue depth. With Oracle Solaris I/O multipathing, a single queue is created for each logical unit regardless of the number of distinct or separate paths it might have to the host.
- Persistent device naming – Maintains device naming of the configured devices even after the reboot or reconfiguration of the system. The only exception to this policy are tape devices found in the `/dev/rmt` file, that do not change unless they are removed and then regenerated.

Benefits of Oracle Solaris I/O Multipathing

- Devices with multiple paths are displayed as a single device instead of one device 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. For example, the `format` command or a volume management product sees a device with multiple paths as one representation of a storage device instead of multiple devices 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 the bandwidth by adding host controllers.
- Oracle Solaris I/O multipathing allows you to combine iSCSI and FC paths to a single target.

Disk Storage Devices and Device Standards

The Oracle Solaris I/O multipathing feature supports both symmetric and asymmetric disk and tape 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, or `active/optimized` and `active/non-optimized` paths.

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

- All T10 Asymmetric Logical Unit Access (ALUA) compliant devices
- All T10 and T11 standards-compliant third-party symmetric devices
- Third-party asymmetric devices

Also, to know if your device is supported, see your storage vendor documentation.

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:

- T10 standards, including SCSI-3, SAM, SPC, and SBC
- T11.3 FC standards, including FCP, 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 (SAS-2)
- iSCSI standards, including RFC 3720 and RFC 7143
- ANSI Standard: Information Technology – SNIA Multipath Management API Specification (ANSI INCITS 412-2006)

♦♦♦ C H A P T E R 3

3

Configuring Oracle Solaris I/O Multipathing Features

The Oracle Solaris I/O multipathing feature is enabled by default and provides automatic configuration. This chapter explains how to configure the Oracle Solaris I/O multipathing feature in the Oracle Solaris OS. It also provides information about the factors to consider while using the multipathing feature on SPARC based systems, x86 based systems, tape drives, third-party storage devices, and on a per-port basis.

In most cases, you do not disable the Oracle Solaris I/O multipathing feature. Therefore, you do not need to perform any configuration steps. If you want to disable the I/O multipathing feature, see your vendor specific documentation.

This chapter provides information about how to display and manage the Oracle Solaris I/O multipathing feature. It contains the following topics:

Displaying Multipathing Information

You can 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 logical unit are consistent with the T10 specification.

Displaying Logical Unit Information

You can display the list of logical units (LUs) by using the `mpathadm list lu` command. The displayed list of LUs contain names that can also be used to display the properties of a particular LU.

Note - If you use the `format` command when the multipathing features are enabled, you can see only one device identifier instance for each LU. When the multipathing feature is not enabled, you can see one identifier for each path.

You can display detailed information of LUs with multiple paths such as path and target port group information.

Note - The name property in the information represents the identifier for this LU, which is derived from the hardware, and used by the 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.

To list all the LUs with multiple paths, use the `mpathadm list lu` command.

For example:

```
# mpathadm list lu
         /dev/rdsk/c0t5000CCA00ABBDDA8d0s2
                Total Path Count: 1
                Operational Path Count: 1
        /dev/rdsk/c0t5000CCA00ABBDC08d0s2
                Total Path Count: 1
                Operational Path Count: 1
        /dev/rdsk/c0t5000CCA00ABC5EDCd0s2
                Total Path Count: 1
                Operational Path Count: 1
        /dev/rdsk/c0t5000CCA00ABCECACd0s2
                Total Path Count: 1
                Operational Path Count: 1
        /dev/rdsk/c0t600144F0DDA8BB1A000057208C880001d0s2
                Total Path Count: 1
                Operational Path Count: 1
        /dev/rdsk/c0t600C0FF0000000000011C51EDAA44600d0s2
                Total Path Count: 3
```

```
                    Operational Path Count: 3
             /dev/rdsk/c0t5000CCA00ABBABDCd0s2
                    Total Path Count: 1
                    Operational Path Count: 1
.
.
.
```

To display detailed information about all LUs, use the `mpathadm show lu` command. To display information about a specific LU, use the `mpathadm show lu` *LU-name* command.

For example:

```
# mpathadm show lu  /dev/rdsk/c0t600144F0DDA8BB1A000057208C880001d0s2
Logical Unit:  /dev/rdsk/c0t600144F0DDA8BB1A000057208C880001d0s2
        mpath-support:  libmpscsi_vhci.so
        Vendor:  SUN
        Product:  ZFS Storage 7420
        Revision:  1.0
        Name Type:  unknown type
        Name:  600144f0dda8bb1a000057208c880001
        Asymmetric:  yes
        Current Load Balance:  shortest-path
        Logical Unit Group ID:  NA
        Auto Failback:  on
        Auto Probing:  NA

        Paths:
                Initiator Port Name:
 iqn.1986-03.com.sun:01:21866c42ab36.5710b544,4000002a00ff
                Target Port Name:  4000002a0000,iqn.1986-03.com.sun:02:56bf24c4-
e3e3-4ff0-e7b0-a6e6880e461d
                Override Path:  NA
                Path State:  OK
                Disabled:  no

        Target Port Groups:
                ID:  0
                Explicit Failover:  no
                Access State:  active optimized
                Target Ports:
                        Name:  4000002a0000,iqn.1986-03.com.sun:02:56bf24c4-e3e3-4ff0-
e7b0-a6e6880e461d
                        Relative ID:  3
```

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

To display information about a specific LU, use the `mpathadm list lu -n` *port-name* command.

For example:

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

To display the detailed configuration information for all LUs, use the `mpathadm show lu` command. For more information, see the mpathadm(8) man page.

Displaying All LUs Associated With a Specific Target Port

To display logical units associated with a specific target, use the `mpathadm list lu -t` *target-port-name* command.

For example:

```
# 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
.
.
.
```

Displaying Properties of a Specific Initiator Port

To list all the initiator ports in the system, use the `mpathadm list initiator-port` command.

For example:

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

To display the properties of a specific initiator port, use the `mpathadm show initiator-port` *initiator-port-name* command.

For example:

```
# 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, all the discovered initiator ports and their properties are displayed.

Displaying the List of Supported Devices

Oracle Solaris supports all T10 ALUA compliant devices by default and do not require any configuration. Also, Oracle Solaris supports a few third-party T10 ALUA noncomplaint devices. You can use the `mpathadm` command to display the list of T10 ALUA noncompliant devices, that are supported by the Oracle Solaris OS.

To identify the multipathing support on your system, use the `mpathadm list mpath-support` command.

For example:

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

To display the multipathing support properties for a specified multipathing support name, use the `mpathadm show mpath-support` *multipathing-support-name* command.

For example:

```
# mpathadm show 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
                shortest-path
        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

                Vendor:  LSI
                Product:  INF-01-00
                Revision:
                Supported Load Balance Types:
                        round-robin

                Vendor:  NETAPP
                Product:  INF-01-00
                Revision:
                Supported Load Balance Types:
                        round-robin
.
.
```

This example does not show the entire output that is displayed when you run the `mpathadm show mpath-support libmpscsi_vhci.so` command. For documentation purposes, it shows

only the initial part of the output that is displayed. For symmetric third-party devices not listed in the library, see "Configuring Third-Party Storage Devices" on page 33.

I/O Latency Distribution

Starting with Oracle Solaris 11.4, you can see the SCSI I/O response time (also known as I/O latency) distribution information for better observability. I/O response time distribution can be used to identify the SCSI response time. The distribution can be displayed by using the `iostat` command. The new -L option is added in conjunction with the -x or -Y options to show the histogram.

EXAMPLE 1   Using `iostat` to Generate Per-LUN Latency Distribution Statistics

The following command generates the per-LUN latency distribution statistics for all devices.

```
% iostat -Lx
                 extended device statistics
device    r/s    w/s   kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b
sd0      0.2    0.0    0.1    0.0  0.0  0.0    0.0    0.0   0   0
latency         range        count     density distribution
              <256us             0       0.00%        0.00%
            256-512us            1       3.85%        3.85%
           512-1024us           24      92.31%       96.15%
              1-2ms              0       0.00%       96.15%
              2-4ms              1       3.85%      100.00%
               >4ms              0       0.00%      100.00%
              total            26
sd1      0.2    0.0    0.1    0.0  0.0  0.0    0.0    0.0   0   0
latency         range        count     density distribution
              <512us             0       0.00%        0.00%
           512-1024us           26      96.30%       96.30%
              1-2ms              0       0.00%       96.30%
              2-4ms              1       3.70%      100.00%
               >4ms              0       0.00%      100.00%
              total            27
sd2      0.1    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0
latency         range        count     density distribution
              <256us             0       0.00%        0.00%
            256-512us            2       9.09%        9.09%
           512-1024us           20      90.91%      100.00%
              >1024us            0       0.00%      100.00%
              total            22
```

The following command generates the per-LUN latency distribution statistics for a specific device.

```
% iostat -Lx c0t5000CCA05635B924d0
                  extended device statistics
device    r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b
sd3       0.1    0.0    0.1     0.0  0.0  0.0    0.0    0.0    0   0
latency         range          count      density distribution
                <64us              0        0.00%        0.00%
              64-128us             1        4.00%        4.00%
             128-256us            22       88.00%       92.00%
             256-512us             2        8.00%      100.00%
               >512us              0        0.00%      100.00%
               total             25
```

You can use the device name to generate the per-LUN latency distribution statistics for a specific device.

```
% iostat -Lx sd3
                  extended device statistics
device    r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b
sd3       0.1    0.0    0.1     0.0  0.0  0.0    0.0    0.0    0   0
latency         range          count      density distribution
                <64us              0        0.00%        0.00%
              64-128us             2        8.00%        8.00%
             128-256us            21       84.00%       92.00%
             256-512us             2        8.00%      100.00%
               >512us              0        0.00%      100.00%
               total             25
```

The following command generates the per-path latency distribution statistics for a specific device.

```
% iostat -LY sd0
                      extended device statistics
device            r/s   w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b
sd0               0.0   0.0    0.0     0.0  0.0  0.0    0.0    0.0    0   0
latency         range          count      density distribution
                <64us              0        0.00%        0.00%
              64-128us             1        4.00%        4.00%
             128-256us            22       88.00%       92.00%
             256-512us             2        8.00%      100.00%
               >512us              0        0.00%      100.00%
               total             25
sd0.t0            0.0   0.0    0.0     0.0  0.0  0.0    0.0    0.2    0   0
sd0.t0.mpt_sas4   0.0   0.0    0.0     0.0  0.0  0.0    0.0    0.2    0   0
latency         range          count      density distribution
                <64us              0        0.00%        0.00%
              64-128us            20       74.07%       74.07%
             128-256us             6       22.22%       96.30%
             256-512us             1        3.70%      100.00%
               >512us              0        0.00%      100.00%
```

```
                      total              27
```

The following command generates the per-path latency distribution statistics for a specific path.

```
% iostat -LY sd0.t0.mpt_sas4
                        extended device statistics
device              r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b
sd0.t0              0.0    0.0    0.0     0.0  0.0  0.0    0.0    0.2    0   0
sd0.t0.mpt_sas4     0.0    0.0    0.0     0.0  0.0  0.0    0.0    0.2    0   0
latency          range           count       density distribution
                 <64us               0         0.00%        0.00%
               64-128us              20        74.07%       74.07%
              128-256us              6         22.22%       96.30%
              256-512us              1          3.70%      100.00%
                 >512us              0          0.00%      100.00%
                 total              27
```

The following command generates path latency for a particular time interval.

```
% iostat -LY sd42.t89.lsc1 5 3
                        extended device statistics
device            r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b
sd42.t89          1.3    0.0    0.8     0.0  0.0  0.0    0.0    0.0    0   0
sd42.t89.lsc1     1.3    0.0    0.8     0.0  0.0  0.0    0.0    0.0    0   0
latency          range           count       density distribution
                 <16us               0         0.00%        0.00%
               16-32us          952535        97.68%       97.68%
               32-64us           19332         1.98%       99.66%
              64-128us             392         0.04%       99.70%
             128-256us            2601         0.27%       99.97%
             256-512us             315         0.03%      100.00%
            512-1024us               3         0.00%      100.00%
                 1-2ms               0         0.00%      100.00%
                 2-4ms               3         0.00%      100.00%
                 4-8ms               9         0.00%      100.00%
                  >8ms               0         0.00%      100.00%
                 total          975190
                        extended device statistics
device            r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b
sd42.t89      15583.6    0.0 7791.8     0.0  0.0  0.4    0.0    0.0    0   0
sd42.t89.lsc1 15583.6    0.0 7791.8     0.0  0.0  0.4    0.0    0.0    0  44
latency          range           count       density distribution
                 <16us               0         0.00%        0.00%
               16-32us           45981        98.49%       98.49%
               32-64us             674         1.44%       99.93%
              64-128us              13         0.03%       99.96%
             128-256us              12         0.03%       99.99%
             256-512us               7         0.01%      100.00%
                 >512us              0          0.00%      100.00%
```

```
                  total         46687
                     extended device statistics
      device          r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b
      sd42.t89      16998.9    0.0 8499.4    0.0  0.0  0.5    0.0    0.0   0   0
      sd42.t89.lsc1 16998.9    0.0 8499.4    0.0  0.0  0.5    0.0    0.0   0  47
      latency         range          count      density distribution
                      <16us              0        0.00%        0.00%
                      16-32us        50705       99.43%       99.43%
                      32-64us          256        0.50%       99.93%
                      64-128us           9        0.02%       99.95%
                      128-256us         18        0.04%       99.99%
                      256-512us          7        0.01%      100.00%
                      >512us             0        0.00%      100.00%
                      total          50995
```

This command provides path latency of `sd42.t89.lsc1` three times every five seconds.

For more information, see the sd(4D) and iostat(8) man pages.

Failover of a Logical Unit

You can failover a logical unit (LU) in a device, which belongs to one of 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 (TPGS) and providing explicit mode ALUA.

▼ How to Failover a Logical Unit

1. **Display the configuration information of a specific logical unit.**

   ```
   # mpathadm show lu  /dev/rdsk/c0t600144F0DDA8BB1A000057208C880001d0s2

   Logical Unit:  /dev/rdsk/c0t600144F0DDA8BB1A000057208C880001d0s2
           mpath-support:  libmpscsi_vhci.so
           Vendor:  SUN
           Product:  ZFS Storage 7420
           Revision:  1.0
           Name Type:  unknown type
   ```

```
            Name:  600144f0dda8bb1a000057208c880001
            Asymmetric:  yes
            Current Load Balance:  shortest-path
            Logical Unit Group ID:  NA
            Auto Failback:  on
            Auto Probing:  NA

            Paths:
                    Initiator Port Name:
 iqn.1986-03.com.sun:01:21866c42ab36.5710b544,4000002a00ff
                    Target Port Name:  4000002a0000,iqn.1986-03.com.sun:02:56bf24c4-
e3e3-4ff0-e7b0-a6e6880e461d
                    Override Path:  NA
                    Path State:  OK
                    Disabled:  no

            Target Port Groups:
                    ID:  0
                    Explicit Failover:  no
                    Access State:  active optimized
                    Target Ports:
                            Name:  4000002a0000,iqn.1986-03.com.sun:02:56bf24c4-e3e3-4ff0-
e7b0-a6e6880e461d
                            Relative ID:  3
```

You can check the current access state of the `Target Port Groups` in the output. To ensure that the storage device supports explicit failover, you must verify that the value of `Explicit Failover` in the output is yes.

2. **Manually force a logical unit failover.**

    ```
    # mpathadm failover lu /dev/rdsk/c0t600144F0DDA8BB1A000057208C880001d0s2
    ```

    If this operation is successful, the access state of the `Target Port Groups` of the device changes as a result of the logical unit failover.

3. **Confirm the access state change.**

    ```
    # mpathadm show lu  /dev/rdsk/c0t600144F0DDA8BB1A000057208C880001d0s2
    Logical Unit:  /dev/rdsk/c0t600144F0DDA8BB1A000057208C880001d0s2
            mpath-support:  libmpscsi_vhci.so
            Vendor:  SUN
            Product:  ZFS Storage 7420
            Revision:  1.0
            Name Type:  unknown type
            Name:  600144f0dda8bb1a000057208c880001
            Asymmetric:  yes
    ```

```
           Current Load Balance:  shortest-path
           Logical Unit Group ID:  NA
           Auto Failback:  on
           Auto Probing:  NA

         Paths:
                 Initiator Port Name:
 iqn.1986-03.com.sun:01:21866c42ab36.5710b544,4000002a00ff
                 Target Port Name:  4000002a0000,iqn.1986-03.com.sun:02:56bf24c4-
e3e3-4ff0-e7b0-a6e6880e461d
                 Override Path:  NA
                 Path State:  OK
                 Disabled:  no

         Target Port Groups:
                 ID:  0
                 Explicit Failover:  yes
                 Access State:  active optimized
                 Target Ports:
                         Name:  4000002a0000,iqn.1986-03.com.sun:02:56bf24c4-e3e3-4ff0-
e7b0-a6e6880e461d
                         Relative ID:  3
```

Disabling and Enabling a Logical Unit Path

You can disable and enable a logical unit path depending on your requirement. The following sections describe how to disable or enable a logical unit path.

Disabling a Logical Unit Path

You can disable an LU path regardless of its operational state. To disable the selected LU path, the full path must be specified by using the initiator port name, target port name, and the LU.

You can use the mpathadm show lu command to list the initiator port name and target port name for a specific path. For more information, see "Displaying Multipathing Information" on page 21.

The following example shows how to disable the LU path:

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

You can use the `mpathadm show lu` command to verify that the path was disabled. For more information, see "Displaying Multipathing Information" on page 21.

Enabling a Logical Unit Path

If the path to an LU is disabled, you can use the `mpathadm enable` command to enable the logical unit path. To enable the LU path, you must specify the full path by using the initiator port name, target port name, and the LU.

The following example shows how to enable the LU path:

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

You can use the `mpathadm show lu` command to verify that the path is enabled. For more information, see "Displaying Multipathing Information" on page 21.

Configuring Third-Party Storage Devices

Before configuring any third-party device, you must ensure that the device is supported by Oracle Solaris. For information about proper vendor and product IDs, modes, and settings required for the device to work with multipathing software, see your third-party user documentation or third-party vendor for information.

`scsi_vhci` is a core kernel module of Oracle Solaris I/O multipathing. Multipath access depends on a device-specific `scsi_vhci` failover operation modules (`fops`). The default is for the `scsi_vhci` code to automatically call a probe function to each `fops` module, looking for the first probe result that indicates the device is supported.

A probe implementation determines support based on some combination of `scsi_inquiry` data. A device with `INQUIRY` data indicating T10 Target-Port-Group-Support (TPGS) (also known as ALUA) compliance will use the standards-based TPGS `fops` module. For more information, see the `scsi_inquiry(9S)` man page.

For noncompliant devices, a `fops` module probe will typically determine support based on a VID/PID match against a private compiled-in table. For more information about how to display the content of the table, see "Displaying the List of Supported Devices" on page 25.

The `scsi_vhci.conf` file supports a `scsi-vhci-failover-override` property, which allows you to add a supported symmetric device which is not listed in the table, or to disable multipath

support for a device, or to override the probe process. For more information about how to add a symmetric device which is not listed in the table see, "Configuring Third-Party Storage Devices" on page 33.

For more information, see the `scsi_vhci(4D)` man page.

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 the `format(8)` man page.

▼ How to Add Third-Party Devices

You can configure multipathing on third-party symmetric storage devices. In a symmetric storage device, all paths to the storage device are active and I/O commands can be issued through any path.

1. Become an administrator.

For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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 must 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 C, "Tuning Disk Target Driver Properties".

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

5. **Reboot the system.**

Configuring Multipathing on Tape Drives

Tape I/O multipathing is disabled by default on SPARC based and x86 based systems. Tape I/O multipathing does not support load-balancing because only one path is used to send I/Os.

Ensure that 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`.

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

You can use the `sg_inq` command to retrieve the inquiry data of the drive.

The instructions to retrieve internal VID and PID table are as follows:

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 36.
2. Reboot the system.

3. Check for the internal vendor and product information.

```
Vendor:  EXAMPLE.Inc
Product:  SAMPLE-4A3
--
Vendor:  EXAMPLE.Inc
Product:  SAMPLE-4DT
```

4. Add the VID and PID pair to the `scsi_vhci.conf` file and specify the `fops` module 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 34.

▼ 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 11.4*.

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 inquiry data of the drive has set any of the TPGS bits, uncomment the following line:

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

   - If the inquiry data of the drive 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 you enable tape I/O multipathing, existing device nodes under `/dev/rmt` are no longer valid because they will point to devices that do not exist after the 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 failover.

Configuring Multipathing on a Port Basis

You can disable or enable multipathing on a specific FC host bus adapter (HBA) controller ports.

Port Configuration Considerations

Before you start configuring a port with multipathing feature, consider the following factors:

- FC global and per-port multipath setting is specified in the `/kernel/drv/fp.conf` file.

  Per-port multipath settings have priority over the global settings. When global multipathing is enabled and if a specific port is disabled for multipathing, that port will not be available in the multipathing configuration. Conversely, when global multipathing is disabled and if a specific port enabled for multipathing, that port is listed in the appropriate `driver.conf` file.

- If a device has more than one path to the host, all paths to the device must either be enabled or disabled for I/O multipathing.

- Configuring multipathing on a port enables the Oracle Solaris I/O multipathing feature to co-exist with other third-party multipathing solutions. However, devices and paths must not be shared between the Oracle Solaris I/O multipathing software and other third-party multipathing solutions.

▼ How to Configure Multipathing on a Port

This procedure describes how to configure multipathing on a port. Depending on how many ports you want the multipathing feature to control; you can disable or enable multipathing globally or for a specific port.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Determine the HBA controller ports that you want the multipathing feature to control.**

To list the available devices, perform an `ls -l` command on the `/dev/cfg` directory. For example:

```
# ls -l
total 16
lrwxrwxrwx   1 root     root          60 Feb 26 15:33 c1 -> ../../devices/pci@400/pci@2/
pci@0/pci@4/scsi@0/iport@40:scsi
lrwxrwxrwx   1 root     root          60 Feb 26 15:33 c3 -> ../../devices/pci@400/pci@1/
pci@0/pci@0/SUNW,qlc@0/fp@0,0:fc
lrwxrwxrwx   1 root     root          62 Feb 26 15:33 c4 -> ../../devices/pci@400/pci@1/
pci@0/pci@0/SUNW,qlc@0,1/fp@0,0:fc
lrwxrwxrwx   1 root     root          59 Feb 26 15:33 c5 -> ../../devices/pci@400/pci@1/
pci@0/pci@4/scsi@0/iport@f:scsi
lrwxrwxrwx   1 root     root          59 Feb 26 15:33 c6 -> ../../devices/pci@400/pci@2/
pci@0/pci@4/scsi@0/iport@f:scsi
lrwxrwxrwx   1 root     root          62 Feb 26 15:33 c7 -> ../../devices/pci@400/pci@2/
pci@0/pci@a/SUNW,emlxs@0/fp@0,0:fc
lrwxrwxrwx   1 root     root          64 Feb 26 15:33 c8 -> ../../devices/pci@400/pci@2/
pci@0/pci@a/SUNW,emlxs@0,1/fp@0,0:fc
drwxr-xr-x   5 root     root           5 Feb 26 15:33 usb0
drwxr-xr-x   4 root     root           4 Feb 26 15:33 usb1
drwxr-xr-x  18 root     root          18 Feb 26 15:33 usb2
```

Note - In the example, 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.

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 multipathing on a specific FC HBA port according to your requirement.**

   - 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";
     ```

     For example, the following entries disable multipathing on all FC HBA controller ports except for the two 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 the dump configuration are updated to reflect the device name changes and the `zpools` are imported from the new device names.

6. **If necessary, configure your applications to use new device names.**

Disabling and Enabling Oracle Solaris I/O Multipathing

Some more considerations while you are disabling or enabling multipathing are as follows:

- Device names in the `/dev` and `/devices` trees change. Applications that use device names directly must be configured to use the new names when you change multipath configuration from enable to disable or vice versa.
- Entries to `/etc/vfstab` entries and dump configuration change. The `/etc/vfstab` file and the dump configuration of the system contain references to device names. The `stmsboot` command automatically updates the `/etc/vfstab` file dump configuration with the new device names.

You can use the `stmsboot` command to enable or disable multipathing for FC and SAS devices. The `stmsboot` command updates the `/etc/vfstab` file and dump configuration to reflect device name changes during the next reboot. The following considerations apply to the `stmsboot` command:

- You must reboot immediately after you run the `stmsboot` command.
- Because the `stmsboot` command reboots the machine to complete the configuration, use the `eeprom` to ensure the system boots from the current boot device.
- The `stmsboot` command saves a copy of `/kernel/drv/fp.conf`, `/kernel/drv/mpt.conf`, and `/etc/vfstab` files before modifying the files. If you encounter unexpected problems while using the `stmsboot` command, see Appendix B, "Troubleshooting Multipathing Problems".

▼ How to Disable Multipathing

This procedure describes how to disable multipathing on all multipath-capable devices. If you want to disable multipathing only on a specific FC or SAS HBA port, see "Configuring Multipathing on a Port Basis" on page 37.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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

   /pci@400/pci@1/pci@0/pci@0/SUNW,qlc@0/fp@0,0
   /pci@400/pci@1/pci@0/pci@0/SUNW,qlc@0,1/fp@0,0
   /pci@400/pci@2/pci@0/pci@a/SUNW,emlxs@0/fp@0,0
   /pci@400/pci@2/pci@0/pci@a/SUNW,emlxs@0,1/fp@0,0
   /pci@400/pci@2/pci@0/pci@4/scsi@0/iport@40
   /pci@400/pci@2/pci@0/pci@4/scsi@0/iport@f
   /pci@400/pci@1/pci@0/pci@4/scsi@0/iport@f

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

   Do you wish to continue? [y/n] (default: y)
   ```

   You are prompted to reboot. During the reboot, the /etc/vfstab file and the dump configuration are updated to reflect the device name changes and the zpools are imported from the new device names.

3. **If necessary, configure your applications to use new device names.**

▼ How to Enable Multipathing

The multipathing feature automatically recognizes Oracle-supported devices. This procedure describes how to enable multipathing on all multipath-capable devices. If you want to enable

multipathing only on a specific FC or SAS HBA port, see "Configuring Multipathing on a Port Basis" on page 37.

1. **Become an administrator.**
   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Enable multipathing.**

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

/pci@400/pci@1/pci@0/pci@0/SUNW,qlc@0/fp@0,0
/pci@400/pci@1/pci@0/pci@0/SUNW,qlc@0,1/fp@0,0
/pci@400/pci@2/pci@0/pci@a/SUNW,emlxs@0/fp@0,0
/pci@400/pci@2/pci@0/pci@a/SUNW,emlxs@0,1/fp@0,0
/pci@400/pci@2/pci@0/pci@4/scsi@0/iport@40
/pci@400/pci@2/pci@0/pci@4/scsi@0/iport@f
/pci@400/pci@1/pci@0/pci@4/scsi@0/iport@f

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

Do you wish to continue? [y/n] (default: y)
```

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

3. **If necessary, configure your applications to use new device names.**

▼ How to Disable Multipathing for a Specific Device

You can disable Oracle Solaris I/O multipathing 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 11.4*.

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 C, "Tuning Disk Target Driver Properties".

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

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

```
# stmsboot -u
```

During the reboot, the /etc/vfstab file and the dump configuration are updated to reflect the device name changes and the zpools are imported from the new device names.

6. **If necessary, configure your applications to use new device names.**

Configuring Automatic Failback

Some storage devices have controllers configured as primary and secondary as a part of the array configuration. The secondary paths might operate at a lower performance level than the

primary paths. The multipathing feature uses the primary path to communicate with the storage device and to keep the secondary path on standby.

When the primary path fails, the Oracle Solaris I/O multipathing feature automatically directs all the I/O traffic over the secondary path, with the primary path taken offline. This process is called as failover operation. When the failure associated with the primary path is repaired, the multipathing feature 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.

Note - The failover and failback operation is applicable only on devices with explicit failover.

You can disable the automatic failback operation so that the multipathing feature does not automatically failback to the primary path. When the failure associated with the primary path is repaired, you can do a manual failback operation by using the `mpathadm failover` command. For more information, see the `mpathadm(8)` 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 11.4*.

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

3. **Disable or enable automatic failback capability by changing the `auto-failback` entry in the `/etc/driver/drv/scsi_vhci.conf` file.**

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

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

5. **Reboot the system.**

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

Configuring Automatic Failback for Multipathing Support

Asymmetric 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. In most cases, the standby path is a suboptimal path. When automatic failback is enabled, the initial path becomes 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 11.4*.

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
  shortest-path
 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

  Vendor:  LSI
  Product:  INF-01-00
  Revision:
  Supported Load Balance Types:
   round-robin

  Vendor:  NETAPP
  Product:  INF-01-00
  Revision:
  Supported Load Balance Types:
   round-robin

  Vendor:  ENGENIO
  Product:  INF-01-00
  Revision:
  Supported Load Balance
.
.
```

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. See .

Configuring Load Balancing Policy for Oracle Solaris I/O Multipathing

I/O multipathing support in Oracle Solaris aggregates the physical paths of a disk device to virtual paths that is managed by the `scsi_vhci` driver. Creating virtual paths are useful for fail

over when a path becomes offline and for load balancing purposes. When there are multiple paths available for I/O, you can specify the load balancing policy by using the load-balance tunable parameter in the `scsi_vhci.conf` file. The policy indicates how a physical path is selected for each I/O instance.

The syntax for the load balancing tunable parameter is as follows:

`load-balance=`*policy*`;`

The following list describes the available load balancing policies:

- `none` or `same-path` – No dynamic load balancing is used for the I/O issued to a given device or a LU. When a physical path is selected, the system continues to use the same path until that path fails. A new path from the other active physical path is selected as a part of the fail over feature. This new path is used for the LU until it fails. The selection mechanism of this physical path is not defined. The policy `same-path` is an alias for `none`.
- `round-robin` – This is the simplest and the most frequently used load balancing policy. When the `round-robin` policy is set, I/O requests are routed to a set of active physical paths in a circular pattern.
- `logical-block` – This policy has a parameter `region-size` in the `scsi_vhci.conf` file. The `region-size` parameter identifies a power-of-two block-size region. The default value of `region-size` is 18, or 128 with 512-byte blocks.

  The logical block address (LBA) space of the device are partitioned into regions with the specified `region-size`. Each region is assigned to one of the physical paths in the active set in a round robin pattern. This policy selects a path that its corresponding regions cover the starting logical block address (LBA) of the I/O in question. If the load-balance policy is `logical-block`, the `region-size` is by default 18 as follows:

  `region-size=18;`

For more information, see "Displaying Logical Unit Information" on page 22.

▼ How to Change the Load Balancing Policy

1. Become an administrator.

For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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

Note - Do not edit the delivered system default version in the `/kernel/drv` directory.

---

3. **Edit the `/etc/driver/drv/scsi_vhci.conf` file to configure the load balancing policy tunable.**

For example, set the value as `round-robin` for the load balancing tunable:

```
load-balance="round-robin";
```

◆ ◆ ◆ **C H A P T E R   4**

4

Configuring Fibre Channel Devices

This chapter provides a high-level overview describing the configuration of Fibre Channel (FC) devices. The Oracle Solaris OS supports automatic configuration of fabric, point-to-point, and Fibre Channel Arbitrated Loop (FC-AL) connected devices. You must disable the automatic configuration feature to manually configure the FC devices.

This chapter contains the following topics:

- "Fibre Channel Features" on page 49
- "Configuring Fibre Channel Devices" on page 50
- "Displaying Fibre Channel Devices" on page 51
- "Removing Fibre Channel Devices" on page 59
- "Booting Oracle Solaris From Fibre Channel Devices" on page 60
- "Configuring Virtual Fibre Channel Ports" on page 67
- "Configuring FCoE Ports" on page 70
- "Configuring Oracle Solaris to Work With CEE-DCBX FCoE Switches" on page 75
- "Configuring IPFC SAN Devices" on page 83

For more information about how to manually configure FC devices, see Appendix A, "Manual Configuration for Fibre Channel Devices".

Fibre Channel Features

FC in Oracle Solaris has the following key features:

- Dynamic storage discovery – Oracle Solaris 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.
- FC booting – osol; supports booting from FC devices.
- Fibre Channel virtualization – N Port ID Virtualization (NPIV) is an extension to the FC standard, which enables an FC port to create many virtual ports on a physical port. This

feature is useful for virtualization environments such as LDoms, and Oracle VM Server for SPARC based and x86 based systems.

- Fibre Channel SR-IOV – This technology is available for Oracle VM Server for SPARC. For more information about Oracle VM server on SPARC, see *Oracle VM Server for SPARC 3.6 Administration Guide*.

- Fibre Channel over Ethernet (FCoE) – A new T11 standard to transport encapsulated FC frames over enhanced Ethernet. The Oracle Solaris FCoE is a software implementation that is designed to work with normal Ethernet controllers.

- IP over fibre channel (IPFC) - Creating IP interface into the fibre channel protocol. See RFC 2625. For more information, see "Configuring IPFC SAN Devices" on page 83.

- OpenBoot PROM (OBP) commands – Can scan and access FC storage devices.

- 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 interfaces (APIs) enable the management of FC HBAs, and provide a standard-based interface for other applications. These FC-HBA APIs can be used to gather information about FC HBAs.

  For more information about common FC-HBA APIs, see the `libhbaapi(3LIB)` man page and Appendix F, "Supported FC-HBA API," in *Writing Device Drivers in Oracle Solaris 11.4*.

Note - Emulex supports IPFC (IP over Fibre Channel) on 4G and 8G version HBAs, and does not support 16G and higher version HBAs.

QLogic supports FCIP 4G version HBAs, and does not support 8G and higher version HBAs.

For more information about FC specifications, go to T11 - Fibre Channel Interfaces.

Configuring Fibre Channel Devices

The basic steps to configure an FC device are as follows:

1. Configure FC switches. You can configure ports and zones according to the vendor-specific documentation for switches.
2. Configure storage devices. You can use LUN masking to enable specific LUNs to be seen by specific hosts. For more information about LUN masking, see your vendor-specific storage documentation.
3. Connect arrays, other storage devices, and Oracle Solaris hosts to a SAN.
4. Configured FC devices are made available to the host automatically during installation, boot time, and run time.

If a new logical unit is added to a storage device during runtime, the new logical unit is configured automatically only if there is I/O traffic to another logical unit in the same storage device. A logical unit cannot be configured automatically if there is no I/O traffic. You can use the `cfgadm` command to manually probe the device.

For example:

```
# cfgadm -c configure c3::10000000c94c0cec
```

Displaying Fibre Channel Devices

The commonly used commands for adding, removing, and displaying FC devices are as follows:

`fcinfo` – Collects and displays administrative information about FC-HBA ports on hosts, remote ports, and configured logical units. For more information, see the `fcinfo`(8) man page. The following examples show the most common administration operations that can be performed by using the `fcinfo` command:

EXAMPLE 2   Display Information of the HBA Ports and Their Link Statistics

To list information of the HBA ports and their link statistics use the `fcinfo hba-port -l` command.

```
# fcinfo hba-port -l
HBA Port WWN: 2100000e1ec95100
        Port Mode: Initiator
        Port ID: 150300
        OS Device Name: /dev/cfg/c6
        Manufacturer: QLogic Corp.
        Model: 7023303
        Firmware Version: 8.05.00
        FCode/BIOS Version:  BIOS: 3.19; fcode: 4.02; EFI: 5.36;
        Serial Number: 463916R+1606293054
        Driver Name: qlc
        Driver Version: 161214-5.06a
        Type: N-port
        State: online
        Supported Speeds: 4Gb 8Gb 16Gb
        Current Speed: 8Gb
        Node WWN: 2000000e1ec95100
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
```

```
                            Loss of Signal Count: 0
                            Primitive Seq Protocol Error Count: 0
                            Invalid Tx Word Count: 0
                            Invalid CRC Count: 0
                    Max NPIV Ports: 253
                    NPIV port list:
            HBA Port WWN: 2100000e1ec95101
                    Port Mode: Initiator
                    Port ID: 20801
                    OS Device Name: /dev/cfg/c7
                    Manufacturer: QLogic Corp.
                    Model: 7023303
                    Firmware Version: 8.05.00
                    FCode/BIOS Version:  BIOS: 3.19; fcode: 4.02; EFI: 5.36;
                    Serial Number: 463916R+1606293054
                    Driver Name: qlc
                    Driver Version: 161214-5.06a
                    Type: NL-port
                    State: online
                    Supported Speeds: 4Gb 8Gb 16Gb
                    Current Speed: 8Gb
                    Node WWN: 2000000e1ec95101
                    Link Error Statistics:
                            Link Failure Count: 0
                            Loss of Sync Count: 0
                            Loss of Signal Count: 0
                            Primitive Seq Protocol Error Count: 0
                            Invalid Tx Word Count: 37825
                            Invalid CRC Count: 0
                    Max NPIV Ports: 253
                    NPIV port list:
```

EXAMPLE 3   Display All The Remote Ports on All the HBAs

To list all the remote ports on all the HBAs, use the `fcinfo remote-port` command.

```
# fcinfo remote-port
HBA Port WWN: 2100000e1ec95100
  Count of Remote Port(s): 4
    Remote Port WWN: 10000000c97ecedc
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port ID: 150400
        Port Symbolic Name:
        Node WWN: 20000000c97ecedc
    Remote Port WWN: 21000024ff41d9a4
```

```
        Active FC4 Types: SCSI
        SCSI Target: no
        Port ID: 150200
        Port Symbolic Name: qlc(0,0)
        Node WWN: 20000024ff41d9a4
    Remote Port WWN: 201600a0b82a38bc
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port ID: 150700
        Port Symbolic Name: SUN     CSM200_R        0660
        Node WWN: 200600a0b82a38bc
    Remote Port WWN: 21000024ff51bb4a
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port ID: 150100
        Port Symbolic Name: qlt0,0
        Node WWN: 20000024ff51bb4a
HBA Port WWN: 2100000e1ec95101
  Count of Remote Port(s): 11
    Remote Port WWN: 10000000c9c07a6d
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port ID: 20b00
        Port Symbolic Name:
        Node WWN: 20000000c9c07a6d
    Remote Port WWN: 21000024ff2cd75a
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port ID: 20200
        Port Symbolic Name:
        Node WWN: 20000024ff2cd75a
    Remote Port WWN: 202200a0b85a8c82
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port ID: 20400
        Port Symbolic Name: SUN     LCSM100_F       0735
        Node WWN: 200200a0b85a8c82
    Remote Port WWN: 21000024ff2cd75b
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port ID: 20300
        Port Symbolic Name:
        Node WWN: 20000024ff2cd75b
    Remote Port WWN: 10000000c97ecedd
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port ID: 20900
        Port Symbolic Name:
```

```
            Node WWN: 20000000c97ecedd
    Remote Port WWN: 202300a0b85a8c82
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port ID: 20600
        Port Symbolic Name: SUN     LCSM100_F       0735
        Node WWN: 200200a0b85a8c82
    Remote Port WWN: 203200a0b85a8c82
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port ID: 20700
        Port Symbolic Name: SUN     LCSM100_F       0735
        Node WWN: 200200a0b85a8c82
    Remote Port WWN: 2100001b321a45e5
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port ID: 20100
        Port Symbolic Name:
        Node WWN: 2000001b321a45e5
    Remote Port WWN: 203300a0b85a8c82
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port ID: 20500
        Port Symbolic Name: SUN     LCSM100_F       0735
        Node WWN: 200200a0b85a8c82
    Remote Port WWN: 2101001b323a45e5
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port ID: 20000
        Port Symbolic Name:
        Node WWN: 2001001b323a45e5
    Remote Port WWN: 10000000c9c07a6c
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port ID: 20a00
        Port Symbolic Name:
        Node WWN: 20000000c9c07a6c
```

EXAMPLE 4   Display All the Remote Ports Visible Through a Local HBA Port and Their Link
                 Statistics

To list all the remote ports visible through a local HBA port and their link statistics, use the
`fcinfo remote-port -ls -p` *PWWN* command.

```
# fcinfo remote-port -ls -p 2100000e1ec95100
HBA Port WWN: 2100000e1ec95100
```

```
Count of Remote Port(s): 4
  Remote Port WWN: 10000000c97ecedc
       Active FC4 Types: SCSI,IP
       SCSI Target: no
       Port ID: 150400
       Port Symbolic Name:
       Node WWN: 20000000c97ecedc
       Link Error Statistics:
              Link Failure Count: 0
              Loss of Sync Count: 6
              Loss of Signal Count: 2
              Primitive Seq Protocol Error Count: 0
              Invalid Tx Word Count: 33
              Invalid CRC Count: 0
  Remote Port WWN: 21000024ff41d9a4
       Active FC4 Types: SCSI
       SCSI Target: no
       Port ID: 150200
       Port Symbolic Name: qlc(0,0)
       Node WWN: 20000024ff41d9a4
       Link Error Statistics:
              Link Failure Count: 0
              Loss of Sync Count: 0
              Loss of Signal Count: 0
              Primitive Seq Protocol Error Count: 0
              Invalid Tx Word Count: 4
              Invalid CRC Count: 0
  Remote Port WWN: 201600a0b82a38bc
       Active FC4 Types: SCSI
       SCSI Target: yes
       Port ID: 150700
       Port Symbolic Name: SUN     CSM200_R        0660
       Node WWN: 200600a0b82a38bc
       Link Error Statistics:
              Link Failure Count: 12
              Loss of Sync Count: 1024
              Loss of Signal Count: 20842
              Primitive Seq Protocol Error Count: 0
              Invalid Tx Word Count: 0
              Invalid CRC Count: 0
  Remote Port WWN: 21000024ff51bb4a
       Active FC4 Types: SCSI
       SCSI Target: yes
       Port ID: 150100
       Port Symbolic Name: qlt0,0
       Node WWN: 20000024ff51bb4a
       Link Error Statistics:
              Link Failure Count: 1
```

```
                        Loss of Sync Count: 0
                        Loss of Signal Count: 0
                        Primitive Seq Protocol Error Count: 0
                        Invalid Tx Word Count: 0
                        Invalid CRC Count: 0
             LUN: 0
               Vendor: SUN
               Product: COMSTAR
               OS Device Name: Unknown
```

EXAMPLE 5   Display All the Logical Units and Its Information

To list all the logical units and its information, use the `fcinfo lu -v` command.

```
OS Device Name: /dev/rmt/3n
        HBA Port WWN: 2100000e1ec95101
        Controller: /dev/cfg/c7
                Remote Port WWN: 500104f000ad3d19
                        LUN: 0
                        State: unknown
                        Class:
        Vendor: SUN
        Product: EXAMPLE-TD7
        Device Type: Tape Device
        Unformatted capacity:  0.000 MBytes
OS Device Name: /dev/rmt/5n
        HBA Port WWN: 2100000e1ec95101
        Controller: /dev/cfg/c7
                Remote Port WWN: 500104f000ad3d0e
                        LUN: 0
                        State: unknown
                        Class:
        Vendor: SUN
        Product: EXAMPLE-TD5
        Device Type: Tape Device
        Unformatted capacity:  0.000 MBytes
```

EXAMPLE 6   Display All the Logical Units on a Remote Port

To list all the logical units on a remote port, use the `fcinfo lu -v -P` *PWWN* command.

```
# fcinfo lu -v -P 500104f000ad3d19
OS Device Name: /dev/rmt/0n
```

```
        HBA Port WWN: 2100000e1ec95101
        Controller: /dev/cfg/c7
                Remote Port WWN: 500104f000ad3cfe
                        LUN: 0
                        State: unknown
                        Class:
        Vendor: STK.EXAMPLE
        Product: T10000A
        Device Type: Tape Device
        Unformatted capacity:  0.000 MBytes
```

cfgadm – Dynamically reconfigures storage devices and FC devices. This command is used most frequently to configure storage devices on a SAN. This command also provides configuration administration operations on dynamically reconfigurable hardware resources. For more information, see the cfgadm(8) man page. The following examples shows the different administration operations that can be performed by using the cfgadm command:

EXAMPLE 7   Display All the Attachment Points

The cfgadm command lists all attachment points except dynamic attachment points.

```
# cfgadm
Ap_Id                      Type        Receptacle   Occupant      Condition
c1                         scsi-sas    connected    configured    unknown
c2                         scsi-sas    connected    configured    unknown
c3                         scsi-sas    connected    configured    unknown
c4                         scsi-sas    connected    configured    unknown
c5                         scsi-sas    connected    configured    unknown
c6                         fc-fabric   connected    unconfigured  unknown
c7                         fc-public   connected    unconfigured  unknown
c8                         scsi-sas    connected    configured    unknown
c9                         scsi-sas    connected    configured    unknown
hca:10E00001430070         IB-HCA      connected    configured    ok
ib                         IB-Fabric   connected    configured    ok
usb0/1                     unknown     empty        unconfigured  ok
usb0/2                     unknown     empty        unconfigured  ok
```

EXAMPLE 8   Display the Current Configurable Hardware Information

To list the current configurable hardware information, including those represented by dynamic attachment points, use the cfgadm -al command.

```
# cfgadm -al
Ap_Id                       Type        Receptacle  Occupant     Condition
c1                          scsi-sas    connected   configured   unknown
c1::dsk/c1t0d0              CD-ROM      connected   configured   unknown
c2                          scsi-sas    connected   configured   unknown
c2::w5000cca0162922a1,0    disk-path   connected   configured   unknown
c3                          scsi-sas    connected   configured   unknown
c3::w5000cca0162b81b5,0    disk-path   connected   configured   unknown
c4                          scsi-sas    connected   configured   unknown
c4::w5000cca0162b899d,0    disk-path   connected   configured   unknown
c5                          scsi-sas    connected   configured   unknown
c5::w5000cca0162a6bcd,0    disk-path   connected   configured   unknown
c6                          fc-fabric   connected   unconfigured unknown
c6::10000000c97ecedc       unknown     connected   unconfigured unknown
c6::201600a0b82a38bc       disk        connected   unconfigured unknown
c6::21000024ff41d9a4       unknown     connected   unconfigured unknown
c6::21000024ff51bb4a       unknown     connected   unconfigured unknown
c7                          fc-public   connected   unconfigured unknown
c7::10000000c97ecedd       unknown     connected   unconfigured unknown
c7::10000000c9c07a6c       unknown     connected   unconfigured unknown
```

EXAMPLE 9 Display the Status of the LUN

To know the status of the LUN, use the `cfgadm -al -o show_SCSI_LUN` command. For
example:

```
# cfgadm -al -o show_SCSI_LUN
Ap_Id                       Type        Receptacle  Occupant     Condition
c6                          fc-fabric   connected   configured   unknown
c6::10000000c97ecedc       unknown     connected   configured   unknown
c6::201600a0b82a38bc       disk        connected   unconfigured unknown
c6::21000024ff41d9a4       unknown     connected   unconfigured unknown
c6::21000024ff51bb4a       unknown     connected   unconfigured unknown
c7                          fc-public   connected   unconfigured unknown
c7::10000000c97ecedd       unknown     connected   unconfigured unknown
c7::10000000c9c07a6c       unknown     connected   configured   unknown
c7::10000000c9c07a6d       unknown     connected   unconfigured unknown
c7::202200a0b85a8c82       disk        connected   unconfigured unknown
c7::202300a0b85a8c82       disk        connected   unconfigured unknown
c7::203200a0b85a8c82       disk        connected   unconfigured unknown
c7::203300a0b85a8c82       disk        connected   unconfigured unknown
c7::2100001b321a45e5       unknown     connected   unconfigured unknown
c7::21000024ff2cd75a       unknown     connected   unconfigured unknown
c7::21000024ff2cd75b       unknown     connected   configured   unknown
c7::2101001b323a45e5       unknown     connected   unconfigured unknown
```

`fcadm` – Performs administrative operations on FC HBA ports on a host. This command can reinitialize the ports, and also create and delete NPIV and FCoE ports. For more information, see "Configuring IPFC SAN Devices" on page 83, "Configuring FCoE Ports" on page 70 and the `fcadm(8)` man page.

`format` – Identifies devices that are connected to the system, and provides the ability to label disk devices. For more information, see the `format(8)` man page.

Removing Fibre Channel Devices

The following procedure assumes that the you have deleted the logical unit by changing the LUN masking configuration on the storage device, so that the logical unit is not visible to the host. This procedure describes how to remove an FC device.

▼ How to Remove a Fibre Channel Device

Before You Begin You must stop using the logical unit device which you want to remove. For example, you can either unmount the file system or export the ZFS pool.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Verify that the LUNs are reported as unusable in your system by using the following command.**

   The removed LUNs are marked as unusable.

```
# cfgadm -al -o show_SCSI_LUN
Ap_Id                        Type         Receptacle   Occupant     Condition
c6                           fc-fabric    connected    configured   unknown
c6::10000000c97ecedc         unknown      connected    configured   unknown
c6::201600a0b82a38bc         disk         connected    unconfigured unknown
c6::21000024ff41d9a4         unknown      connected    unconfigured unknown
c6::21000024ff51bb4a         unknown      connected    unconfigured unknown
c7                           fc-public    connected    unconfigured unknown
c7::10000000c97ecedd         unknown      connected    unconfigured unknown
c7::10000000c9c07a6c         unknown      connected    unusable     unknown
c7::10000000c9c07a6d         unknown      connected    unconfigured unknown
c7::202200a0b85a8c82         disk         connected    unconfigured unknown
```

```
c7::202300a0b85a8c82          disk         connected    unconfigured unknown
c7::203200a0b85a8c82          disk         connected    unconfigured unknown
c7::203300a0b85a8c82          disk         connected    unconfigured unknown
c7::2100001b321a45e5          unknown      connected    unconfigured unknown
c7::21000024ff2cd75a          unknown      connected    unconfigured unknown
c7::21000024ff2cd75b          unknown      connected    unconfigured unknown
c7::2101001b323a45e5          unknown      connected    unconfigured unknown
```

3. **Remove the unusable entries from the host configuration.**

   ```
   # cfgadm -c unconfigure -o unusable_SCSI_LUN c7::10000000c9c07a6c
   ```

▼ How to Remove an Fibre Channel Target Port

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **If the target port that you want to remove is the last path to the device, unmount the file systems or export the ZFS pools for all logical units used by the host on the storage device.**

3. **Unconfigure the target port from your system by using the `cfgadm -c unconfigure` *target-port*command. For example:**

   ```
   # cfgadm -c unconfigure c7::10000000c9c07a6c
   ```

   > **Note -** The changes made by the `cfgadm -c unconfigure` command do not persist after a reboot.

4. **(Optional) To make the devices permanently unavailable to the system, you can use switch zoning or LUN masking on the storage device.**

Booting Oracle Solaris From Fibre Channel Devices

You can set up a SPARC or an x86 server with Oracle Solaris to boot from an FC disk device. On a SPARC server, the installer configures OBP automatically. On an x86 server, you must configure the HBA BIOS manually to boot Oracle Solaris from an FC device.

See your hardware provider for instructions about installing HBA hardware.

Configuring Fibre Channel Boot Devices on the SPARC Platform

You can set up a SPARC server with Oracle Solaris to boot from a FC disk device. FC devices are made available automatically to the host during the boot time allowing you to access FC boot devices during Oracle Solaris installation as an internal boot device. For more information, see *Automatically Installing Oracle Solaris 11.4 Systems*.

Configuring Fibre Channel Boot Devices on an x86 Platform

You can set up an x86 server to boot from an FC device, which can be accessed during Oracle Solaris installation as an internal boot device. You can choose one of the FC devices to install the OS. For more information about Oracle Solaris installation, see *Automatically Installing Oracle Solaris 11.4 Systems*.

After the OS installation and before rebooting, you must gather the configuration information by issuing the `fcinfo` command on the newly installed LUN. The `fcinfo` command output provides a map from OS device name (c#t#d) to the World Wide Name (WWN) of the target port and LUN. You must record the HBA port WWN, target port WWN, and LUN information.

During the reboot after the installation, you must change the x86 BIOS and FC HBA BIOS setup to identify the FC initiator that must be used to boot the OS. Use the WWN and LUN to set the HBA BIOS for each HBA port, which can be used for booting. Change the system BIOS to perform booting from a disk.

If you are using a custom disk, do not remove the overlap (s2) partition. The x86 installer has a dependency on this partition.

Note - Console access is required to perform HBA and system BIOS changes.

▼ How to Perform OS Installation on FC Devices on an x86 Platform

This procedure describes how to perform a DVD installation or a network-based installation of Oracle Solaris on an x86 based system for FC devices. For more information, see *Automatically Installing Oracle Solaris 11.4 Systems*.

1. **Install the Oracle Solaris OS by selecting one of the following methods:**

   - Automatic installation – You must choose network boot option from the BIOS of the x86 based system to start the automated installation.
   - Text installation – You must choose the following option from the GRUB menu when booted from media or when booted from an installation server to start the text installation.

     ```
     Oracle Solaris 11.4 Text Installer and command line
     ```

   For more information, see *Automatically Installing Oracle Solaris 11.4 Systems*.

2. **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.

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

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

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

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

3. **Before rebooting after the installation completes, issue the `fcinfo lu` command on the logical unit that was selected during installation.**

   ```
   # fcinfo lu /dev/rdsk/c0t600015D0002028000000000000001142d0s2
   ```

   You can use the sample output in the figure to map the OS device name to the HBA WWN and array WWN as follows:

   - OS Device Name = c0t600015d0002028000000000000001142d0

- Array WWN = 201600A0B811BF96, LUN 0
- HBA WWNs = 210000E08B8300CE

4. **Reboot the system.**

During the reboot process, enter the HBA #1 BIOS screen and specify the boot device to be the FC LUN on which you have just installed Oracle Solaris.

Follow this step for each HBA to be used for multipathing, and specify the boot device to be the FC LUN on which you have installed Oracle Solaris.

- **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.**

```
**************************************************************************
**********************************Select Host Adapter*********************************
***********  **Adapter Type  Address    Slot Bus Device Function **  *
***********  *                                                    *  *
***********  * QLA2462       9400       01   02  01      1        *  *
***********  * QLA2462       9800       01   02  01      0        *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  *                                                    *  *
***********  ***************************************************** *
**************************************************************************
**************************************************************************
 Use <Arrow keys> to move cursor, <Enter> to select option, <Esc> to backup
```

    c. **Configure the boot device.**

i **Select Configuration Settings.**

```
**************************************************************************
**********************************Selected Adapter***********************************
***********      Adapter Type   Address     Slot Bus Device Function    ************
***********      QLA2462         9400        01  02  01      1           ************
***********************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**********************************Fast!UTIL Options**********************************
********************                                       *************************
********************            **************************  *************************
********************            * Configuration Settings  *  *************************
********************            * Scan Fibre Devices       *  *************************
********************            * Fibre Disk Utility       *  *************************
********************            * Loopback Data Test       *  *************************
********************            * Select Host Adapter      *  *************************
********************            * Exit Fast!UTIL           *  *************************
********************            **************************  *************************
********************                                       *************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
      Use <Arrow keys> to move cursor, <Enter> to select option, <Esc> to backup
```

ii **Choose Selectable Boot Settings.**

```
**************************************************************************
*********************************Selected Adapter***********************************
***********      Adapter Type   Address     Slot Bus Device Function    ************
***********      QLA2462         9800        01  02  01      0           ************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
********************************Selectable Boot Settings********************************
********                                                            *********
********* Selectable Boot:                      Enabled              *********
********* (Primary) Boot Port Name,Lun:         201600A0B811BF96,  0  *********
*********            Boot Port Name,Lun:         0000000000000000,  0  *********
*********            Boot Port Name,Lun:         0000000000000000,  0  *********
*********            Boot Port Name,Lun:         0000000000000000,  0  *********
********                                                            *********
********            Press "C" to clear a Boot Port Name entry       *********
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
      Use <Arrow keys> and <Enter> to change settings, <Esc> to exit
```

iii **Make sure that Selectable Boot is set to enable.**

In this menu, you can select the boot device or 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.**

```
********************************************************************
*****************************Selected Adapter***********************
***********    Adapter Type  Address    Slot Bus Device Function   *
***********    QLA2462        9800        01  02  01      0         *
********************************************************************
********************************************************************
*****************************Adapter Settings**********************
***************                                                   *
***************    BIOS Address:              CF000               *
***************    BIOS Revision:             1.24                *
***************    Adapter Serial Number:     C49344              *
***************    Interrupt Level:           11                  *
***************    Adapter Port Name:         210000E08B8300CE*
***************    Host Adapter BIOS:         Enabled             *
***************    Frame Size:                2048                *
***************    Loop Reset Delay:          5                   *
***************    Adapter Hard Loop ID:      Disabled            *
***************    Hard Loop ID:              0                   *
***************    Spinup Delay:              Disabled            *
***************    Connection Options:        2                   *
***************    Fibre Channel Tape Support:Enabled             *
***************    Data Rate:                 2                   *
********************************************************************
       Use <Arrow keys> and <Enter> to change settings, <Esc> to exit
```

```
********************************************************************
*****************************Selected Adapter***********************
***********    Adapter Type  Address    Slot Bus Device Function   *
***********    QLA2462        9800        01  02  01      0         *
********************************************************************
********************************************************************
********************************************************************
********************************************************************
*******************************Selectable Boot Settings************
********                                                          *
******** Selectable Boot:                    Enabled              *
******** (Primary) Boot Port Name,Lun:       201600A0B811BF96,  0 *
********           Boot Port Name,Lun:        0000000000000000,  0 *
********           Boot Port Name,Lun:        0000000000000000,  0 *
********           Boot Port Name,Lun:        0000000000000000,  0 *
********                                                          *
********           Press "C" to clear a Boot Port Name entry      *
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
       Use <Arrow keys> and <Enter> to change settings, <Esc> to exit
```

The figure shows the following modifications:

- Selectable boot = Enabled
- ARRAY WWN = 201600A0B811BF96
- ARRAY LUN = 0
- HBA WWN = 210000E08B8300CE

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

6. **Provide the system BIOS as per the access method of the vendor and specify the boot device to be the FC LUN on which you have installed Oracle Solaris.**

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

Configuring Virtual Fibre Channel Ports

NPIV is a Fibre Channel facility that enables one FC 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 can be used to group ports together by port WWN. It is the preferred method of zoning.

This section provides the steps used to configure N_Port ID Virtualization (NPIV) ports, also known as virtual FC ports.

Limitations of NPIV

NPIV limitations when virtualizing FC 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. Some of the targets in a SAN 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.
- Oracle Solaris I/O multipathing can be used with NPIV. Although for real path redundancy you must ensure that different paths are physically redundant.
- NPIV is supported only in a Fabric topology, and not in an FC-AL or point-to-point topology.

- 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 support up to 255 virtual ports, though this capability is defined by the resources on the switch. Switches might need to be updated with the latest firmware levels for NPIV support.

For more information about booting from vHBA, see "Booting From a Virtual LUN" in *Oracle VM Server for SPARC 3.6 Administration Guide*.

Managing NPIV Ports

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

You can use `fcinfo` and `fcadm` status commands to determine the status of NPIV ports, regardless of whether the ports are created by using the `fcadm` command. These commands also report the relationship between the physical port and the virtual ports hosted on a specific HBA port.

Other FC commands, such as `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 name and a 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 about 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 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` output.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. Create an NPIV port.

fcadm create-npiv-port -p *Virtual_Port_WWN* -n *Virtual_Node_WWN*
PhysicalPort_port_WWN

If you omit the -p and -n options, a random WWN will be assigned for the virtual port and
virtual node.

Example 10 Creating an NPIV Port

The following example creates an NPIV port on a physical HBA port with a WWN of
2100000e1ec95100, a virtual port WWN set to c300000000005100, and a virtual node WWN
set to c0007d32aa4c2b59.

```
# fcadm create-npiv-port -p c300000000005100 -n c0007d32aa4c2b59 2100000e1ec95100
```

▼ 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 11.4*.

2. Display the currently configured NPIV ports.

```
# fcinfo hba-port
```

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

Example 11 Displaying the Status of NPIV Ports

The following example shows that HBA port 2100000e1ec95100 has three virtual ports.

```
# fcinfo hba-port
HBA Port WWN: 2100000e1ec95100
        Port Mode: Initiator
        Port ID: 150300
        OS Device Name: /dev/cfg/c6
        Manufacturer: QLogic Corp.
        Model: 7023303
        Firmware Version: 8.05.00
        FCode/BIOS Version:  BIOS: 3.19; fcode: 4.02; EFI: 5.36;
        Serial Number: 463916R+1606293054
        Driver Name: qlc
        Driver Version: 161214-5.06a
```

```
Type: N-port
State: online
Supported Speeds: 4Gb 8Gb 16Gb
Current Speed: 8Gb
Node WWN: 2000000e1ec95100
Max NPIV Ports: 253
NPIV port list:
  Virtual Port1:
          Node WWN: c0007d32aa4c2b59
          Port WWN: c300000000005100
  Virtual Port2:
          Node WWN: c0007d1d0e534e62
          Port WWN: c200000000005100
  Virtual Port3:
           Node WWN: c0007db21db5c9b7
           Port WWN: c100000000005100
```

▼ 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. For more information, see "Displaying Fibre Channel Devices" on page 51.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Delete an NPIV port.**

   # **fcadm delete-npiv-port -p** *Virtual_Port_WWN PhysicalPort_port_WWN*

Example 12   Deleting an NPIV Port

The following example deletes an NPIV port on a physical HBA port with a WWN of 2100000e1ec95100.

fcadm delete-npiv-port -p c300000000005100 2100000e1ec95100

Configuring FCoE Ports

Fibre Channel Over Ethernet (FCoE) is a T11 standard that transports encapsulated FC frames over enhanced Ethernet. FCoE enables network convergence and cost-effective SAN expansion in large deployments.

This section provides information about how to configure the software FCoE port hosted on a normal Ethernet interface, and hardware and software information about FCoE ports on converged network adapters (CNAs).

Configuring Converged Network Adapter Based FCoE Ports

CNAs combine the functionality of FC-HBAs and Ethernet network interface card (NIC) 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.

Benefits of CNAs

Second-generation CNAs are generally used to set up the SAN. Using CNAs have the following advantages:

- Using CNAs in storage network reduces the number of adapters used in a storage network, which helps 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 (PCI-E) 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

Configuring Software FCoE Ports

The Oracle Solaris FCoE initiator is a software implementation that works with normal Ethernet controllers.

Limitations of FCoE

Oracle Solaris FCoE implementations have the following limitations:

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

FCoE is not supported on all hardware. FCoE works with Ethernet controllers that support 802.3x (PAUSE) and jumbo frames.

Managing 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 FC commands, such as `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. An FC 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 name 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 about 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.

■ Ensure the following services are enabled:

```
# svcs svc:/system/fcoe_initiator:default
STATE          STIME   FMRI
online         Dec_09  svc:/system/fcoe_initiator:default
```

As an administrator, you can use 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, you can use the following command to delete an FCoE port:

```
# 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, you can use the following command to display the status of currently configured FCoE ports:

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

For example:

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

You can use the following command 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

If new devices are added to an FC SAN or if there is a device misbehaving on a SAN, you can force an FCoE port reinitialization. In many cases, this operation can resolve problems in an FC SAN.

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

When you connect an FC switch or other FC ports in a SAN, the system generates a remote state change notification (RSCN). Other initiators rediscover the port after the RSCN is generated, and the FC login session is established or reused. The `fcadm force-lip` command is disruptive to I/Os, but I/Os continue to communicate. This command does not cause any data loss.

As an administrator, you can use the following command to force a link that is connected to a port to reinitialize:

 # `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 to 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 8, "Managing Converged Networks by Using Data Center Bridging" in *Managing Network Datalinks in Oracle Solaris 11.4*.

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

- "Using Priority-Based Flow Control Capable Ethernet Adapters" on page 76
- "Using Converged Network Adapters" on page 78
- "Using a Non CEE-DCBX FCoE Switch" on page 79

Using Priority-Based Flow Control Capable Ethernet Adapters

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   2   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 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 Oracle Solaris 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
- CNAs

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

FIGURE 3   Connection With CEE-DCBX FCoE Switch by Using CNAs

Using a 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 4   Connection With Non CEE-DCBX FCoE Switch

▼ How to Configure the 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 dcbx-version 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 an FCoE Switch" on page 81.

For more information about LLDP, see Chapter 7, "Exchanging Network Connectivity Information With Link Layer Discovery Protocol" in *Managing Network Datalinks in Oracle Solaris 11.4*.

Verifying and Configuring an FCoE Switch

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 that 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. It 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 can be set in the ixgbe.conf file.

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

TABLE 1   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
`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 IPFC SAN Devices

This section explains how to configure Internet Protocol over Fibre Channel (IPFC) in Oracle Solaris. It provides the considerations while using IPFC and describes how to identify and configure IPFC instances. The IPFC driver is based on RFC 2625 and allows IP traffic to run over FC.

IPFC Considerations

Note the following considerations for IPFC:

- Promiscuous mode is not supported. The `snoop` utility cannot be used.
- IP multicast is not supported (RFC 2625 does not define IP multicast support).
- IPFC network interfaces cannot be used to forward IP datagrams. Oracle Solaris cannot function as an IP router with an IPFC interface.
- IPFC interfaces cannot be administered using the `dladm` command.
- IPFC interfaces do not use vanity names.

Determining Fibre Channel Adapter Port Instances

Each IPFC instance communicates through a specific fibre channel adapter port. To configure an IPFC instance, it is first necessary to identify the FC HBA port instance (`fp` driver instance).

You can determine the fibre channel adapter port instance based on any of the following:

- WWN of a remote device or IPFC instance visible by the HBA port. For more information, see "Determining the HBA Port Instance Number From a Physical Device Path" on page 86.
- WWN of the HBA port. For more information, see "Determining the HBA Port WWN From a Remote WWN" on page 84.
- Controller number of the HBA port. For more information, see "Determining the Controller Number from an HBA Port WWN" on page 85.
- Oracle Solaris physical device path of the HBA port. For more information, see "Determining the Physical Device Path From the Controller Number" on page 85.

Tip - You can determine the physical path using the controller number. The controller number can be determined from the WWN of the HBA port, and the WWN of the HBA port can be determined using WWN of a remote device or IPFC instance visible by the HBA port.

Determining the HBA Port WWN From a Remote WWN

If you know the WWN of a remote FC device or IPFC instance, you can determine the local HBA port WWN using the `fcinfo` command. In the following example, the WWN of the remote IPFC instance is 2100001b328512fb:

```
# fcinfo remote-port 2100001b328512fb
HBA Port WWN: 10000000c9722de0
  Count of Remote Port(s): 8
    Remote Port WWN: 2100001b328512fb
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port ID: 10300
        Port Symbolic Name: qlc(0,0)
        Node WWN: 2000001b328512fb
    Remote Port WWN: 2101001b32a512fb
```

You can see that the remote port 2100001b328512fb is accessible through the HBA port with WWN 10000000c9722de0.

Determining the Controller Number from an HBA Port WWN

From the WWN of the HBA port, you can determine the controller number or the controller path. In this example, the HBA port WWN is 10000000c9722de0 (as in the previous section):

```
# fcinfo hba-port 10000000c9722de0
HBA Port WWN: 10000000c9722de0
        Port Mode: Initiator
        Port ID: 10100
        OS Device Name: /dev/cfg/c4
        Manufacturer: Emulex
        Model: LPe11002-S
        Firmware Version: LPe11002-S 2.82a4
        FCode/BIOS Version: Boot:5.02a1
        Serial Number: 0999BG0-07500002OH
        Driver Name: emlxs
        Driver Version: 3.1.01.0 (2016.05.23.12.00)
        Type: N-port
        State: online
        Supported Speeds: 1Gb 2Gb 4Gb
        Current Speed: 4Gb
        Node WWN: 20000000c9722de0
        NPIV Not Supported
-bash-4.3#
```

The line labelled `OS Device Name` indicates that the controller number is 4 (the controller device path is `/dev/cfg/c4`).

Determining the Physical Device Path From the Controller Number

From the controller number or the controller device path, you can determine the physical device path. In this example the controller number is 4 (the controller device path is `/dev/cfg/c4`).

```
# ls -l /dev/cfg/c4
lrwxrwxrwx   1 root     root          60 Sep 26  2014 /dev/cfg/c4 -> ../../devices/
pci@0,0/pci10de,377@f/pci10df,fc22@0/fp@0,0:fc
```

Removing `../../devices` and `:fc` from the output, gives you the physical device path.

```
/pci@0,0/pci10de,377@f/pci10df,fc22@0/fp@0,0
```

Determining the HBA Port Instance Number From a Physical Device Path

It is possible to determine the correspondence between HBA port WWNs and `fp` instance numbers using the following command:

```
# for dev in $(devprop -a initiator-port | grep fp@); do devprop -n $dev initiator-port;
 prtconf $dev; echo; done
10000000c9722de0
fp, instance #2

10000000c9722ddf
fp, instance #0
```

Here, HBA port WWN 10000000c9722de0 corresponds to `fp` instance 2 and HBA port WWN 10000000c9722ddf corresponds to `fp` instance 0.

You can determine the HBA port instance number from the physical device path. In this example, the physical device path is `/pci@0,0/pci10de,377@f/pci10df,fc22@0/fp@0,0`.

```
# grep '"/pci@0,0/pci10de,377@f/pci10df,fc22@0/fp@0,0"' /etc/path_to_inst
"/pci@0,0/pci10de,377@f/pci10df,fc22@0/fp@0,0" 2 "fp"
```

The HBA port instance number is 2.

Configuring IPFC Instances

Any standard network commands can be used after IPFC is attached. There are no usage differences when commands such as `ping` or `ftp` are used in an Ethernet setup. This section describes how to configure and use the IPFC instances.

▼ How to Configure an IPFC Instance

Use this procedure to configure an IPFC interface using the instance number N determined in the previous section. The configuration operation takes immediate effect and persists across reboots. This procedure assumes that the instance is 2.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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 2 and an IP address of `192.0.2.1`.

```
# ipadm create-ip fcip2
# ipadm create-addr -a local=192.0.2.1/24 fcip2/example
```

For more information, see the ipadm(8) man page.

3. Verify if the configuration operation was successful.

Use the `ipadm` command to verify the configuration operation succeeded.

```
# ipadm
NAME            CLASS/TYPE STATE       UNDER     ADDR
bge0            ip         ok          --        --
   bge0/a       static     ok          --        192.168.100.60/24
bge1            ip         down        --        --
fcip2           ip         ok          --        --
   fcip2/example static    ok          --        192.168.2.1/24
lo0             loopback   ok          --        --
   lo0/v4       static     ok          --        172.16.0.1/8
   lo0/v6       static     ok          --        ::1/128
```

For more information, see *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

♦ ♦ ♦ **C H A P T E R   5**

5

Configuring an Oracle Solaris iSCSI Initiator

This chapter describes how to configure Oracle Solaris iSCSI initiator in the Oracle Solaris OS. For information about the procedures associated with configuring iSCSI initiator, see "Configuring an iSCSI Initiator" on page 91.

This chapter contains the following topics:

- "About iSCSI in Oracle Solaris" on page 89
- "Configuring an iSCSI Initiator" on page 91
- "Configuring Authentication in an iSCSI-Based Storage Network" on page 105
- "Tuning iSCSI Initiator and Target Parameters" on page 108
- "Troubleshooting iSCSI Configuration Problems" on page 113

For information about configuring Oracle Solaris iSCSI targets with COMSTAR, see Chapter 8, "Configuring Storage Devices With COMSTAR" in *Managing Devices in Oracle Solaris 11.4*.

For more information about installing and booting an iSCSI disk, *Automatically Installing Oracle Solaris 11.4 Systems*.

About iSCSI in Oracle Solaris

iSCSI protocol enables you to access storage devices as if they are connected to a local system. An Oracle Solaris system can act as either an iSCSI client (initiator) or a server (target).

TCP connections or iSCSI Extensions for RDMA (iSER) can be used to transfer iSCSI data in Oracle Solaris. iSER accelerates the iSCSI protocol by using Remote Direct Memory Access (RDMA) protocol to transfer data directly between the iSCSI nodes without intermediate data copies.

In Oracle Solaris, I/O multipathing is enabled by default and must not be disabled for iSCSI. When multiple iSCSI sessions are configured per target (MS/T), I/O multipathing combines these sessions to a single iSCSI device.

Benefits of iSCSI Devices

Using iSCSI initiator 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.
  - You can use the existing infrastructure and management tools for IP networks.
- iSCSI can utilize InfiniBand (IB) infrastructure to access storage devices. You can use iSCSI on an existing IB infrastructure.
- You can install the Oracle Solaris OS on an iSCSI disk. This iSCSI disk can also be used as a root file system to boot the OS. For more information about installing the OS on an iSCSI disk, see *Automatically Installing Oracle Solaris 11.4 Systems*.

Limitations of iSCSI Devices

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

- iSCSI targets cannot be configured as dump devices.
- Sharing iSCSI traffic with other traffic can affect performance.
- 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. This limitation does not apply to kernel zones.

iSCSI Software and Hardware Requirements

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

- Oracle Solaris storage software and devices.

  This requirement includes the `system/storage/iscsi/iscsi-initiator` software package for the iSCSI initiator management utilities.
- If you are using TCP connection, use any supported NIC.
- If you are using iSER, configure IB. For more information, see Chapter 3, "Using InfiniBand Devices" in *Managing Devices in Oracle Solaris 11.4*.

Best Practices for Configuring iSCSI

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

- Deploy iSCSI devices in a 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 per Target" on page 96.

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 "IPMP Support in Oracle Solaris" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.
- 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 93.
- Use CHAP authentication for additional security. For more information about configuring CHAP authentication, see "Configuring Authentication in an iSCSI-Based Storage Network" on page 105.

Configuring an iSCSI Initiator

This section describes the tasks associated with configuring an iSCSI initiator.

- "Target Discovery Methods" on page 92
- "Enabling Multiple iSCSI Sessions for a Single Target" on page 96
- "Displaying iSCSI Configuration" on page 99
- "Accessing iSCSI Disks" on page 103
- "Removing Discovered Target Devices" on page 104

You can also find related information in the following documents:

- *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*

■ Chapter 8, "Configuring Storage Devices With COMSTAR" in *Managing Devices in Oracle Solaris 11.4*

Target Discovery Methods

The three target discovery methods available in Oracle Solaris are as follows:

■ `sendtargets` discovery method
■ Internet Storage Name Service (iSNS) discovery method
■ Static discovery method

Note - Do not configure an iSCSI target to be discovered by more than one discovery method.

The following sections describe how to configure these device discovery methods.

Administering `sendtargets` Discovery Method

`sendtargets` is one of the dynamic target discovery methods. This is a simple discovery method to configure an iSCSI initiator to connect to all the targets on a particular storage device.

To enable an iSCSI initiator to use the `sendtargets` discovery method, you must specify the IP address and port number of the iSCSI target node. The target node sends the list of all available targets and portals to the initiator. After receiving the target and portal information, the initiator tries to establish sessions with all of those targets through all the portals.

If the initiator does not have access to all the target portals or if you do not want the initiator to establish sessions with all available targets then, use Static configuration or iSNS discovery method.

▼ **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 11.4*.

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

   ```
   initiator# svcs network/iscsi/initiator
   STATE          STIME    FMRI
   online         17:33:28 svc:/network/iscsi/initiator:default
   ```

3. **Obtain the target IP address and port number from the iSCSI storage device management interface.**

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

4. **To configure the `sendtargets` discovery method, provide the discovery address:**

   ```
   initiator# iscsiadm add discovery-address 192.0.2.10
   ```

   If no port number is provided, the default port 3260 is used.

5. **Review the `sendtargets` configuration information.**

   ```
   initiator# iscsiadm list target
   Target: iqn.1986-03.com.sun:02:02380f80-ab3c-47a9-8882-d3cafc2c49ae
           Alias: -
           ISID: 4000002a0000
           Connections: 0
   Target: iqn.1986-03.com.sun:02:3b8eb56d-051a-43ef-b56e-d2b81c94959b
           Alias: -
           ISID: 4000002a0000
           Connections: 0
   ```

   You can also use the `iscsiadm list target -v` command for information about the targets.

6. **Enable the `sendtargets` discovery method.**

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

   To list the discovered targets see, . For more information about the `iscsiadm` command, see the `iscsiadm(8)` man page.

Administering Internet Storage Name Service Target Discovery Method

The Internet Storage Name Service (iSNS) enables an iSCSI initiator to discover the targets to which it has access, by using as little configuration information as possible. It also provides state change notifications to notify an iSCSI initiator when there are changes in the operational state of storage nodes. To use the iSNS discovery method, you must provide the iSNS server address and port number combination. This enables the iSCSI initiator to query the iSNS server that you specified to perform the device discovery. The default port for the iSNS server is 3205.

The iSNS discovery method is suitable for large iSCSI setup with multiple iSCSI initiator hosts or multiple iSCSI target nodes or when state change notifications are required. iSNS provides similar functionality as FC fabric services.

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 11.4*.

▼ 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 11.4*.

2. Verify that the iSCSI initiator service is online.

```
initiator# svcs network/iscsi/initiator
STATE          STIME    FMRI
online         17:33:28 svc:/network/iscsi/initiator:default
```

3. Obtain the target IP address and port number from the iSNS server management interface.
If you are using the Oracle iSNS server, see Chapter 9, "Configuring and Managing the Oracle Solaris Internet Storage Name Service (iSNS)" in *Managing Devices in Oracle Solaris 11.4*.
If you are using a third-party iSNS server, see the specific vendor documentation.

4. Configure the iSNS discovery method by providing the IP address of the iSNS server.

```
initiator# iscsiadm add isns-server 198.51.100.2
```

5. Review the iSNS configuration information.

```
initiator# iscsiadm list isns-server -v
iSNS Server IP Address: 203.0.113.66/27:3205
```

6. Enable the iSNS discovery method.

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

To list the discovered targets, see "Displaying iSCSI Configuration" on page 99.

Administering Static Configuration Discovery Method

You can use the static configuration discovery method in the followings scenarios:

- When only a few iSCSI targets must be discovered.

- To restrict the initiator to access only a specific number of targets presented by the `sendtargets` discovery.
- When the initiator must not access all the target portals of the storage device.

You can determine the static target address from the management tool of the array. For more information, see the `iscsiadm(8)` man page.

Note - Do not configure an iSCSI target to be discovered by both static and dynamic device discovery methods. Usage of redundant discovery methods reduces the performance, when an initiator is communicating with iSCSI target devices.

▼ How to Configure an iSCSI Initiator With Static Configuration

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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

   ```
   initiator# svcs network/iscsi/initiator
   STATE          STIME    FMRI
   online         17:33:28 svc:/network/iscsi/initiator:default
   ```

3. **Obtain the target IP address and iSCSI 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 target to be statically discovered.**

   ```
   initiator# iscsiadm add static-config target-name,target-address[:port-number][,tpgt]
   ```

   For example:

   ```
   initiator# iscsiadm add static-config iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-
   c3b36c123456, 192.0.2.2:3260
   ```

   For more information, see the `iscsiadm(8)` man page.

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,
   ```

```
192.0.2.2:3260
```

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

6. **Enable the static configuration discovery method.**

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

    To list the discovered targets, see .

Enabling Multiple iSCSI Sessions for a Single Target

In Oracle Solaris, the multiple sessions per target (MS/T) strategy provides better performance than multiple connections per session. Using MS/T and creating multiple TCP connections ensures better usage of the networking stack. This feature also ensures better performance by using multiple send and receive threads.

You can use the following 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. You can achieve higher bandwidth by using multiple NICs on the host side to connect to multiple portals on the same target. When multiple iSCSI sessions are configured per target (MS/T), I/O multipathing combines these sessions to a single iSCSI device.

If there are multiple target portal groups configured on a target device, the configured number of sessions are created to each of these target portal groups. `sendtarget` discovery automatically detects the existence of multiple target portal group tag (TPGTs) and creates multiple target sessions.

▼ How to Enable Multiple iSCSI Sessions per Target

Before You Begin   Confirm that the multipathing 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";
```

Also, confirm that multiple network connections are available and are configured by using IPMP.

```
# ipadm show-addr
```

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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

   - List the current parameter values of 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
     ```

   - List the current parameter values of the iSCSI target.

     For example:

     ```
     initiator# iscsiadm list target-param iqn.1986-03.com.sun:02:02380f80-ab3c-47a9-8882-
     d3cafc2c49ae
     Target: iqn.1986-03.com.sun:02:02380f80-ab3c-47a9-8882-d3cafc2c49ae
             Alias: -
             .
             .
             .
             Configured Sessions: 1
     ```

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

3. **Modify the number of configured sessions. You can modify the configured session by using either of the following methods.**

   - Configure the initiator node to apply to all targets (globally) or apply to a specific target by providing the number of the configured session.

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

       ```
       initiator# iscsiadm modify initiator-node -c Number-of-configured session
       ```

For example:

```
initiator# iscsiadm modify initiator-node -c 4
```

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

```
initiator# iscsiadm modify target-param -c Number-of-configured session  target-name
```

For example:

```
initiator# iscsiadm modify target-param -c 4  iqn.1992-08.com.abcstorage:sn.
84186266
```

■ Configure the initiator node to apply to all targets (globally) or apply to a specific target by providing one or more local IP addresses.

In this method, you must supply one or more local IP addresses in a comma-separated list. Each IP address represents an iSCSI session. TCP connections associated with the session will bind to the particular local IP address. Binding multiple local addresses is a preferred configuration when used with IPMP.

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

```
initiator# iscsiadm modify initiator-node -c IP address1, IP address2
```

For example:

```
initiator# iscsiadm modify initiator-node -c 192.0.2.1,198.51.100.1
```

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

```
initiator# iscsiadm modify target-param -c IP address1, IP address2  target-name
```

For example:

```
initiator# iscsiadm modify target-param -c 192.0.2.1,198.51.100.1 iqn.1992-08.com.
abcstorage:sn.84186266
```

For information about routing when using multiple local addresses, see "Maintaining IP Connectivity and Routing While Deploying IPMP" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

4. Verify if the parameter is modified.

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: 4
```

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: 4
```

Creating multiple sessions present multiple paths to Oracle Solaris I/O multipathing. To list the number of paths for each LU, use the following command.

```
initiator# mpathadm list lu
```

Displaying iSCSI Configuration

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

The following example shows how to display information about which discovery method is used.

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

The following examples show how to display discovered targets.

The following example shows how to display information about an iSCSI initiator by using the iscsiadm list initiator-node command.

```
# 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 the parameter values for a specific iSCSI target.

```
# iscsiadm list target
Target: iqn.1986-03.com.sun:02:02380f80-ab3c-47a9-8882-d3cafc2c49ae
        Alias: -
        ISID: 4000002a0000
        Connections: 0
Target: iqn.1986-03.com.sun:02:3b8eb56d-051a-43ef-b56e-d2b81c94959b
        Alias: -
        ISID: 4000002a0000
        Connections: 0
```

```
# iscsiadm list target-param iqn.1986-03.com.sun:02:02380f80-ab3c-47a9-8882-d3cafc2c49ae
```

```
Target: iqn.1986-03.com.sun:02:02380f80-ab3c-47a9-8882-d3cafc2c49ae
Alias: -
```

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 (-). The following example shows the parameters that were negotiated between the target and the initiator.

```
# iscsiadm list target-param -v
Target: iqn.1986-03.com.sun:02:02380f80-ab3c-47a9-8882-d3cafc2c49ae
        Alias: -
        Bi-directional Authentication: disabled
        Authentication Type: NONE
        Login Parameters (Default/Configured):
                Data Sequence In Order: yes/-
                Data PDU In Order: yes/-
                Default Time To Retain: 20/-
                Default Time To Wait: 2/-
```

```
                    Error Recovery Level: 0/-
                    First Burst Length: 65536/-
                    Immediate Data: yes/-
                    Initial Ready To Transfer (R2T): yes/-
                    Max Burst Length: 262144/-
                    Max Outstanding R2T: 1/-
                    Max Receive Data Segment Length: 8192/-
                    Max Connections: 65535/-
                    Header Digest: NONE/-
                    Data Digest: NONE/-
            Tunable Parameters (Default/Configured):
                    Session Login Response Time: 60/-
                    Maximum Connection Retry Time: 180/-
                    Login Retry Time Interval: 60/-
            Configured Sessions: 1

Target: iqn.1986-03.com.sun:02:3b8eb56d-051a-43ef-b56e-d2b81c94959b
        Alias: -
        Bi-directional Authentication: disabled
        Authentication Type: NONE
        Login Parameters (Default/Configured):
                    Data Sequence In Order: yes/-
                    Data PDU In Order: yes/-
                    Default Time To Retain: 20/-
                    Default Time To Wait: 2/-
                    Error Recovery Level: 0/-
                    First Burst Length: 65536/-
                    Immediate Data: yes/-
                    Initial Ready To Transfer (R2T): yes/-
                    Max Burst Length: 262144/-
                    Max Outstanding R2T: 1/-
                    Max Receive Data Segment Length: 8192/-
                    Max Connections: 65535/-
                    Header Digest: NONE/-
                    Data Digest: NONE/-
        Tunable Parameters (Default/Configured):
                    Session Login Response Time: 60/-
                    Maximum Connection Retry Time: 180/-
                    Login Retry Time Interval: 60/-
            Configured Sessions: 1
```

The iscsiadm list target-param -v *target-name* command provides information about the parameters of a specific target.

For example:

```
# 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 `iscsiadm list target -v` *target-name* command provides detailed information about a target.

```
# iscsiadm list target -v eui.50060e8004275511
Target: eui.50060e8004275511
        TPGT: 1
        ISID: 4000002a0000
        Connections: 1
                CID: 0
                  IP address (Local): 198.51.100.71:32813
                  IP address (Peer): 198.er51.100.40:3260
                  Transport Type: socket
                  Discovery Method: Static
                  Login Paramets (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
```

Accessing iSCSI Disks

You can create a ZFS 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/c0t600144F0D1720B000000540094860001d0s2
```

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

```
initiator# format
       0. c0t600C0FF0000000000011C51EDAA44600d0 <SUN-StorEdge 3510-415F-272.96GB>
          /scsi_vhci/ssd@g600c0ff0000000000011c51edaa44600
       1. c0t5000CCA00ABB24F0d0 <HITACHI-H103030SCSUN300G-A2A8-279.40GB>  OSOL11
          /scsi_vhci/disk@g5000cca00abb24f0
       2. c0t5000CCA00ABBAA9Cd0 <HITACHI-H103030SCSUN300G-A2A8-279.40GB>  OSOL10u11
          /scsi_vhci/disk@g5000cca00abbaa9c
       3. c0t5000CCA00ABBAAD0d0 <HITACHI-H103030SCSUN300G-A2A8-279.40GB>  OSOL11u4
          /scsi_vhci/disk@g5000cca00abbaad0
       4. c0t5000CCA00ABBABDCd0 <HITACHI-H103030SCSUN300G-A2A8-279.40GB>
          /scsi_vhci/disk@g5000cca00abbabdc
       5. c0t5000CCA00ABBAC6Cd0 <HITACHI-H103030SCSUN300G-A2A8-279.40GB>
          /scsi_vhci/disk@g5000cca00abbac6c
Specify disk (enter its number): 0
selecting c0t600144F0B5418B0000004DDAC7C10001d0
[disk formatted]
```

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

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

```
# zpool create iscsi-pool c0t600144F0B5418B0000004DDAC7C10001d0
# zfs create iscsi-pool/data-fs
```

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 11.4*.

Removing Discovered Target Devices

The initiator is removed from the associated targets after you perform any one of the following actions:

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

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

To remove any discovered targets, you must be an administrator and must be logged in to a local system, where you can access the configured target devices.

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

- To remove an iSCSI SendTargets discovery entry:

  initiator# **iscsiadm remove discovery-address IP-address:3260**
- To remove an iSCSI iSNS discovery entry:

  initiator# **iscsiadm remove isns-server server-address:3205**
- To remove a static iSCSI discovery entry:

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

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
```

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/Os on the LU and 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 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, the 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 or the initiator to authenticate the identity of the target.
- *Bidirectional* authentication adds a second level of security by adding authentication on both directions.

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.

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 107.

▼ **How to Configure CHAP Authentication for an iSCSI Initiator**

This procedure shows how to configure a CHAP identity in iSCSI. To enable the target to authenticate the initiator, you must set the CHAP identity on the initiator.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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 the initiator node to have 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 want to set up bidirectional CHAP authentication.

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

   Enable CHAP authentication for a specific iSCSI target by configuring iSCSI initiator to perform CHAP authentication of an iSCSI 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
```

For more information about how to configure CHAP authentication for an iSCSI target, see "How to Configure CHAP Authentication for an iSCSI Target" in *Managing Devices in Oracle Solaris 11.4*.

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 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 11.4*.

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, node IP address)
   - The shared secret key that the node uses to communicate with the RADIUS server
   - The CHAP name of a target (for example, `iqn` name of a target) and the secret key for each target that needs to be authenticated

   For more information about how to configure a RADUIS server on an iSCSI target, see "How to Configure a RADIUS Server for an iSCSI Target" in *Managing Devices in Oracle Solaris 11.4*.

Tuning iSCSI Initiator and Target Parameters

You can modify parameters for both an iSCSI initiator node (globally) and for a iSCSI target device. The following parameters can be modified for an iSCSI initiator:

- iSCSI initiator node name - If you change an initiator node name, the targets that were discovered by iSNS might be removed from the initiator's target list. The removal of the targets from the target list of the initiator depends on the discovery domain configuration on the iSNS server at the time the name was changed.
- 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 106.

The iSCSI driver provide default values for an iSCSI initiator and iSCSI target device parameters. If you modify the parameters of an iSCSI initiator, the modified parameters are inherited by the iSCSI target device. If the iSCSI target device already has a different value, it does not inherit the iSCSI initiator parameter values. If you change the initiator node name, the targets that are discovered by iSNS log out and are removed from the target list of the initiator. Also, if the new name does not belong to the same discovery domain as that of the targets, the targets are logged out and removed from the target list of the initiator. But 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 are not removed.

> ⚠️ **Caution -** Ensure that the target software supports the parameter to be modified. Otherwise, you might not be able to log in to the iSCSI target device. For more information about the list of supported parameters, see vendor array documentation.

You can modify the iSCSI parameters only when there is no active I/O communication between the initiator and the target. After the parameter is modified by using the `iscsiadm modify` command, the iSCSI driver reconnects the session.

For more information about all the parameters, see the `iscsiadm(8)` man page.

▼ How to Modify an iSCSI Initiator Node and Target Parameters

This procedure describes how modified parameters of the iSCSI initiator are inherited by the iSCSI target device and how to modify parameters on the iSCSI target device.

This following 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 11.4*.

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.

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 (-).

3. Modify the parameter of the iSCSI initiator.

For example, set the header digest to CRC32.

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

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(8) man page.

Tuning Other iSCSI Parameters

You can modify parameters to adjust various response or connection time values of the iSCSI initiator either for all targets the initiator system is connected to or 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

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 a specific target:

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

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 2   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 the 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 the target I/O times out or the connection fails.	0 – 3600	60

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 tool is a GUI tool for dumping and analyzing network traffic. For more information, see https://www.wireshark.org/.
- tshark – This is a CLI tool for dumping and analyzing network traffic.

For example, tracing iSCSI packets on an iSCSI port, which is 3260 by default.

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

Troubleshooting Oracle Solaris iSCSI and RADIUS Server Issues

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" in *Managing Devices in Oracle Solaris 11.4*.

```
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" in *Managing Devices in Oracle Solaris 11.4*.

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 iSCSI port of the storage device by using the telnet command to ensure that the iSCSI service is available. The default port is 3260.

  In addition, check the log file of the storage device 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: 203.0.113.56/27:3205
        Target name: iqn.1992-08.com.xyz:sn.1234566
                Target address:   203.0.113.35/27:3260, 1
        Target name: iqn.2003-10.com.abc:group-0:154:abc-65-01
                Target address:   203.0.113.66/27:3260, 1
        Target name: iqn.2003-10.com.abc:group-0:154:abc-65-02
                Target address:   203.0.113.66/27:3260, 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 the 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 the target during enumeration.

- Review the /var/adm/messages file to see if an error is reported. If you think a LUN should be listed but it is not, then check the /var/adm/messages log file.
- Check the log files of the storage device 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#/   These values are returned in an iSCSI login response as defined by
STATUS-DETAIL#   Internet Small Computer Systems Interface (iSCSI), 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 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) 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) 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 non-default 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(8) 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.

◆ ◆ ◆   **C H A P T E R   6**

6

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 does 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, SR-IOV SAS controllers.

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

It contains the following topics:

- "Dynamic Discovery of SAS Devices" on page 123
- "Displaying SAS Configuration Information" on page 124

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` and `format` commands. 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(8)` and `format(8)` man pages.

Displaying SAS Configuration Information

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(8)` man page.

You can display the configuration information about the SAS devices by using the `sasinfo` command.

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
```

7

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 contains the following topics:

-
-

About Persistent Binding

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(8) man page.

For any tape drive visible to Oracle Solaris 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 11.4*.

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

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

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

   For more information about creating entries in the `devlink.tab` file, see "Creating Tape Links" on page 128.

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(8)` man page.

♦♦♦ **A P P E N D I X   A**

Manual Configuration for Fibre Channel Devices

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

This appendix covers the following topics:

- "Checking the Health of FC Devices" on page 131
- "Manually Configuring FC Devices" on page 133
- "Configuring Fabric Device Nodes" on page 134
- "Configuring Device Nodes Without Multipathing Enabled" on page 136
- "Configuring Device Nodes With Multipathing Enabled" on page 139
- "Unconfiguring Fabric Devices" on page 143

Checking the Health of FC Devices

You can use the following commands to perform health checks on the devices in a SAN:

- Use the `fmadm faulty` command, which provides status information for resources that the Fault Manager currently associated with a fault, defect, or alert. If the `fmadm faulty` command does not throw any output, it means there are no faulty devices. You can also use the `fmadm` command to know the following configuration information:

  - View the set of diagnosis engines and agents that are currently participating in fault management.
  - View the list of system components that have been diagnosed as associated with a fault, defect, or alert.

  For more information about resolving faulty devices, see "Resolving Faulty Devices" in *Managing Devices in Oracle Solaris 11.4*.

  For more information, see the `fmadm(8)` man page.

- Use the zpool status *zpool-name* command, which displays the detailed health status about the specified pool. If no pool is specified, then the status of each pool in the system is displayed. For example:

```
# zpool status test_pool
  pool: test_pool
 state: ONLINE
  scan: none requested
config:

        NAME                       STATE     READ WRITE CKSUM
        test_pool                  ONLINE       0     0     0
          mirror-0                 ONLINE       0     0     0
            c0t5000CCA00ABBAA9Cd0  ONLINE       0     0     0
            c0t5000CCA00ABCE8DCd0  ONLINE       0     0     0

errors: No known data errors
```

- Use the mpathadm show lu *logical-unit* command, which provides the configuration information about the specified LU.

```
# mpathadm show lu /dev/rdsk/c0t5000CCA00ABBAA9Cd0
Logical Unit:  /dev/rdsk/c0t5000CCA00ABBAA9Cd0s2
        mpath-support:  libmpscsi_vhci.so
        Vendor:  HITACHI
        Product:  H103030SCSUN300G
        Revision:  A2A8
        Name Type:  unknown type
        Name:  5000cca00abbaa9c
        Asymmetric:  no
        Current Load Balance:  shortest-path
        Logical Unit Group ID:  NA
        Auto Failback:  on
        Auto Probing:  NA

        Paths:
                Initiator Port Name:  w5080020000d67ae9
                Target Port Name:  w5000cca00abbaa9d
                Override Path:  NA
                Path State:  OK
                Disabled:  no

        Target Ports:
```

```
        Name:  w5000cca00abbaa9d
        Relative ID:  0
```

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 an FC Device

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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 the `cfgadm_fp(8)` and `fp(4D)` man pages.

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 Device Nodes Without Multipathing Enabled" on page 136
   - "Configuring Device Nodes With Multipathing Enabled" on page 139

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

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

    ```
    # manual_configuration_only=1;
    ```

7. **Reboot the system.**

    ```
    # init 6
    ```

Configuring Fabric Device Nodes

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

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

▼ How to Ensure That 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 11.4*.

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 an Oracle Solaris System

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

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

---

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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

   ```
   # cfgadm -l
   Ap_Id              Type        Receptacle   Occupant     Condition
   c0              fc-fabric    connected    unconfigured unknown
   c1              fc-private   connected    configured   unknown
   ```

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

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

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

   ```
   # cfgadm -al
   Ap_Id                 Type        Receptacle    Occupant    Condition
   c0                 fc-fabric    connected    unconfigured unknown
   c0::50020f2300006077 disk        connected    unconfigured unknown
   ```

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

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

Configuring Device Nodes Without Multipathing Enabled

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

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

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

▼ How to Manually Configure an FC Device Without Multipathing

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

1. **Become an administrator.**

    For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2.   Identify the device to be configured.

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

3.   Configure the fabric device.

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

4.   Verify that the selected fabric device is configured.

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

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

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

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

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

```
c0::50020f2300006077,2 disk      connected    configured   unknown
c0::50020f2300006077,3 disk      connected    configured   unknown
```

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

▼ How to Configure Multiple FC Devices Without Multipathing

Make sure you first identify the devices visible to the system with the procedure "How to Ensure That LUN Level Information Is Visible" on page 134. 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 11.4*.

2. Identify the devices to be configured.

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

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

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

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

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

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

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

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

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

Configuring Device Nodes With 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(8)`. The

procedures in this section illustrate steps to detect fabric devices that are visible on a system and to configure them as multipathing devices to make them available to the system.

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

▼ How to Configure Individual Multipathed FC Devices

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

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

1.   Become an administrator.

For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2.   Identify the PWWN 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 following two multipathed devices are created for the device represented by `c0::50020f2300006077` and `c2::50020f2300006107:`.

▼ How to Configure Multiple Multipathed FC Devices

Before you configure or remove device nodes, be sure to first identify the fabric devices by using the procedures in .

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

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

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Identify the fabric-connected host port to be configured.**

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

   Devices represented by `Ap_Ids` `c0::50020f2300006077` and `c2::50020f2300006107` are two paths to the same physical device, with `c0::50020f2300006077` already configured. Configure the unconfigured devices on the selected port. This operation repeats the `configure` command of an individual device for all the devices on `c2`. This can be time-consuming if the number of devices on `c2` is large.

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

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

   ```
   # cfgadm -al
   Ap_Id                 Type        Receptacle  Occupant      Condition
   c0                    fc-fabric   connected   configured    unknown
   c0::50020f2300006077 disk         connected   configured    unknown
   ```

```
c0::50020f23000063a9 disk        connected   configured   unknown
c1                    fc-private connected   configured   unknown
c1::220203708b69c32b disk        connected   configured   unknown
c1::220203708ba7d832 disk        connected   configured   unknown
c1::220203708b8d45f2 disk        connected   configured   unknown
c1::220203708b9b20b2 disk        connected   configured   unknown
c2                    fc-fabric  connected   configured   unknown
c2::50020f2300005f24 disk        connected   configured   unknown
c2::50020f2300006107 disk        connected   configured   unknown
```

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

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

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

Unconfiguring Fabric Devices

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

▼ How to Manually Unconfigure an FC Device

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

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

1. **Become an administrator.**

For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Identify the device to be unconfigured.**

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

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

3. **Unconfigure the fabric device.**

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

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

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

▼ How to Unconfigure All FC Devices on an FC Host Port

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

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

1. **Become an administrator.**

For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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

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

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

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

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

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

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

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

5. **Verify that all the devices on `c0` are unconfigured.**

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

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

B

Troubleshooting Multipathing Problems

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

This appendix covers the following topics:

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

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

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

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

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

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

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

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

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

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

   b. **Update the boot archive.**

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

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

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

9. **Unmount the BE and the USF root file system.**

```
# beadm umount solaris
# umount /a
```

10. **Reboot the system.**

C

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 use the following prefix categories:

- `BCD` – Binary-Coded-Decimal as shown in Table 3, "Supported Tunables and Their Flag Values on Different Platforms," on page 156
- `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 13, "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 I/O multipathing is enabled on the device. You can use the `prtconf` command to check the device configuration information. See the `prtconf(8)` man page.

EXAMPLE 13   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 14   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

Note - The bit mask values in the bit-masking format can represent different tunable parameters among the SPARC sd(4D) drivers, the solid-state SPARC ssd(4D) drivers, and the X86 sd(4D) drivers. To prevent misconfiguration when tuning parameters in the sd.conf and ssd. conf files, use the JSON-text format, as described in "*Name:Value* Pair Format to Tune Disk Drivers" on page 152.

The bit-masking format, also known as the Version1 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 156 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`
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   15   Configuring the Tunable Parameters for an `sd` Driver on an x86 Platform

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

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

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 16 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*.

D

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 https://www.oracle.com/servers/network-adapters/

Index