Managing Devices in Oracle® Solaris 11.2

Part No: E36834
July 2014

ORACLE

Contents

Using This Documentation

- **Overview** – Describes how to manage removable media, disks, and devices.
- **Audience** – System administrators.
- **Required knowledge** – Basic Oracle Solaris or UNIX system administration experience.

Product Documentation Library

Late-breaking information and known issues for this product are included in the documentation library at http://www.oracle.com/pls/topic/lookup?ctx=solaris11.

Access to Oracle Support

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

Feedback

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

♦ ♦ ♦ CHAPTER 1

Managing Devices in Oracle Solaris

This chapter provides overview information about device management in Oracle Solaris. Oracle Solaris supports various peripheral devices such as disks, DVD drives, and tape devices.

This chapter covers the following topics:

What's New in Device Management?

This release includes the Oracle Hardware Management Pack features. Previously, this package was available as a separate download. These features provide cross platform components to better manage your hardware such as displaying hardware configuration and status information. These features are included in the `system/management` package with names similar to the following:

```
system/management/fwupdate
system/management/ipmitool
system/management/raidconfig
system/management/ubiosconfig
```

For more information about these features, go to http://www.oracle.com/technetwork/documentation/sys-mgmt-networking-190072.html#hwmgmt.

For a complete listing of new Oracle Solaris features and a description of Oracle Solaris releases, see "Oracle Solaris 11.2 Release Notes".

```
This is a test of the msgtest.
```

Description: Looks like the msgset element is not being formatted correctly in html.

Cause: We did not know about this.

Solution: Let tools know.

Example: We need examples.

About Device Management

Device management in the Oracle Solaris typically involves tasks such as the following:

- Adding and removing peripheral devices from systems
- Possibly adding a third-party device driver to support a device
- Displaying system configuration information.

Note - If you are using an x86 based system, use the device detection tool to determine if the hardware is supported in this Oracle Solaris release. For more information, go to http://www.oracle.com/webfolder/technetwork/hcl/devicelist/index.html.

About Device Drivers

A computer uses a wide range of peripheral devices and mass-storage devices such as drives, printers, and so on. The Oracle Solaris software does not directly communicate with all of these devices. Each type of device requires different data formats, protocols, and transmission rates.

A *device driver* is a low-level program that enables the operating system to communicate with a specific piece of hardware. The driver serves as the operating system's “interpreter” for that piece of hardware.

You can customize a driver configuration by adding or modifying either a per-device parameter or a global property in that driver's configuration file. The addition or modification impacts all devices in the system. In the Oracle Solaris 11 release, driver customizations are made in the `/etc/driver/drv` directory rather than in the `/kernel` directory as in previous releases. Files in the `/etc/driver/drv` directory are preserved during the upgrade. Thus, your driver customizations are not overwritten when the system is upgraded.

▼ How to Customize a Driver Configuration

1. **Become an administrator.**
2. **Copy the original vendor supplied *driver.conf* file to the `/etc/driver/drv` directory. For example:**

```
# cp /kernel/drv/sd.conf /etc/driver/drv/sd.conf
```

3. **Modify the parameter entry and save the file.**

   For example, the `sd.conf` includes the following entry for `sd` device at target 0, lun 0:

   ```
   name="sd" class="scsi" target=0 lun=0;
   ```

   To add the `retries` parameter for this device, modify the existing entry as follows:

   ```
   name="sd" class="scsi" target=0 lun=0 retries=4;
   ```

4. **Display the customized property value. For example:**

   ```
   # prtconf -vu
   sd, instance #1
   Admin properties:
   name='retries' type=int items=1
   value=00000004
   ```

Automatic Configuration of Devices

The kernel consists of a small generic core with a platform-specific component and a set of modules. A *kernel module* is a software component that performs a specific task on the system. A device driver that is loaded when the device is accessed is an example of a *loadable* kernel module.

The following table lists the contents of kernel modules.

TABLE 1-1 Description of Solaris Kernel Modules

Location	Directory Contents
`/platform/`*arch*`/kernel` where *arch* is the information that is displayed by the `uname -m` command.	Platform-specific kernel components
`/kernel`	Kernel components common to all platforms that are needed for booting the system
`/usr/kernel`	Kernel components common to all platforms within a particular instruction set

The system determines what devices are attached to it at boot time. Then, the kernel configures itself dynamically, loading needed modules into memory. At this time, device drivers are loaded when devices, such as disk devices and tape devices, are accessed. This process is called *autoconfiguration* where all kernel modules are loaded automatically when they are needed.

Autconfiguration provides the following benefits:

- Main memory is used more efficiently because modules are loaded when needed.
- Kernel reconfiguration is unnecessary when new devices are added to the system.
- You can load and test drivers without having to rebuild the kernel.
- You can add a new device and its driver without having to perform a reconfiguration boot afterwards.

You can add, remove, or replace devices in Oracle Solaris OS while the system is still running, provided that the system components support hot-plugging. For information about hot-plugging devices, see Chapter 2, “Dynamically Configuring Devices”.

You can also customize the way in which kernel modules are loaded by modifying the `/etc/system` file. For instructions on modifying this file, see `system`(4).

Installing Unsupported Devices

Oracle Solaris includes all the device drivers necessary to support a wide range of standard devices. These drivers can be found in the `/kernel/drv` and `/platform/`uname -m`/kernel/drv` directories.

However, for an unsupported device, the manufacturer should provide the necessary software to properly install, maintain, and administer the device. At a minimum, this software consists of the following:

- Device driver
- Associated configuration file that will reside in the `/kernel/drv` directories
- Necessary custom maintenance and administrative utilities, in case the device is incompatible with Oracle Solaris utilities

For more information about what you need for unsupported devices, contact your device manufacturer.

Displaying Device Configuration Information

The following table describes the three commands that are used to display system and device configuration information.

Command	Description	Man Page
`prtconf`	Displays system configuration information, including the total amount of memory and the device configuration, as described by the system's device hierarchy. The output that is displayed by this command depends upon the type of system.	`prtconf`(1M)

Command	Description	Man Page
`sysdef`	Displays device configuration information, including system hardware, pseudo devices, loadable modules, and selected kernel parameters.	`sysdef`(1M)
`dmesg`	Displays system diagnostic messages, as well as a list of devices that are attached to the system since the last reboot.	`dmesg`(1M)

For information about the device names that are used to identify devices on the system, see "Device Naming Conventions" on page 26.

When you run the `prtconf` and `sysdef` commands, the output might include the following driver-related message line next to specific device instances:

device, `instance` *#number* `(driver not attached)`

The absence of a driver can mean one of the following:

- A driver is unavailable for the device.
- No driver is currently attached because the device is not in use. When a device is unused, its driver is automatically unloaded.
- No driver is currently attached because the device itself does not exist at that device instance.

  Because system processes always monitor system devices, the message usually indicates the absence of the device.

The following utilities detect when a specified device is in use:

- `dumpadm`
- `format`
- `mkfs` and `newfs`
- `swap`

The utilities might detect some of the following usage scenarios:

- Device is part of a ZFS storage pool
- Device is a dump or swap device
- Mounted file system or an entry for the device exists in the `/etc/vfstab` file

For example, using the `format` utility to access an active device might generate a message similar to the following:

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
0. c1t0d0 <FUJITSU-MAY2073RCSUN72G-0401 cyl 8921 alt 2 hd 255 sec 63>
/pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@0,0
/dev/chassis/SYS/HD0/disk
```

```
1. c1t1d0 <FUJITSU-MAY2073RCSUN72G-0401-68.37GB>
/pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@1,0
/dev/chassis/SYS/HD1/disk
Specify disk (enter its number): 0
selecting c1t0d0
[disk formatted]
/dev/dsk/c1t0d0s0 is part of active ZFS pool rpool. Please see zpool(1M).


FORMAT MENU:
.
.
.
```

The following examples show different uses of the `prtconf`, `sysdef`, and other commands to display system configuration information.

EXAMPLE 1-1 Displaying All the Devices of a SPARC Based System

```
$ /usr/sbin/prtconf -v | more
Memory size: 32640 Megabytes
System Peripherals (Software Nodes):

SUNW,SPARC-Enterprise-T5220
.
.
.
location: /dev/chassis/SUN-Storage-J4400.0918QAKA24/SCSI_Device__2/disk
Device Minor Nodes:                          The detailed disk information is described in this section.
dev=(27,40)
dev_path=/pci@0/pci@0/pci@8/pci@0/pci@a/LSILogic,sas@0/sd@2,0:a
spectype=blk type=minor
dev_link=/dev/dsk/c4t2d0s0
dev_path=/pci@0/pci@0/pci@8/pci@0/pci@a/LSILogic,sas@0/sd@2,0:a,raw
spectype=chr type=minor
dev_link=/dev/rdsk/c4t2d0s0
Device Minor Layered Under:
mod=zfs accesstype=blk
dev_path=/pseudo/zfs@0
Minor properties:
name='Nblocks' type=int64 items=1 dev=(27,40)
value=0000000074702c8f
name='Size' type=int64 items=1 dev=(27,40)
value=000000e8e0591e00
.
.
.
```

EXAMPLE 1-2 Displaying Information About a Specific Device of a SPARC Based System

In this example, the `sd` instance number for `/dev/dsk/c4t2d0s0` is displayed.

```
# prtconf -v /dev/dsk/c4t2d0s0 | grep instance
sd, instance #5
```

EXAMPLE 1-3 Displaying Only the Devices That Are Attached to the System

```
# prtconf | grep -v not
```

EXAMPLE 1-4 Displaying Device Usage Information

In this example, the `fuser` command is used to display which processes are accessing the `/dev/console` device.

```
# fuser -d /dev/console
/dev/console:     5742o    2269o   20322o     858o
```

EXAMPLE 1-5 Displaying System Configuration Information of an x86 Based System

```
# prtconf
System Configuration:  Oracle Corporation  i86pc
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

i86pc
scsi_vhci, instance #0
pci, instance #0
pci108e,4843, instance #0
pci8086,25e2, instance #0
pci8086,3500, instance #7
pci8086,3510, instance #9
pci8086,3518, instance #10
pci108e,4843, instance #0
pci108e,4843, instance #1
pci8086,350c, instance #8
pci8086,25e3 (driver not attached)
pci8086,25f8, instance #2
pci108e,286, instance #0
disk, instance #0
disk, instance #2
disk, instance #3
disk, instance #1
pci8086,25e5 (driver not attached)
pci8086,25f9 (driver not attached)
pci8086,25e7 (driver not attached)
pci108e,4843, instance #0 (driver not attached)
pci108e,4843, instance #1
pci108e,4843, instance #2 (driver not attached)
pci108e,4843 (driver not attached)
pci108e,4843 (driver not attached)
pci108e,4843 (driver not attached)
pci108e,4843 (driver not attached)
pci8086,2690, instance #6
pci108e,125e, instance #2
pci108e,125e, instance #3
pci108e,4843, instance #0
pci108e,4843, instance #1
device, instance #0
keyboard, instance #0
```

```
mouse, instance #1
pci108e,4843, instance #2
pci108e,4843, instance #3
pci108e,4843, instance #0
storage, instance #0
disk, instance #4
.
.
.
```

EXAMPLE 1-6 Displaying Device Configuration Information of an x86 Based System

```
# sysdef
* Hostid
*
29f10b4d
*
* i86pc Configuration
*
*
* Devices
*
+boot (driver not attached)
memory (driver not attached)
aliases (driver not attached)
chosen (driver not attached)
i86pc-memory (driver not attached)
i86pc-mmu (driver not attached)
openprom (driver not attached)
options, instance #0
packages (driver not attached)
delayed-writes (driver not attached)
itu-props (driver not attached)
isa, instance #0
motherboard (driver not attached)
pnpADP,1542, instance #0
asy, instance #0
asy, instance #1
lp, instance #0 (driver not attached)
fdc, instance #0
fd, instance #0
fd, instance #1 (driver not attached)
kd (driver not attached)
kdmouse (driver not attached)
.
.
.
```

Resolving Faulty Devices

A device retirement mechanism isolates a device that is flagged as *faulty* by the fault management framework (FMA). This feature allows faulty devices to be safely and automatically inactivated to avoid data loss, data corruption, or panics and system down time.

The retirement process takes into account the stability of the system after the device has been retired.

Critical devices are never retired. If you need to manually replace a retired device, use the `fmadm repair` command after the device replacement so that system knows that the device is replaced.

For more information, see `fmadm`(1M).

When a device is retired, a message similar to the following is displayed on the console and recorded on the `/var/adm/messages` file.

```
Aug 9 18:14 starbug genunix: [ID 751201 kern.notice] \
     NOTICE: One or more I/O devices have been retired
```

You can use the `prtconf` command to identify specific retired devices. For example:

```
# prtconf
.
.
.
pci, instance #2
scsi, instance #0
disk (driver not attached)
tape (driver not attached)
sd, instance #3
sd, instance #0 (retired)
scsi, instance #1 (retired)
disk (retired)
tape (retired)
pci, instance #3
network, instance #2 (driver not attached)
network, instance #3 (driver not attached)
os-io (driver not attached)
iscsi, instance #0
pseudo, instance #0
.
.
.
```

▼ How to Resolve a Faulty Device

Use the steps that follow to resolve a faulty device or a device that has been retired.

Note - For ZFS device problem or failure information, see Chapter 10, “Oracle Solaris ZFS Troubleshooting and Pool Recovery,” in “Managing ZFS File Systems in Oracle Solaris 11.2 ”.

---

1. **Identify the faulted device with the `fmadm faulty` command. For example:**

```
# fmadm faulty
--------------- ------------------------------------  -------------- ---------
TIME            EVENT-ID                              MSG-ID SEVERITY
--------------- ------------------------------------  -------------- ---------
Jun 20 16:30:52 55c82fff-b709-62f5-b66e-b4e1bbe9dcb1  ZFS-8000-LR Major

Problem Status    : solved
Diag Engine       : zfs-diagnosis / 1.0
System
Manufacturer  : unknown
Name          : ORCL,SPARC-T3-4
Part_Number   : unknown
Serial_Number : 1120BDRCCD
Host_ID       : 84a02d28

----------------------------------------
Suspect 1 of 1 :
Fault class : fault.fs.zfs.open_failed
Certainty   : 100%
Affects     : zfs://pool=86124fa573cad84e/vdev=25d36cd46e0a7f49/
pool_name=pond/vdev_name=id1,sd@n5000c500335dc60f/a
Status      : faulted and taken out of service

FRU
Name             : "zfs://pool=86124fa573cad84e/vdev=25d36cd46e0a7f49/
pool_name=pond/vdev_name=id1,sd@n5000c500335dc60f/a"
Status        : faulty

Description : ZFS device 'id1,sd@n5000c500335dc60f/a' in pool 'pond' failed to
open.

Response    : An attempt will be made to activate a hot spare if available.

Impact      : Fault tolerance of the pool may be compromised.

Action      : Use 'fmadm faulty' to provide a more detailed view of this event.
Run 'zpool status -lx' for more information. Please refer to the
associated reference document at
http://support.oracle.com/msg/ZFS-8000-LR for the latest service
procedures and policies regarding this diagnosis.
```

2. **Replace the faulty or retired device or clear the device error. For example:**

```
# zpool clear pond c0t5000C500335DC60Fd0
```

If an intermittent device error occurred but the device was not replaced, you can attempt to clear the previous error.

3. **Clear the FMA fault. For example:**

```
# fmadm repaired zfs://pool=86124fa573cad84e/vdev=25d36cd46e0a7f49/ \
pool_name=pond/vdev_name=id1,sd@n5000c500335dc60f/a
fmadm: recorded repair to of zfs://pool=86124fa573cad84e/vdev=25d36cd46e0a7f49/
pool_name=pond/vdev_name=id1,sd@n5000c500335dc60f/a
```

4. **Confirm that the fault is cleared.**

```
# fmadm faulty
```

If the error is cleared, the `fmadm faulty` command returns nothing.

Adding a Peripheral Device to a System

To adding a new peripheral device that is not hot-pluggable, you perform the following general steps:

- Shut down the system
- Connect the device to the system
- Reboot the system

In some cases, you might have to add a third-party device driver to support the new device.

For information on hot-plugging devices, see Chapter 2, “Dynamically Configuring Devices”.

▼ How to Add a Peripheral Device

This procedure applies to the following devices that are not hot-pluggable:

- DVD drive
- Secondary disk drive
- Tape drive

1. **Become an administrator.**

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

2. **(Optional) If you need to add a device driver to support the device, complete the procedure “How to Add a Device Driver” on page 24.**

3. **Shut down the system.**

```
# shutdown -i0 -g30 -y
```

`-i0`	Brings the system to the 0 init state, which is the appropriate state for turning the system power off for adding and removing devices.
`-g30`	Shuts the system down in 30 seconds. The default is 60 seconds.

`-y` Continues the system shutdown without user intervention. Otherwise, you are prompted to continue the shutdown process.

4. **Select one of the following to turn off power to the system after it is shut down:**

   - For SPARC platforms, turn off power when the `ok` prompt is displayed.
   - For x86 platforms, turn off power when the `type any key to continue` prompt is displayed.

5. **Turn off power to all peripheral devices.**

   For the location of power switches on any peripheral devices, refer to the hardware installation guides that accompany your peripheral devices.

6. **Install the peripheral device, making sure that the hardware configuration adheres to manufacturer specifications.**

   Refer to the hardware installation guide that accompanies the peripheral device for information on installing and connecting the device.

7. **Turn on the power to the system.**

   The system boots to multiuser mode, and the login prompt is displayed.

8. **To verify that the peripheral device has been added, access the device.**

   For information on accessing the device, see "Accessing Devices" on page 25.

▼ How to Add a Device Driver

In the Oracle Solaris 11 release, the `pkg` commands are used to add packages to the system. Device driver packages might still be in SRV4 package format so the steps below use the `pkgadd` command instead of the `pkg install` command.

This procedure assumes that the device has already been added to the system. If not, see "How to Add a Peripheral Device" on page 23.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in "Securing Users and Processes in Oracle Solaris 11.2 ".

2. **Place the media into the drive.**

3. **Install the driver.**

```
# pkgadd [-d] device package-name
```

4. **Verify that the package has been added correctly.**

```
# pkgchk package-name
#
```

If the package is correctly installed, the command generates no output.

Example 1-7 Adding a Device Driver

The following example shows how to install and verify a package called `XYZdrv`.

```
# pkgadd XYZdrv
(licensing messages displayed)
.
.
.
Installing XYZ Company driver as <XYZdrv>
.
.
.
Installation of <XYZdrv> was successful.
# pkgchk XYZdrv
#
```

Accessing Devices

To manage disks, file systems, and other devices, you must know how to specify device names. In most cases, you can use logical device names to represent devices that are connected to the system. Both logical and physical device names are represented on the system by logical and physical device files.

How Device Information Is Created

When a system is booted for the first time, a device hierarchy is created to represent all the devices connected to the system. The kernel uses the device hierarchy information to associate drivers with their appropriate devices. The kernel also provides a set of pointers to the drivers that perform specific operations.

The device hierarchy consists of the following:

- The `/devices` directory – the name space of all devices on the system. This directory represents the *physical* devices that consists of actual bus and device addresses. The `devfs` file system manages this directory.

- The `/dev` directory – the name space of *logical* device names. The `dev` file system manages this directory.

The `devfsadm` command manages system devices by performing the following operations:

- It attempts to load every driver in the system and attach to all possible device instances.
- It creates the device files in the `/devices` directory and the logical links in the `/dev` directory.
- It maintains the `path_to_inst` instance database.

The `devfsadmd` daemon automatically updates the `/dev` and `/devices` directories in response to dynamic reconfiguration events or file system accesses. This daemon is started by the service management facility when a system is booted.

For more information, see the following references:

- `devfsadm`(1M)
- `dev`(7FS)
- `devfs`(7FS)
- `path_to_inst`(4)

Device Naming Conventions

In Oracle Solaris, devices are referenced in one of three ways:

- **Physical device name** – Represents the full device path name in the device information hierarchy. The physical device name is created by when the device is first added to the system. Physical device files are found in the `/devices` directory.
- **Instance name** – Represents the kernel's abbreviation name for every possible device on the system. For example, `sd0` and `sd1` represent the instance names of two disk devices. Instance names are mapped in the `/etc/path_to_inst` file.
- **Logical device name** – The logical device name is created by when the device is first added to the system. Logical device names are used with most file system commands to refer to devices. For a list of file commands that use logical device names, see Table 1-2. Logical device files in the `/dev` directory are symbolically linked to physical device files in the `/devices` directory.

Device name information is displayed with the following commands:

- `dmesg`
- `format`
- `sysdef`

- `prtconf`

Logical Disk Device Names

Logical device names are used to access disk devices when you perform the following tasks:

- Add a new disk to the system.
- Move a disk from one system to another system.
- Access or mount a file system residing on a local disk.
- Back up a local file system.

Names of logical devices use the format `/dev/[r]dsk/c`*n*`t`*n*`d`*n*`[s`*n*`,p`*n*`]`.

`dev`	Devices directory
`[r]dsk`	Raw disk device subdirectory
`c`*n*	Logical controller number
`t`*n*	Physical bus target number
`d`*n*	Drive number
`[s`*n*`,p`*n*`]`	Slice number (`s0` to `s7`) or `fdisk` partition number (`p0` to `p4`).

The following sections explain some of these name components in detail.

Disk Subdirectory

Some disk and file administration commands require the use of either a *raw* (or *character*) device interface, or a *block* device interface. Raw device interfaces transfer only small amounts of data at a time. Block device interfaces include a buffer from which large blocks of data are read at once.

The disk subdirectory you specify depends on the device interface required by the command.

- When a command requires the raw device interface, specify the `/dev/rdsk` subdirectory. (The “r” in `rdsk` stands for “raw.”)
- When a command requires the block device interface, specify the `/dev/dsk` subdirectory.
- When you are not sure whether a command requires use of `/dev/dsk` or `/dev/rdsk`, check the man page for that command.

The following table shows which interface is required for some commonly used disk and file system commands.

TABLE 1-2 Device Interface Type Required by Some Frequently Used Commands

Command Reference	Interface Type	Example of Use
dumpadm(1M)	Block	`dumpadm -d /dev/zvol/dsk/rpool/dump`
prtvtoc(1M)	Raw	`prtvtoc /dev/rdsk/c0t0d0s0`
swap(1M)	Block	`swap -a /dev/zvol/dsk/rpool/swap`

Direct and Bus-Oriented Controllers

You might access disk partitions or slices differently depending upon whether the disk device is connected to a direct or bus-oriented controller. Generally, direct controllers do not include a *target* identifier in the logical device name.

Note - Controller numbers are assigned automatically during system initialization. The numbers are strictly logical and imply no direct mapping to physical controllers.

- To specify a slice on a disk with an IDE controller, use the naming convention `c`*n*`d`*n*`[s`*n*`,p`*n*`]`.

  `c`*n* Logical controller number

  `d`*n* Drive number

  `[s`*n*`,p`*n* Slice number (`s0` to `s7`) or `fdisk` partition number (`p0` to `p4`)

  To indicate the entire `fdisk` partition, specify slice 2 (`s2`).
- To specify a slice on a disk with a bus-oriented controller, SCSI for instance, use the naming convention `c`*n*`t`*n*`d`*n*`[s`*n*`,p`*n*`]`.

  `c`*n* Logical controller number

  `w`*n* Physical bus target number

  `d`*n* Drive number

  `[s`*n*`,p`*n* Slice number (`s0` to `s7`) or `fdisk` partition number (`p0` to `p4`)

To indicate the whole disk, specify slice 2 (`s2`).

Logical Tape Device Names

Logical tape device files are found in the `/dev/rmt/*` directory as symbolic links from the `/devices` directory. The naming convention is `/dev/rmt/`*nd*.

`dev`	Devices directory
`rmt`	Raw magnetic tape device directory
n	Drive number (`0`-*n*
d	Optional density, which can be `l` (low), `m` (medium), `h` (high), `u` (ultra), or `c` (compressed).

The first tape device connected to the system is 0 (`/dev/rmt/0`). Tape density values are described in Chapter 11, “Managing Tape Drives Tasks”.

Logical Removable Media Device Names

Since removable media is managed by removable media management services, the logical device name is usually not used unless you want to mount the media manually.

The logical device name that represents the removable media devices on a system are described in “Accessing Removable Media” on page 242.

Where to Find Additional Device Management Tasks

The following table describes where to find step-by-step instructions for hot-plugging devices and adding serial devices, such as printers and modems.

TABLE 1-3 Where to Find Instructions for Adding a Device

Device Management Task	For More Information
Add a disk that is not hot-pluggable.	Chapter 7, “Setting Up ZFS on Disks” or “Setting Up Disks for ZFS File Systems” on page 134
Hot-plug a SCSI or PCI device.	“Performing SCSI Hot-Plugging Operations” on page 33 or “PCI or PCIe Hot-Plugging With the `cfgadm` Command” on page 43

Device Management Task	For More Information
Hot-plug a USB device.	“Managing USB Mass Storage Devices” on page 210
Add terminal services.	Chapter 5, “Managing the System Console, Terminal Devices, and Power Services,” in “Managing System Information, Processes, and Performance in Oracle Solaris 11.2 ”
Add a printer.	“Configuring and Managing Printing in Oracle Solaris 11.2 ”
Secure a device.	“Securing Systems and Attached Devices in Oracle Solaris 11.2 ”

♦ ♦ ♦ CHAPTER 2

Dynamically Configuring Devices

This chapter describes dynamic reconfiguration (DR) in Oracle Solaris and provides instructions for performing DR operations on SPARC based or x86 based systems. The following topics are covered:

For information about hot-plugging other device types, see the following chapters or sections:

- USB devices – "Hot-Plugging USB Devices With the `cfgadm` Command" on page 226
- InfiniBand devices – Chapter 3, "Using InfiniBand Devices"
- Accessing devices – "Accessing Devices" on page 25

Dynamic Reconfiguration and Hot-Plugging

Hot-plugging is an operation in which you add, remove, or replace system components while the system is running. *Dynamic reconfiguration* refers to the ability to adjust configuration of hot-plugged components. This term also refers to the general ability to move both hardware and software system resources around in the system or to disable them in some way without physically removing them from the system.

In Oracle Solaris, you can add, remove, or replace devices while the system is still running provided that the system components support hot-plugging. Without the support, new devices are configured at boot time, after the new components are installed on the system.

You can hot-plug bus types such as USB, Fibre Channel, SCSI, and so on. Additionally, you can hot-plug devices such as PCI and PCIe, USB, InfiniBand, and so on.

To perform hot-plugging and DR, you typically use the `cfgadm` command. This command also guides you through the steps to complete these tasks. With the command, you can perform the following:

- Display system component status
- Test system components
- Change component configurations
- Display configuration help messages

Performing DR and hot-plugging require administrative privileges that are not generally granted to user accounts. Therefore, you must obtain the appropriate rights for these tasks. For more information, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

Use DR in conjunction with additional layered products from Oracle, such as alternate pathing or fail over software. These products work together to provide fault tolerance in the event of a device failure and thus ensure higher availability of the systems.

Without high availability software, you replace a failed device only by manually stopping the appropriate applications, unmounting noncritical file systems, and then proceeding with the device replacement.

Note - Some systems have a mix of slots that are hot-pluggable and not hot-pluggable. Refer to your hardware documentation for information about hot-plugging devices on your specific system.

Attachment Points

Attachment points are locations on the system where DR can occur.

An attachment point consists of the following parts:

- *Occupant* – a hardware component that can be configured into the system. An occupant's state can either be configured or unconfigured.
- *Receptacle* – the location that accepts the occupant. A receptacle's state can either be connected or disconnected. The empty state also exists but applies only to non SCSI host bus adapters (HBAs).

The following table shows the combined states of occupants and receptacles on attachment points and the corresponding states of a device.

Occupant and Receptacle Combined States	Description of Device State
Unconfigured/Empty	Device is not physically connected (applies to non SCSI HBAs only.
Unconfigured/Disconnected	Device is logically disconnected and unavailable, even though the device might be physically connected.
Unconfigured/Connected	Device is logically connected but unavailable. Device is included in the `prtconf` command output.
Configured/Connected	Device is connected and available.

About Attachment Point Identification

Attachment points are represented by physical and logical attachment point IDs (`Ap_Ids`). The physical `Ap_Id` is the physical path name of the attachment point. The logical `Ap_Id` is a user-friendly alternative for the physical `Ap_Id`. For more information on `Ap_Ids`, refer to `cfgadm`(1M) man page.

The logical `Ap_Id` for a device consists of the combination of the HBA `Ap_Id` and the device identifier, and follows the format *HBA-apid::device-identifier*.

For example, the `Ap_Id` of a SCSI HBA is normally the controller number, such as `c0`. If the device identifier on the HBA is `dsk`, then that device's logical `Ap_Id` would be `c0::dsk`.

The device identifier is derived from the logical device name in the `/dev` directory. For example, a tape device with logical device name, `/dev/rmt/1`, would have the device identifier `rmt/1`. Thus, the tape device's logical `Ap_Id` would be `c0::rmt/1`.

If an HBA `Ap_Id` has no controller number, an internally generated unique identifier is provided, such as `fas1:scsi`. If a device identifier cannot be derived from the logical name in the `/dev` directory, then an internally generated unique identifier is also provided. For example, for the `/dev/rmt/1` tape device, the logical name might be `st4` and the logical `Ap_Id` would be `c0::st4`.

For more information about SCSI `Ap_Ids`, refer to `cfgadm_scsi`(1M).

Performing SCSI Hot-Plugging Operations

This section provides various procedures and examples for performing SCSI hot-plugging operations as listed in the following task map.

TABLE 2-1 SCSI Hot-Plugging With the `cfgadm` Command Task Map

Task	Description	For Instructions
Display information about SCSI devices.	Display information about SCSI controllers and devices.	"Displaying Information About SCSI Devices" on page 34
Configure or unconfigure a SCSI controller.	Perform configuration or unconfiguration operations on a SCSI controller.	"Configuring or Unconfiguring a SCSI Controller" on page 35
Connect or disconnect a SCSI controller.	Perform connect or disconnect operations on a SCSI controller.	"Connecting or Disconnecting a SCSI Controller" on page 36
Configure or unconfigure a SCSI device.	Perform configuration or unconfiguration operations on a specific SCSI device.	"Configuring or Unconfiguring a SCSI Device" on page 37
Add a SCSI device to a SCSI bus.	Add a specific SCSI device to a SCSI bus.	"How to Add a SCSI Device to a SCSI Bus" on page 37
Replace an identical device on a SCSI controller.	Replace a device on the SCSI bus with another device of the same type.	"How to Replace an Identical Device on a SCSI Controller" on page 39
Remove a SCSI device.	Remove a SCSI device from the system.	"How to Remove a SCSI Device" on page 40
Troubleshoot SCSI configuration problems.	Resolve a failed SCSI unconfigure operation.	"Resolving a Failed SCSI Unconfigure Operation" on page 43

Note - The SCSI framework generally supports hot-plugging of SCSI devices. However, always consult your hardware documentation to confirm whether hot-plugging is supported for your specific SCSI devices.

Displaying Information About SCSI Devices

As a useful rule, display information about the system devices before and after the actual hot-plugging step. The information helps you in the following ways:

- You can properly identify the bus or device to dynamically reconfigure.
- You can verify from the state of the bus or device that the reconfiguration completed successfully.

To display device information, you use the `cfgadm -l[a]` command. You can choose to display only the attachment point information, or you can include other information such as the controllers and attached devices.

Note - The command excludes any unsupported SCSI device from the information display.

The following examples show how to display SCSI device information.

- Showing the state and condition of attachment points on the system.

```
# cfgadm -l
Ap_Id    Type         Receptacle   Occupant     Condition
c2       scsi-bus     connected    configured   unknown
c3       scsi-sas     connected    configured   unknown
```

- Showing the state and condition of controllers and attached devices

```
# cfgadm -al
Ap_Id                Type         Receptacle   Occupant     Condition
c2                   scsi-bus     connected    configured   unknown
c2::dsk/c2t0d0       CD-ROM       connected    configured   unknown
c3                   scsi-sas     connected    configured   unknown
c3::dsk/c3t0d0       disk         connected    configured   unknown
c3::dsk/c3t1d0       disk         connected    configured   unknown
c3::dsk/c3t2d0       disk         connected    configured   unknown
c3::dsk/c3t3d0       disk         connected    configured   unknown
```

Configuring or Unconfiguring a SCSI Controller

Use one of the following commands depending on the operation you want to perform:

- **cfgadm -c configure** *controller* – configures a controller.
- **cfgadm -c unconfigure** *controller* – unconfigures a controller.

In both commands, the *controller* refers to the controller number. For more information about the controller number, see “About Attachment Point Identification” on page 33.

The following example unconfigures the SCSI controller `c2`.

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

The following example shows the state of the `c2` controller after it has been unconfigured. In the displayed information, the `Occupant` field indicates the controller's new state.

```
# cfgadm -al
Ap_Id            Type         Receptacle   Occupant       Condition
c2               scsi-bus     connected    unconfigured   unknown     The controller is unconfigured.
c2::dsk/c2t0d0   unavailable  connected    unconfigured   unknown
c3               scsi-sas     connected    configured     unknown
```

```
c3::dsk/c3t0d0   disk         connected    configured     unknown
c3::dsk/c3t1d0   disk         connected    configured     unknown
c3::dsk/c3t2d0   disk         connected    configured     unknown
c3::dsk/c3t3d0   disk         connected    configured     unknown
```

Note - If an unconfigure operation fails, see "Resolving a Failed SCSI Unconfigure Operation" on page 43.

Connecting or Disconnecting a SCSI Controller

Use one of the following commands depending on the operation you want to perform.

- **`cfgadm -c connect`** *controller* – connects a controller.
- **`cfgadm -c disconnect`** *controller* – disconnects a controller.

Disconnecting the controller suspends all I/O activity on the SCSI bus until the `cfgadm -c connect` command is issued. The DR software initiated by the command to disconnect performs some basic checking to prevent critical partitions from being disconnected. However, the software cannot detect all cases. Some controllers on the system might be for disks that contain critical components of the root file system. Disconnecting these controllers would cause the system to hang and would require a system boot.

Thus, as a precaution during disconnect operations, a confirmation prompt always appears after you issue the command to disconnect. The command is executed only after you provide the confirmation.

Caution - Because of the risks in disconnect operations, always exercise caution to avoid causing the system to fall into an unknown state.

The following example disconnects the SCSI controller `c2` and also shows the confirmation prompt that follows the command.

```
# cfgadm -c disconnect c2
WARNING: Disconnecting critical partitions may cause system hang.
Continue (yes/no)?
```

The following example shows the state of the `c2` controller after it has been disconnected. In the displayed information, the `Receptacle` field indicates the controller's new state. Further, the devices that are attached to it are also disconnected.

```
# cfgadm -al
Ap_Id           Type         Receptacle   Occupant     Condition
c2              unavailable  disconnected configured   unknown     The controller is disconnected.
c2::dsk/c2t0d0  unavailable  disconnected configured   unknown     Attached device is disconnected.
```

```
c3              scsi-sas     connected    configured   unknown
c3::dsk/c3t0d0  disk         connected    configured   unknown
c3::dsk/c3t1d0  disk         connected    configured   unknown
c3::dsk/c3t2d0  disk         connected    configured   unknown
c3::dsk/c3t3d0  disk         connected    configured   unknown
```

Configuring or Unconfiguring a SCSI Device

Use one of the following commands depending on the operation you want to perform.

- **`cfgadm -c configure`** *device* – configures a device.
- **`cfgadm -c unconfigure`** *device* – unconfigures a device.

In both commands, *device* refers to the device's logical `Ap_Id`. For more information about a device's logical `Ap_Id`, see "About Attachment Point Identification" on page 33.

The following example configures the `c2t0d0` device. Based on the system's device configuration information, the device's logical `Ap_Id` is `c2::dsk/c2t0d0`.

```
# cfgadm -c configure c2::dsk/c2t0d0
```

The following example shows the state of the `c2t0d0` device after it has been configured. In the displayed information, the `Occupant` field indicates the device's new state.

```
# cfgadm -al
Ap_Id            Type      Receptacle   Occupant     Condition
c2               scsi-bus  connected    configured   unknown
c2::dsk/c2t0d0   CD-ROM    connected    configured   unknown     The device is configured.
c3               scsi-sas  connected    configured   unknown
c3::dsk/c3t0d0   disk      connected    configured   unknown
c3::dsk/c3t1d0   disk      connected    configured   unknown
c3::dsk/c3t2d0   disk      connected    configured   unknown
c3::dsk/c3t3d0   disk      connected    configured   unknown
```

▼ SPARC: How to Add a SCSI Device to a SCSI Bus

Note - When you add devices, specify the `Ap_Id` of the SCSI HBA (controller) to which the device is attached, not the `Ap_Id` of the device itself.

---

1. (Optional) Display the SCSI configuration to identify the bus to which to add the device.

```
# cfgadm -al
```

2. **Add the device.**

   a. **Type the following command:**

      # **cfgadm -x insert_device** *controller*

   b. **Type `y` at the prompt to proceed.**

      I/O activity on the SCSI bus is suspended while the hot-plug operation is in progress.

   c. **Insert the device and switch its power on.**

   d. **Confirm that the operation has been completed.**

3. **Verify that the device has been added.**

   # **cfgadm -al**

Example 2-1 Adding a SCSI Device

In this example, a device is added to the controller `c3`.

```
# cfgadm -al
Ap_Id            Type         Receptacle   Occupant     Condition
c2               scsi-bus     connected    configured   unknown
c2::dsk/c2t0d0   CD-ROM       connected    configured   unknown
c3               scsi-sas     connected    configured   unknown
c3::dsk/c3t0d0   disk         connected    configured   unknown
c3::dsk/c3t1d0   disk         connected    configured   unknown
c3::dsk/c3t2d0   disk         connected    configured   unknown

# cfgadm -x insert_device c3
Adding device to SCSI HBA: /devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2
This operation will suspend activity on SCSI bus: c3
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

After the device is added and power to the device is turned on, the operation continues with a prompt.

```
Enter y if operation is complete or n to abort (yes/no)? y

# cfgadm -al
Ap_Id            Type         Receptacle   Occupant     Condition
c2               scsi-bus     connected    configured   unknown
c2::dsk/c2t0d0   CD-ROM       connected    configured   unknown
c3               scsi-sas     connected    configured   unknown
c3::dsk/c3t0d0   disk         connected    configured   unknown
c3::dsk/c3t1d0   disk         connected    configured   unknown
c3::dsk/c3t2d0   disk         connected    configured   unknown
c3::dsk/c3t3d0   disk         connected    configured   unknown     A disk is added to c3.
```

▼ SPARC: How to Replace an Identical Device on a SCSI Controller

Before You Begin Review the following conditions when replacing an identical device on a SCSI controller:

- If you are replacing a mirrored or unmirrored boot device that is part of a ZFS root pool, see http://www.oracle.com/technetwork/articles/servers-storage-admin/o11-091-sol-dis-recovery-489183.html.
- If the device is controlled by legacy volume management software, consult your volume management documentation for the specific steps to replace a device. If you have an active Oracle support plan, see the following references:
  - If you are using Solaris Volume Manager (SVM), see MOS document 1010753.1.
  - If you are using Veritas Volume Manager (VxVM), see MOS document 1003122.1, 1011782.1, and 1002285.1.

1. **(Optional) Display the SCSI configuration to identify the device to be replaced.**

   ```
   # cfgadm -al
   ```

2. **Replace a device on the SCSI bus with another device of the same type.**

   a. **Type the following command.**

      ```
      # cfgadm -x replace_device device-apid
      ```

   b. **Type `y` at the prompt to proceed.**

      I/O activity on the SCSI bus is suspended while the hot-plug operation is in progress.

   c. **Power off the device to be removed and remove it.**

   d. **Add the replacement device. Then, power it on.**

      The replacement device should be of the same type and at the same address (target and LUN) as the device to be removed.

   e. **Confirm that the operation has been completed.**

3. **Verify that the device has been replaced.**

   ```
   # cfgadm -al
   ```

Example 2-2 Replacing Devices of the Same Type on a SCSI Bus

In this example, the disk `c3t3d0` is replaced by an identical device.

```
# cfgadm -al
```

```
c2               scsi-bus     connected    configured   unknown
c2::dsk/c2t0d0   CD-ROM       connected    configured   unknown
c3               scsi-sas     connected    configured   unknown
c3::dsk/c3t0d0   disk         connected    configured   unknown
c3::dsk/c3t1d0   disk         connected    configured   unknown
c3::dsk/c3t2d0   disk         connected    configured   unknown
c3::dsk/c3t3d0   disk         connected    configured   unknown

# cfgadm -x replace_device c3::dsk/c3t3d0
Replacing SCSI device: /devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@3,0
This operation will suspend activity on SCSI bus: c3
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

Power to the device is turned off and the device is replaced. After the replacement device is installed and power is turned on, the operation continues with a prompt.

```
Enter y if operation is complete or n to abort (yes/no)? y

# cfgadm -al
Ap_Id            Type         Receptacle   Occupant     Condition
c2               scsi-bus     connected    configured   unknown
c2::dsk/c2t0d0   CD-ROM       connected    configured   unknown
c3               scsi-sas     connected    configured   unknown
c3::dsk/c3t0d0   disk         connected    configured   unknown
c3::dsk/c3t1d0   disk         connected    configured   unknown
c3::dsk/c3t2d0   disk         connected    configured   unknown
c3::dsk/c3t3d0   disk         connected    configured   unknown      The device is replaced.
```

▼ SPARC: How to Remove a SCSI Device

1. Identify the current SCSI configuration.

```
# cfgadm -al
```

2. Remove the SCSI device from the system.

a. Type the following command.

```
# cfgadm -x remove_device device
```

b. Type `y` at the prompt to proceed.

I/O activity on the SCSI bus is suspended while the hot-plug operation is in progress.

c. Power off the device to be removed and remove it.

d. Confirm that the operation has been completed.

Note - This step must be performed if you are removing a SCSI RAID device from a SCSI RAID array.

3. **Verify that the device has been removed from the system.**

```
# cfgadm -al
```

Example 2-3 Removing a SCSI Device

In this example, the disk `c3t3d0` is removed.

```
# cfgadm -al
Ap_Id            Type         Receptacle   Occupant     Condition
c2               scsi-bus     connected    configured   unknown
c2::dsk/c2t0d0   CD-ROM       connected    configured   unknown
c3               scsi-sas     connected    configured   unknown
c3::dsk/c3t0d0   disk         connected    configured   unknown
c3::dsk/c3t1d0   disk         connected    configured   unknown
c3::dsk/c3t2d0   disk         connected    configured   unknown
c3::dsk/c3t3d0   disk         connected    configured   unknown

# cfgadm -x remove_device c3::dsk/c3t3d0
Removing SCSI device: /devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@3,0
This operation will suspend activity on SCSI bus: c3
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

After power to the device has been turned off and the device is removed, the operation continues with a prompt.

```
Enter y if operation is complete or n to abort (yes/no)? y

# cfgadm -al
Ap_Id            Type         Receptacle   Occupant     Condition
c2               scsi-bus     connected    configured   unknown
c2::dsk/c2t0d0   CD-ROM       connected    configured   unknown
c3               scsi-sas     connected    configured   unknown
c3::dsk/c3t0d0   disk         connected    configured   unknown
c3::dsk/c3t1d0   disk         connected    configured   unknown
c3::dsk/c3t2d0   disk         connected    configured   unknown
```

Troubleshooting SCSI Configuration Problems

This section explains error messages and provides possible solutions to SCSI configuration problems. For more information about troubleshooting SCSI configuration problems, see `cfgadm`(1M).

Error Message

```
cfgadm: Component system is busy, try again: failed to offline:
device-path
Resource             Information
------------------  -------------------------
/dev/dsk/c1t0d0s0   mounted filesystem "/file-system"
```

Cause

You attempted to remove or replace a device with a mounted file system.

Solution

Unmount the file system that is listed in the error message and retry the `cfgadm` operation.

Error Message

```
cfgadm: Component system is busy, try again: failed to offline:
device-path
Resource             Information
------------------  -------------------------
/dev/dsk/device-name   swap area
```

Cause

If you use the `cfgadm` command to remove a system resource, such as a swap device or a dedicated dump device, a similar error message is displayed if the system resource is still active.

Solution

Unconfigure the swap areas on the device that is specified and retry the `cfgadm` operation.

Error Message

```
cfgadm: Component system is busy, try again: failed to offline:
device-path
Resource             Information
------------------  -------------------------
/dev/dsk/device-name   dump device (swap)
```

Cause

You attempted to remove or replace a dump device that is configured on a swap area.

Solution

Unconfigure the dump device that is configured on the swap area and retry the `cfgadm` operation.

Error Message

```
cfgadm: Component system is busy, try again: failed to offline:
device-path
Resource             Information
```

```
-----------------  -------------------------
/dev/dsk/device-name   dump device (dedicated)
```

Cause

You attempted to remove or replace a dedicated dump device.

Solution

Unconfigure the dedicate dump device and retry the `cfgadm` operation.

Resolving a Failed SCSI Unconfigure Operation

If you perform a SCSI unconfigure operation and one or more targeted devices are busy, the operation fails. Subsequent DR operations on this controller and target devices will also fail. Instead, the following message is displayed:

```
dr in progress
```

To resolve the failed operation, reconfigure the controller.

```
# cfgadm -c configure device-apid
```

After verifying that the device has been configured, you can proceed with unconfiguring the device again.

PCI or PCIe Hot-Plugging With the `cfgadm` Command

You can hot-plug PCI adapter cards only if their corresponding device drivers support hot-plugging. The following are additional conditions for PCI or PCIe hot-plugging:

- If you are adding adapter cards, make sure that slots are available.
- If you are removing adapter cards, ensure the following:
  - The adapter card is hosting only nonvital system resources.
  - Critical resources are accessible through an alternate pathway.

    For example, if a system has only one Ethernet card installed on it, you cannot remove the Ethernet card. Otherwise, network connection is lost. In this case, you must have additional layered software support to keep the network connection active.

Note - The `cfgadm` command displays only hot-pluggable PCI devices and slots. It does not provide information about non hot-pluggable PCI devices.

You can also use the prtconf command to display additional configuration information that pertains to the hardware such as verifying that an added hardware is correctly configured.

For example, after a configure operation, use the prtconf -D command to verify that the driver is attached to the newly installed hardware device. If the device driver has not been added to the system prior to hardware configuration, you can manually add it by using the add_drv command.

For more information, see prtconf(1M) and add_drv(1M).

Observe the LED indicators on the system to get a visual indication about the status of the slot's hot-plug operation. In the case of PCI Express devices, the LED behavior matches the behavior defined in the PCI Express specification. Otherwise, the behavior might be platform dependent. Refer to your platform guide for specific details.

For PCI Express devices, when the Attention Button is pressed, the power indicator blinks to indicate the beginning of a state transition. The blinking ends when the state transition has ended.

For brevity, the examples on this section list only PCI attachment points. The attachment points that are displayed on your system depend on your system configuration.

The following task map describes the tasks for managing PCI or PCIe devices on your system.

TABLE 2-2 PCI or PCIe Hot-Plugging With the cfgadm Command Task Map

Task	Description	For Instructions
Display PCI slot configuration information.	Display the status of PCI hot-pluggable devices and slots on the system.	"Displaying PCI Slot Configuration Information" on page 44
Remove a PCI adapter card.	Unconfigure the card, disconnect power from the slot, and remove the card from the system.	"How to Remove a PCI Adapter Card" on page 46
Add a PCI adapter card.	Insert the adapter card into a hot-pluggable slot, connect power to the slot, and configure the card.	"How to Add a PCI Adapter Card" on page 48
Identify error message and possible solutions to resolve PCI configuration problems.	Troubleshoot PCI configuration problems.	"Troubleshooting PCI Configuration Problems" on page 49

Displaying PCI Slot Configuration Information

This section provides examples for display the status of PCI hot pluggable devices and slots on the system.

Note - You must have the appropriate administrator rights to run these commands.

- Displaying PCI slot configuration information.

```
# cfgadm
Ap_Id                Type         Receptacle   Occupant     Condition
pci1:hpc0_slot0      unknown      empty        unconfigured unknown
pci1:hpc0_slot1      unknown      empty        unconfigured unknown
pci1:hpc0_slot2      unknown      empty        unconfigured unknown
pci1:hpc0_slot3      ethernet/hp  connected    configured   ok
pci1:hpc0_slot4      unknown      empty        unconfigured unknown
```

- Displaying specific PCI device information.

```
# cfgadm -s "cols=ap_id:type:info" pci
Ap_Id                Type         Information
pci1:hpc0_slot0      unknown      Slot 7
pci1:hpc0_slot1      unknown      Slot 8
pci1:hpc0_slot2      unknown      Slot 9
pci1:hpc0_slot3      ethernet/hp  Slot 10
pci1:hpc0_slot4      unknown      Slot 11
```

  From the output, `pci1:hpc0_slot0` is the logical `Ap_Id` for the hot-pluggable slot, `Slot 7`. The component `hpc0` indicates the hot-pluggable adapter card for this slot, and `pci1` indicates the PCI bus instance. The `Type` field indicates the type of PCI adapter card that is present in the slot.

- Displaying PCIe slot configuration information.

```
# cfgadm pci
Ap_Id                Type         Receptacle   Occupant     Condition
pcie1                unknown      empty        unconfigured unknown
pcie2                unknown      empty        unconfigured unknown
pcie3                unknown      empty        unconfigured unknown
pcie4                etherne/hp   connected    configured   ok
pcie5                pci-pci/hp   connected    configured   ok
pcie6                unknown      disconnected unconfigured unknown
```

- Displaying specific PCIe device information.

```
# cfgadm -s "cols=ap_id:busy:o_state" pci
Ap_Id               Busy     Occupant
pcie1               n        unconfigured
pcie2               n        unconfigured
pcie3               n        unconfigured
pcie4               n        configured
pcie5               n        configured
pcie6               n        configured
```

Note - The logical `Ap_Id` in most cases should match the slot label that is silk-screened on the system chassis. Refer to your platform guide for the `cfgadm` output of the hot-pluggable slots. The `Busy` field can be displayed to ensure that the `Ap_Id` is not transitioning to another state before a hot-plug operation is attempted.

How to Remove a PCI Adapter Card

1. **Determine which slot the PCI adapter card is in.**

   ```
   # cfgadm pci
   ```

2. **Stop any application that has the device open.**

   For example, if the device is an Ethernet card with a configured IP interface, remove the IP interface.

   ```
   # ipadm delete-ip ip-interface
   ```

3. **Unconfigure the device manually by using one of the following methods:**

   - For a PCIe adapter card, use the auto-configuration method, such as pressing the slot's Attention Button as defined by your platform guide.
   - Issue the following command:

     ```
     # cfgadm -c unconfigure pci-device
     ```

4. **Confirm that the device has been unconfigured.**

   ```
   # cfgadm pci
   ```

5. **Disconnect the power to the slot manually. If the auto-configuration method is used, this step is not necessary.**

   ```
   # cfgadm -c disconnect pci-device
   ```

   Refer to your platform guide for more information.

6. **Confirm that the device has been disconnected.**

7. **Follow appropriate instructions in your platform guide to remove the PCI adapter card. After the card is removed, the Receptacle state is empty.**

   ```
   # cfgadm pci
   ```

Note - The auto-configuration method can be enabled or disabled at boot, depending on the platform implementation. Set the auto-configuration method as appropriate for your environment.

Example 2-4 Removing a PCI Adapter Card

In this example, the card on as `pcie4` is removed.

```
# cfgadm pci
Ap_Id         Type         Receptacle   Occupant     Condition
pcie1         unknown      empty        unconfigured unknown
pcie2         unknown      empty        unconfigured unknown
pcie3         unknown      empty        unconfigured unknown
pcie4         etherne/hp   connected    configured   ok
pcie5         pci-pci/hp   connected    configured   ok
pcie6         unknown      disconnected unconfigured unknown

# ipadm delete-ip net1

# cfgadm -c unconfigure pcie4

# cfgadm pci
Ap_Id         Type         Receptacle   Occupant     Condition
pcie1         unknown      empty        unconfigured unknown
pcie2         unknown      empty        unconfigured unknown
pcie3         unknown      empty        unconfigured unknown
pcie4         unknown      connected    unconfigured unknown     The device is unconfigured.
pcie5         pci-pci/hp   connected    configured   ok
pcie6         unknown      disconnected unconfigured unknown

# cfgadm -c disconnect pcie4

# cfgadm pci
Ap_Id         Type         Receptacle   Occupant     Condition
pcie1         unknown      empty        unconfigured unknown
pcie2         unknown      empty        unconfigured unknown
pcie3         unknown      empty        unconfigured unknown
pcie4         unknown      disconnected unconfigured unknown     Power is disconnected.
pcie5         pci-pci/hp   connected    configured   ok
pcie6         unknown      disconnected unconfigured unknown
```

After the device is removed, the procedure continues.

```
# cfgadm pci
Ap_Id         Type         Receptacle   Occupant     Condition
pcie1         unknown      empty        unconfigured unknown
pcie2         unknown      empty        unconfigured unknown
pcie3         unknown      empty        unconfigured unknown
pcie4         unknown      empty        unconfigured unknown     The receptacle is empty.
pcie5         pci-pci/hp   connected    configured   ok
pcie6         unknown      disconnected unconfigured unknown
```

How to Add a PCI Adapter Card

The following procedure applies to both PCI or PCIe cards when you add them to the system.

1. **Identify the hot-pluggable slot and open latches.**

   ```
   # cfgadm pci
   ```

2. **Follow the appropriate instructions in your platform guide to insert a PCI adapter card into the slot.**

3. **Determine which slot the PCI adapter card occupies after it is inserted.**

   ```
   # cfgadm pci
   ```

4. **Connect the power to the slot by choosing one of the following methods:**

   - Issue the `cfgadm` command.

     ```
     # cfgadm -c connect pci-device
     ```

   - If working on a PCIe adapter card, use the auto-configuration method, such as by pressing the slot's Attention Button as defined by your platform guide.

5. **Confirm that the attachment point is connected.**

   ```
   # cfgadm pci
   ```

6. **Configure the PCI adapter card manually by using the `cfgadm` command.**

   If you are using the auto-configuration method, this step might not be necessary. Refer to your platform guide for more information.

   ```
   # cfgadm -c configure pci-device
   ```

7. **Verify the configuration of the PCI adapter card in the slot.**

8. **Configure any supporting software if this device is a new device.**

   For example, if this device is an Ethernet card, use the `ipadm` command to set up the interface, as follows:

   ```
   # ipadm create-addr ip-interface
   ```

Note - The auto-configuration method can be enabled or disabled at boot, depending on the platform implementation. Set the auto-configuration method as appropriate for your environment.

Example 2-5 Adding a PCI Adapter Card

In this example, the `pcie3` card is added to the system.

```
# cfgadm pci
Ap_Id   Type         Receptacle   Occupant     Condition
pcie1   unknown      empty        unconfigured unknown
pcie2   unknown      empty        unconfigured unknown
pcie3   unknown      disconnected unconfigured unknown
pcie4   unknown      empty        unconfigured unknown
pcie5   pci-pci/hp   connected    configured   ok
pcie6   unknown      disconnected unconfigured unknown

# cfgadm -c connect pcie3

# cfgadm pci
Ap_Id   Type         Receptacle   Occupant     Condition
pcie1   unknown      empty        unconfigured unknown
pcie2   unknown      empty        unconfigured unknown
pcie3   unknown      connected    unconfigured unknown     Device is added.
pcie4   unknown      empty        unconfigured unknown
pcie5   pci-pci/hp   connected    configured   ok
pcie6   unknown      disconnected unconfigured unknown

# cfgadm -c configure pcie3

# cfgadm pci
Ap_Id   Type         Receptacle   Occupant     Condition
pcie1   unknown      empty        unconfigured unknown
pcie2   unknown      empty        unconfigured unknown
pcie3   ethernet/hp  connected    configured   unknown     Device is configured.
pcie5   pci-pci/hp   connected    configured   ok
pcie6   unknown      disconnected unconfigured unknown

# ipadm create-addr -a 192.168.1.10 net1
```

Troubleshooting PCI Configuration Problems

This section describes problems with PCI configuration as indicated by their error messages.

Error Message

```
cfgadm: Configuration operation invalid: invalid transition
```

Cause

An invalid transition was attempted.

Solution

Check whether the `cfgadm -c` command was issued appropriately. Use the `cfgadm` command to check the current receptacle and occupant state and to make sure that the `Ap_Id` is correct.

Error Message

```
cfgadm: Attachment point not found
```

Cause

The specified attachment point was not found.

Solution

Check whether the attachment point is correct. Use the `cfgadm` command to display a list of available attachment points. Also check the physical path to see if the attachment point is still there.

PCIe Hot-Plugging With the `hotplug` Command

You can use the `hotplug` command with PCI Express (PCIe) and PCI SHPC devices to manage hot pluggable connections only. A connection can be a connector or port. A hotplug connector is a physical representation in the system where a component is inserted or removed. A hotplug port is a logical representation in the system device tree where the connection of a device to the system is managed.

You can use the hotplug features to take a device, including an on-board device, online or offline without physically adding or removing the device from the system.

The hotplug service must be enabled to manage devices with the `hotplug` command. Except on sun4v platforms, the service is disabled by default on most platforms. To enable the hotplug service, type the following command:

```
# svcadm enable svc:/system/hotplug:default
```

The following examples show different uses of the `hotplug` command:

- Displaying all the system's PCI/PCIe hot-pluggable connectors or ports. These ports can be virtual or physical.

  ```
  # hotplug list -lv
  ```

- Configuring or unconfiguring PCI/PCIe devices.

  Use one of the following commands:

  - **`hotplug enable`** *path connector*
  - **`hotplug disable`** *path connector*

  The following example configures an Ethernet card in a PCIe slot.

  ```
  # hotplug enable /pci0,0 pcie0
  ```

- Attaching or detaching the device driver for a PCI device node.

Use one of the following commands:

- **`hotplug online`** *path port*
- **`hotplug offline`** *path port*

The following example detaches the device driver for a PCI device node to take it offline.

```
# hotplug offline /pci0,0/pci1 pci.0,2
```

- Installing or uninstalling services that can be supported by the drivers of the device on a specific port

  Use one of the following commands:

  - **`hotplug install`** *path port*
  - **`hotplug uninstall`** *path port*

  The following example installs dependent ports of an IOV physical function.

  ```
  # hotplug install /pci@400/pci@1/pci@0/pci@4 pci.0,1
  ```

- Showing information about all connectors, ports, and their associated devices in verbose mode

  ```
  # hotplug list -v path connection
  ```

  The following example displays IOV virtual functions that were probed after the installation operation illustrated in the previous example.

  ```
  # hotplug list -v /pci@400/pci@1/pci@0/pci@4 pci.0,1
  <pci.0,1>  (ONLINE)
  { IOV physical function }
  { IOV virtual function 'pci.0,81' }
  { IOV virtual function 'pci.0,83' }
  { IOV virtual function 'pci.0,85' }
  { IOV virtual function 'pci.0,87' }
  <pci.0,81>  (OFFLINE)
  ethernet@0,81
  <pci.0,83>  (OFFLINE)
  ethernet@0,83
  <pci.0,85>  (OFFLINE)
  ethernet@0,85
  <pci.0,87>  (OFFLINE)
  ethernet@0,87
  ```

Troubleshooting PCI Hot Plug Operations

The following error messages might be displayed while you are performing PCI hot plug operations:

- About maintenance states for an attached device on a hot-pluggable port

```
/pci@0,0/pci10de,5d@e <pci.a,1> (MAINTENANCE)
.
.
./pci@0,0/pci108e,534a@d <pci.0,0> (MAINTENANCE-SUSPENDED)
```

These messages indicate that a fault event or a maintenance operation occurred. The `MAINTENANCE` states mean that a device in use is not fully operational. The `MAINTENANCE-SUSPENDED` state means that the device is *live suspended* due to a maintenance operation such as the reconfiguration of a device hardware.

- About service unavailability

```
ERROR: hotplug service is not available.
```

The message means that the hotplug service is disabled and you cannot perform hotplug operations. To use the `hotplug` command, the `svc:/system/hotplug:default` must be running.

- About problems displaying connections

```
ERROR: there are no connections to display.
(See hotplug(1m) for more information.)
```

The message means that the system does not have any supported I/O buses. It could also mean that the system might have other hot-pluggable I/O devices, but you need to use the `cfgadm` command rather than the `hotplug` command to manage these devices.

- About busy resources

```
# hotplug uninstall /pci@400/pci@1/pci@0/pci@4 pci.0,0
ERROR: devices or resources are busy.
ethernet@0,81:
...
```

The message means that you are uninstalling dependent ports of an IOV physical function while a dependent IOV virtual function is busy.

Performing SATA Hot-Plugging Operations

SATA controller and port multiplier device ports are represented by attachment points in the device tree. SATA devices that are connected and configured on the system are shown as attachment point name extensions. The terms attachment point and SATA port can be used interchangeably.

The `cfgadm` syntax to display device information is the same as for SCSI devices, which is `cfgadm -al`. However, for brevity, the examples in this section use the `grep` utility to directly display a specific device's information.

The following example shows how SATA information is displayed by the `cfgadm` command.

```
% cfgadm -al
Ap_Id                Type   Receptacle   Occupant     Condition
sata0/0::dsk/c7t0d0  disk   connected    configured   ok
sata0/1::dsk/c7t1d0  disk   connected    configured   ok
sata0/2::dsk/c7t2d0  disk   connected    configured   ok
sata0/3::dsk/c7t3d0  disk   connected    configured   ok
.
.
.
```

Configuring Or Unconfigure a SATA Device

Use one of the following command syntaxes depending on the operation you want to perform:

- **`cfgadm -c configure`** *controller* – configures a SATA controller.
- **`cfgadm -c unconfigure`** *controller* – unconfigures a SATA controller.

In both commands, *controller* refers to the SATA HBA `Ap_Id`, such as `sata0/0`. For more information about the `Ap_Id`, see “About Attachment Point Identification” on page 33.

The following example configures `c7t0d0`.

```
# cfgadm -c configure c7t0d0
```

The following example shows the state of `c7t0d0` after it has been configured.

```
# cfgadm | grep sata0/0
sata0/0   disk         connected    unconfigured ok
```

▼ How to Replace a SATA Device

You must unconfigure a SATA device first before replacing it. After a device is physically removed or replaced, then configure the device.

1. **Become an administrator.**
2. **Identify the device to be replaced.**

   ```
   # cfgadm -al
   ```

3. **Unconfigure the device.**

```
# cfgadm -c unconfigure controller
```

4. **Confirm that the device is unconfigured.**

```
# cfgadm | grep controller
```

5. **Remove the device and replace it with a device of a similar type.**

6. **Configure the replaced device.**

```
# cfgadm -c configure controller
```

7. **Confirm that the device is configured.**

```
# cfgadm | grep controller
```

Example 2-6 Replacing a SATA Device

In this example, the `c7t3d0` device is replaced with a device of a similar type..

```
# cfgadm -al
Ap_Id                  Type         Receptacle   Occupant     Condition
sata0/0::dsk/c7t0d0    disk         connected    configured   ok
sata0/1::dsk/c7t1d0    disk         connected    configured   ok
sata0/2::dsk/c7t2d0    disk         connected    configured   ok
sata0/3::dsk/c7t3d0    disk         connected    configured   ok
.
.
.
# cfgadm -c unconfigure sata0/3

# cfgadm | grep sata0/3
sata0/3                disk         connected    unconfigured ok
```

After the `c7t3d0` is replaced, configure the replacement device.

```
# cfgadm -c configure sata0/3

# cfgadm | grep sata0/3
sata0/3::dsk/c7t3d0      disk         connected   configured   ok
```

Troubleshooting SATA Configuration Problems

While performing hot-plugging operations with SATA devices, you might see the following error messages.

- About non existent file errors

```
do_control_ioctl: open failed: errno:2
```

```
cfgadm: Library error: Cannot open ap_id: /devices/pci@0,0/pci10...
No such file or directory
```

The message might indicate that you are configuring or unconfiguring the device by specifying the device's logical `Ap_Id` with the `cfgadm` command, such as `sata0/0::dsk/c7t0d0`. Use the HBA `Ap_Id` or controller number instead.

- About hardware specific failure

```
Unconfigure the device at: /devices/pci@2,0/pci1022...
This operation will suspend activity on the SATA device
Continue (yes/no)? y
cfgadm: Hardware specific failure: Failed to unconfig device \
                        at ap_id: /devices/pci@2,0/pci10...
```

The message indicates that you are unconfiguring a device that is part of an active ZFS storage pool. Remove that device from the pool first.

Reconfiguration Coordination Manager (RCM) Script Overview

The Reconfiguration Coordination Manager (RCM) is the framework that manages the dynamic removal of system components. By using RCM, you can register and release system resources in an orderly manner.

The RCM script feature simplifies and better controls the DR process. By creating an RCM script, you can do the following:

- Automatically release a device when you dynamically remove a device. This process also closes the device if the device is opened by an application.
- Run site-specific tasks when you dynamically remove a device from the system.
- Write customized scripts to perform other shutdown operations when a reconfiguration request is received. and which impacts the resources that are registered in the script.

Note - The `cfgadm -f` command can force a reconfiguration operation. However, this command might leave applications in an unknown state. Manually releasing resources from applications commonly causes errors. Thus, avoid using this syntax for to perform reconfiguration.

About RCM Scripts

An RCM script can be any of the following:

- An executable shell script (Perl, `sh`, `csh`, or `ksh`) or binary program that the RCM daemon runs. Perl is the recommended language.
- A script that runs in its own address space by using the user ID of the script file owner.
- A script that is run by the RCM daemon when you use the `cfgadm` command to dynamically reconfigure a system resource.

You can use an RCM script to release a device from an application when you dynamically remove a device. If the device is currently open, the RCM script also closes the device.

For example, an RCM script for a tape backup application can inform the tape backup application to close the tape drive or shut down the tape backup application.

Run an RCM script as follows:

```
$ script-name command [args ...]
```

An RCM script performs the following basic steps:

1. Takes the RCM command from command-line arguments.
2. Executes the command.
3. Writes the results to `stdout` as name-value pairs.
4. Exits with the appropriate exit status.

The RCM daemon runs one instance of a script at a time. Thus, if a script is running, the RCM daemon does not run the same script until the first script exits.

RCM Script Commands

You must include the following RCM commands in an RCM script:

- `scriptinfo` – gathers script information
- `register` – registers interest in resources
- `resourceinfo` – gathers resource information

You might include some or all of the following RCM commands:

- `queryremove` – queries whether the resource can be released
- `preremove` – releases the resource
- `postremove` – provides post-resource removal notification
- `undoremove` – undoes the actions done in `preremove`

For a complete description of these RCM commands, see `rcmscript(4)` man page.

When you dynamically remove a device, the RCM daemon runs the following:

- The script's `register` command to gather the list of resources (device names) that are identified in the script.
- The script's `queryremove` and `preremove` commands prior to removing the resource if the script's registered resources are affected by the dynamic remove operation.
- The script's `postremove` command if the remove operation succeeds. However, if the remove operation fails, the RCM daemon runs the script's `undoremove` command.

RCM Script Directories

The following table shows the locations where you can store the RCM scripts.

TABLE 2-3 RCM Script Directories

Directory Location	Script Type
`/etc/rcm/scripts`	Scripts for specific systems
`/usr/platform/`uname -i`/lib/rcm/scripts`	Scripts for a specific hardware implementation
`/usr/platform/`uname -m`/lib/rcm/scripts`	Scripts for a specific hardware class
`/usr/lib/rcm/scripts`	Scripts for any hardware

Using RCM Scripts

This section provides procedures for using RCM scripts.

▼ How to Prepare to Write an RCM Script for an Application

This procedure lists some preparatory steps to take before writing an RCM script for applications. After you have completed writing the script, install and test it.

To install the script, see "How to Install an RCM Script" on page 59. To test the script, see "How to Test an RCM Script" on page 59. For an example of an RCM script, see "Tape Backup RCM Script Example" on page 60.

1. **Identify the resources the application uses.**

   These are resources that you can potentially remove which would affect the application.

   ```
   # cfgadm -al
   ```

2. **Identify the application commands for releasing the resource.**

   These commands instruct the application to release the resource in an orderly fashion. Consult your application's documentation for these commands.

3. **Identify the script commands that remove the resource.**

   These commands include notifying the application that the resource has been removed. Refer to the `rcmscript(4)` man page.

4. **Identify the script commands that would notify the application about the availability of the resource, if the resource is replaced.**

   The commands should also include those that automatically run if the resource removal fails.

5. **Write the RCM script and assign it a name the uses the following format:**

   vendor`,`*service*

   - *vendor* – the stock symbol of the vendor that provides the script, or any distinct name that identifies the vendor.
   - *service* – the service that the script represents.

How to Prepare to Write an RCM Script for Specific Site Use

This procedure lists some preparatory steps to take before writing an RCM script for site customization. After you have completed writing the script, install and test it.

To install the script, see "How to Install an RCM Script" on page 59. To test the script, see "How to Test an RCM Script" on page 59. For an example of an RCM script, see "Tape Backup RCM Script Example" on page 60.

1. **Identify the resources to be dynamically removed.**

   Make sure to list the devices' logical `Ap_Id`.

   ```
   # cfgadm -al
   ```

2. **Identify the applications and the commands to stop them before resources can be removed.**

   Consult the applications' documentation for these commands.

3. **Identify the script commands to run before the resources are removed.**

   Refer to the `rcmscript(4)` man page.

4. **Identify the script commands to run after the resources are removed.**

   Refer to the `rcmscript(4)` man page.

How to Install an RCM Script

1. **Become an administrator.**

2. **Copy the script to the appropriate directory.**

   See Table 2-3.

   For example:

   ```
   # cp ABC,sample.pl /usr/lib/rcm/scripts
   ```

3. **Change the user ID and the group ID of the script to the desired values.**

   ```
   # chown user:group /usr/lib/rcm/scripts/ABC,sample.pl
   ```

4. **Send `SIGHUP` to the RCM daemon.**

   ```
   # pkill -HUP -x -u root rcm_daemon
   ```

How to Test an RCM Script

1. **Set environment variables, such as `RCM_ENV_FORCE`, in the command-line shell before running your script.**

   For example, in the Korn shell, use the following:

   ```
   $ export RCM_ENV_FORCE=TRUE
   ```

2. **Test the script by running the script commands manually from the command line.**

   For example:

   ```
   $ script-name scriptinfo
   $ script-name register
   $ script-name preremove resource-name
   $ script-name postremove resource-name
   ```

3. **Make sure that each RCM script command in your script prints appropriate output to `stdout`.**

4. **Install the script in the appropriate script directory.**

For more information, see "How to Install an RCM Script" on page 59.

5. **Test the script by initiating a dynamic remove operation.**

   For example, assume your script registers the device, `/dev/dsk/c1t0d0s0`. Try these commands.

```
$ cfgadm -c unconfigure c1::dsk/c1t0d0
$ cfgadm -f -c unconfigure c1::dsk/c1t0d0
$ cfgadm -c configure c1::dsk/c1t0d0
```

Caution - Make sure that you are familiar with these commands because they can alter the state of the system and cause system failures.

How to Remove an RCM Script

1. **Become an administrator.**

2. **Remove the script from the RCM script directory.**

   For example:

```
# rm /usr/lib/rcm/scripts/ABC,sample.pl
```

3. **Send `SIGHUP` to the RCM daemon.**

```
# pkill -HUP -x -u root rcm_daemon
```

Tape Backup RCM Script Example

This example illustrates how to use an RCM script for tape backups.

What the Tape Backup RCM Script Does

The tape backup RCM script performs the following steps:

1. Sets up a dispatch table of RCM commands.
2. Calls the dispatch routine that corresponds to the specified RCM command and exits with status 2 for unimplemented RCM commands.
3. Sets up the `scriptinfo` section.

```
rcm_script_func_info=Tape backup appl script for DR
```

4. Registers all tape drives in the system by printing all tape drive device names to `stdout`.

   ```
   rcm_resource_name=/dev/rmt/$f
   ```

   If an error occurs, the script prints the error information to `stdout`.

   ```
   rcm_failure_reason=$errmsg
   ```

5. Sets up the resource information for the tape device.

   ```
   rcm_resource_usage_info=Backup Tape Unit Number $unit
   ```

6. Sets up the `preremove` information by checking if the backup application is using the device. If the backup application is not using the device, the dynamic reconfiguration operation continues. If the backup application is using the device, the script checks `RCM_ENV_FORCE`. If `RCM_ENV_FORCE` is set to `FALSE`, the script denies the dynamic reconfiguration operation and prints the following message:

   ```
   rcm_failure_reason=tape backup in progress pid=...
   ```

   If `RCM_ENV_FORCE` is set to `TRUE`, the backup application is stopped, and the reconfiguration operation proceeds.

Outcomes of the Tape Backup Reconfiguration Scenarios

Here are the various outcomes if you use the `cfgadm` command to remove a tape device without the RCM script:

- If you use the `cfgadm` command and the backup application is not using the tape device, the operation succeeds.
- If you use the `cfgadm` command and the backup application is using the tape device, the operation fails.

Here are the various outcomes if you use the `cfgadm` command to remove a tape device with the RCM script.

- If you use the `cfgadm` command and the backup application is not using the tape device, the operation succeeds.
- If you use the `cfgadm` command without the `-f` option and the backup application is using the tape device, the operation fails with an error message similar to the following:

  ```
  tape backup in progress pid=...
  ```

- If you use the `cfgadm -f` command and the backup application is using the tape device, the script stops the backup application and the `cfgadm` operation succeeds.

Example – Tape Backup RCM Script

```
#! /usr/bin/perl -w
```

```
#
# A sample site customization RCM script.
#
# When RCM_ENV_FORCE is FALSE this script indicates to RCM that it cannot
# release the tape drive when the tape drive is being used for backup.
#
# When RCM_ENV_FORCE is TRUE this script allows DR removing a tape drive
# when the tape drive is being used for backup by killing the tape
# backup application.
#

use strict;

my ($cmd, %dispatch);
$cmd = shift(@ARGV);
# dispatch table for RCM commands
%dispatch = (
"scriptinfo"    =>      \&do_scriptinfo,
"register"      =>      \&do_register,
"resourceinfo"  =>      \&do_resourceinfo,
"queryremove"   =>      \&do_preremove,
"preremove"     =>      \&do_preremove
);


if (defined($dispatch{$cmd})) {
&{$dispatch{$cmd}};
} else {
exit (2);
}

sub do_scriptinfo
{
print "rcm_script_version=1\n";
print "rcm_script_func_info=Tape backup appl script for DR\n";
exit (0);
}

sub do_register
{
my ($dir, $f, $errmsg);

$dir = opendir(RMT, "/dev/rmt");
if (!$dir) {
$errmsg = "Unable to open /dev/rmt directory: $!";
print "rcm_failure_reason=$errmsg\n";
exit (1);
}

while ($f = readdir(RMT)) {
# ignore hidden files and multiple names for the same device
if (($f !~ /^\./) && ($f =~ /^[0-9]+$/)) {
print "rcm_resource_name=/dev/rmt/$f\n";
}

}

closedir(RMT);
```

```
exit (0);
}
sub do_resourceinfo
{
my ($rsrc, $unit);

$rsrc = shift(@ARGV);
if ($rsrc =~ /^\/dev\/rmt\/([0-9]+)$/) {
$unit = $1;
print "rcm_resource_usage_info=Backup Tape Unit Number $unit\n";
exit (0);
} else {
print "rcm_failure_reason=Unknown tape device!\n";
exit (1);
}
}

sub do_preremove
{
my ($rsrc);

$rsrc = shift(@ARGV);

# check if backup application is using this resource
#if (the backup application is not running on $rsrc) {
# allow the DR to continue
#        exit (0);
#}
#
# If RCM_ENV_FORCE is FALSE deny the operation.
# If RCM_ENV_FORCE is TRUE kill the backup application in order
# to allow the DR operation to proceed
#
if ($ENV{RCM_ENV_FORCE} eq 'TRUE') {
if ($cmd eq 'preremove') {
# kill the tape backup application
}
exit (0);
} else {
#
# indicate that the tape drive can not be released
# since the device is being used for backup by the
# tape backup application
#
print "rcm_failure_reason=tape backup in progress pid=...\n"
;
exit (3);
}
}
```

♦ ♦ ♦ CHAPTER 3

Using InfiniBand Devices

This chapter provides general overview and step-by-step instructions for using InfiniBand (IB) devices in your network. The following topics are covered:

For general information about dynamic reconfiguration and hot-plugging, see Chapter 2, "Dynamically Configuring Devices".

About InfiniBand Devices

InfiniBand (IB) is an I/O technology based on switched fabrics. It provides a high bandwidth and low latency interconnect for attaching I/O devices to hosts and for host-to-host communication. IB devices are managed by the Solaris IB nexus driver.

Oracle Solaris supports the following devices:

- IP over IB (IPoIB) devices – Enables transporting the IP packets over IB connections. This feature is implemented by the `ibp` driver. See the `ibp`(7D) man page for details.
- Ethernet over IB (EoIB) - Enables transporting Ethernet frames over IB fabric.
- Socket Direct Protocol (SDP) – Supports sockets over IB
- Reliable Datagram Service version 3 (RDSv3) and Reliable Datagram Service (RDS)
- NFS over Remote Direct Memory Access (NFSoRDMA) - Provides NFS services over IB using RDMA
- iSCSI Extension for RDMA (iSER) – Provides RDMA data transfer capability to the iSCSI protocol
- User Direct Access Programming Language (uDAPL)

- Open Fabric User Verb (OFUV)

The IB nexus driver queries the Solaris IB Device Manager (IBDM) for *communication services* to enumerate the IB Port, HCA_SVC, and IB VPPA devices, where VPPA refers to a virtual physical point of attachment.

The IB partition link represents a new *part* class of data link and is managed by using `dladm` subcommands. The partition links are used for data transfers. You can create an IB partition link on top of an IB physical link, one per each partition key (`P_Key`) on the port.

Port devices bind a communication service to a specific port of a Host Channel Adapter (HCA). A port is represented by a port number. HCA_SVC devices bind a communication service to a specific HCA. VPPA devices bind a communication service to a combination of a port and a partition key (`P_key`). Note that Port devices and HCA_SVC devices always use a partition key whose value is zero. Port, HCA_SVC, and VPPA devices are children of the HCA and are enumerated through the `ib.conf` file. For more information, see the `ib`(7D) man page.

Input Output Controller (IOC) devices are children of the IB nexus driver and are part of an I/O unit. Pseudo devices are also children of the IB nexus driver and refer to all other devices that provide their own configuration files to enumerate. For more information, see the `ib`(4) man page.

The following list shows possible IB device tree path names.

- IOC device – `/ib/ioc@1730000007F510C,173000007F50`
- IB pseudo device – `/ib/`*driver*`@`*unit-address*
- IB VPPA device – `/pci@1f,2000/pci@1/pci15b3,5a44@0/ibport@,`*port#*`,`*P_key*`,`*service*
- IB HCA_SVC device –
- IB Port device – `/pci@1f,2000/pci@1/pci15b3,5a44@0/ibport@<port#>,0,`*service*
- HCA – `/pci@1f,2000/pci@1/pci15b3,5a44@0`

Note - For IB HCA_SVC devices, the port number and the value of *P_key* is zero.

The following list describes some of the IB components that are part of the path names.

services — A communication service. For example, `ipib` is the communication service used by the `ibd` kernel client driver.

P_key — The partition link key value being used.

port — The port number.

unit-address — The IB kernel client driver's property by this name as specified in its `driver.conf` file. For more information, see `driver.conf`(4).

EoIB uses the services provided by the Oracle Solaris IB framework. However, EoIB is not a child of the IB nexus driver. Oracle Solaris EoIB is a single-instance pseudo device driver with a device tree path of `/pseudo/eib@0`. An EoIB datalink is a special `eoib` class of virtual links created and managed by `dladm` subcommands. You can create an EoIB datalink over an existing IB physical link, one for each EoIB Gateway discovered on the IB fabric from that IB physical link. To display a list of discovered EoIB Gateways, use the `dladm` subcommands.

For information about using IB diagnostic commands and utilities, see "Monitoring and Troubleshooting IB Devices" on page 85.

InfiniBand Software Packages

The IB related software packages are as follows:

- `driver/infiniband/connectx` – Mellanox ConnectX Family InfiniBand HCA and 10GbE NIC drivers
- `system/io/infiniband/ethernet-over-ib` – Network device driver implementing Ethernet over InfiniBand
- `system/io/infiniband/ib-device-mgt-agent` – InfiniBand Device Manager Agent
- `system/io/infiniband/ib-sockets-direct` – InfiniBand layered Sockets Direct Protocol
- `system/io/infiniband/ip-over-ib` – Network device driver supporting the IP over InfiniBand (IPoIB) protocol
- `system/io/infiniband/open-fabrics` – Open Fabrics kernel components
- `system/io/infiniband/reliable-datagram-sockets-v3` – Reliable Datagram Sockets (RDSv3)
- `system/io/infiniband/reliable-datagram-sockets` – Reliable Datagram Sockets
- `system/io/infiniband/rpc-over-rdma` – InfiniBand RPC over RDMA Driver
- `system/io/infiniband/udapl` – UDAPL library and commands
- `system/io/infiniband` – InfiniBand Framework

Dynamically Reconfiguring IB Devices

You can configure or unconfigure an IB device on a system by using the `cfgadm` command. The command manages dynamic reconfiguration (DR) of the entire IB fabric as seen by a host. In addition, the command enables you to do the following:

- Display the IB fabric
- Manage communication services

- Update `P_key` table databases

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

All the IB devices, such as Port, VPPA, HCA_SVC, IOC, and pseudo devices support `cfgadm` operations.

Note - When performing configuration procedures on IB devices, you must have the appropriate administrative privileges. For more information, see "Using Your Assigned Administrative Rights" in "Securing Users and Processes in Oracle Solaris 11.2 ".

Displaying IB Device Information

To display IB device information, use the following command syntax:

```
# cfgadm -[a]l
```

where the `-a` option adds more information that is displayed than by using the `-l` option alone.

In the following sample output, information is truncated to show relevant IB information only.

```
# cfgadm -al
Ap_Id                       Type         Receptacle     Occupant     Condition
ib                          IB-Fabric    connected      configured   ok
hca:21280001A0A478          IB-HCA       connected      configured   ok
ib::21280001A0A47A,0,ipib   IB-PORT      connected      configured   ok
ib::21280001A0A479,0,ipib   IB-PORT      connected      configured   ok
ib::1730000008070,0,hnfs    IB-HCA_SVC   connected      configured   ok
ib::daplt,0                 IB-PSEUDO    connected      configured   ok
ib::iser,0                  IB-PSEUDO    connected      configured   ok
ib::rdsib,0                 IB-PSEUDO    connected      configured   ok
ib::rdsv3,0                 IB-PSEUDO    connected      configured   ok
ib::rpcib,0                 IB-PSEUDO    connected      configured   ok
ib::sdpib,0                 IB-PSEUDO    connected      configured   ok
ib::sol_umad,0              IB-PSEUDO    connected      configured   ok
ib::sol_uverbs,0            IB-PSEUDO    connected      configured   ok
```

The `cfgadm` command displays information about attachment points, which are locations in the system where DR operations can occur.

For more information about attachment points, see "Attachment Points" on page 32. For more information about attachment points supported by the `cfgadm` command, see the `cfgadm_ib`(1M) man page.

An attachment point consists of a receptacle and an occupant. The following table describes the state of a device depending on the corresponding combined states of the receptacle, occupant, and condition:

Receptacle/Occupant/Condition Combined State	Description
`connected/configured/ok`	The device is connected and available. The `devinfo` node is present.
`connected/unconfigured/unknown`	The device is unavailable and no `devinfo` node or device driver exists for this device. Or, the device was never configured for use by `ib` nexus driver. The device might be known to the IB Device Manager.

Devices are listed by their attachment point IDs (`Ap_Id`). Note that all IB `Ap_Id`s are shown as `connected`. The following list explains the information for a specific `Ap_Id` in the sample output.

`ib::21280001A0A47A,0,ipib` Identifies an IB port device that is connected to port GUID and is bound to the `ipib` service.

`ib::sdpib,0` Identifies a pseudo device.

`hca:21280001A0A478` Identifies an HCA device.

`ib::1730000008070,0,hnfs` Identifies an IB HCA_SVC device that is bound to the `hnfs` service.

`ib::ibgen,0` Identifies a pseudo device.

You can use other options of the `cfgadm` command to further customize the information to be displayed. The following examples show the use of selected options.

EXAMPLE 3-1 Displaying Information About a Port Device

This example shows information about the port device `ib::21280001A0A47A,0,ipib`.

```
# cfgadm -al -s "cols=ap_id:info" ib::21280001A0A47A,0,ipib
Ap_Id                          Information
ib::21280001A0A47A,0,ipib      ipib
```

EXAMPLE 3-2 Displaying Port and GUID Information of an HCA Device

This example shows the number of ports and their respective GUIDs for the HCA device `hca::1730000008070`.

```
# cfgadm -al -s "cols=ap_id:info" hca::1730000008070
Ap_Id                          Information
hca::1730000008070             VID: 0x15b3, PID: 0x5a44, #ports: 0x2,
port1 GUID: 0x1730000008071, port2 GUID: 0x1730000008072
```

EXAMPLE 3-3 Displaying Kernel Clients of an HCA Device

This example shows the kernel clients of the HCA `hca:173000007F50`.

```
$ cfgadm -x list_clients hca:173000007F50
Ap_Id                       IB Client         Alternate HCA
ib::1730000007F51D0            ibgen              no
ib::1730000007F51D1            ibgen              no
ib::1730000007F51,0,ipib       ibd                no
ib::ibgen,0                    ibgen              no
-                              ibdm               no
-                              ibmf               no
-                              nfs/ib             no
```

In the sample output, if a kernel IB client uses an HCA other than `hca:173000007F50`, the entry under the column `Alternate HCA` would indicate `yes`. If IB Managers and kernel clients that do not use the HCA, their `Ap_Id`s would not be listed. Moreover, although EoIB uses the Solaris IB framework, the command does not list EoIB as a kernel IB client.

EXAMPLE 3-4 Displaying Supported Communication Services

The following example lists IB communication services currently used by the InfiniBand Transport Framework (IBTF).

```
# cfgadm -x list_services ib
Port communication services:
srp
VPPA communication services:
ibd
HCA_SVC communication services:
hnfs
```

Using the `prtconf` Command

You can also use the `prtconf` command to display general information about IB devices. In the following example, `pci15b3,673c` refers to an IB HCA.

```
$ prtconf
.
.
.
ib, instance #0
rpcib, instance #0
rdsib, instance #0
daplt, instance #0
rdsv3, instance #0
sdpib, instance #0
sol_umad, instance #0
sol_uverbs, instance #0
iser, instance #0
```

```
.
.
.
pci15b3,673c, instance #0
ibport, instance #0
ibport, instance #1
```

Configuring or Unconfiguring IB Devices

You use the following commands to configure or unconfigure IB devices. The commands apply to IB port, HCA_SVC, or VPPA devices.

- `cfgadm -c configure` *device* configures the IB device.
- `cfgadm -c unconfigure` *device* unconfigures the IB device.

In both commands, the *device* is represented by its `Ap_Id`.

Note - To perform DR on HCAs, you use the bus-specific `cfgadm` plugin such as the `cfgadm_pci` command for a PCI based HCA. However, the procedures are beyond the scope of this documentation. For more information, refer to the appropriate man page, for example, the `cfgadm_pci`(1M) man page.

As a useful practice, you should display information about the system devices before and after the actual step of configuring or unconfiguring a device. The information helps you in the following ways:

- Properly identify the bus or device to dynamically reconfigure.
- Verify from the state of the bus or device that the reconfiguration completed successfully.

The following examples show how to use the `cfgadm` command to perform DR on IB devices.

EXAMPLE 3-5 Configuring an IB Port Device

The following example configures the IB port device `ib::1730000007F51,*0*,ipib` and includes verification of the operation.

```
# cfgadm -c configure ib::1730000007F51,*0*,ipib
# cfgadm -a ib::1730000007F51,*0*,ipib
Ap_Id                      Type      Receptacle Occupant   Condition
ib::1730000007F51,*0*,ipib  IB-Port   connected  configured ok     The port device is configured.
```

Note that if EoIB datalinks were configured over the IB Port when the port was previously unconfigured, you must re-create the vnics corresponding to those EoIB datalinks on the

EoIB Gateway with the new IB HCA port GUIDs. You must re-create the vnics before re-configuring the IB Port. For information about how to create vnics on the Gateway, refer to the Sun Network QDR InfiniBand Gateway Switch Administration Guide (http://docs.oracle.com/cd/E36256_01/pdf/E36260.pdf).

EXAMPLE 3-6 Unconfiguring an IB Port Device

The following example unconfigures the IB port device `ib::1730000007F51,*0*,ipib` and includes verification of the operation.

```
# cfgadm -c unconfigure ib::1730000007F51,*0*,ipib
Unconfigure the device: /devices/ib:fabric::1730000007F51,*0*,ipib
This operation will suspend activity on the IB device
Continue (yes/no)? Y
# cfgadm -a ib::1730000007F51,*0*,ipib
Ap_Id                      Type    Receptacle Occupant     Condition
ib::1730000007F51,*0*,ipib IB-Port  connected  unconfigured unknown     The port device is unconfigured.
```

EXAMPLE 3-7 Configuring an IB Pseudo Device

The following example configures the pseudo device `ib::ibgen,0` and includes verification of the operation.

```
# cfgadm -yc configure ib::ibgen,0
# cfgadm -a ib::ibgen,0
Ap_Id                 Type       Receptacle Occupant   Condition
ib::ibgen,0           IB-PSEUDO  connected  configured ok     The device is configured.
```

EXAMPLE 3-8 Unconfiguring an IB Pseudo Device

The following example unconfigures the pseudo device `ib::ibgen,0` and includes verification of the operation.

```
# cfgadm -c unconfigure ib::ibgen,0
Unconfigure the device: /devices/ib:fabric::ibgen,0
This operation will suspend activity on the IB device
Continue (yes/no)? Y
# cfgadm -a ib::ibgen,0
Ap_Id                 Type      Receptacle Occupant     Condition
ib::ibgen,0           IB-PSEUDO connected  unconfigured unknown     The device is unconfigured.
```

Modifying IB Configurations

This section provides examples of the use of the following `cfgadm` command options to modify existing IB configurations:

- `cfgadm -o` *hardware-options*
- `cfgadm -x` *hardware-functions* (also used to customize information display. See "Displaying IB Device Information" on page 68.)

EXAMPLE 3-9 Updating the IB `P_key` Tables

When you enable additional `P_keys` or disable them, the `P_key` table information of an HCA's ports changes. For consistency, corresponding updates must also occur in the internal `P_key` databases of the InfiniBand Transport Framework (IBTF) and IBDM.

The following example shows how to update the `P_key` databases of IBTF and IBDM.

```
# cfgadm -x update_pkey_tbls -y ib
```

For more information, see the `ibtl`(7D) and `ibdm`(7D) man pages.

EXAMPLE 3-10 Adding a Communication Service

The following example adds a VPPA communication service to the IB device and also includes verification of the operation.

```
# cfgadm -o comm=vppa,service=new -x add_service ib
# cfgadm -x list_services ib
Port communication services:
srp
VPPA communication services:
ibd
new      The service has been added.
HCA_SVC communication services:
nfs_service
```

EXAMPLE 3-11 Remove an Existing Communication Service

This example removes the communication service that was added in the previous example. The name of the communication service is `new`. After the removal, the completion of the operation is verified.

```
# cfgadm -o comm=vppa,service=new -x delete_service ib
# cfgadm -x list_services ib
Port communication services:
srp
VPPA communication services:
ibd
HCA_SVC communication services:
hnfs
```

Using the uDAPL API With InfiniBand Devices

User Direct Access Programming Library (uDAPL) is a standard API that enhances performance of data center applications for data messaging and provides scalability and reliability of Remote Direct Memory Access (RDMA) capable interconnections such as InfiniBand. The uDAPL interface is defined by the DAT Collaborative organization (http://www.datcollaborative.org).

uDAPL features in Oracle Solaris include the following:

- A standard DAT registry library, `libdat`. For more information, see the `libdat`(3LIB) man page.
- A standard service provider registration file, `dat.conf`. For more information, see the `dat.conf`(4) man page.
- Support for multiple service providers so that each provider specifies its own uDAPL library path, version number, and so on, in its `service_provider.conf` file. For more information, see, the `service_provider.conf`(4) man page.
- An administrative tool, the `datadm` command, to configure `dat.conf`. For more information, see the `datadm`(1M) man page.
- A new resource control property, `project.max-device-locked-memory`, to regulate the amount of locked down physical memory.
- A naming scheme that uses either IPv4 or IPv6 addresses that leverage the IP infrastructure, such as ARP in IPv4 and neighbor discovery in IPv6, for address resolution. The Solaris uDAPL Interface Adapter directly maps to an IPoIB device instance.
- Support for the standard Address Translation Scheme that is used by the DAT collaborative community.
- A uDAPL service provider library to support the hermon Host Channel Adapter with automatic registration to the `dat.conf` registration file. For more information, see the `hermon`(7D) man page.
- Support for both the SPARC platform and x86 platforms.

▼ How to Enable uDAPL

1. **Become an administrator.**
2. **(Optional) Verify that the following packages are installed:**
   - Mellanox ConnectX Family InfiniBand HCA and 10GbE NIC drivers
   - Network device driver supporting the IP over InfiniBand (IPoIB) protocol

- User Direct Access Programming Library (UDAPL) Direct Access Transport (DAT) facility

By default, these packages are included in a normal Oracle Solaris installation. Verification should generate the following output:

```
$ pkg verify -v connectx ip-over-ib udapl
PACKAGE                                                 STATUS
pkg://solaris/driver/infiniband/connectx                    OK
pkg://solaris/system/io/infiniband/ip-over-ib               OK
pkg://solaris/system/io/infiniband/udapl                    OK
```

3. Create the IPoIB interface.

```
# ipadm create-ip name
# ipadm create-addr -a address [address-object]
```

name — Name of the IPoIB interface.

address — A valid IP address. By default, this address is assumed to be a static address.

address-object — A name that identifies the IP address in association with the IP interface. If *address-object* is not specified, the OS automatically assigns a name using the format *ip-name*/*protocol*.

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

4. Enumerate device entries in the service provider's configuration file into a list of interface adapters in the registry.

```
# datadm -a serviceprovider.conf
```

where *serviceprovider.conf* is the configuration file from the service provider. The file contains information about the device types that the provider supports. For more information, see the `service_provider.conf`(4) man page.

Example 3-12 Enabling uDAPL After IPoIB Interface Configuration

The following example configures `ibd1` and updates the `dat.conf` registry with entries from the configuration file `ABCDudaplt.conf`.

In the interface configuration, the address object is automatically assigned.

```
# ipadm create-ip ibd1
# ipadm create-addr -a 192.168.0.1/24
# ipadm show-addr
ADDROJB       TYPE      STATE   ADDR
ibd1/v4       static    ok      192.168.0.1/24
...
```

```
# datadm -a /usr/share/dat/ABCDudaplt.conf
```

Updating the DAT Static Registry

Updating the `dat.conf` file occurs when IPoIB changes occur on the system. You update the registry in two ways.

You can add or remove a list of adapters from the registry that correspond to a device entry in the service provider's configuration file. For this type of updating, you use one of the following commands:

- `datadm -a` *serviceprovider.conf* adds a list to the registry. For example, to add a list of adapters from a device entry in `ABCDudaplt.conf`, type the following commands. The second command displays the updated registry.

  ```
  # datadm -a /usr/share/dat/ABCDudaplt.conf
  # datadm -v
  ```

- `datadm -r` *serviceprovider.conf* removes a list from the registry. For example, to remove a list of adapters that correspond to a device entry in `ABCDudapl.conf`, type the following commands. The second command displays the updated registry.

  ```
  # datadm -r /usr/share/dat/ABCDudaplt.conf
  # datadm -v
  ```

You can also Make the registry reflect the current state of the system with an up to date list of interface adapters for the service providers that are listed in the registry. For this type of updating, you use the following command:

```
datadm -u
```

For more information, see the `service_provider.conf`(4) man page.

Administering IPoIB Devices

For every HCA, one physical data link per port is created by default. The physical links can be used as administrative and observability data points. You can create IB partition links over the physical data links, similar to a creating VNICs over a NIC. Physical data links are not used for data transfers, thus, plumbing and assigning IP addresses is not supported on these links. Data is transferred on the partition data links.

For information about configuring your network components, see “Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.2 ”.

Displaying Physical Data Link Information

The physical link state directly corresponds to the IB HCA port state. To display physical data link information, use the following commands:

- `dladm show-phys [`*link*`]`

  If you do not specify *link*, then information about all the links is displayed.

  This command displays basic information about links on the system such as the media type, the current state, and the speed capacity. The same command is used to display information about links of other media types.

  The following examples have truncated outputs to show information relevant to InfiniBand only.

```
# dladm show-phys
LINK     MEDIA        STATE      SPEED     DUPLEX     DEVICE
ibp0    Infiniband     up        8000      unknown    ibp0
ibp1    Infiniband    down       8000      unknown    ibp1
ibp2    Infiniband    down       8000      unknown    ibp2
ibp3    Infiniband     up        8000      unknown    ibp3
```

- `dladm show-ib [`*link*`]`

  If you do not specify *link*, then information about all the links is displayed.

  This command displays the physical links, port GUIDs, port# HCA GUIDs, `P_Keys` present on the port at the time the command is running, and the EoIB gateways that are discovered from each IB port.

  The following example shows four gateway Ethernet ports discovered from the IB HCA port `ibp1`. All four are on the gateway `nm2gw-1`.

```
# dladm show-ib
LINK    HCAGUID        PORTGUID       PORT STATE GWNAME  GWPORT   PKEYS
ibp0    212800013F2F5A 212800013F2F5B 1    down  --      --       FFFF
ibp1    212800013F2F5A 212800013F2F5C 2    up    nm2gw-1 0a-eth-1 FFFF,8001
                                                 nm2gw-1 0a-eth-2
                                                 nm2gw-1 0a-eth-3
                                                 nm2gw-1 0a-eth-4
```

▼ How to Create IB Partition Links

One or more IB partition data links can be created on the top of the IB physical links with the same `P_Key`. The partition data links are used for data transfers.

1. **Become an administrator.**

2. Create new IB partition links.

```
# dladm create-part -l link -P key partition
```

link Name of the physical data link.

key Value of the partition key (`P_key`).

partition Name of the partition link and using the format *P_key*.*link*

The command assumes that the port is up, the `P_Key` is present on the port, and IPoIB is successfully initialized. If these three conditions do not exist, the command fails.

Note - Even if the conditions do not exist, you can still create a partition link by using the `-f` option. Suppose that the partition key is not configured on the port or the port is marked as `down`. To create the partition link, type the following:

```
# dladm create-part -f -l link -P key name
```

In this case, the link state transitions to `up` when the partition key is added to the port and the port is activated.

---

3. Display the IB partition link information.

```
# dladm show-part
```

4. Plumb and assign an IP address to an IB partition link.

```
# ipadm create-ip name
# ipadm create-addr -a address [address-object]
```

name Name of the partition link.

address A valid IP address. By default, this address is assumed to be a static address.

address-object A name that identifies the IP address in association with the IP interface. If *address-object* is not specified, the OS automatically assigns a name using the format *ip-name*/*protocol*

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

Example 3-13 Configuring a Partition Link

This example show how to perform the following task:

- Create partition links over the physical links `ibp0` and `ibp2`
- Configure an IPoIB interface is over `ibp2`.

The values of `P_key` are `0x8001` for `ibp0` and `0x9000` for `ibp2`.

```
# dladm create-part -l ibp0 -P 0x8001 p8001.ibp0
# dladm create-part -l ibp0 -P 0x9000 p9000.ibp2

# dladm show-part
LINK           P_Key      OVER       STATE      FLAGS
p8000.ibp0       8001      ibp0      unknown     ----
p9000.ibp2       9000      ibp2      unknown     ----

# ipadm create-ip p9000.ibp2
# ipadm create-addr -a 1.1.1.1 p9000.ibp2/ipv4
# ipadm show-addr
ADDROJB           TYPE      STATE   ADDR
p9000.ibp2/ipv4   static    ok      1.1.1.1/24

# dladm show-part
LINK           P_Key      OVER       STATE      FLAGS
p8000.ibp0       8001      ibp0      unknown     ----
p9000.ibp2       9000      ibp2           up     ----
```

▼ How to Remove an IB Partition Link

Before You Begin Make sure that no IPoIB interface is configured over the partition link. If an interface exists, remove it first.

1. Become an administrator.

2. Remove an IB partition link.

```
# dladm delete-part partition
```

In this example, the partition link `p8001.ibp0` is removed after ensuring that it has no existing interface.

```
# dladm delete-part p8001.ibp0
```

3. Review the partition link information.

The following example shows the remaining partition link on the system after `p8001.ibp0` is removed.

```
# dldam show-part
LINK        P_Key      OVER     STATE      FLAGS
p9000.ibp2   9000      ibp2      down      f---
```

Administering EoIB Datalinks

The per IB-port physical datalinks described in “Administering IPoIB Devices” on page 76 can also be used as administrative end points for EoIB datalinks. On Oracle Solaris 11.2, you can create an EoIB datalink as a special `eoib` class object over an IB physical link using the `dladm create-eoib` command. These `eoib` class EoIB datalinks can host Ethernet data transfers just as regular Ethernet datalinks do. Additionally, you can also build VNICs and VLANs over these EoIB datalinks by using the `dladm create-vnic` and `dladm create-vlan` commands.

Note - EoIB datalinks can be created arbitrarily on Oracle Solaris. However, for data transfers to be successful, a Sun Network QDR InfiniBand Gateway Switch must be properly configured on the InfiniBand fabric. For information about configuring the gateway, see the Sun Network QDR InfiniBand Gateway Switch Administration Guide (http://docs.oracle.com/cd/E36256_01/pdf/E36260.pdf).

For information about configuring the network components, see “Configuring and Administering Network Components in Oracle Solaris 11.2 ”.

▼ How to Create and Configure an EoIB Datalink

You can create EoIB datalinks over IB physical links on the system, one for each `GWPORT` discovered on the HCA port corresponding to the IB physical link. You can configure IP, VNIC, and VLAN interfaces over these EoIB datalinks and use them for data transfer, similar to the way regular Ethernet datalinks are used.

1. **Become an administrator.**

2. **Display information about a chosen IB physical link.**

   ```
   # dladm show-ib link
   ```

   If the output shows that the link's port is down, you can still proceed with the datalink creation and configuration. Then when the causes of the down state are resolved, the datalink will automatically transition to the up state. For more information about the information that this command displays, see “Displaying Physical Data Link Information” on page 77.

3. **Create a new EoIB datalink.**

   ```
   # dladm create-eoib -l link -g gway -c gway-port eoib-link
   ```

   link Name of the physical link.

gway Gateway name.

gway-port Gateway Ethernet port.

eoib-link Name of the EoIB datalink.

The command creates an EoIB datalink and binds the host IB port to the Ethernet port on the EoIB gateway.

4. **Create an IP interface over the datalink and assign an IP address to the interface.**

```
# ipadm create-ip name
# ipadm create-addr -a address [address-object]
```

name Name of the EoIB interface.

address A valid IP address. By default, this address is assumed to be a static address.

address-object A name that identifies the IP address in association with the IP interface. If *address-object* is not specified, the OS automatically assigns a name using the format *name*/*protocol*

5. **(Optional) Display the interface information.**

```
# ipadm show-addr
```

6. **Display EoIB datalink information.**

In the following two commands, the second shows more IB related information about the link than the first command that shows general information.

```
# dladm show-link [eoib-link]

# dladm show-eoib [eoib-link]
```

A typical output of the `dladm show-eoib` command would be similar to the following:

```
LINK    GWNAME  GWPORT   GWID FLAGS  SPEED MACADDRESS     OVER
eib1    nm2gw-1 0a-eth-2 1A8  aHnU-- 10000 0:25:8b:60:2:3 ibp1
```

The GWID value `1A8` is the unique identifier associated with the gateway and Ethernet port `{nm2gw-1, 0a-eth-2}` on the IB fabric. The MAC address is the address specified as a parameter when the vnic associated with this datalink was created on the gateway. For more detailed explanation of the meaning of the rest of the information provided in the output, refer to the `dladm`(1M) man page.

Note - The vnic in this context refers to the entity created on the gateway switch using the `createvnic` gateway command. This vnic is different from the entity that is created with Oracle Solaris `dladm` subcommands.

For details about the `createvnic` command, see the Sun Network QDR InfiniBand Gateway Switch Command Reference (http://docs.oracle.com/cd/E36256_01/pdf/E36263.pdf).

Example 3-14 Creating and Configuring an EoIB Datalink

In this example, an EoIB link is configured over `ibp1`.

```
# dladm show-ib ibp1
LINK    HCAGUID        PORTGUID       PORT STATE GWNAME  GWPORT   PKEYS
ibp1    212800013F2F5A 212800013F2F5C 1    up    nm2gw-1 0a-eth-1 FFFF,8001
                                                   nm2gw-1 0a-eth-2
                                                   nm2gw-1 0a-eth-3
                                                   nm2gw-1 0a-eth-4

# dladm create-eoib -l ibp1 -g nm2gw-1 -c 0a-eth-2 eib1

# ipadm create-ip eib1
# ipadm create-addr -a=192.168.99.142/24
# ipadm show-addr eib1
ADDROBJ   STATE    TYPE   ADDR
eib1/v4   static   ok     192.168.99.142.24

# dladm show-link eib1
LINK    CLASS  MTU   STATE  OVER
eib1    eoib   1500  up     ibp1

# dladm show-eoib eib1
LINK    GWNAME  GWPORT   GWID FLAGS  SPEED MACADDRESS     OVER
eib1    nm2gw-1 0a-eth-2 1A8  aHnU-- 10000 0:25:8b:60:2:3 ibp1
```

▼ How to Remove an EoIB Datalink

You can remove an existing EoIB datalink that has no IP interfaces, VLANs, or VNICs currently built over it. The following procedure removes the EoIB datalink `eib1` created in the previous procedure.

1. **Become an administrator.**

2. **Identify the EoIB datalink to remove.**

```
# dladm show-eoib
```

3. **Verify that no VNICs, VLANs, and IP interfaces are built over the EoIB datalink.**

```
# dladm show-vlan link
# dladm show-vnic -l link
# dladm show-if interface
```

where *link* is the EoIB datalink you want to remove and *interface* is the IP interface that might be configured over the EoIB datalink. Typically, the interface name is identical with the datalink name over which the interface is created.

4. **Remove any interfaces built over the EoIB datalink.**

```
# ipadm delete-ip interface
```

5. **Remove the EoIB datalink.**

```
# dladm delete-eoib link
```

6. **Verify that the EoIB datalink is removed.**

```
# dladm show-eoib link
```

Example 3-15 Removing an EoIB Datalink

In this example, the IP interface is first removed from the datalink `eib1` before the datalink itself is removed. No VLANs nor VNICs are configured over the link.

```
# dladm show-eoib eib1
LINK    GWNAME  GWPORT   GWID FLAGS  SPEED MACADDRESS     OVER
eib1    nm2gw-1 0a-eth-2 1A8  aHnU-- 10000 0:25:8b:60:2:3 ibp1

# dladm show-vlan eib1
dladm: failed to show vlan eib1: object not found
# dladm show-vnic -l eib1
     no output generated
# ipadm show-if eib1
IFNAME  CLASS  STATE  ACTIVE  OVER
eib1    ip     ok     yes     --

# ipadm delete-ip eib1
# dladm delete-eoib eib1
# dladm show-eoib eib1
dladm: non-existent datalink 'eib1'
```

How to Restore an EoIB Datalink That Failed During Migration

EoIB datalinks in previous Oracle Solaris releases existed as `phys` class datalinks similar to regular Ethernet datalinks. If you upgrade to Oracle Solaris 11.2 , existing EoIB datalinks on the system are automatically migrated to the new `eoib` class datalinks.

If you also plan to upgrade the gateway firmware, upgrade the OS first to ensure the proper migration of the Oracle Solaris EoIB instances to the new datalink class. Then, upgrade the gateway firmware.

Note that automatic migration only partially succeeds if the original gateway Ethernet port instances corresponding to the existing `phys` class datalinks are unavailable at the time of the upgrade. You must complete the migration manually.

1. **Become an administrator.**

2. **Identify any EoIB datalink that has not been fully migrated.**

   ```
   # dladm show-eoib
   ```

   Information would be missing for datalinks that have not successfully migrated, such as in the following example:

   ```
   # dladm show-eoib
   LINK   GWNAME GWPORT GWID FLAGS  SPEED MACADDRESS  OVER
   eib2   ?      ?      1A8  ------ 10000 0:0:0:0:0:0 ibp1
   ```

3. **Determine the gateway system name and Ethernet port corresponding to the GWID.**

   The gateway/IB fabric administrator can determine this information from knowledge of the configuration and by using appropriate tools on the gateway.

4. **Delete any existing VNICs, VLANs, or IP interfaces that might have been configured over the datalink.**

   Use the appropriate `dladm` and `ipadm` subcommands to remove existing VNICs, VLANs, and IP interfaces.

5. **Delete the partially migrated EoIB datalink.**

   ```
   # dladm delete-eoib link
   ```

6. **Recreate the EoIB datalink with the correct information.**

   ```
   # dladm create-eoib -l link -g gway -c gway-port eoib-link
   ```

link	Name of the physical link.
gway	Gateway name.
gway-port	Gateway Ethernet port.
eoib-link	Name of the EoIB datalink.

7. Recreate the VNICs, VLANs, or IP interfaces that you might have previously removed in Step 4.

Use the appropriate `dladm` and `ipadm` subcommands to recreate deleted VNICs, VLANs, and IP interfaces.

Example 3-16 Restoring an EoIB Datalink That Failed Migration

In this example, `eib2` did not completely migrate. You have determined that `eib2` had the following configurations prior to the migration:

- `eib2` was configured over the physical link `ibp1`.
- The GWID is `1A8`.
- The gateway name is `nm2gw-2`.
- The gateway port is `0a-eth-1`.

After determining that all existing configurations over `eib2` have been deleted, you proceed as follows:

```
# dladm delete-eoib eib2
# dladm create-eoib -l ibp1 -g nm2gw-2 -c 0a-eth-1 eib2
```

Monitoring and Troubleshooting IB Devices

In the Oracle Solaris 11 release, new commands and utilities enable you to manage the IB fabric more effectively. These commands are included in the `system/io/infiniband/open-fabrics` package and the man pages are installed automatically when the `open-fabrics` package is installed. For example:

```
% man rping
Reformatting page.  Please Wait... done

librdmacm                                              RPING(1)

NAME
rping - RDMA CM connection and RDMA ping-pong test.
```

```
SYNOPSIS
rping -s [-v] [-V] [-d] [-P] [-a address] [-p port]
[-C message_count] [-S message_size]
rping -c [-v] [-V] [-d] -a address [-p port]
[-C message_count] [-S message_size]
.
.
.
```

The following new commands and utilities provide the ability to list and query IB devices, diagnose and trouble shoot IB fabric issues, and measure IB performance.

TABLE 3-1 General IB Monitoring Commands

Command	Description
`ibv_asyncwatch`	Monitors InfiniBand asynchronous events
`ibv_devices or ibv_devinfo`	Lists InfiniBand devices or device information
`ibv_rc_pingpong`, `ibv_srq_pingpong`, or `ibv_ud_pingpong`	Tests node to node connectivity by using RC connection, SRQs, or UD connection
`mckey`	Tests RDMA CM multicast setup and simple data transfer
`rping`	Tests RDMA CM connection and attempts RDMA ping-pong
`ucmatose`	Tests RDMA CM connection and attempts simple ping-pong
`udaddy`	Tests RDMA CM datagram setup and attempts simple ping-pong

TABLE 3-2 General IB Performance Testing Commands

Command	Description
`rdma_bw or rdma_lat`	Tests RDMA write transactions for streaming bandwidth or latency.
`ib_read_bw` or `ib_read_lat`	Tests RDMA read transactions for bandwidth or latency.
`ib_send_bw` or `ib_send_lat`	Tests RDMA send transactions for bandwidth or latency.
`ib_write_bw` or `ib_write_bw_postlist`	Tests RDMA write transactions for bandwidth that displays one I/O request at a time or post list bandwidth that displays a list of I/O requests.
`ib_write_lat`	Tests RDMA write transactions for latency.
`ib_clock_test`	Tests accuracy of system clock
`qperf`	Measures socket and RDMA performance

TABLE 3-3 RDS Monitoring and Testing Tools

Command	Description
`rds-info`	Displays RDS kernel module information
`rds-ping`	Determines if remote node over RDS is reachable
`rds-stress`	Sends message between processes over RDS sockets

Note that RDSv3 does not support unconfiguring HCAs. If the system has an RDSv3 driver installed at the time of DR, unconfiguring the HCA fails with an error message such as the following example for `ib::rdsv3,0`.

```
# cfgadm -c unconfigure ib::rdsv3,0
This operation will suspend activity on the IB device
Continue (yes/no)? yes
cfgadm: Hardware specific failure: unconfigure operation
failed ap_id: /devices/ib:fabric::rdsv3,0

# cfgadm -c unconfigure PCI-EM0
cfgadm: Component system is busy, try again: unconfigure failed
```

Workaround:

Remove the RDSv3 driver and reboot the system before performing the HCA DR operation.

```
# rem_drv rdsv3
Device busy
Cannot unload module: rdsv3
Will be unloaded upon reboot.

# init 6
```

TABLE 3-4 Fabric Diagnostic Tools

Command	Description
`ibdiagnet`	Performs diagnostic check of the entire fabric
`ibaddr`	Queries InfiniBand address or addresses
`ibnetdiscover`	Discovers remote InfiniBand topology
`ibping`	Validates connectivity between IB nodes
`ibportstate`	Queries physical port state and link speed of an IB port
`ibroute`	Displays InfiniBand switch forwarding tables
`ibstat` or `ibsysstat`	Query status of InfiniBand device or devices or the status of a system on an IB address
`ibtracert`	Traces an IB path

Command	Description
`perfquery` or `saquery`	Queries IB port counters or sIB subnet administration attributes
`sminfo`	Queries IB SMInfo attribute
`smpquery` or `smpdump`	Queries or dumps IB subnet management attributes
`ibcheckerrors` or `ibcheckerrs`	Validates IB port (or node) or IB subnet and reports errors
`ibchecknet`, `ibchecknode`, or `ibcheckport`	Validates IB subnet, node, or port and reports errors
`ibcheckportstate`, `ibcheckportwidth`, `ibcheckstate`, or `ibcheckwidth`	Validates IB port that are link up but not active, ports for 1x (2.0 Gbps) link width, ports in IB subnet that are link up but not active, or lx links in IB subnet
`ibclearcountersibclearerrors` or `ibclearerrors`	Clears port counters or error counters in IB subnet
`ibdatacountersibdatacounts`, or `ibdatacounts`	Queries for data counters in IB subnet or IB port data counters
`ibdiscover.pl`	Annotates and compares IB topology
`ibhosts`	Displays IB host nodes in topology
`iblinkinfo.pl` or `iblinkinfo`	Displays link information for all links in the fabric
`ibnodes`	Displays IB nodes in topology
`ibprintca.pl`	Displays either the CA specified or the list of CAs from the `ibnetdiscover` output
`ibprintrt.pl`	Displays either only the router specified or a list of routers from the `ibnetdiscover` output
`ibprintswitch.pl`	Displays either the switch specified or a list of switches from the `ibnetdiscover` output
`ibqueryerrors.pl`	Queries and report non-zero IB port counters
`ibrouters`	Displays IB router nodes in topology
`ibstatus`	Queries basic status of IB devices
`ibswitches`	Displays IB switch nodes in topology
`ibswportwatch.pl`	Polls the counters on the specified switch or port and report rate of change information
`set_nodedesc.sh`	Sets or displays node description string for IB Host Controller Adapters (HCA)s
`dump2psl.pl`	Dumps PSL file based on `opensm` output file that is used for credit loop checking
`dump2slvl.pl`	Dumps SLVL file based on `opensm` output file that is used for credit loop checking

Command	Description
`ibis`	An extended TCL shell for IB management inband services

♦ ♦ ♦ CHAPTER 4

Managing Disks in Oracle Solaris

Disk management involves a range of tasks from creating appropriate disk slices and file systems during an Oracle Solaris installation to adding or replacing disks. This chapter provides overview information about managing disks on the system. The following topics are covered:

This chapter covers the following topics:

For instructions on how to add a disk to your system, see Chapter 7, "Setting Up ZFS on Disks".

Disk Management Features

This section describes features in Oracle Solaris for managing disks on your system.

Installing on Large Disks

You can install and boot Oracle Solaris from a disk that is up to 2 TB in size. This support applies to the following systems:

- SPARC platforms with an updated OBP. However, the SPARC boot loader remains unchanged.
- x86 platforms that use GRUB 2 as the default system boot loader.

On both system types, the two-terabyte disk must be connected to a system with a minimum of 1.5 GB of memory.

With EFI (GPT) partitioning, all of the disk space on the boot device can be used for Oracle Solaris installations.

For more information, see Chapter 2, “x86: Administering the GRand Unified Bootloader (Tasks),” in “Booting and Shutting Down Oracle Solaris 11.2 Systems ”.

Disk management utilities have been enhanced to accommodate this feature. For example, the `fdisk` utility can support up to 2-TB partitions. However, limits might be imposed on other non-EFI partitions. If you run the utility on a disk that is greater than 2TB in size, the utility warns you that you cannot create a non-EFI partition that is greater than 2 TB.

Note - You cannot move a disk over 1 TB with a legacy MBR or a legacy VTOC to a previous Oracle Solaris release. EFI labeled disks continue to work as in previous Solaris releases.

Using Whole Disks for a ZFS Root Pool Disk or Disks

The Oracle Solaris installation program can install an EFI (GPT) disk label on a ZFS root pool disk or disks by using DVD, USB, and automated installation methods. UEFI firmware support and the introduction of GRUB 2 provide the ability to boot from a GPT labeled disk. Thus, you can use whole disks for a ZFS root pool disk or disks on the following platforms:

- SPARC based systems with GPT enabled firmware
- Most x86 based systems

Note - For more information about disk labels, see “EFI (GPT) Disk Label” on page 94.

On SPARC based systems, the root file system is contained in slice 0. On x86 based systems, the root file system is contained in partition 1.

The `zpool` command can support EFI (GPT) labels. To recreate a root pool or create an alternate root pool, use the `zpool create -B` command. The command option creates the required slices and information for booting. If you use the `zpool replace` command to replace a disk in a root pool that has an EFI (GPT) labeled disk, you must also reinstall the boot loader.

Using Advanced Format Disks

Oracle Solaris can support large capacity disks, also known as advanced format (AF) disks. AF disks are hard disk drives that exceed the traditional 512-byte block size that previous releases support.

AF disks are generally in the 4-KB block size range, but vary as follows:

- A 4-KB native disk (4kn) has a physical and logical block size of 4 KB
- A 512-byte emulation (512e) has a physical block size of 4 KB but reports a logical block size of 512 bytes

Oracle Solaris also supports the 512-byte native (512n) disk, which is a traditional disk with 512-byte block size.

Before purchasing advanced format drives, confirm with your device manufacturer that their 512e devices have a power-safe feature. This feature prevents data loss after a power failure while data is still in transit.

To determine if your system has AFD-supported disks, use the `devprop` command.

```
# devprop -n device-path
```

The following examples show the command output for different disk types.

- For a 4-Kb native disk

```
# devprop -n /dev/rdsk/c0t0d0s0 device-blksize device-pblksize
4096
4096
```

- For a 512n disk

```
# devprop -n /dev/rdsk/c1t0d0s0 device-blksize device-pblksize
512
512
```

- For a 512e disk

```
# devprop -n /dev/rdsk/c2t0d0s0 device-blksize device-pblksize
512
4096
```

iSNS Support in the Solaris iSCSI Target and Initiator

Oracle Solaris provides support for the Internet Storage Name Service (iSNS) protocol in the Solaris iSCSI target and initiator software. The iSNS protocol enables automated discovery, management, and configuration of iSCSI devices on a TCP/IP network.

- For information about configuring the Oracle Solaris iSCSI target to use a third-party iSNS server, see Chapter 8, “Configuring Storage Devices With COMSTAR ”.
- For information about configuring the Solaris iSCSI target with a Solaris iSNS server in the Oracle Solaris release, see Chapter 9, “Configuring and Managing the Oracle Solaris Internet Storage Name Service (iSNS)”.

Identifying Devices by Physical Locations

The `/dev/chassis` directory provides device names that include physical locations. You can use this information to help you identify where devices are physically located if they need to be replaced or changed. For a list of commands that display disk information as well as examples to obtain physical location information, see “Displaying Disk Physical Locations” on page 114.

Concepts and Terminology

This section does not discuss all the terminology that is used for disks, such as *track*, *cylinder*, *sector*, and so on. For explanations of the meaning of these terms, refer to numerous information sources about hard disks, such as http://en.wikipedia.org/wiki/Hard_disk_drive, or the disk manufacturer's documentation.

The following concepts are useful to understand disk management in Oracle Solaris.

EFI (GPT) Disk Label

A disk *label* stores information about the disk's controller, geometry, and slices. A disk label is also called the *VTOC* (*Volume Table of Contents*) label.

Labeling a disk means writing slice information onto the disk. You usually label a disk after you change its slices or partitions. By means of the label, the OS is informed about the slices. If you fail to label a disk after you create slices, the slices will be unavailable.

Oracle Solaris supports the following disk labels:

- SMI – Traditional VTOC label for disks that are less than 2 TB in size.
- EFI – Label for disks that are larger than 2 TB. However, the Extensible Firmware Interface GUID Partition Table (EFI GPT) disk label is also available for disks less than 2 TB.

Note - You can continue to use the VTOC label on any size disk, but the addressable space by the VTOC is limited to 2 TB.

In Oracle Solaris, an EFI (GPT) labeled disk is installed by default on the following systems:

- SPARC systems with GPT enabled firmware

  On SPARC T4 servers, the Sun System Firmware must be at least version 8.4.0. On SPARC T5 and SPARC M5 servers, the firmware must be at least version 9.1.0.

- x86 systems

The Oracle Solaris ZFS file system supports file systems that are greater than 1 TB in size.

Note - The legacy Solaris Volume Manager software can also be used to manage disks that are greater than 1 TB, but a root disk that is managed by Solaris Volume Manager cannot be used for booting the Oracle Solaris 11 release.

The following additional features differentiate the EFI disk label from the VTOC disk label:

- Provides usable slices 0-6, where partition 2 is just another slice.
- Prohibits overlap of partitions or slices with a primary or backup label, or with other partitions. The size of the EFI label is usually 34 sectors, so partitions usually start at sector 34. Thus, no partition can start at sector zero (0). The entire disk is represented by c*x*t*y*d*z*.
- Does not use the notion of geometry. Partitions in EFI (GPT) labeled disks are defined based on logical blocks. Thus the EFI disk label provides information about disk or partition sizes in sectors and blocks, but not in cylinders and heads.
- Stores information in the last two cylinders of a disk or partition instead of in an alternate cylinder area.
- Supports reassigning partition tags after partition sizes are changed, except the `unassigned` partition tag which is assigned only to partitions with sizes equal to zero.

Before deciding to use EFI disks, consider that layered software products that are intended for systems with VTOC-labeled disks might be incapable of accessing a disk with an EFI disk label.

About Disk Slices or Partitions

Files on a disk are contained in file systems. Each file system on a disk is assigned to a *slice* comprising of a group of sectors. Slices are sometimes referred to as *partitions*. Certain interfaces, such as the `format` utility, refer to slices as partitions.

Each disk slice appears as a separate disk drive.

When setting up slices, remember these rules:

- Each disk slice holds only one file system.
- No file system can span multiple slices.

For information about file systems, see “Managing File Systems in Oracle Solaris 11.2 ”.

Third-party database applications typically create raw data slices. These applications must not use block 0 or slice 2 for raw slices. Block 0 stores the disk label while slice 2 represents the entire disk with a VTOC label. Creating raw slices on these two locations overwrites the disk label and renders data on the disk inaccessible.

Using the Free Hog Slice

When you use the `format` utility to change the size of one or more disk slices, you designate a temporary slice that will expand and shrink to accommodate the resizing operations.

This temporary slice donates, or "frees," space when you expand a slice, and receives, or "hogs," the discarded space when you shrink a slice. For this reason, the donor slice is sometimes called the *free hog*.

The free hog slice exists only during installation or when you run the `format` utility. There is no permanent free hog slice during day-to-day operations.

For information on using the free hog slice, see "How to Replace a ZFS Root Pool (VTOC)" on page 132 or "How to Replace a ZFS Root Pool (EFI (GPT))" on page 137.

Where to Find Disk Management Tasks

Use these references to find step-by-step instructions for managing disks.

Disk Management Task	For More Information
Display physical disk information.	"Obtaining Disk Information" on page 108
Configure disks.	"Configuring Disks" on page 116
Display information about disk usage.	"Managing Disk Use" on page 97
Set up ZFS pools on disks.	Chapter 7, "Setting Up ZFS on Disks"
Hot-plug a SCSI or PCI disk.	Chapter 2, "Dynamically Configuring Devices"

♦ ♦ ♦ CHAPTER 5

Managing the Use of System's Disks

This chapter describes how to optimize disk space by locating unused files and large directories. This is a list of the information in this chapter:

Managing Disk Use

Use the following task map for a list of management tasks you can perform on disks.

TABLE 5-1 Managing Disk Use Task Map

Task	Description	For Instructions
Display information about the use of disk space on the system.	Show how disk space is currently being used.	"Displaying Information About Disk Space" on page 98
Display information about sizes of files that use up disk space.	Show sizes and other information about files that are stored on the system disks.	"Displaying Information About File Sizes" on page 99
Display information about the size of directories and subdirectories.	Display the size of one or more directories, subdirectories, and files by using the `du` command.	"Displaying Information About Directory Sizes" on page 101
Perform disk maintenance.	Ensure disk availability by regularly removing unnecessary files and directories. Identify old files by listing them based on time stamps.	"Maintaining Disks" on page 102
Find and remove old or inactive files.	Use the `find` to identify inactive files that can be marked for deletion.	"Removing Old or Inactive Files" on page 102

Task	Description	For Instructions
Clear out temporary directories.	Locate temp directories, then use the `rm -r *` command to remove the entire directory.	"Emptying Temporary Directories" on page 104
Find and delete core files.	Find and delete core files by using the `find . -name core -exec rm {} \;` command.	"Removing Dump Files" on page 105

The following list summarizes the commands available for displaying information about file size and disk space.

- `zpool list` – displays pool space size, how much space is allocated to datasets and internal metadata and how much space is unallocated. See the `zpool`(1M) man page.
- `zfs list` – displays amount of space consumed by datasets and any descendents, the amount of space available to the dataset and its descendents, and the amount of data that is referenced by this dataset, which may or may not be shared with other datasets in the pool. See the `zfs`(1M) man page.
- `df` – reports the number of free disk blocks and files. See the `df`(1M) man page.
- `du` – summarizes disk space allocated to each subdirectory. See the `du`(1) man page.
- `find -size` – searches recursively through a directory based on the size specified with the `-size` option. See the `find`(1) man page.
- `ls` – lists the size of a file in the power of 1024 scaling. See the `ls`(1) man page.

The `zpool list` and `zfs list` commands are better than the legacy `df` and `du` commands for determining your available ZFS storage pool and file system space. With the legacy commands, you cannot easily discern between pool and file system space, nor do the legacy commands account for space that is consumed by descendent file systems or snapshots.

Displaying Information About Disk Space

The command you use to obtain information about the use of disk space on your system depends on the file system. The following sections show how to use the appropriate command for the file system to display data about disk use.

Use of Disk Space on ZFS File Systems

To display information about how ZFS disk space is used, use the `zpool` command as follows:

```
# zpool list root-pool
```

where *root-pool* is the name of the root pool on the system. The following example provides information about `rpool`, which has 10.0 Gbyte allocated and 580 Gbyte free.

```
# zpool list rpool
NAME   SIZE  ALLOC   FREE  CAP  HEALTH  ALTROOT
rpool   68G  10.0G  58.0G  14%  ONLINE  -
```

The command's `-r` option enables you to compare available space with actually used space in the root pool.

```
# zfs list -r rpool
NAME                       USED  AVAIL  REFER  MOUNTPOINT
rpool                     10.2G  56.8G   106K  /rpool
rpool/ROOT                5.04G  56.8G    31K  legacy
rpool/ROOT/solaris        5.04G  56.8G  5.04G  /
rpool/dump                1.00G  56.8G  1.00G  -
rpool/export                63K  56.8G    32K  /export
rpool/export/home           31K  56.8G    31K  /export/home
rpool/swap                4.13G  56.9G  4.00G  -
```

Use of Disk Space on UFS File Systems

To display information about how UFS disk space is used, use the `df` command.

```
$ df
/                  (/dev/dsk/c0t0d0s0 ):  101294 blocks   105480 files
/devices           (/devices          ):       0 blocks        0 files
/system/contract   (ctfs               ):       0 blocks 2147483578 files
/proc              (proc               ):       0 blocks     1871 files
/etc/mnttab        (mnttab             ):       0 blocks        0 files
/etc/svc/volatile  (swap               ):  992704 blocks    16964 files
/system/object     (objfs              ):       0 blocks 2147483530 files
```

To find out how disk space is used per user, use the `quot` command.

```
# quot /ufsfs
/dev/rdsk/c3t3d0s0:
21048   root
350   amy
250   rory
```

Note - The `quot` command works only on local legacy UFS file systems.

Displaying Information About File Sizes

You can check the size of files and sort them by using the `ls` command. You can find files that exceed a size limit by using the `find` command. For more information, see the `ls`(1) and `find`(1) man pages.

Note - If you run out of space in the `/var` directory, do not symbolically link the `/var` directory to a directory on a file system with more disk space. Doing so, even as a temporary measure, might cause problems for certain daemon processes and utilities.

Using the `ls` Command

The `ls` commands displays a list of files on a specific directory. The following `ls` options are useful for obtaining information about file sizes on the system.

- `-l` – displays a list of files and directories in long format and shows the sizes in bytes.
- `-h` – scales file sizes and directory sizes into KB, MB, GB, or TB when the file or directory size is larger than 1024 bytes.
- `-s` – displays a list of the files and directories and shows the sizes in blocks.

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

The following example shows that the `lastlog` and `messages` files are the two largest files in the `/var/adm` directory.

```
$ cd /var/adm
$ ls -lh
total 148
-rw-------   1 uucp     bin            0 Nov 26 09:25 aculog
-r--r--r--   1 root     other       342K Nov 26 13:56 lastlog
-rw-r--r--   1 root     root         20K Nov 26 13:55 messages
-rw-r--r--   1 root     bin         3.3K Nov 26 13:56 utmpx
-rw-r--r--   1 adm      adm          19K Nov 26 13:56 wtmpx
```

In the following example that uses the `-sh` options, the first column of the output indicates the blocks that are used by the files.

```
$ ls -sh
880 -r--r--r--   1 root     other       342K Nov 26 13:56 lastlog
 25 -rw-r--r--   1 root     root         20K Nov 26 13:55 messages
  7 -rw-r--r--   1 root     bin         3.3K Nov 26 13:56 utmpx
 24 -rw-r--r--   1 adm      adm          19K Nov 26 13:56 wtmpx
```

You can combine the `ls` command with the `sort` command to display a directory's files in an ascending or descending order according to a criterion such as file size. For more information about the `sort` command, see the `sort`(1) man page.

Using the `find` Command

The `find` command enables you to search for files that fit a specific search criteria. For example, to find files that exceed a specific file size, you use the following command syntax:

```
$ find directory -size +nnn
```

where *directory* specifies the directory that you want to search and *nnn* is a number that represents a size in 512-byte blocks which you specify for the `-size` option.

The following example shows how to find files larger than 400 blocks in the current working directory. The `-print` option displays the output of the `find` command.

```
$ find . -size +400 -print
./Howto/howto.doc
./Howto/howto.doc.backup
./Howto/howtotest.doc
./Routine/routineBackupconcepts.doc
./Routine/routineIntro.doc
```

Displaying Information About Directory Sizes

You can display the size of directories by using the `du` command and options. Additionally, you can find the amount of disk space used by user accounts on local UFS file systems by using the `quot` command. For more information about these commands, see `du`(1) and `quot`(1M).

The `du` command displays the size of the directory you specify as well as all the subdirectories. You can use the command with the following options:

- `-a` – displays the size of each file and subdirectory, and the total number of blocks that are contained in the specified directory.
- `-s` – displays the total number of blocks that are contained in the specified directory.
- `-H` – displays the size of each directory in 1000–byte blocks.

Use the following command syntax:

```
$ du [options] [directory1 directory2 ...]
```

The following example shows the size of `/var/adm` and its subdirectories:

```
$ du /var/adm
2       /var/adm/acct/nite
2       /var/adm/acct/sum
8       /var/adm/acct
2       /var/adm/sa
2       /var/adm/sm.bin
258     /var/adm
```

To compare sizes of different directories, specify those directories in the command. The following example shows the comparison of sizes between `/var/adm` and `/var/spool/lp`.

```
$ du /var/adm /var/spool/lp
2       /var/adm/acct/nite
2       /var/adm/acct/sum
8       /var/adm/acct
2       /var/adm/sa
2       /var/adm/sm.bin
258     /var/adm
4       /var/spool/lp/admins
2       /var/spool/lp/requests/printing....
4       /var/spool/lp/requests
4       /var/spool/lp/system
2       /var/spool/lp/fifos
24      /var/spool/lp
```

The following example shows directory sizes in 1024–byte blocks:

```
$ du -h /usr/share/audio
796K   /usr/share/audio/samples/au
797K   /usr/share/audio/samples
798K   /usr/share/audio
```

Maintaining Disks

Perform regular disk maintenance to ensure that efficient use of disk space. To create additional disk space, you can do delete files and directories that you no longer need. This section discusses some disk maintenance tasks.

Caution - Deleting files and directories from the system have permanent results. Unless you have an archive system that backs up all your machines, deleted files become irrecoverable. You must ensure that the correct files and directories are being removed.

Because file deletion is a critical operation, you must have the appropriate rights to be able to perform system wide deletions. For more information, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

Removing Old or Inactive Files

With the `ls -t` command, you can generate a list of files in a directory and sort them according to their respective time stamps. By default, the files will be listed from the newest files to the oldest. The following example lists the files in `/var/adm` starting from the most recent file.

```
$ ls -t /var/adm
total 134
-rw-------   1 root     root         315 Sep 24 14:00 sulog
-r--r--r--   1 root     other     350700 Sep 22 11:04 lastlog
-rw-r--r--   1 root     bin         4464 Sep 22 11:04 utmpx
-rw-r--r--   1 adm      adm        20088 Sep 22 11:04 wtmpx
-rw-r--r--   1 root     root       11510 Sep 10 16:13 messages.1
drwxrwxr-x   5 adm      adm          512 Sep 10 15:19 acct
drwxrwxr-x   2 adm      sys          512 Sep 10 15:19 sa
drwxr-xr-x   2 adm      adm          512 Sep 10 15:17 log
```

You can then remove old files that you determine as no longer needed.

▼ How to Find Old or Inactive Files for Deletion

The `find` command enables you to search for files within a defined time range that you can mark for deletion.

1. **Become an administrator.**

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

2. **Find files that have not been accessed for a specified number of days and list them in a file.**

   ```
   # find directory -type f[-atime +nnn] [-mtime +nnn] -print > output-file &
   ```

directory	Identifies the directory you want to search. Subdirectories below are also searched.
`-atime +`*nnn*	Specifies the number of days that files that have not been accessed.
`-mtime +`*nnn*	Specifies the number of days that files have not been modified.
output-file	Refers to the file to which the output of the command will be written.

3. **Ensure that the list of files in the output file can be safely removed.**

   If some files in the list are still needed, remove their filenames from the output file.

4. **Remove the files that are listed in the output file.**

   ```
   # rm `cat output-file`
   ```

Example 5-1 Finding and Removing Old or Inactive Files

The following example shows files in the `/var/adm` directory and the subdirectories that have not been accessed in the last 60 days. The `/var/tmp/deadfiles` file contains the list of inactive files. The `rm` command removes these inactive files.

```
# find /var/adm -type f -atime +60 -print > /var/tmp/deadfiles &
# more /var/tmp/deadfiles
/var/adm/aculog
/var/adm/spellhist
/var/adm/wtmpx
/var/adm/sa/sa13
/var/adm/sa/sa27
/var/adm/sa/sa11
/var/adm/sa/sa23
/var/adm/sulog
/var/adm/vold.log
/var/adm/messages.1
/var/adm/messages.2
/var/adm/messages.3
# rm `cat /var/tmp/deadfiles`
#
```

Emptying Temporary Directories

The `/var/tmp` and `/var/spool` directories are locations for temporary files which are not necessary to store over long periods of time. You can remove these after determining that the files are no longer needed. Similarly, empty other directories that you know contain temporary files.

To remove unneeded subdirectories, use the `rm -r` command syntax. The following example shows how to empty a user created temporary directory `mywork` that also contains temporary subdirectories.

```
# cd mywork
# ls
filea.000
fileb.000
filec.001
tempdir/
drafts/
# rm -r *
# ls
#
```

Removing Dump Files

Dump files contain raw data that get generated at the instant that computer program crashes. These files are also called crash dumps, memory dumps, or system dumps. The files are important for diagnosing problems with the programs that generated the dumps. The relevance and importance of these dump files are connected only to the instant of the program's abnormal termination. Thus, these files have no permanent value. You should not store these files over time, especially after you have completed diagnosing and resolving the problem of the program crash. Because of the temporariness of the value of these dump files as well as their typically large sizes, you can delete them safely.

All dump files are named `core`. These files can be generated in any random directory. You can delete these files in any of the following ways:

- Go to the `/var/crash/`*system* directory and remove the core files. In this path, *system* identifies the system that generated the dump files. For example:

  ```
  # cd /var/crash/system01
  # rm *
  ```

- Search a specific directory and remove any core files that are found there.

  The following example uses the `find` command to remove core files in the `/home/jones` user account directory.

  ```
  # cd /home/jones
  # find . -name core -exec rm {} \;
  ```

♦ ♦ ♦ CHAPTER 6

Administering the System's Disks

This chapter provides procedures to administer disks on the system.

The following topics are covered:

For overview information about disk management, see Chapter 4, "Managing Disks in Oracle Solaris".

Tools for Managing Disks

The Format utility is the central tool for administering disks. The utility enables you to perform a range of tasks from discovering the types of disks to verifying that these disks are known to the system.

Note - To use the utility, you must have the appropriate administrative rights. Refer to "Using Your Assigned Administrative Rights" in "Securing Users and Processes in Oracle Solaris 11.2 ".

Launch the utility by issuing the `format` command. The command displays a list of disks on the system and prompts you to select a disk. After you select the disk, the Format Menu appears as follows:

```
FORMAT MENU:
disk       - select a disk
type       - select (define) a disk type
partition  - select (define) a partition table
current    - describe the current disk
format     - format and analyze the disk
fdisk      - run the fdisk program (x86 only)
repair     - repair a defective sector
```

```
label      – write label to the disk
analyze    – surface analysis
defect     – defect list management
backup     – search for backup labels
verify     – read and display labels
save       – save new disk/partition definitions
inquiry    – show vendor, product and revision
volname    – set 8-character volume name
!<cmd>     – execute <cmd>, then return

format>
```

At the `format >` prompt, type the action that you want to perform. Depending on the action that you choose, you are prompted to type further actions. To exit, type **`quit`**.

For more information, see the `format`(1M) man page as well as Chapter 10, "The Format Utility Reference".

The `format -e` command launches the utility that enables you to write EFI-compliant disk labels to support disks with capacities greater than 1 Tbyte. However, many software components are still limited to 1 Tbye size or less. Therefore, use this command syntax with caution. For examples of how to use the `format -e` command, see "Creating Disk Labels" on page 117.

The following additional commands are also available for administering disks such as displaying disk information:

- `croinfo`(1M)
- `diskinfo`(1M)
- `zpool`(1M)
- `prtvtoc`(1M)
- `prtconf`(1M)
- `fdisk`(1M) for x86 systems

Obtaining Disk Information

To facilitate disk identification, assign aliases to the disks. Use the following command:

```
# fmadm add-alias chassis-name.chassis-serial alias-id
```

You can set a naming standard so that the alias name maps to a specific disk's chassis name and chassis serial number. The following example maps the disk with chassis name SUN-Storage-J4200 and serial number 0905QAJ00E to an alias.

```
# fmadm add-alias SUN-Storage-J4200.0905QAJ00E J4200@RACK10:U24-25
```

For more information about assigning aliases to disks, see the `fmadm`(1M) man page.

Identifying Disks on a System

To identify disks, launch the Format utility by issuing the `format` command. The command displays the disks in the system similar to the following example:

```
# format
AVAILABLE DISK SELECTIONS:
0. c2t0d0 <SUN36G cyl 14087 alt 2 hd 24 sec 424>
/pci@1c,600000/scsi@2/sd@0,0
/dev/chassis/J4200@RACK10:U24-25/SCSI_Device__0/disk
1. c2t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
/pci@1c,600000/scsi@2/sd@1,0
/dev/chassis/J4200@RACK10:U24-25/SCSI_Device__1/disk
Specify disk (enter its number):
```

The command displays a list of recognized disks under `AVAILABLE DISK SELECTIONS`. In the example's first entry, disk 0 or target 0 is connected to the second SCSI host adapter (`scsi@2`). In turn, the host adapter is connected to the second PCI interface (`/pci@1c,600000/...`). The output also associates both the physical and logical device name to the disk's marketing name, `SUN36G`, which is always listed in angle brackets `<>`.

The association enables you to identify which logical device names represent the disks that are connected to your system. For a description of logical and physical device names, see “Device Naming Conventions” on page 26.

Some disks do not have marketing names. In this case, use the utility to label the disk and, if necessary, identify its type. See “Creating Disk Labels” on page 117.

You can use a wildcard to display disk information. For example, to display the disks that are connected to controller 0 in the previous sample output, type the following:

```
# format /dev/rdsk/c2t*
```

If the Format utility does not recognize a disk, connect the disk to the system by consulting your disk hardware documentation. Or, use the alternative procedures in the following sections to identify disks:

- Chapter 7, “Setting Up ZFS on Disks”.
- “How to Label a Disk” on page 117.

Displaying Slice or Partition Information

Two options in the Format utility enable you to administer disk partitions or slices, namely `partition` and `fdisk`. The `fdisk` option is used specifically to manage partitions on x86 based systems. Thus, these partitions are also called `fdisk` partitions.

Note - Solaris slices are also called partitions. Certain interfaces might refer to a *slice* as a *partition*. To avoid confusion, Oracle Solaris documentation tries to distinguish between `fdisk` partitions and the entities within the Solaris `fdisk` partition. These entities might be called slices or partitions.

For easier management, use whole disks for creating ZFS storage non-root pools instead of disk slices . You only need to use a disk slice if the disk is intended for the ZFS root pool. When you create a pool with whole disks, an EFI label is applied. If you need to prepare a disk for use as a root pool disk, create a slice 0 that contains the entire disk capacity, as shown in Example 6-1.

For information about setting up disks for use with ZFS storage pools, see Chapter 7, "Setting Up ZFS on Disks".

To display slice or partition information, perform the following general steps after you launch the Format utility:

1. At the Format Menu `format >` prompt, type **`partition`**.

   If you are using an x86 based system, you can also type **`fdisk`**.
2. At the `partition >` prompt, type **`print`**.

   If you typed **`fdisk`** at the `format >` prompt, you do not need to type **`print`**

The following list explains the meaning of the partition information. The actual partition information that is displayed varies depending on the label.

`Part`	Partition or slice number. For VTOC labeled disks, the partitions range from 0–7. For EFI labeled disks, the partitions range from 0–6.
`Tag`	File system that is mounted on the partition.
`Flag`	Any combination of the following states that apply to a specific partition: writable (`w`), mountable (`m`), readable (`r`), and unmountable (`u`). `wu_rm` for example are the flags for partitions reserved for swap areas.
`Cylinders`	Applies only to VTOC-labeled disks and refers to the starting and ending cylinder number for the slice.
`Size`	Slice or partition size in MB.
`Blocks`	Applies only to VTOC-labeled disks and refers to the total number of cylinders and the total number of sectors per slice.
`First Sector`	Applies only to EFI labeled disks and refers to the starting block number.
`Last Sector`	Applies only to EFI labeled disks and refers to the ending block number.

EXAMPLE 6-1 Slice Information on a Disk With a VTOC Label

This example assumes that the disk being administered is `c2t3d0`.

```
format> partition
partition> print
Current partition table (c2t3d0):
Total disk cylinders available: 14087 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
0       root    wm       0 - 14086      136.71GB    (14087/0/0) 286698624
1       swap    wu       0                0         (0/0/0)             0
2     backup    wu       0 - 14086      136.71GB    (14087/0/0) 286698624
3 unassigned    wm       0                0         (0/0/0)             0
4 unassigned    wm       0                0         (0/0/0)             0
5 unassigned    wm       0                0         (0/0/0)             0
6        usr    wm       0                0         (0/0/0)             0
7 unassigned    wm       0                0         (0/0/0)             0
partition> q
format> q
```

EXAMPLE 6-2 Slice Information on a Disk With an EFI Label

This example assumes that the disk being administered is `c2t3d0`.

```
format> partition
partition> print
Current partition table (default):
Total disk sectors available: 286722878 + 16384 (reserved sectors)

Part      Tag    Flag     First Sector         Size         Last Sector
0        usr    wm                34      136.72GB          286722911
1 unassigned    wm                 0           0               0
2 unassigned    wm                 0           0               0
3 unassigned    wm                 0           0               0
4 unassigned    wm                 0           0               0
5 unassigned    wm                 0           0               0
6 unassigned    wm                 0           0               0
7   reserved    wm         286722912        8.00MB          286739295
partition> q
format> q
```

EXAMPLE 6-3 Slice Information on a Disk With an EFI (GPT) Label

This example assumes that the disk being administered is `c2t0d0`.

```
format> partition
partition> print
Current partition table (original):
Total disk sectors available: 27246525 + 16384 (reserved sectors)
Part      Tag    Flag     First Sector        Size        Last Sector
0  BIOS_boot    wm               256     256.00MB         524543
1        usr    wm            524544      12.74GB         27246558
2 unassigned    wm                 0          0              0
```

```
3 unassigned    wm                0          0             0
4 unassigned    wm                0          0             0
5 unassigned    wm                0          0             0
6 unassigned    wm                0          0             0
7   reserved    wm          27246559       8.00MB        27262942
partition> q
format> q
```

EXAMPLE 6-4 Partition Information From the `fdisk` Command

The `fdisk` option of the Format utility shows similar partition information as the `partition` option but also includes partition types. In the following example, the disk has an EFI and a Solaris partition, and the Solaris partition is active.

```
format> fdisk
Part      Tag    Flag     First Sector         Size         Last Sector
  0  BIOS_boot    wm               256      256.00MB          524543
  1        usr    wm            524544       68.11GB          143358320
  2 unassigned    wm                 0          0              0
  3 unassigned    wm                 0          0              0
  4 unassigned    wm                 0          0              0
  5 unassigned    wm                 0          0              0
  6 unassigned    wm                 0          0              0

           Total disk size is 8924 cylinders
           Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1                 EFI               1     6       6      0
          2       Active    Solaris2          7  8925    8919    100
```

Displaying Disk Label Information

To display disk label information, use the `prtvtoc` command.

```
# prtvtoc path/device-name
```

where *device-name* is the raw disk device in the *patch* that you want to examine.

Note - To use this command, you must have the appropriate administrative rights. Refer to "Using Your Assigned Administrative Rights" in "Securing Users and Processes in Oracle Solaris 11.2 ".

The information varies depending on the current label of the disk. On VTOC labeled disks, information about tracks and cylinders is included. On EFI labeled disk, no track or cylinder information is provided.

EXAMPLE 6-5 Label Information on a Disk With a VTOC Label

```
# prtvtoc /dev/rdsk/c2t3d0s0
* /dev/rdsk/c2t3d0s0 partition map
*
* Dimensions:
*     512 bytes/sector
*     848 sectors/track
*      24 tracks/cylinder
*   20352 sectors/cylinder
*   14089 cylinders
*   14087 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
0      2    00          0 286698624 286698623
2      5    01          0 286698624 286698623
```

EXAMPLE 6-6 Label Information on a Root Pool Disk With an EFI Label

```
# prtvtoc /dev/dsk/c7t0d0s1
* /dev/dsk/c7t0d0s1 partition map
*
* Dimensions:
*     512 bytes/sector
* 156301488 sectors
* 156301421 accessible sectors
*
* Flags:
*   1: unmountable
*  10: read-only
*
* Unallocated space:
*       First     Sector    Last
*       Sector     Count    Sector
*          34       222       255
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
0     24    00        256    524288    524543
1      4    00     524544 155760527 156285070
8     11    00  156285071     16384 156301454
~#
```

EXAMPLE 6-7 Label Information on a Non-root Pool Disk With an EFI Label

```
# prtvtoc /dev/dsk/c8t3d0
* /dev/dsk/c8t3d0 partition map
*
* Dimensions:
```

```
*     512 bytes/sector
* 143374738 sectors
* 143374671 accessible sectors
*
* Flags:
*   1: unmountable
*  10: read-only
*
* Unallocated space:
*       First     Sector    Last
*       Sector     Count    Sector
*          34       222       255
*
*                           First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
0      4    00        256 143358065 143358320
8     11    00  143358321     16384 143374704
```

Displaying Disk Physical Locations

To display physical locations of disks, use the `croinfo` command which provides information about the chassis, receptacle, and occupant relative to the specific disk.

```
$ croinfo
D:devchassis-path                t:occupant-type   c:occupant-compdev
-------------------------------------------------  ------------------
/dev/chassis/SYS/HDD0/disk       disk              c2t0d0
/dev/chassis/SYS/HDD1/disk       disk              c2t1d0
/dev/chassis/SYS/HDD2/disk       disk              c2t2d0
```

You can use different options with the `croinfo` command to display information only about a specific disk.

- `croinfo -c` *disk* displays information only about a specific disk. *disk* refers to the component listed under the `c:occupant-compdev` column.
- `croinfo -c` *disk* `-o cp` displays the path which the specific disk occupies when it was installed on the system.

For other options, see the `croinfo`(1M) man page.

Note - The `format` command also provides physical device location information. See the sample output in "Identifying Disks on a System" on page 109.

Other commands display locations of devices as well. The following examples show how these commands are used to identify and locate devices.

EXAMPLE 6-8 Using the `diskinfo` Command

This example assumes that you assigned the alias `J4200@RACK10:U24-25` to the disk `SUN-Storage-J4200.0905QAJ00E`. For more information about disk aliases, see "Obtaining Disk Information" on page 108.

```
$ diskinfo
D:devchassis-path                                  c:occupant-compdev
-------------------------------------------------  ------------------
/dev/chassis/J4200@RACK10:U24-25/SCSI_Device__0/disk   c2t0d0
/dev/chassis/J4200@RACK10:U24-25/SCSI_Device__1/disk   c2t1d0
/dev/chassis/J4200@RACK10:U24-25/SCSI_Device__2/disk   c2t2d0
```

EXAMPLE 6-9 Using the `prtconf` Command

```
$ prtconf -l | more
System Configuration:  Oracle Corporation  sun4v
Memory size: 523776 Megabytes
System Peripherals (Software Nodes):

ORCL,SPARC-T3-4 location: /dev/chassis//SYS/MB/HDD0/disk
scsi_vhci, instance #0 location: /dev/chassis//SYS/MB/HDD0/disk
disk, instance #4 location: /dev/chassis//SYS/MB/HDD4/disk
disk, instance #5 location: /dev/chassis//SYS/MB/HDD5/disk
disk, instance #6 location: /dev/chassis//SYS/MB/HDD6/disk
```

EXAMPLE 6-10 Using the `zpool` Command

```
% zpool status -l export
pool: export
state: ONLINE
scan: none requested
config:

NAME                          STATE     READ WRITE CKSUM
export                        ONLINE       0     0     0
mirror-0                      ONLINE       0     0     0
/dev/chassis//SYS/MB/HDD0/disk  ONLINE       0     0     0
/dev/chassis//SYS/MB/HDD1/disk  ONLINE       0     0     0
mirror-1                      ONLINE       0     0     0
/dev/chassis//SYS/MB/HDD2/disk  ONLINE       0     0     0
/dev/chassis//SYS/MB/HDD3/disk  ONLINE       0     0     0
mirror-2                      ONLINE       0     0     0
/dev/chassis//SYS/MB/HDD4/disk  ONLINE       0     0     0
/dev/chassis//SYS/MB/HDD5/disk  ONLINE       0     0     0

errors: No known data errors
```

Configuring Disks

This section covers disk configuration tasks such as formatting, labeling, or partitioning disks. Disks are typically formatted and configured by the manufacturers and the default configurations are generally sufficient. However, disk configuration might be required on certain circumstances such as disk corruption and recovery.

Note that configuring a disk such as formatting would destroy existing data on the disk. Relabeling would remove existing partition information. Make sure to perform the necessary backups to ensure that no data loss occurs if you reconfigure a disk.

Procedures to configure disks on both SPARC based and x86 based systems have similar steps. However, with x86 based systems, you use the `fdisk` option of the Format utility for disk operations.

To configure a disk, the disk must be inactive. Thus, if the disk to be configured contains Oracle Solaris, then you must boot from a different media. For more information about booting Oracle Solaris systems, see topics on booting from a local media or from the network in "Booting and Shutting Down Oracle Solaris 11.2 Systems ".

Formatting a Disk

An unformatted disk is unusable. The Format utility can detect formatted disks. Select the disk that you want to verify, as shown in the following example for the `c2t1d0` disk. After you select the disk, the utility indicates whether the disk is formatted.

```
# format
AVAILABLE DISK SELECTIONS:
0. c2t0d0 <SUN36G cyl 14087 alt 2 hd 24 sec 424>
/pci@1c,600000/scsi@2/sd@0,0
/dev/chassis/J4200@RACK10:U24-25/SCSI_Device__0/disk
1. c2t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
/pci@1c,600000/scsi@2/sd@1,0
/dev/chassis/J4200@RACK10:U24-25/SCSI_Device__1/disk
Specify disk (enter its number): 1      Disk c2t1d0 is selected.
selectin c2t1d0
[disk formatted]      The selected disk is already formatted.
```

The Format utility automatically configures any unlabeled SCSI disk. The utility would display information about the formatted disk as follows:

```
c2t3d0: configured with capacity of 136.73GB
```

Note - Because formatting disks destroys data, if you suspect that a disk is corrupted, you can select `analyze` from the Format Menu. This functionality performs a surface analysis of the disk. After the test, you can determine if formatting is necessary.

EXAMPLE 6-11 Formatting Disk `c2t1d0`

The following example shows how to format `c2t1d0` that you selected after launching the Format utility.

```
format> format      Formatting is selected.
The protection information is not enabled
The disk will be formatted with protection type 0

Ready to format.  Formatting cannot be interrupted
and takes 169 minutes (estimated). Continue? yes

Beginning format. The current time is Fri Apr 1 ...

Formatting...
done

Verifying media...
pass 0 - pattern = 0xc6dec6de
14086/23/734

pass 1 - pattern = 0x6db6db6d
14086/23/734

Total of 0 defective blocks repaired.

format> q
```

Creating Disk Labels

Use this procedure under the following circumstances:

- During a system installation.
- When adding new disks.
- If the disk label becomes corrupted.
- If you change the disk type.

▼ How to Label a Disk

1. Launch the Format utility in Oracle Solaris.

```
# format
```

Note - To apply an EFI label, use the `format -e` command syntax instead. See Example 6-13.

A numbered list of disks is displayed.

2. **When prompted, type the number of the disk that you want to label.**

3. **If necessary, specify the disk type by performing the following substeps.**

   Do these substeps only if the utility does not recognize the disk type or if you want to change the disk type. Otherwise, proceed to the next step.

   a. **At the `format >` prompt, choose `type`.**

   b. **From the available options, type the number that corresponds to the disk's type.**

      If your disk is a SCSI-2 disk, you can choose to auto configure the disk by typing zero (`0`).

4. **Label the disk.**

   Performing one of the following depending on which prompt is displayed:

   - At the `format >` prompt, type **`label`**, then proceed as prompted.
   - At the confirmation prompt, type **`y`**.

5. **(Optional) To verify the results of the labeling process, at the `format >` prompt, type `verify`.**

6. **Exit the Format utility.**

Example 6-12 Labeling a Disk

The following example shows how to automatically configure a new and unlabeled disk, `c2t1d0`. The example begins after you have issued the `format` command and a list of disks is displayed.

```
# format
...
Specify disk (enter its number): 1
c2t1d0: configured with capacity of 33.92GB
Disk not labeled.  Label it now?  yes
format> verify
     Information is displayed.
format> q
```

Example 6-13 Labeling a Disk With an EFI Label

The following example shows how to use the `format -e` command to label a disk with an EFI label. Make sure to verify that your layered software products will continue to work on systems

with EFI-labeled disks. For general information on EFI label features, see “EFI (GPT) Disk Label” on page 94.

This example assumes that the selected disk to label is `c2t3d0`.

```
# format -e
...
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[0]: 1
Ready to label disk, continue? yes
format> quit
```

Example 6-14 Labeling a Disk With an SMI Label

The following example applies an SMI label to `c2t0d0`. Because the disk previously had an EFI label, the example includes a warning. Otherwise, the labeling would proceed without any further warnings.

```
# format -e
...
Specify disk (enter its number): 3
selecting c2t0d0
[disk formatted]
...
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0
Warning: This disk has an EFI label. Changing to SMI label will erase all
current partitions.
Continue? yes
Auto configuration via format.dat[no]?
Auto configuration via generic SCSI-2[no]?
```

Modifying Slices or Partitions

In most cases, disks are formatted and partitioned by the manufacturer. Thus, you configure disk partitions or slices if you want to modify partition configuration, either by changing the slice or partition size, or by changing the partition type.

SPARC: Using the `partition` Option

The Format utility's `partition` option configures disk slices on SPARC based systems. This section provides examples on the use of the `partition` option.

EXAMPLE 6-15 SPARC: Resizing a Disk Slice

In this example, the root pool disk's size is insufficient. For an optimal size, the bulk of the disk space must be in slice 0. You change the partition size by **modifying** the current partition.

```
...      Format Menu
format> partition
partition> print
Current partition table (default):
Total disk cylinders available: 14085 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
0       root    wm       1 -    13      129.19MB    (13/0/0)       264576
1       swap    wu      14 -    26      129.19MB    (13/0/0)       264576
2     backup    wu       0 - 14086      136.71GB    (14087/0/0) 286698624
3 unassigned    wm       0               0         (0/0/0)             0
4 unassigned    wm       0               0         (0/0/0)             0
5 unassigned    wm       0               0         (0/0/0)             0
6        usr    wm      27 - 14084      136.43GB    (14058/0/0) 286108416
7 unassigned    wm       0               0         (0/0/0)             0
8       boot    wu       0 -     0        9.94MB    (1/0/0)         20352
9 alternates    wm       0               0         (0/0/0)             0

partition> modify      Change partition size.
Select partitioning base:
0. Current partition table (default)
1. All Free Hog
Choose base (enter number) [0]? 1

Part      Tag    Flag     Cylinders         Size            Blocks
0       root    wm       0               0         (0/0/0)             0
1       swap    wu       0               0         (0/0/0)             0
2     backup    wu       0 - 14084      136.69GB    (14085/0/0) 286657920
3 unassigned    wm       0               0         (0/0/0)             0
4 unassigned    wm       0               0         (0/0/0)             0
5 unassigned    wm       0               0         (0/0/0)             0
6        usr    wm       0               0         (0/0/0)             0
7 unassigned    wm       0               0         (0/0/0)             0
8       boot    wu       0 -     0        9.94MB    (1/0/0)         20352
9 alternates    wm       0               0         (0/0/0)             0

Do you wish to continue creating a new partition
table based on above table[yes]? yes
Free Hog partition[6]? 0
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:

Part      Tag    Flag     Cylinders         Size            Blocks
0       root    wm       1 - 14084      136.68GB    (14084/0/0) 286637568
1       swap    wu       0               0         (0/0/0)             0
2     backup    wu       0 - 14084      136.69GB    (14085/0/0) 286657920
3 unassigned    wm       0               0         (0/0/0)             0
```

```
4 unassigned    wm       0               0         (0/0/0)             0
5 unassigned    wm       0               0         (0/0/0)             0
6        usr    wm       0               0         (0/0/0)             0
7 unassigned    wm       0               0         (0/0/0)             0
8       boot    wu       0 -     0       9.94MB    (1/0/0)         20352
9 alternates    wm       0               0         (0/0/0)             0

Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "c2t0d0"
Ready to label disk, continue? yes
partition> q
format> q
```

x86: Using the `fdisk` Option

You use the Format utility's `fdisk` option to administer `fdisk` partitions on x86 based systems.

You can use x86-type disks for ZFS storage pools, provided that the following requirements are met:

- If the disk has multiple partitions, one of partitions must be a Solaris partition.

  With the `fdisk` option, you can determine if a Solaris partition exists. If not, create one, as shown in Example 6-16
- The Solaris partition must be the active partition on the disk.

  The active partition is partition whose operating system will be booted by default at system startup.
- Solaris `fdisk` partitions must begin on cylinder boundaries.

  The Solaris `fdisk` partitions must not begin at cylinder 0 on the first disk, which is reserved for storing additional boot information, including the master boot record.
- The Solaris `fdisk` partition can be the entire disk or only part of the disk to leave space for other partitions.

  Provided the disk has sufficient space, you can create new partitions without having to reconfigure existing partitions on the disk.

All `fdisk` partitions have identifiers. A Solaris partition has two identifiers:

- `Solaris` uses the `0x82` identifier.
- `Solaris2` uses the `0xbf` identifier.

All Oracle Solaris commands, utilities, and drivers have been updated to work with either identifier with no impact on the `fdisk` functionality. Thus, you can switch between the two identifiers by selecting the appropriate option from the `fdisk` menu as follows:

```
format > fdisk
```

```
...
SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs      Toggle between identifiers
5. Edit/View extended partitions
6. Exit (update disk configuration and exit)
7. Cancel (exit without updating disk configuration)
```

Note - You can change the `fdisk` identifier even when the file system that is contained in the partition is mounted.

EXAMPLE 6-16 x86: Creating a Solaris `fdisk` Partition That Spans the Entire Drive

The following example shows how to create a Solaris `fdisk` partition that spans the entire `c8t3d0` drive.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
0. c8t0d0 <SEAGATE-ST973401LSUN72G-0556 cyl 8921 alt 2 hd 255 sec 63>
/pci@0,0/pci1022,7458@11/pci1000,3060@4/sd@0,0
/dev/chassis/SYS/HD0/disk
1. c8t1d0 <SEAGATE-ST973401LSUN72G-0556 cyl 8921 alt 2 hd 255 sec 63>
/pci@0,0/pci1022,7458@11/pci1000,3060@4/sd@1,0
/dev/chassis/SYS/HD1/disk
2. c8t2d0 <SEAGATE-ST973401LSUN72G-0556-68.37GB>
/pci@0,0/pci1022,7458@11/pci1000,3060@4/sd@2,0
/dev/chassis/SYS/HD2/disk
3. c8t3d0 <SEAGATE-ST973401LSUN72G-0556 cyl 14087 alt 2 hd 24 sec 424>
/pci@0,0/pci1022,7458@11/pci1000,3060@4/sd@3,0
/dev/chassis/SYS/HD3/disk
Specify disk (enter its number): 3
selecting c8t3d0
[disk formatted]
No Solaris fdisk partition found.
format> fdisk
No fdisk table exists. The default partitioning for your disk is:

a 100% "SOLARIS System" partition.

Type "y" to accept the default partition, otherwise type "n" to edit the
partition table. y

format> label
Ready to label disk, continue? yes
format> quit
```

EXAMPLE 6-17 x86: Converting an x86 Partition to a Solaris Partition

This example shows how to convert an existing partition to a Solaris partition. By default, disks have the EFI label on most x86 based systems. To change a partition type, you first destroy the existing one. When you change partition types, the labels are also automatically changed. Because the disk is originally an EFI disk, then you use the `format -e` command to launch the Format utility.

```
# format -e
...
format> fdisk
FORMAT MENU:
disk       - select a disk
type       - select (define) a disk type
Total disk size is 17833 cylinders
Cylinder size is 16065 (512 byte) blocks

Cylinders
Partition   Status    Type          Start   End    Length    %
=========   ======    ============  =====   ===    ======   ===
1                     EFI           0       17833  17834    100

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Edit/View extended partitions
6. Exit (update disk configuration and exit)
7. Cancel (exit without updating disk configuration)
Enter Selection: 3
Specify the partition number to delete (or enter 0 to exit): 1
This will make all files and
programs in this partition inaccessible (type "y" or "n"). y
Enter Selection: 1
Select the partition type to create:
1=SOLARIS2   2=UNIX      3=PCIXOS     4=Other        5=DOS12
6=DOS16      7=DOSEXT    8=DOSBIG     9=DOS16LBA     A=x86 Boot
B=Diagnostic C=FAT32     D=FAT32LBA   E=DOSEXTLBA    F=EFI (Protective)
G=EFI_SYS    0=Exit? 1
Specify the percentage of disk to use for this partition
(or type "c" to specify the size in cylinders). 100
Should this become the active partition? If yes, it  will be activated
each time the computer is reset or turned on.
Please type "y" or "n". y
Enter Selection: 6
Partition 1 is now the active partition.
```

Recovering a Corrupted Disk Label

A power or system failure might cause a disk's label to become unrecognizable. A corrupted disk label doesn't always mean that the slice information or the disk's data must be re-created or restored. However, the label must be restored.

Recovering a corrupted disk label requires two general steps: specifying the correct type for the disk and then recovering the backup label. Both steps are part of the Format utility.

▼ How to Recover a Corrupted Disk Label

1. **Boot the system to single-user mode.**

   Use either a local Oracle Solaris DVD or the network to boot and access the disk.

   For information on booting the system, see "Booting and Shutting Down Oracle Solaris 11.2 Systems ".

2. **Launch the Format utility.**

   ```
   # format
   ```

3. **Type the number of the disk that you need to recover.**

   ```
   Specify disk (enter its number):
   ```

   If the utility can configure the selected, the utility displays the following message:

   disk`: configured with capacity of` *size*

4. **Select one of the following to determine how to label the disk.**

   - **If the disk was configured successfully, do the following:**

     a. **At the `format >` prompt, type `verify`.**

        The verification process includes searching for the disk's backup label.

     b. **Ensure that the contents of the retrieved backup label appear satisfactory.**

     c. **At the `format >` prompt, type `backup` to apply the label.**

     d. **When prompted, confirm the command.**

   - **If the disk was *not* configured successfully, do the following:**

a. **At the `format >` prompt of the Format Menu, type `type`.**

   The Available Drives Type menu is displayed.

b. **Type the number that corresponds to the disk's type.**

c. **When prompted to proceed with labeling the disk, type `no`.**

d. **At the `format >` prompt, type `verify`.**

   The verification process includes searching for the disk's backup label.

e. **Ensure that the contents of the retrieved backup label appear satisfactory.**

f. **At the `format >` prompt, type `backup` to apply the label.**

g. **When prompted, confirm the command.**

5. **After the label is restored, exit the Format utility.**

6. **Verify the file systems on the recovered disk.**

   For information about using the `zpool scrub` command for ZFS file systems, see the `zpool`(1M) man page.

Example 6-18 Recovering the Label of an Autoconfigured Disk

This example assumes that you have already selected the disk whose label you want to recover.

```
format> verify
Warning: Could not read primary label.
Warning: Check the current partitioning and 'label' the disk or
use the 'backup' command.
Backup label contents:
Volume name = <        >
ascii name  = <FUJITSU-MAP3147N SUN146G-0501>
pcyl        = 14089
ncyl        = 14087
acyl        =    2
nhead       =   24
nsect       =  848
Part      Tag    Flag     Cylinders         Size            Blocks
0       root    wm       0 - 14086      136.71GB    (14087/0/0) 286698624
1       swap    wu       0                0         (0/0/0)             0
2     backup    wu       0 - 14086      136.71GB    (14087/0/0) 286698624
3 unassigned    wm       0                0         (0/0/0)             0
4 unassigned    wm       0                0         (0/0/0)             0
5 unassigned    wm       0                0         (0/0/0)             0
```

```
6        usr    wm       0               0         (0/0/0)            0
7 unassigned    wm       0               0         (0/0/0)            0
```

After the backup label is retrieved and you ensure that the contents are satisfactory, proceed with applying the label.

```
format> backup
Disk has a primary label, still continue? y

Searching for backup labels...found.
Restoring primary label

format> q
```

Example 6-19 Recovering the Label of a Disk Not Automatically Configurable

This example assumes that you have already selected the disk whose label you want to recover.

```
format> type
Specify disk type (enter its number)[12]: 12
Disk not labeled.  Label it now?  no

format> verify
Warning: Could not read primary label.
Warning: Check the current partitioning and 'label' the disk
or use the 'backup' command.
Backup label contents:
Volume name = <        >
ascii name  = <FUJITSU-MAP3147N SUN146G-0501>
pcyl        = 14089
ncyl        = 14087
acyl        =    2
nhead       =   24
nsect       =  848
Part      Tag    Flag     Cylinders         Size            Blocks
0       root    wm       0 - 14086      136.71GB    (14087/0/0) 286698624
1       swap    wu       0                 0        (0/0/0)             0
2     backup    wu       0 - 14086      136.71GB    (14087/0/0) 286698624
3 unassigned    wm       0                 0        (0/0/0)             0
4 unassigned    wm       0                 0        (0/0/0)             0
5 unassigned    wm       0                 0        (0/0/0)             0
6        usr    wm       0                 0        (0/0/0)             0
7 unassigned    wm       0                 0        (0/0/0)             0
```

After the backup label is retrieved and you ensure that the contents are satisfactory, proceed with applying the label.

```
format> backup
Disk has a primary label, still continue? y
Searching for backup labels...found.
Restoring primary label

format> q
```

Adding a Third-Party Disk

Oracle Solaris supports many third-party disks. However, for the disk to be recognized, you might need to supply a device driver.

Other options for adding disks are as follows:

- If you are adding a SCSI disk, you might to try the Format utility's automatic configuration feature.
- You might try hot-plugging a PCI, SCSI, or USB disk. For more information, see Chapter 1, “Managing Devices in Oracle Solaris”.

Note - Oracle cannot guarantee that its Format utility will work properly with all third-party disk drivers. If the disk driver is not compatible with the Format utility, the disk drive vendor should supply you with a custom disk formatting program.

Typically, you discover that software support is missing when you invoke the Format utility and find that the disk type is not recognized.

Refer to the appropriate configuration procedure for adding system disks or secondary disks in Chapter 7, “Setting Up ZFS on Disks”.

♦ ♦ ♦ CHAPTER 7

Setting Up ZFS on Disks

This chapter describes how to set up disks for ZFS file systems on both SPARC based and x86 based systems. The following topics are covered:

For overview information about disk management, see Chapter 4, "Managing Disks in Oracle Solaris".

About ZFS Disk Setup on Oracle Solaris Systems

You can use any disk on which to set up a ZFS file system. However, a ZFS file system is not directly mapped to a disk or a disk slice. You must create a ZFS storage pool before creating a ZFS file system. For more information about ZFS storage pools, see Chapter 3, "Managing Oracle Solaris ZFS Storage Pools," in "Managing ZFS File Systems in Oracle Solaris 11.2 ".

Note - You must properly configure your system disks before you can set up storage pools. For instructions on disk configuration, see "Configuring Disks" on page 116.

A root pool contains the root file system that is used to boot Oracle Solaris. A disk that is used in a non-root pool usually contains user or data files. You can attach additional disks to a root pool or a non-root pool for increased disk space.

To reduce system down time due to hardware failures, create a redundant root pool. Without a redundant root pool, if a root pool disk becomes damaged, the system might not boot. For redundant root pool configurations, Oracle Solaris supports only a mirrored root pool. In a mirrored root pool, you can add, replace, or detach disks to manage a pool's size.

To recover from a damaged root pool disk, choose one of the following:

- Reinstall the entire Oracle Solaris OS.
- Replace the root pool disk and restore your file systems from snapshots or from a backup medium.

You can replace a disk in a redundant pool provided that enough redundancy exists among the other devices. In a non-redundant pool, you can replace a disk only if all of the devices are in the ONLINE status.

Preparing a Disk for a ZFS Root File System

The following list specifies root pool disk requirements for SPARC based and x86 based systems:

- The disk to use must either be a single disk or part of a mirrored configuration. Non-redundant configurations and RAIDZ configurations are unsupported for the root pool.
- All subdirectories of the root file system except `/var` that are part of the OS image must be in the same dataset as the root file system.
- All Oracle Solaris components except the swap and dump devices must reside in the root pool.
- For a root pool disk that is VTOC labeled, create a disk slice with the bulk of disk space in slice 0, if you need to replace a root pool disk.

For a pool to be efficient, do not use different slices on a disk to share among different operating systems or with a different ZFS storage pool or storage pool components.

Oracle Solaris installs an EFI (GPT) label for the root pool disk or disks on most x86 based systems and on SPARC based systems with updated GPT firmware. However, the SMI (VTOC) label is still available and supported.

If your SPARC based systems' firmware is not updated for GPT support, you must create a disk slice for a disk that is intended for a ZFS root pool. However, on x86 based systems with a root pool disk that is labeled with EFI, the correct boot partitions are created automatically.

Adding or Replacing Disks for ZFS File Systems

Installing a new disk to add capacity or replace a failed disk depends on the hardware and system capabilities, such as support for dynamic reconfiguration (DR). Review your hardware documentation when adding or replacing a disk on your system.

The following procedure provides general steps to change disks. You must consult your hardware documentation for the correct method. For information about DR, including procedures and examples of DR operations, read Chapter 2, “Dynamically Configuring Devices”.

▼ How to Add Disks

1. **Become an administrator.**
2. **If required, bring the failed disk offline.**

   ```
   # zpool offline root-pool disk
   ```

3. **If required, unconfigure the disk.**

   ```
   # cfgadm -c unconfigure disk-apid
   ```

   where *disk-apid* refers to the device's logical attachment point ID. For more information about these IDs, see “About Attachment Point Identification” on page 33. See also “Configuring or Unconfiguring a SCSI Device” on page 37 for additional information about unconfiguring and configuring disks.

   The following example uses the attachment point ID to unconfigure the disk `c2t1d0`.

   ```
   # cfgadm -c unconfigure c2::dsk/c2t1d0
   ```

4. **Install the new disk.**
5. **If required, configure the new disk.**

   ```
   # cfgadm -c configure disk-apid
   ```

SPARC: Setting Up Disks for ZFS File Systems

This section contains procedures to set up ZFS storage pools after you have completed configuring your system's disks. These procedures apply to SPARC based systems only.

To perform any of the following tasks, you must boot from a separate Oracle Solaris installation DVD that is either accessed locally or from the network. For instructions, see topics on booting from a local media or from the network in “Booting and Shutting Down Oracle Solaris 11.2 Systems ”.

▼ SPARC: How to Recreate the ZFS Root Pool (EFI (GPT))

Use the following procedure to recreate the ZFS root pool or to create an alternate root pool. The `zpool create` command automatically creates a EFI (GPT) labeled disk with the correct boot information.

1. **Become an administrator.**

2. **Identify the disks for the root pool.**

   Issue the `format` command to launch the Format utility. The following is a sample output of the command.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
AVAILABLE DISK SELECTIONS:
       0. c3t0d0 <FUJITSU-MAV2073RCSUN72G-0301 cyl 14087 alt 2 hd 24 sec 424>
          /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@0,0
       1. c3t1d0 <FUJITSU-MAV2073RCSUN72G-0301 cyl 14087 alt 2 hd 24 sec 424>
          /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@1,0
       2. c3t2d0 <FUJITSU-MAV2073RCSUN72G-0301-68.37GB>
          /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@2,0
       3. c3t3d0 <FUJITSU-MAV2073RCSUN72G-0301-68.37GB>
          /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@3,0
```

3. **To recreate the root pool, exit the Format utility and type:**

```
# zpool create -B root-pool mirror disk1 disk2
```

   where *root-pool* is the name of the root pool.

   The following example sets and mirrors `rpool` on `c3t0d0` and `c3t1d0`.

```
# zpool create -B rpool mirror c3t0d0 c3t1d0
```

4. **Restore the root pool snapshots, if necessary.**

   For information about complete ZFS root pool recovery, see "Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.2 ".

▼ SPARC: How to Replace a ZFS Root Pool (VTOC)

In general, the root pool disk is installed automatically when the system is installed. Use this procedure if you need to replace a root pool disk or attach a new disk as a mirrored root pool disk.

Before You Begin Before you perform this procedure, ensure that you have completed the following tasks:

- Installed the new or replacement disk. See "Adding or Replacing Disks for ZFS File Systems".
- Verified that the disk has a VTOC label.

  Use the `prtvtoc` *path/disk-name* command to verify. See "Displaying Disk Label Information" on page 112 for other commands you can use. If you need to change the

label from EFI to VTOC, see “Creating Disk Labels” on page 117 for instructions with an example in Example 6-14.

- Verified that slice 0 uses most of the disk's size for optimal configuration.

  Use the Format utility to verify. If you need to change the slice configuration, see “Modifying Slices or Partitions” on page 119 with an example in Example 6-15.

1. **Become an administrator.**

2. **Replace the root pool on the new disk.**

   ```
   # zpool replace root-pool disk
   ```

3. **Depending on the specific task, choose one of the following sets of steps.**

   - Perform the following steps if you are replacing a failed disk with a new disk.
     1. **If required, bring the new disk online.**

        ```
        # zpool online root-pool disk
        ```

     2. **Verify that the new disk is resilvered.**

        ```
        # zpool status root-pool
        ```

     3. **Skip this step and proceed to the next step if you do not want to install Oracle Solaris on the new disk. Otherwise, install Oracle Solaris and then boot the system.**
     4. **Apply the boot blocks after the new disk is resilvered.**

        ```
        # bootadm install-bootloader
        ```

        For more information about booting Oracle Solaris systems, see the following resources:

        - `installboot`(1M) man page.
        - “Booting and Shutting Down Oracle Solaris 11.2 Systems ”.

     5. **Verify that the boot blocks are installed by rebooting the system to run level 3.**

        ```
        # init 6
        ```

   - Perform the following steps if you are attaching a new disk to create a mirrored root pool or attaching a larger disk to replace a smaller disk.
     1. **Attach the new disk to the ZFS pool.**

        ```
        # zpool attach root-pool disk new-disk
        ```

        The following example attaches the new disk `c2t1d0s0` to the current `rpool` on `c2t0d0s0`.

        ```
        # zpool attach rpool c2t0d0s0 c2t1d0s0
        ```

The `zpool attach` command also automatically applies the boot blocks.

2. **Verify that you can boot from the new disk.**
3. **If you are replacing an old disk with the new, then after the system boots from the new disk, detach the old disk.**

   ```
   # zpool detach root-pool old-disk
   ```

   For example, if you are replacing `c2t0d0s0` with `c2t1d0s0`, you would type the following:

   ```
   # zpool detach rpool c2t0d0s0
   ```

4. **Set up the system to boot automatically from the new disk, either by using the eeprom command or the `setenv` command from the SPARC boot PROM.**

x86: Setting Up Disks for ZFS File Systems

This section contains procedures to set up ZFS storage pools after you have completed configuring your system's disks. These procedures apply to x86 based systems only.

To perform any of the following tasks, you must boot from a separate Oracle Solaris installation DVD that is either accessed locally or from the network. For instructions, see topics on booting from a local media or from the network in "Booting and Shutting Down Oracle Solaris 11.2 Systems ".

▼ x86: How to Recreate the ZFS Root Pool (EFI (GPT))

Use the following procedure to recreate the ZFS root pool or to create an alternate root pool. The `zpool create` command automatically creates a EFI (GPT) labeled disk with the correct boot information.

Before You Begin Ensure that the disk has a Solaris partition that is also selected as the active partition. Use the `fdisk` option of the Format utility to view partition information. Example 6-4 partially shows the information that the option displays.

If no Solaris partition exists, create one. See Example 6-16 as a guide.

1. **Become an administrator.**
2. **(Optional) Identify the disks for the root pool.**

Issue the `format` command to launch the Format utility. The following is a sample output of the command.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
0. c6t0d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@0,0
1. c6t1d0 <FUJITSU-MAV2073RCSUN72G-0301-68.37GB>
/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@1,0
2. c6t2d0 <FUJITSU-MAV2073RCSUN72G-0301-68.37GB>
/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@2,0
3. c6t3d0 <FUJITSU-MAV2073RCSUN72G-0301 cyl 14087 alt 2 hd 24 sec 424>
/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@3,0
Specify disk (enter its number):
```

3. **To recreate the root pool, exit the Format utility and type:**

```
# zpool create -B root-pool mirror disk1 disk2
```

where *root-pool* is the name of the root pool.

The following example sets and mirrors `root-pool` on `c6t0d0` and `c6t1d0`.

```
# zpool create -B root-pool mirror c6t0d0 c6t1d0
```

4. **Restore the root pool snapshots, if necessary.**

For information about complete ZFS root pool recovery, see "Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.2 ".

▼ x86: How to Replace a ZFS Root Pool Disk (VTOC)

In general, the root pool disk is installed automatically when the system is installed. Use this procedure if you need to replace a root pool disk or attach a new disk as a mirrored root pool disk.

Before You Begin Before you perform this procedure, ensure that you have completed the following tasks:

- Installed the new or replacement disk. See "Adding or Replacing Disks for ZFS File Systems".
- Verified that the disk has a VTOC label.

  Use the `prtvtoc` *path/disk-name* command to verify. See "Displaying Disk Label Information" on page 112 for other commands you can use. If you need to change the label from EFI to VTOC, see "Creating Disk Labels" on page 117 for instructions with an example in Example 6-14.
- Ensured that the disk has a Solaris partition that is also selected as the active partition.

Use the `fdisk` option of the Format utility to view partition information. Example 6-4 partially shows the information that the option displays.

If no Solaris partition exists, create one. See Example 6-16 as a guide.

1. **Become an administrator.**

2. **Replace the root pool on the new disk.**

   ```
   # zpool replace root-pool disk
   ```

   The following example replaces `rpool` on the new disk `c8t1d0s0`.

   ```
   # zpool replace rpool c8t1d0s0
   ```

3. **Depending on the specific task, choose one of the following sets of steps.**

   - Perform the following steps if you are replacing a failed disk with a new disk.
     1. **If required, bring the new disk online.**

        ```
        # zpool online root-pool disk
        ```

     2. **Verify that the new disk is resilvered.**

        ```
        # zpool status root-pool
        ```

     3. **Skip this step and proceed to the next step if you do not want to install Oracle Solaris on the new disk. Otherwise, install Oracle Solaris and then boot the system.**
     4. **Apply the boot blocks after the new disk is resilvered.**

        ```
        # bootadm install-bootloader
        ```

        For more information about booting Oracle Solaris systems, see the following resources:

        - `installboot`(1M) man page.
        - "Booting and Shutting Down Oracle Solaris 11.2 Systems ".

        If you want to install the GRUB Legacy boot loader, you must first remove all GRUB 2 boot environments from your system and then use the `installgrub` command. For instructions, see "x86: Installing GRUB Legacy on a System That Has GRUB 2 Installed" in "Booting and Shutting Down Oracle Solaris 11.2 Systems ".

     5. **Verify that the boot blocks are installed by rebooting the system to run level 3.**

        ```
        # init 6
        ```

   - Perform the following steps if you are attaching a new disk to create a mirrored root pool or attaching a larger disk to replace a smaller disk.
     1. **Attach the new disk to the ZFS pool.**

        ```
        # zpool attach root-pool old-disk new-disk
        ```

The following example attaches the new disk `c8t1d0s0` to the current `rpool` on `c8t0d0s0`.

```
# zpool attach rpool c8t0d0s0 c8t1d0s0
```

The `zpool attach` command automatically applies the boot blocks as well.

2. **Verify that you can boot from the new disk.**
3. **If you are replacing an old disk with the new, then after the system boots from the new disk, detach the old disk.**

```
# zpool detach root-pool old-disk
```

For example, if you are replacing `c8t0d0s0` with `c8t1d0s0`, you would type the following:

```
# zpool detach rpool c8t0d0s0
```

4. **Set up the system to boot automatically from the new disk by reconfiguring the system's BIOS.**

▼ x86: How to Replace a ZFS Root Pool (EFI (GPT))

In general, the root pool disk is installed automatically when the system is installed. Also, in most cases, an EFI (GPT) disk label is installed on the root pool disk.

Use this procedure to replace a root pool disk or attach a new disk as a mirrored root pool disk.

Before You Begin Before you perform this procedure, ensure that you have completed the following tasks:

- Installed the new or replacement disk. See "Adding or Replacing Disks for ZFS File Systems".
- Ensured that the disk has a Solaris partition that is also selected as the active partition.

  Use the `fdisk` option of the Format utility to view partition information. Example 6-4 partially shows the information that the option displays.

  If no Solaris partition exists, create one. See Example 6-16 as a guide.

1. **Become an administrator.**
2. **Identify the disks for the root pool.**

Issue the `format` command to launch the Format utility. The following is sample output of the command.

```
# format -e
```

```
AVAILABLE DISK SELECTIONS:
1. c8t0d0 <Sun-STK RAID INT-V1.0 cyl 17830 alt 2 hd 255 sec 63>
/pci@0,0/pci10de,375@f/pci108e,286@0/disk@0,0
2. c8t1d0 <Sun-STK RAID INT-V1.0-136.61GB>
/pci@0,0/pci10de,375@f/pci108e,286@0/disk@1,0
3. c8t2d0 <Sun-STK RAID INT-V1.0-136.61GB>
/pci@0,0/pci10de,375@f/pci108e,286@0/disk@2,0
4. c8t3d0 <Sun-STK RAID INT-V1.0-136.61GB>
/pci@0,0/pci10de,375@f/pci108e,286@0/disk@3,0
```

3. **To replace the root pool, exit the Format utility and type:**

   ```
   # zpool replace root-pool disk
   ```

   The following example replaces `rpool` on the new disk `c8t1d0s0`.

   ```
   # zpool replace rpool c8t1d0s0
   ```

4. **Depending on the specific task, choose one of the following sets of steps.**

   - Perform the following steps if you are replacing a failed disk with a new disk.
     1. **If required, bring the new disk online.**

        ```
        # zpool online root-pool disk
        ```

     2. **Verify that the new disk is resilvered.**

        ```
        # zpool status root-pool
        ```

     3. **Skip this step and proceed to the next step if you do not want to install Oracle Solaris on the new disk. Otherwise, install Oracle Solaris and then boot the system.**
     4. **Apply the boot blocks after the new disk is resilvered.**

        ```
        # bootadm install-bootloader
        ```

        For more information about booting Oracle Solaris systems, see the following resources:

        - `installboot`(1M) man page.
        - "Booting and Shutting Down Oracle Solaris 11.2 Systems ".

        If you want to install the GRUB Legacy boot loader, you must first remove all GRUB 2 boot environments from your system and then use the `installgrub` command. For instructions, see "x86: Installing GRUB Legacy on a System That Has GRUB 2 Installed" in "Booting and Shutting Down Oracle Solaris 11.2 Systems ".

     5. **Verify that the boot blocks are installed by rebooting the system to run level 3.**

        ```
        # init 6
        ```

   - Perform the following steps if you are attaching a new disk to create a mirrored root pool or attaching a larger disk to replace a smaller disk.
     1. **Attach the new disk to the ZFS pool.**

```
# zpool attach root-pool old-disk new-disk
```

The following example attaches the new disk `c8t1d0s0` to the current `rpool` on `c8t0d0s0`.

```
# zpool attach rpool c8t0d0s0 c8t1d0s0
```

The `zpool attach` command automatically applies the boot blocks as well.

2. **Verify that you can boot from the new disk.**
3. **If you are replacing an old disk with the new, then after the system boots from the new disk, detach the old disk.**

```
# zpool detach root-pool old-disk
```

For example, if you are replacing `c8t0d0s0` with `c8t1d0s0`, you would type the following:

```
# zpool detach rpool c8t0d0s0
```

5. **Set up the system to boot automatically from the new disk by reconfiguring the system's BIOS.**

Configuring a Disk for a ZFS Non-Root File System

A disk that is set up for a non-root ZFS file system is relabeled automatically when the pool is created or when the disk is added to the pool. If whole disks are used either to create a pool or to add to a ZFS storage pool, an EFI label is applied.

Generally, most modern bus types support hot-plugging. This feature enables a system to immediately recognize a disk that you insert in to an empty slot. For more information about hot-plugging devices, see Chapter 2, “Dynamically Configuring Devices”.

▼ How to Set Up a Disk for a ZFS Non-Root File System

This procedure applies to both SPARC based systems and x86 based systems. It assumes that you have installed the replacement disk on the system.

For information about adding disks, see “Adding or Replacing Disks for ZFS File Systems” on page 130.

1. **Become an administrator.**
2. **To verify that the disk is accessible, launch the Format utility.**

   ```
   # format
   ```

   Ensure that the new disk is included in the command output.
3. **Exit the Format utility.**
4. **Replace the root pool on the new disk.**

   ```
   # zpool replace non-root disk
   ```
5. **If required, bring the new disk online.**

   ```
   # zpool online non-root disk
   ```
6. **Verify that the new disk is resilvered.**

   ```
   # zpool status non-root
   ```
7. **Attach the new disk to the ZFS pool.**

   ```
   # zpool attach root-pool old-disk new-disk
   ```

Example 7-1 Setting Up a Disk for a ZFS Non Root Pool

In this example, a new disk `c1t1d0` is installed for a ZFS non root pool `tank`.

```
# zpool replace tank c1t1d0
# zpool online tank c1t1d0
# zpool status tank
# zpool attach tank mirror c1t1d0 c2t0d0
```

♦ ♦ ♦ CHAPTER 8

Configuring Storage Devices With COMSTAR

This chapter describes how to configure Common Multiprotocol SCSI TARget, or COMSTAR, a software framework that enables you to convert any Oracle Solaris 11 host into a SCSI target device that can be accessed over a storage network by initiator hosts.

This means you can make storage devices on a system available to Linux, Mac OS, or Windows client systems as if they were local storage devices. Supported storage protocols are iSCSI, FC, iSER, and SRP.

For information about configuring iSCSI initiators, see Chapter 4, “Configuring Solaris iSCSI Initiators,” in “Managing SAN Devices and Multipathing in Oracle Solaris 11.2 ”.

For information about the iSNS support in Oracle Solaris, see Chapter 9, “Configuring and Managing the Oracle Solaris Internet Storage Name Service (iSNS)”.

For troubleshooting general iSCSI configuration problems in Oracle Solaris, see “Troubleshooting iSCSI Configuration Problems” in “Managing SAN Devices and Multipathing in Oracle Solaris 11.2 ”.

Overview of COMSTAR Features

COMSTAR uses a SCSI Target Mode Framework (STMF) to manage target storage devices with the following components:

- Port providers (or plug-ins) – Implement protocols, such as Fibre Channel (FC) and iSCSI.
- Logical unit providers – Emulate various SCSI devices, such as disk and tape devices.
- The `libstmf` management library – Provides the COMSTAR management interface. The modules that implement the iSCSI functionality do not interact directly with the underlying transport. In a similar way, the modules that implement the transport protocol are unaware of the SCSI-level functionality that is inherent in the packets they are transporting. Some transport examples are Fibre Channel and iSCSI. The framework separates the execution and cleanup of SCSI commands and the associated resources. This separation simplifies the task of writing SCSI or transport modules.

  Use the following to administer these features:

- The `itadm` command manages Internet SCSI (iSCSI) nodes within the SCSI target mode framework.
- The `stmfadm` command configures logical units within the SCSI target mode framework.
- The `srptadm` command manages SCSI RDMA Protocol (SRP) target ports within the SCSI target mode framework.

Identifying COMSTAR Software and Hardware Requirements

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

Configuring Storage Devices With COMSTAR Task Map

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

- “Identifying COMSTAR Software and Hardware Requirements” on page 142
- “Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.2 ”
- “Configuring Storage Devices With COMSTAR” on page 144
- “Configuring Dynamic or Static Target Discovery” on page 143
- Chapter 4, “Configuring Solaris iSCSI Initiators,” in “Managing SAN Devices and Multipathing in Oracle Solaris 11.2 ”
- “How to Access iSCSI Disks” on page 151
- “Configuring Authentication in Your iSCSI-Based Storage Network” in “Managing SAN Devices and Multipathing in Oracle Solaris 11.2 ”
- “Setting Up iSCSI Multipathed Devices in Oracle Solaris” in “Managing SAN Devices and Multipathing in Oracle Solaris 11.2 ”
- “Monitoring Your iSCSI Configuration” in “Managing SAN Devices and Multipathing in Oracle Solaris 11.2 ”

COMSTAR Terminology

Review the following terminology before configuring target devices with COMSTAR.

Term	Description
Discovery	The process that presents the initiator with a list of available targets.
Discovery method	The way in which the iSCSI targets can be found. Three methods are currently available: ■ Internet Storage Name Service (iSNS) – Potential targets are discovered by interacting with one or more iSNS servers. ■ SendTargets – Potential targets are discovered by using a *discovery-address.* ■ Static – Static target addressing is configured.
Initiator	The driver that initiates SCSI requests to the iSCSI target.
Initiator group	A set of initiators. When an initiator group is associated with a LUN, only initiators from that group may access the LUN.
`iqn` or `eui` address format	An `iqn` (iSCSI qualified name) address is the unique identifier for a device in an iSCSI network using the form *iqn.date.authority:uniqueid*. An iSCSI initiator or target is assigned an IQN name automatically when the iSCSI initiator or target is initialized. An `eui` (extended unique identifier) address consists of 16 hexadecimal digits, and identifies a class of GUIDs that is used in both the SCSI and InfiniBand standards. SRP devices use the `eui` address format.
Logical unit	A uniquely numbered component in a storage system. When a LUN is associated with one or more SCSI targets, the target can be accessed by one or more SCSI initiators.
Target device	The iSCSI storage component.
Target group	A set of targets. A can be made available to all targets in one target group.
Target portal group	A list of IP addresses that determines which interfaces a specific iSCSI target will listen to. A TPG contains IP addresses and TCP port numbers

Configuring Dynamic or Static Target Discovery

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

- **Dynamic device discovery** – Two dynamic device discovery methods are available:
  - SendTargets – If an iSCSI node exposes a large number of targets, such as an iSCSI to Fibre-Channel bridge, you can supply the iSCSI node IP address/port combination

and allow the iSCSI initiator to use the SendTargets features to perform the device discovery.

- iSNS – The Internet Storage Name Service (iSNS) allows the iSCSI initiator to discover the targets to which it has access using as little configuration information as possible. It also provides state change notification to notify the iSCSI initiator when changes in the operational state of storage nodes occur. To use the iSNS discovery method, you can supply the iSNS server address/port combination and allow the iSCSI initiator to query the iSNS servers that you specified to perform the device discovery. The default port for the iSNS server is `3205`. For more information about iSNS, see RFC 4171:

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

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

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

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

  target,*target-address*[:*port-number*]

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

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

Configuring Storage Devices With COMSTAR

You can set up and configure a COMSTAR Internet SCSI (iSCSI) target and make it available over the network. The iSCSI features can work over a normal Internet connection (such as Ethernet) using the standard iSCSI protocol. The iSCSI protocol also provides naming and discovery services, authentication services using CHAP and RADIUS, and centralized management through iSNS.

If the system has at least one InfiniBand (IB) Host Channel Adapter (HCA) and connectivity exists between the initiator and the target, the iSCSI connection uses iSCSI Extensions for RDMA (iSER) for enhanced data throughput. iSER used with an IB HCA provides high bandwidth, low CPU utilization, and a single network connection that multiple protocols can share.

The iSER target and initiator both use a component called iSCSI Data Mover to provide iSCSI connection services. Use of iSER is automatic whenever both the initiator and the target are configured to use IP addresses that correspond to IB-capable devices.

How to Enable the STMF Service

COMSTAR uses SMF to store its current, persistent configuration, such as logical unit mapping, host group definitions, and target group definitions. When the service is enabled during boot or when using the `svcadm` command, it clears any stale configuration data inside the kernel framework, and then reloads the configuration from the SMF repository into the driver. After the configuration is loaded, any changes that are made to the configuration are automatically updated inside the driver database, as well as inside the SMF repository. For example, any changes made through the `stmfadm` command are automatically updated in both areas.

The COMSTAR target mode framework runs as the `stmf` service. By default, the service is disabled. You must enable the service to use COMSTAR functionality. You can identify the service with the `svcs` command. If you have not rebooted the server since installing the `group/feature/storage-server` package, the service might not be enabled correctly.

1. **Install the COMSTAR storage server software.**

```
target# pkg install group/feature/storage-server
Packages to install:    75
Create boot environment:    No
Services to restart:     7
DOWNLOAD                                  PKGS       FILES    XFER (MB)
Completed                                 75/75   9555/9555  105.7/105.7

PHASE                                       ACTIONS
Install Phase                             13347/13347

PHASE                                         ITEMS
Package State Update Phase                    75/75
Image State Update Phase                        2/2
Loading smf(5) service descriptions: 17/17
Loading smf(5) service descriptions: 3/3

PHASE                                         ITEMS
Reading Existing Index                          8/8
Indexing Packages                             75/75
Indexing Packages                             75/75
Optimizing Index...

PHASE                                         ITEMS
Indexing Packages                           573/573
```

2. **Enable the `stmf` service.**

```
target# svcadm enable stmf
target# svcs stmf
STATE          STIME    FMRI
online         09:42:32 svc:/system/stmf:default
```

How to Back Up and Restore a COMSTAR Configuration

After you complete your COMSTAR configuration, make a copy that can be restored, if needed.

1. **Become an administrator.**
2. **Export the current COMSTAR configuration.**

   ```
   # svccfg export -a stmf > COMSTAR.backup
   ```

3. **If necessary, restore the exported configuration.**

   ```
   # svccfg import COMSTAR.backup
   ```

How to Create a Logical Unit

The logical unit provider for creating disk-type LUs is called `sbd`. However, you must initialize the storage for the logical unit before you can share a disk-type LU.

The disk volume provided by the server is referred to as the *target*. When the LU is associated with an iSCSI target, it can be accessed by an iSCSI initiator.

The process for creating SCSI LUs is as follows:

- Initialize the storage for the LU, also known as the *backing store*.
- Create a SCSI LU by using the backing store.

When a LU is created, it is assigned a global unique identifier (GUID), for example, `600144F0B5418B0000004DDAC7C10001`. The GUID is used to refer to the LU in subsequent tasks, such as mapping a LU to select hosts.

The following steps are completed on the system that is providing the storage device.

1. **Create a ZFS storage pool.**

   ```
   target# zpool create sanpool mirror c2t3d0 c2t4d0
   ```

2. **Create a ZFS volume to be used as a SCSI LU.**

```
target# zfs create -V 2g sanpool/vol1
```

3. **Create a LU for the ZFS volume.**

```
target# stmfadm create-lu /dev/zvol/rdsk/sanpool/vol1
Logical unit created: 600144F0B5418B0000004DDAC7C10001
```

You can find the device path for the ZFS volume in the `/dev/zvol/rdsk/`*pool-name/* directory.

4. **Confirm that the LU has been created.**

```
target# stmfadm list-lu
LU Name: 600144F0B5418B0000004DDAC7C10001
```

5. **Add the LU view.**

This command makes the LU accessible to all systems.

```
target# stmfadm add-view 600144F0B5418B0000004DDAC7C10001
```

If you want to restrict the LU view to specific systems, see "How to Restrict Logical Unit Access to Selected Systems" on page 153.

6. **Verify the LU configuration.**

```
target# stmfadm list-view -l 600144F0B5418B0000004DDAC7C10001
View Entry: 0
Host group   : All
Target group : All
LUN          : 0
```

How to Create an iSCSI Target

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

1. **Enable the iSCSI target service.**

```
target# svcadm enable -r svc:/network/iscsi/target:default
```

Confirm that the service is enabled.

```
target# svcs -l iscsi/target
fmri         svc:/network/iscsi/target:default
name         iscsi target
enabled      true
state        online
next_state   none
state_time   Mon May 23 14:48:59 2011
```

```
logfile      /var/svc/log/network-iscsi-target:default.log
restarter    svc:/system/svc/restarter:default
dependency   require_any/error svc:/milestone/network (online)
dependency   require_all/none svc:/system/stmf:default (online)
```

2. **Create the iSCSI target.**

```
target# itadm create-target
Target iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-c3b36c039405 successfully created
```

3. **Display the iSCSI target information.**

```
target# itadm list-target -v
TARGET NAME                                              STATE    SESSIONS
iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-c3b36c039405  online   0
alias:                  -
auth:                   none (defaults)
targetchapuser:         -
targetchapsecret:       unset
tpg-tags:               default
```

How to Enable iSNS Discovery for the Target Device

Use this method if you have at least one iSNS server on your network. This method enables the iSCSI initiator to discover targets with as little configuration as possible, while also providing state-change notification. State-change functionality notifies the iSCSI initiator when there are changes in the operational state of a target. For this method, you must supply the iSNS server address and port combination. The iSCSI initiator can then query the specified iSNS server to perform device discovery. The default port for an iSNS server is 3205.

1. **Become an administrator.**

2. **Add the iSNS server information.**

```
target# itadm modify-defaults --isns-server ip-address
```

Identify the *ip-address* of the iSNS server in your network.

This step adds the iSNS server information to all of the Oracle Solaris iSCSI targets.

3. **Enable iSNS server discovery.**

```
target# itadm modify-defaults --isns enable
```

▼ How to Configure an IB HCA for iSER

An InfiniBand (IB) Host Channel Adapter (HCA) is required to take full advantage of the iSCSI Extensions for RDMA (iSER) capabilities. To use iSER, you must configure the HCA on both the target and the initiator.

1. **Become an administrator on the host (initiator) system.**

2. **Connect the HCA to an IB switch.**

   See the vendor documentation for details.

3. **Configure the target and the initiator for the HCA.**

   The target and the initiator must be on the same subnet. This example uses `ibd0` as the driver.

   ```
   # ipadm create-ip ibd0
   ```

4. **Configure the IP address and port combination for the HCA.**

   ```
   # ipadm create-addr -T static -a local=10.1.190.141/24 ibd0/v4addr
   ```

5. **Verify the interface configuration.**

   ```
   # ipadm show-addr
   ADDROBJ           TYPE     STATE        ADDR
   lo0/v4            static   ok           127.0.0.1/8
   e1000g0/_b        dhcp     ok           10.1.190.141/24
   lo0/v6            static   ok           ::1/128
   e1000g0/_a        addrconf ok           fe80::214:4fff:fe27:360c/10
   ```

6. **Become an administrator on the target system and repeat steps 3-5 for all other HCA hosts on the network.**

7. **Verify connectivity on both the target and the initiator.**

   ```
   target# ping initiator-ip
   initiator# ping target-ip
   ```

Creating iSCSI Target Portal Groups

You can create a target portal group (TPG) to manage the discovery of multiple iSCSI and iSER targets. A TPG is a list of IP addresses to determine upon which interfaces a specific iSCSI target will listen.

A TPG contains IP addresses and TCP port numbers. To use this capability, you need to do the following:

- Create a TPG as a list of *ip-address:port* specifiers by using the `itadm create-tpg` command.
- Bind a specific iSCSI target to a TPG by using the `itadm modify-target -t` command.
- When an iSCSI target is made active, an iSCSI listener is created for each IP address and port belonging to a TPG associated with that target.

A TPG is an efficient way to control which targets are discovered through specific ports. For example, you could restrict your iSCSI target so that it is available only through one specific IP address or only through a set of iSER-capable IP addresses.

Note - Do not confuse target portal groups with target groups. A target group is a list of SCSI target ports that are all treated the same when creating views. Creating a view can help you facilitate LU mapping. Each view entry specifies a target group, a host group, and a LU. For more information on Target Groups and LUN mapping, see "Making SCSI Logical Units Available" on page 152 and `stmfadm`(1M).

To learn about static and iSNS target discovery, see "Configuring Dynamic or Static Target Discovery" on page 143. The iSCSI initiator uses the `iscsiadm` command to discover TPGs. For more information, see `iscsiadm`(1M) and `itadm`(1M).

Using TPGs with iSER

When you use the SendTargets discovery and iSER at the same time, a common convention is to use a TPG to associate a specific iSCSI target port with only iSER-capable IP addresses. For example, if a target system has four IP addresses, A, B, C, and D, and only addresses B and C are iSER-capable, then addresses B and C could be added to a TPG, and assigned to a target T.

An iSCSI initiator with both Ethernet and InfiniBand (IB) interfaces could use the SendTargets discovery method to discover the possible storage targets. Without the use of TPGs, the initiator might always prefer the use of the Ethernet interfaces over the IB interfaces. By associating target T only with the IB interfaces, the initiator correctly prefers using its IB-capable interface when connecting to target T.

▼ How to Create a Target Portal Group for iSCSI Targets

You can create a target portal group (TPG) by providing a unique name, and a TPG Tag (ranging from 2–65535) is automatically generated. TPG Tag 1 is reserved for the default TPG that is used when you do not explicitly set a TPG on the target. The portal for the default TPG matches requests from all network interfaces on port `3260`.

The following steps shows how to create two TPGs, TPGA and TPGB, that use port `8000` for the IP addresses in TPGB.

1. **Become an administrator.**

2. **Create two TPGs.**

```
target# itadm create-tpg TPGA 192.168.0.1 192.168.0.2
target# itadm create-tpg TPGB 192.168.0.2:8000 192.168.0.2:8000
```

Note - IPv4 portals are specified in dotted address notation (for example, `192.168.0.1`). IPv6 portal addresses must be enclosed in square brackets.

---

3. **Configure an existing iSCSI target to use the TPGs, TPGA and TPGB.**

```
# itadm modify-target -t TPGA,TPGB eui.20387ab8943ef7548
```

4. **Verify the TPGs that you created.**

```
# itadm list-tpg -v
```

You can remove a TPG with the `itadm delete-tpg` command.

How to Access iSCSI Disks

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

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

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

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

```
initiator# format
0. c0t600144F0B5418B0000004DDAC7C10001d0 <SUN-COMSTAR-1.0 cyl 1022 alt 2 hd 128 sec 32>
/scsi_vhci/disk@g600144f0b5418b0000004ddac7c10001
1. c8t0d0 <Sun-STK RAID INT-V1.0 cyl 17830 alt 2 hd 255 sec 63>
/pci@0,0/pci10de,375@f/pci108e,286@0/disk@0,0
2. c8t1d0 <Sun-STK RAID INT-V1.0-136.61GB>
/pci@0,0/pci10de,375@f/pci108e,286@0/disk@1,0
3. c8t2d0 <Sun-STK RAID INT-V1.0-136.61GB>
/pci@0,0/pci10de,375@f/pci108e,286@0/disk@2,0
```

```
4. c8t3d0 <Sun-STK RAID INT-V1.0 cyl 17830 alt 2 hd 255 sec 63>
/pci@0,0/pci10de,375@f/pci108e,286@0/disk@3,0
Specify disk (enter its number): 0
selecting c0t600144F0B5418B0000004DDAC7C10001d0
[disk formatted]
```

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

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

```
initiator# zpool create pool-name c0t600144F0B5418B0000004DDAC7C10001d0
initiator# zfs create pool-name/fs-name
```

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

Making SCSI Logical Units Available

Simply registering a logical unit (LU) with the STMF framework does not make it available to hosts (initiators) on the network. This section describes how to make LUs visible to initiator hosts for the following configurations.

For iSCSI, Fibre Channel, and FCoE configurations, a LU must be mapped before it can be accessed. You can choose one of the following methods, both of which use the `stmfadm` command:

- **Simple mapping** – Exposes the LU to all initiators through all the ports, using one command. Making LUs available to all hosts uses this method.
- **Selective mapping**– Enables you to specify the hosts that can access the LU. Making LUs available to selected hosts uses this method. This process includes the following steps:
  1. Defining host groups – A host group is a name given to a set of hosts (initiators) that are allowed to access the same LUs. This step is not needed if the same set of LUs is visible to all the hosts, as in simple mapping.
  2. Defining target groups – A target group is a name given to a set of target ports that export the same set of LUs to the same set of host groups. This step is not needed if the same set of LUs is visible to all the hosts, as in simple mapping.
  3. Adding one or more views for each logical unit – Adding a view creates a bridge between the LU and the host initiator. When an initiator from the host group logs in to a target port contained in the target group, the LU is visible.

Note - Don't confuse a target group with a target portal group (TPG). A TPG is a list of IP addresses that an iSCSI target listens to. A TPG can help you restrict an iSCSI target so that it is available only through one specific IP address. For more information on target groups, see `stmfadm`(1M).

A view entry consists of four components: host group, target group, logical unit number (LUN), and LU identifier. Of these four components, only the LU identifier is required. If the other components are omitted, the following default values are assumed:

- If the host group is omitted, the all initiators value is assumed.
- If the target group is omitted, the all targets value is assumed.
- If the LUN is omitted, the system chooses a suitable LUN for the entry.

How to Make a Logical Unit Available to All Systems

This procedure makes a LU available to all initiator hosts on a storage network.

1. **Obtain the Global Unique Identification (GUID) number for the LU.**

```
# stmfadm list-lu -v
```

2. **Add a view for the logical unit.**

```
3 # stmfadm add-view GUID-number
```

How to Restrict Logical Unit Access to Selected Systems

Use this procedure to restrict LU access to selected hosts on a storage network. If you are using Fibre Channel ports, first identify the World Wide Names (WWN). Then, selectively map a logical unit number (LUN) to the ports on `host-a`, for example. A target group (`targets-0`) is also defined for a given set of target ports that export the same set of LUs to the same host group.

For information about configuring a target group, see `stmfadm`(1M).

1. **Become an administrator.**

2. **Identify the Fibre Channel (FC) port on the initiator.**

```
initiator# fcinfo hba-port
HBA Port WWN: 210000e08b195dae
Port Mode: Initiator
Port ID: 0
OS Device Name: /dev/cfg/c8
Manufacturer: QLogic Corp.
Model: 375-3108-xx
Firmware Version: 03.03.28
FCode/BIOS Version:  fcode: 1.13;
Serial Number: not available
Driver Name: qlc
Driver Version: 20100408-3.01
Type: unknown
State: offline
Supported Speeds: 1Gb 2Gb
Current Speed: not established
Node WWN: 200000e08b195dae
NPIV Not Supported
.
.
.
```

3. **Create a host group.**

```
target# stmfadm create-hg host-a
```

4. **Add the WWNs identified in the preceding output as members of the host group.**

```
target# stmfadm add-hg-member -g host-a  210000e08b195dae 210100e08b395dae
```

5. **Create a target group.**

```
target# stmfadm create-tg targets-0
```

6. **Specify the target group members by adding the target names.**

   Each SCSI target can be a member of only one target group.

```
target# stmfadm add-tg-member -g targets-0 wwn.guid-number
```

7. **Identify the GUID number for the LU.**

```
target# stmfadm list-lu -v
```

8. **Make the LU available by adding a view entry, specifying the host group name and the LU GUID number.**

```
target# stmfadm add-view -h host-a -t targets-0 -n 1 guid-number
```

Configuring Fibre Channel Devices With COMSTAR

You can set up and configure a COMSTAR target on a SPARC system or x86 system in a FC network environment and make it accessible to the storage network. Ensure that the logical unit has been created. For instructions, see “How to Create a Logical Unit” on page 146,

Configuring Fibre Channel Ports With COMSTAR

The Fibre Channel (FC) port provider can use the same HBAs that are used for the FC initiators. A given FC port can be used as an initiator or as a target, but not as both. You can also configure a dual port or quad port FC HBA so that a subset of the HBA ports is in target mode and the rest of the ports are in initiator mode.

The procedures in this section are specific to QLogic HBAs. The driver to use with HBAs in initiator mode is `qlc`, which is a QLogic driver. The `qlc` driver works only in initiator mode, and cannot be used for target mode. The COMSTAR driver for a target QLogic 4G HBA is `qlt`.

Because initiator mode and target mode use different drivers, the driver you attach to an HBA port defines its function as a target or initiator. You can specify a driver for all the ports by specifying the PCI device ID of the HBA. Or, you can configure the driver on a port-by-port basis. Both methods use the `update_drv` command and are described in this section. For more information, see `update_drv`(1M).

▼ How to Display Existing FC Port Bindings

Before making changes to the HBA ports, first check the existing port bindings.

1. **Become an administrator.**
2. **Display what is currently bound to the port drivers.**

   In this example, the current binding is `pciex1077,2432`.

```
# mdb -k
Loading modules: [ unix krtld genunix specfs ...
> ::devbindings -q qlc
30001617a08 pciex1077,2432, instance #0 (driver name: qlc)
300016177e0 pciex1077,2432, instance #1 (driver name: qlc)
> $q
```

▼ How to Set All FC Ports to a Specific Mode

This procedure changes all ports on all HBAs with a specific PCI device ID to target mode. The PCI device ID number binds a driver to a port, thereby setting all the HBA ports with that PCI device ID, for example, all QLogic 4G PCI express HBAs, to target mode.

1. **Become an administrator.**

2. **Remove the current binding.**

   In this example, the `qlc` driver is actively bound to `pciex1077,2432`. You must remove the existing binding for `qlc` before you can add that binding to a new driver. Single quotation marks are required in this syntax.

   ```
   # update_drv -d -i 'pciex1077,2432' qlc
   Cannot unload module: qlc
   Will be unloaded upon reboot.
   ```

   This message does *not* indicate an error. The configuration files have been updated but the `qlc` driver remains bound to the port until the system is rebooted.

3. **Establish the new binding.**

   In this example, `qlt` is updated. Single quotes are required in this syntax.

   ```
   # update_drv -a -i 'pciex1077,2432' qlt
   devfsadm: driver failed to attach: qlt
   Warning: Driver (qlt) successfully added to system but failed to attach
   ```

   This message does not indicate an error. The `qlc` driver remains bound to the port, until the system is rebooted. The `qlt` driver attaches when the system is rebooted.

4. **Reboot the system to attach the new driver. Then, recheck the bindings.**

   ```
   # init 6
   .
   .
   .
   # mdb -k

   Loading modules: [ unix krtld genunix specfs dtrace ...
   > ::devbindings -q qlt
   30001615a08   pciex1077,2432, instance #0 (driver name: qlt)
   30001615e0    pciex1077,2432, instance #1 (driver name: qlt)
   > $q
   ```

5. **Verify that the target mode framework has access to the HBA ports.**

   ```
   # stmfadm list-target -v
   Target: wwn.210100E08BA54E60
   Operational Status : Offline
   Provider Name : qlt(1)
   ```

```
Alias : -
Sessions : 0
Target: wwn.210100E08BA54E60
Operational Status : Offline
Provider Name : qlt(0)
Alias : -
Sessions : 0
```

▼ How to Set Selected FC Ports to Initiator or Target Mode

This procedure uses *path-based bindings*. It shows you how to use a specific device path to bind a port to a driver that is different from the driver to which it is currently bound.

1. **Become an administrator.**

2. **Display a list of the HBA ports and their respective device paths.**

   This example shows the device paths for a single HBA with two ports.

   ```
   # luxadm -e port
   /devices/pci@780/QLGC,qlc@0,1/fp@0,0:devctl CONNECTED
   /devices/pci@780/QLGC,qlc@0/fp@0,0:devctl CONNECTED
   ```

3. **Set the top port to target mode, and leave the bottom port in initiator mode.**

   Remove the initial `/devices` portion of the path, and include everything up to `/fp@0...`. The path with the `/devices` portion removed is the path to which the system binds the `qlt` driver.

   Single quotation marks are required in this syntax.

   ```
   # update_drv -a -i '/pci@780/QLGC,qlc@0,1' qlt
   devfsadm: driver failed to attach: qlt
   Warning: Driver (qlt) successfully added to system but failed to attach.
   ```

   This message does not indicate an error. The **`qlc`** driver remains bound to the port until reboot. The **`qlt`** driver attaches during reboot.

4. **Reboot the system to attach the new driver. Then, recheck the bindings.**

   You should see that the port changed from initiator mode (`qlc`) to target mode (`qlt`).

   ```
   # init 6
   .
   .
   .
   # mdb -k
   .
   .
   .
   > $q
   ```

5. **Verify that the target mode framework has access to the HBA ports.**

```
# stmfadm list-target -v
```

Making Logical Units Available for FC and FCoE

Simply registering a logical unit (LU) with the STMF framework does not make it available to hosts (initiators) on the network. You must make logical units visible to initiator hosts for Fibre Channel and FCoE configurations by mapping the logical unit. To determine which method to use and how to map the logical unit, see "How to Make a Logical Unit Available to All Systems" on page 153. Both methods use the `stmfadm` command. The additional steps below are for FC and FCoE configurations.

▼ How to Make Logical Units Available for FC and FCoE

This procedure makes the LU available to all hosts or selected hosts for FC or FCoE configurations on a storage network. The steps are run on the host.

1. **Become an administrator.**

2. **Make a LU available to hosts.**

   Obtain the global unique identification (GUID) number for the LU.

```
# sbdadm list-lu
# stmfdm list-lu -v
```

   Identify the WWNs for the FC or FCoE ports of the host.

```
# fcinfo hba-port
HBA Port WWN: *210000e08b83378d*
OS Device Name: /dev/cfg/c4
Manufacturer: QLogic Corp.
Model: QLA2462
Firmware Version: 4.0.27
Fcode/BIOS Version: N/A
Type: N-port
State: online
Supported Speeds: 1Gb 2Gb 4Gb
Current Speed: 4Gb
Node WWN: 210000e08b83378d
HBA Port WWN: *210100e08ba3378d*
OS Device Name: /dev/cfg/c5
Manufacturer: QLogic Corp.
Model: QLA2462
Firmware Version: 4.0.27
Fcode/BIOS Version: N/A
Type: N-port
```

```
State: online
Supported Speeds: 1Gb 2Gb 4Gb
Current Speed: 4Gb
Node WWN: 210100e08ba3378d
```

3. **Add a view and perform mapping.**

   Follow the instructions in "How to Make a Logical Unit Available to All Systems" on page 153.

4. **Verify that the LU is visible on an Oracle Solaris initiator host by running the following script.**

```
#!/bin/ksh
fcinfo hba-port |grep "^HBA" | awk '{print $4}' | while read 1n
do
fcinfo remote-port -p $1n -s >/dev/null 2>&amp;1
done
```

   All subsequent LUNs appear in the `format` output, because the script forces the initiator to touch all the LUNs through all the ports. If you do not see the LUNs, run the `format` command again. If you still do not see the LUNs, ensure that the service is enabled on the target by using the `svcs stmf` command. Also ensure that you added view entries for the LU, as described in "How to Make a Logical Unit Available to All Systems" on page 153.

5. **Verify that the LU is visible on other systems.**

   - For a Linux initiator host, verify that the LU is visible by running the utility provided by the HBA vendor. The utility scans for configuration changes.
   - For a Windows initiator host, verify that the logical unit is visible by selecting Control Panel → Administrative Tools → Computer Management → Disk Management. Then, from the Action menu, choose Rescan Disks.

Configuring FCoE Devices With COMSTAR

You can set up and configure a COMSTAR target in a Fibre Channel over Ethernet (FCoE) network environment, then make it accessible to the storage network. Ensure that your system meets the necessary prerequisites before you begin:

- "How to Create a Logical Unit" on page 146
- "How to Create an iSCSI Target" on page 147
- Chapter 4, "Configuring Solaris iSCSI Initiators," in "Managing SAN Devices and Multipathing in Oracle Solaris 11.2 "

Configuring FCoE Ports

FCoE functionality is provided through Ethernet interfaces. Fibre Channel over Ethernet (FCoE) ports are logical entities associated with Ethernet interfaces. Within an Oracle Solaris system, a one-to-one mapping of FCoE ports and Ethernet interfaces exist. You can associate only one FCoE port with a given Ethernet interface. FCoE and IP cannot share the same Ethernet interface. So, before you create an FCoE port on an Ethernet interface, ensure that the interface has been unplumbed.

The FCoE port configuration is persistent across reboots. All configured FCoE ports are created and placed online automatically after the system reboots.

For FCoE target ports, you must enable the following service before creating FCoE target ports to get persistent behavior.

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

Enabling 802.3x PAUSE and Jumbo Frames on the Ethernet Interface

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

Perform one of the following prerequisite steps before you create an FCoE port on an Ethernet interface:

- Enable 802.3x (also called PAUSE) on the Ethernet interface.

  Doing so ensures a consistent Ethernet transport.
- Enable jumbo frames (> 2.5 Kbyte) on the Ethernet interface.

  A Fibre Channel data frame can be as large as 2136 bytes.

▼ How to Create FCoE Target Ports

1. **Create an FCoE target port on a specified network interface.**

   ```
   # fcadm create-fcoe-port -t nxge0
   ```

If the Ethernet interface you selected does not support multiple unicast address (for example, the VMware network interface), you are prompted to explicitly enable promiscuous mode on that interface.

2. **Enable promiscuous mode, if prompted.**

```
# fcadm create-fcoe-port -t -f e1000g0
```

If no error messages appear, the FCoE target port is created, and the FCoE target is online.

▼ How to Verify That an FCoE Target Port Is Working

1. **Display the FCoE ports that you created.**

```
# fcadm list-fcoe-ports
HBA Port WWN: 200000144fda7f66
Port Type: Target
MAC Name: nxge0
MTU Size: 9194
MAC Factory Address: 00144fda7f66
MAC Current Address: 0efc009a002a
Promiscuous Mode: On
```

2. **Display all target mode Fibre Channel HBA ports on the host.**

```
# fcinfo hba-port -t
HBA Port WWN: 200000144fda7f66
Port Mode: Target
Port ID: 9a002a
OS Device Name: Not Applicable
Manufacturer: Sun Microsystems, Inc.
Model: FCoE Virtual FC HBA
Firmware Version: N/A
FCode/BIOS Version: N/A
Serial Number: N/A
Driver Name: COMSTAR FCOET
Driver Version: 1.0
Type: F-port
State: online
Supported Speeds: 1Gb 10 Gb
Current Speed: 10Gb
Node WWN: 100000144fda7f66
```

3. **View a list of the FCoE target ports.**

The -v option displays additional information about the target, along with SCSI session information for logged-in initiators.

```
# stmfadm list-target -v
Target wwn.200000144FDA7F66
Operational Status: Online
```

```
Provider Name     : fcoet
Alias             : fcoet1
Sessions          : 1
Initiator: wwn.210000E08B818343
Alias: #QLA2342 FW:v3.03.25 DVR:v8.02.14.01
Logged in since: ...
```

▼ How to Delete FCoE Target Ports

You can disable FCoE functionality when needed.

1. **Take the FCoE target port offline.**

   # **stmfadm offline-target** *wwn.200000144fda7f66*

2. **Remove the FCoE target port.**

   # **fcadm delete-fcoe-port** *nxge0*

Configuring SRP Devices With COMSTAR

The SCSI RDMA Protocol accelerates the SCSI protocol by mapping the SCSI data transfer phases to Infiniband (IB) Remote Direct Memory Access (RDMA) operations. As a result, an SRP initiator can read and write data from a COMSTAR SRP target at high data rates with relatively low CPU utilization.

You can set up and configure a COMSTAR SRP target and make it available over an Infiniband (IB) fabric. The SRP target is available wherever a supported IB Host Channel Adapter (HCA) is installed on the target system.

- **One SCSI target per IB HCA** – The COMSTAR SRP target uses a simple model in which each supported IB HCA is registered as a SCSI target. The SCSI target is a virtual object that contains a task router and acts as a the connection between the SCSI transport (in this case, SRP) and the SCSI back end (STMF and SBD).

  An HCA can contain multiple physical ports. The same SCSI target is shared between all the ports of an HCA. The SCSI target representing the HCA is automatically available for incoming connections through all the active ports of that HCA.
- **SRP target eui identifier** – In the IB architecture, each HCA and each port is assigned a 64-bit GUID by the manufacturer. The COMSTAR SCSI target created for each HCA is given a name corresponding to the GUID of that HCA, which has the format: `eui.`*HCA-GUID*. For example, if the target system includes a supported IB HCA with an HCA GUID of `0003BA0001002E48`, then a SCSI target will be created using the name `eui.0003BA0001002E48`. The string `eui` stands for *extended unique identifier* and names a class of GUIDs that is used in both the SCSI and IB standards.

- **SRP initiator eui identifier** – Similarly, SRP uses a 64-bit initiator GUID to identify the initiator system. The choice of which GUID to use is determined by the SRP initiator implementation. Many initiators use the GUID of the HCA that is being used for the outgoing connection. For example, an initiator using the GUID of `0003BA0001002EA5` is known to COMSTAR as `eui.0003BA0001002EA5`.

Using COMSTAR Views With SRP

The COMSTAR view facility can be used to create target groups and host groups that restrict and configure which logical units (LUs) can be accessed through each SCSI target or initiator as described in "Making SCSI Logical Units Available" on page 152. The `eui` identifier of the SRP initiator is added to a host group. The `eui` identifier of the SRP SCSI Target is added to a target group. The view entries for each LU then determine the particular set of LUs that each initiator can access.

▼ How to Enable the SRP Target Service

The COMSTAR port provider for the COMSTAR SRP target is managed by the Service Management Facility (SMF). The primary SRP target service is `svc:/system/ibsrp/target:default`, which can be abbreviated to `ibsrp/target`.

The SRP package is `storage/scsi-rdma/scsi-rdma-target`.

1. **Become an administrator.**
2. **Recursively enable the SRP target service.**

```
# svcadm enable -r ibsrp/target
```

3. **Display the SRP target service information.**

```
# svcs -l ibsrp/target
```

▼ How to Verify SRP Target Status

1. **Become an administrator.**
2. **Verify the presence of the expected SRP SCSI target on the system.**

```
# srptadm list-target
Target HCA 21280001A0D0F0:
Enabled            : true
```

```
SRP Target Name     : eui.0021280001A0D0F0
Operational Status  : online
```

♦ ♦ ♦ CHAPTER 9

Configuring and Managing the Oracle Solaris Internet Storage Name Service (iSNS)

This chapter provides an overview of the Internet Storage Name Service (iSNS), and describes how to configure the Oracle Solaris iSNS server, manage the iSNS server, and manage iSNS clients.

This is a list of the information in this chapter:

About the iSNS Technology

The Internet Storage Name Service (iSNS) is a protocol that allows dynamic discovery of iSCSI initiators and targets within an IP storage area network *SAN*. The iSNS protocol enables identification, connection to, and management of iSCSI devices by providing the following services:

- **Name registration and discovery**: The source of data that is to be stored (known as the *initiator*) and the storage object (known as the *target*) register their attributes and address, and then can obtain information about accessible storage devices dynamically.
- **Discovery domains and logon control**: Resources in a typical storage network are divided into groups called *discovery domains*, which can be administered through network management applications. Discovery domains enhance security by providing access control to targets that are not enabled with their own access controls, while limiting the logon process of each initiator to a relevant subset of the available targets in the network.
- **State-change notification**: The iSNS server notifies relevant iSNS clients of network events, for example, a newly created disk Logical Unit Number (LUN), storage resources going offline, discovery domain membership changes and link failures in a network. These notifications let a network quickly adapt to changes in topology, which is key to scalability and availability. This is an optional service.
- **Entity status inquiry**: The iSNS server verifies that a iSNS client is available. As a result, a status change notification might be issued. This is an optional service.

In a simple configuration, the source of data that is to be stored (the initiator) exchanges data with a storage object (the target). The initiator can locate the target and the target always recognizes the initiator. For example, the Oracle StorageTek™ 5320 Network Attached Storage (NAS) appliance is a iSCSI target because it stores data. The data comes from various iSCSI clients such as a data management applications or network interface cards which act as initiators. However, in large and complex configurations, it is difficult and time-consuming to configure every initiator for every target and for every target to recognize every initiator. The iSNS server resolves this by using discovery and security mechanisms to dynamically and automatically identify initiators and targets, and manage their connections to authorized resources.

After a Oracle Solaris system has been configured as an iSNS server, all targets and initiators can register with the server. The targets and initiators become iSCSI *clients* or *nodes* of the iSNS server. These clients are members of the *default discovery domain*, the only domain in the *default discovery domain set*. When you enable the default discovery domain set, the iSNS server can provide the iSCSI Name Service (iSNS) for the clients in a simple manner.

To take advantage of the iSCSI Name Service's abilities, create several discovery domain sets and discovery domains. Then assign the clients to different domains, overlapping their memberships. The iSNS server keeps track of the clients' status as a member of one or more discovery domains. For example, when a new storage device is added to the storage network and is registered with the iSNS server, it is in the default discovery domain in the default discovery domain set. You then assign this target to the discovery domains whose initiators will use it as a resource. The iSNS server then removes this target as a member of the default discovery domain in the default discovery domain set.

All initiators and targets are assigned to at least one discovery domain. Assigning an initiator to one discovery domain restricts its access to those targets in the same discovery domain set. Assigning an initiator to several discovery domains allows it to find and use targets in all of the discovery domain sets that include the initiator's discovery domain. You can manage access to clients by disabling and enabling their discovery domain sets without affecting the clients in other discovery domain sets.

For example, a site has two discovery domain sets in addition to the default one: Production and Research. Within the two discovery domain sets are three domains in addition to the default one: Development, Operations, and Finance. The Development discovery domain is in the Research discovery domain set, Operations is in the Production domain set, and Finance is a member of both discovery domain sets. Each client has been assigned to the discovery domain set that uses it the most. A data application in the Operations discovery domain can locate and get access to storage devices in the Production discovery domain set because it is a member of that discovery domain set but it cannot get access to a storage device in the Research discovery domain set. A data application in the Finance discovery domain can locate storage devices in both the Production and Research discovery domain sets because it is a member of both sets. If the Research discovery domain set were disabled, initiators in the Finance discovery domain would not have access to the Research storage devices but would continue to have access to those in the Production discovery domain set.

Configuring the iSNS Server

You can configure the iSNS server using as described in the following task maps and sections.

Task		For Instructions
1. Accept the default properties of the iSNS server or change them.		
	A. Notification of state changes of the server	“How to Set Notifications for Changes in Server State” on page 169
	B. Number of attempts to determine a client's availability	“How to Set the Number of Retries for Client Inquiries” on page 169
	C. Location of file that stores client data.	“How to Specify the Data Store Location” on page 170
2. Enable the iSNS server and display the settings.		“How to Install the iSNS Server Package” on page 168 “How to Display the Current Server Configuration” on page 171
3. Register all clients with the iSNS server.		Use the client's management interface's iSCSI configuration function to specify the IP address of the i SNS server and to allow discovery.
4. Enable the default discovery domain set.		“How to Enable the Default Discovery Domain Set” on page 171

After these tasks, the iSNS server is operating in a minimal manner. All of the clients are in the default discovery domain and are unassigned. Each client can identify and get access to all of the other clients.

5. Create the discovery domain sets for your site.	“How to Create the Discovery Domain Sets” on page 171
6. Create the discovery domains for your site.	“How to Create the Discovery Domains” on page 172
7. Add each discovery domain to one or more discovery domain set.	“How to Add a Discovery Domain to a Discovery Domain Set” on page 173
8. Assign clients to one or more discovery domains.	“How to Assign Clients to a Discovery Domain” on page 174
9. Verify the membership of clients in discovery domains and the membership of discovery domains in discovery domain sets.	“How to Display the Status of a Discovery Domain Set” on page 175 “How to Display the Status of a Discovery Domain” on page 176 “How to Display the Status of Clients” on page 176

The next section provides instructions for setting up the iSNS environment. The following topics are discussed:

- “Setting Up the iSNS Administrative Settings” on page 168
- “Using the Command Line Interface to Configure iSNS” on page 170

Setting Up the iSNS Administrative Settings

This section provides the procedures for changing the default administrative settings of the iSNS service and for starting the iSNS daemon. If you change a setting after the iSNS server has been started, you need to refresh the iSNS server. If you change the data store location, you need to restart the iSNS server.

The following tasks are described in this section:

- “How to Install the iSNS Server Package” on page 168
- “How to Set Notifications for Changes in Server State” on page 169
- “How to Set the Number of Retries for Client Inquiries” on page 169
- “How to Specify the Data Store Location” on page 170

For more details about these operations, see the `isns`(1M) man page.

▼ How to Install the iSNS Server Package

Install the iSNS server package and start the iSNS service.

1. **Become an administrator.**

2. **Install the iSNS server package.**

```
# pkg install service/storage/isns
```

3. **Enable the iSNS service.**

```
# svcadm enable isns_server
```

4. **Verify that the service is running.**

```
# svcs svc:/network/isns_server:default
STATE          STIME    FMRI
online         16:10:49 svc:/network/isns_server:default
```

How to Set Notifications for Changes in Server State

By default, all clients are notified when the iSNS server is not available. To disable these notifications, change the `Management_SCNs_Enabled` property.

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`
   - `solaris.smf.manage.isns`
   - `solaris.smf.value.isns`

   For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **Use the `svccfg` command to disable the property:**

   ```
   # svccfg -s svc:/network/isns_server setprop config/Management_SCN_Enabled=false
   ```

3. **Reload the server configuration:**

   ```
   # svcadm refresh svc:/network/isns_server
   ```

How to Set the Number of Retries for Client Inquiries

The default number of retries is 3. If the server does not get a response to three inquiries, it registers that client as unavailable. To change the number of retries, change the value of the `ESI Retry Threshold` property.

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`
   - `solaris.smf.manage.isns`
   - `solaris.smf.value.isns`

   For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **Use the `svccfg` command to change the property to, for example, 6 retries:**

   ```
   # svccfg -s svc:/network/isns_server setprop config/ESI_retry_threshold_count=6
   ```

3. **Reload the server configuration:**

```
# svcadm refresh svc:/network/isns_server
```

▼ How to Specify the Data Store Location

The default location and name for the file that contains the client data is `/etc/isns/isnsdata.xml` If you have a complex network environment that includes one or more backup iSNS servers, the data store must reside in a common location so that all servers can use it. Use the `data_store_location` property to specify the new location. You can also change the name of the file.

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`
   - `solaris.smf.manage.isns`
   - `solaris.smf.value.isns`

   For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **Use the `svccfg` command to change the location to, for example, `/etc/isns2/isns_data.xml`:**

```
# svccfg -s svc:/network/isns_server setprop config/data_store_location="/etc/isns2/
isns_data.xml"
```

3. **If you change the data store location after the server has been enabled, you must restart the server:**

```
# svcadm restart svc:/network/isns_server
```

Using the Command Line Interface to Configure iSNS

This section provides the procedures for configuring the iSNS server using the command line interface.

The following tasks are described in this section:

- “How to Display the Current Server Configuration” on page 171
- “How to Enable the Default Discovery Domain Set” on page 171

- "How to Create the Discovery Domain Sets" on page 171
- "How to Create the Discovery Domains" on page 172
- "How to Add a Discovery Domain to a Discovery Domain Set" on page 173
- "How to Assign Clients to a Discovery Domain" on page 174

These procedures use the `isnsadm`(1M) command. See the man page for a complete description of all of the command options.

▼ How to Display the Current Server Configuration

- **The following command shows the properties of the iSNS server:**

```
# isnsadm show-config
Data Store Location: /etc/isns/isnsdata.xml
Entity Status Inquiry Non-Response Threshold: 3
Management SCN Enabled: yes
Authorized Control Node Names: -
```

▼ How to Enable the Default Discovery Domain Set

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`
   - `solaris.smf.manage.isns`
   - `solaris.smf.value.isns`

   For more information about roles, see Chapter 3, "Assigning Rights in Oracle Solaris," in "Securing Users and Processes in Oracle Solaris 11.2 ".

2. **Activate the default discovery domain set:**

```
# isnsadm enable-dd-set Default
```

▼ How to Create the Discovery Domain Sets

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`

- `solaris.smf.manage.isns`
- `solaris.smf.value.isns`

For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **Create a discovery domain set:**

```
# isnsadm create-dd-set set_name
```

3. **Enable the discovery domain set:**

```
# isnsadm enable-dd-set set_name
```

4. **View all the discovery domain sets, including the new one:**

```
# isnsadm list-dd-set -v
DD Set name: Default
State: Enabled
DD Set name:set_name
State: Enabled
```

The list of discovery domain sets includes the default discovery domain set as well as the new one.

▼ How to Create the Discovery Domains

New discovery domains are members of the default discovery domain set. After you create them, you add them to the new discovery domain set.

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

- `solaris.isnsmgr.write`
- `solaris.smf.manage.isns`
- `solaris.smf.value.isns`

For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **Create the discovery domain:**

```
# isnsadm create-dd domain_name
```

3. **View the new discovery domain in the Default discovery domain set:**

```
# isnsadm list-dd-set
DD name: name
DD set(s): Default
```

4. **Create other discovery domains.**

▼ How to Add a Discovery Domain to a Discovery Domain Set

This task removes the discovery domain from the default discovery domain set and adds it the discovery domain set that you specify. Because the new discovery domain set has been enabled, all the clients in its discovery domains can be discovered by the iSNS server.

You do not need to have privileges to list the members of the discovery domains and discovery domain sets.

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`
   - `solaris.smf.manage.isns`
   - `solaris.smf.value.isns`

   For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **List the discovery domains to identify the one you want to add.**

   ```
   # isnsadm list-dd -v
   ```

3. **List the discovery domain sets to identify the one you want as the container for the new discovery domain.**

   ```
   # isnsadm list-dd-set
   ```

4. **Move the discovery domain to the discovery domain set that you want:**

   ```
   # isnsadm add-dd domain_name -s set_name
   ```

5. **View the new addition to the discovery domain set:**

   ```
   # isnsadm list-dd-set -v domain_name
   ```

▼ How to Assign Clients to a Discovery Domain

Before You Begin Use the client's management interface to register the client. Using the iSCSI configuration function, specify the IP address of the iSNS server and allow discovery of the client by the iSNS server.

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`
   - `solaris.smf.manage.isns`
   - `solaris.smf.value.isns`

   For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **Verify that the client has been registered with the iSNS server:**

```
# isnsadm list-node
iSCSI Name: iqn.1986-03.com.sun:01:000e0c9f10da.45173FEA.engr
Alias: STK5320_NAS
Type: Target
.
iSCSI Name: iqn.1986-03.com.sun:01:000e0c9f10da.454F00A2.acct
Alias:
Type: Initiator
```

   The output shows the clients' iSCSI names.

3. **Verify the name of the discovery domain:**

```
# isnsadm list-dd
```

4. **Add the client to the discovery domain:**

```
# isnsadm add-node  -d domain_name  iSCSI_Name
```

   For example, to add the target called “STK5320_NAS” to the Eng-dd discovery domain:

```
# isnsadm add-node -d Eng-dd  iqn.1986-03.com.sun:01:000e0c9f10da.454F00A2.engr
```

5. **List all the clients in the discovery domain to verify the client has been added:**

```
# isnsadm list-dd -v domain_name
```

   For example, to check the Eng-dd discovery domain:

```
# isnsadm list-dd -v Eng-dd
DD name: Eng-dd
DD set: Development-dds
iSCSI Name: iqn.1986-03.com.sun:01:000e0c9f10da.45173FEA.engr
```

```
iSCSI Name: iqn.1986-03.com.sun:01:000e0c9f10da.454F00A2.acct
iSCSI name: iqn.1986-03.com.sun:01:e00000000000.46fd8e2b
```

Managing the iSNS Server and Clients

This section describes how to maintain the iSNS discovery domain sets and their members, the initiators and targets.

As the site grows, continue to add clients, discovery domains, and discovery domain sets as described in the following sections:

- “How to Create the Discovery Domain Sets” on page 171
- “How to Create the Discovery Domains” on page 172
- “How to Add a Discovery Domain to a Discovery Domain Set” on page 173
- “How to Assign Clients to a Discovery Domain” on page 174

This section provides the other procedures for managing the iSNS server, using the command line interface.

The following tasks are described in this section:

- “How to Display the Status of a Discovery Domain Set” on page 175
- “How to Display the Status of a Discovery Domain” on page 176
- “How to Display the Status of Clients” on page 176
- “How to Remove a Client from a Discovery Domain” on page 176
- “How to Remove a Discovery Domain from a Discovery Domain Set” on page 177
- “How to Disable a Discovery Domain Set” on page 177
- “How to Remove a Discovery Domain Set” on page 178

▼ How to Display the Status of a Discovery Domain Set

- **Show the status of the discovery domain set and list the discovery domains that are its members:**

```
# isnsadm list-dd-set -v set_name
```

How to Display the Status of a Discovery Domain

- **Show the status of the discovery domain and lists the clients that are its members:**

  ```
  # isnsadm list-dd -v domain_name
  ```

How to Display the Status of Clients

- **Select one of the following to display client status:**

  - **Show the status of all clients:**

    ```
    # isnsadm list-node -v
    ```

  - **Show the status of only the clients that are targets, that is, storage objects:**

    ```
    # isnsadm list-node -t
    ```

How to Remove a Client from a Discovery Domain

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`
   - `solaris.smf.manage.isns`
   - `solaris.smf.value.isns`

   For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **List the clients to identify the one you want to remove.**

   ```
   # isnsadm list-node -v
   iSCSI Name: iqn.1986-03.com.sun:01:000e0c9f10da.45173FEA.engr
   Alias: STK5320_NAS
   Type: Target
   Network Entity: SE5310
   Portal: 172.20.57.95:3260
   Portal Group: 1
   Portal: 172.20.56.95:3260
   Portal Group: 1
   DD Name: Research,Finance
   ```

The output shows the client's iSCSI name and the name of the discovery domains of which it is a member.

3. **Remove the client from the discovery domain.**

```
# isnsadm remove-node -d domain_name iSCSI_name
```

How to Remove a Discovery Domain from a Discovery Domain Set

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`
   - `solaris.smf.manage.isns`
   - `solaris.smf.value.isns`

   For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **List the discovery domains to identify the one you want to remove.**

```
# isnsadm list-dd -v
```

3. **Remove the discovery domain from the discovery domain set.**

```
# isnsadm remove-dd set_name domain_name
```

How to Disable a Discovery Domain Set

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`
   - `solaris.smf.manage.isns`
   - `solaris.smf.value.isns`

   For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **Deactivate a discovery domain set:**

   ```
   # isnsadm disable-dd-set set_name
   ```

3. **Verify that the state of the discovery domain set has changed to Disabled:**

   ```
   # isnsadm list-dd-set set_name
   ```

How to Remove a Discovery Domain Set

After you remove a discovery domain set, its discovery domains remain. A discovery domain must be a member of at least one discovery domain set.

1. **Use the following profiles to obtain the authorizations needed for managing the iSNS service:**

   - `solaris.isnsmgr.write`
   - `solaris.smf.manage.isns`
   - `solaris.smf.value.isns`

   For more information about roles, see Chapter 3, “Assigning Rights in Oracle Solaris,” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **List the discovery domain sets to identify the one you want to remove.**

   ```
   # isnsadm list-dd-set -v
   ```

3. **Remove the discovery domain set.**

   ```
   # isnsadm remove-dd-set set_name
   ```

♦ ♦ ♦ CHAPTER 10

The Format Utility Reference

This chapter describes the Format utility's menus and commands.

This is a list of the reference information in this chapter.

For a description of the use of the Format utility, see Chapter 6, "Administering the System's Disks".

Recommendations and Requirements for Using the Format Utility

You must assume the `root` role or become an administrator to use the Format utility. See "Using Your Assigned Administrative Rights" in "Securing Users and Processes in Oracle Solaris 11.2 ". Otherwise, the following error message is displayed when you try to use the `format` utility.

```
$ format
Searching for disks...done
No permission (or no disks found)!
```

Format Menu and Command Descriptions

The `format` menu contents are displayed as follows:

```
FORMAT MENU:
disk       - select a disk
type       - select (define) a disk type
partition  - select (define) a partition table
current    - describe the current disk
```

```
format     – format and analyze the disk
fdisk      – run the fdisk program (x86 only)
repair     – repair a defective sector
label      – write label to the disk
analyze    – surface analysis
defect     – defect list management
backup     – search for backup labels
verify     – read and display labels
save       – save new disk/partition definitions
inquiry    – show vendor, product and revision
volname    – set 8&minus;character volume name
!<cmd>     – execute &lt;cmd>, then return
quit
format>
```

The following table describes the main menu items for the `format` utility.

TABLE 10-1 The Main Menu Item Descriptions for the `format` Utility

Menu Item	Command or Menu?	Description
`disk`	Command	Lists all of the system's drives. Also lets you choose the disk you want to use in subsequent operations. This disk is referred to as the current disk.
`type`	Command	Identifies the manufacturer and model of the current disk. Also displays a list of known drive types. Choose the `Auto configure` option for all SCSI-2 disk drives.
`partition`	Menu	Creates and modifies slices. For more information, see "`partition` Menu" on page 181.
`current`	Command	Displays the following information about the current disk: ■ Device name and device type ■ Number of cylinders, alternate cylinders, heads and sectors ■ Physical device name
`format`	Command	Formats the current disk by using one of these sources of information in this order: 1. Information that is found in the `format.dat` file 2. Information from the automatic configuration process 3. Information that you type at the prompt if no `format.dat` entry exists This command does not apply to IDE disks. IDE disks are preformatted by the manufacturer.
`fdisk`	Menu	x86 platform only: Runs the `fdisk` program to create a Solaris `fdisk` partition. The `fdisk` command cannot be used on disks with an EFI label that are greater than 1 terabyte in size.
`repair`	Command	Repairs a specific block on the current disk.

Menu Item	Command or Menu?	Description
`label`	Command	Writes a new label to the current disk.
`analyze`	Menu	Runs read, write, and compare tests. For more information, see "`analyze` Menu" on page 183.
`defect`	Menu	Retrieves and displays defect lists. For more information, see "`defect` Menu" on page 184. This feature does not apply to IDE disks. IDE disks manage defects automatically.
`backup`	Command	**VTOC** – Searches for backup labels. **EFI** – Not supported.
`verify`	Command	Displays the following information about the current disk: ■ Device name and device type ■ Number of cylinders, alternate cylinders, heads and sectors ■ Partition table
`save`	Command	**VTOC** – Saves new disk and partition information. **EFI** – Not applicable.
`inquiry`	Command	**SCSI disks only** – Displays the vendor, product name, and revision level of the current drive.
`volname`	Command	Labels the disk with a new eight-character volume name that you specify.
`quit`	Command	Exits the Format menu.

`partition` Menu

The `partition` menu contents are displayed as follows:

```
format> partition
PARTITION MENU:
0      - change `0' partition
1      - change `1' partition
2      - change `2' partition
3      - change `3' partition
4      - change `4' partition
5      - change `5' partition
6      - change `6' partition
7      - change `7' partition
select - select a predefined table
modify - modify a predefined partition table
name   - name the current table
print  - display the current table
label  - write partition map and label to the disk
quit
```

```
partition>
```

The following table describes the `partition` menu items.

TABLE 10-2 Descriptions for `partition` Menu Items

Subcommand	Description
`change ` *n* ` partition`	Enables you to specify the following information for the new partition: ■ Identification tag ■ Permission flags ■ Starting cylinder ■ Size
`select`	Enables you to choose a predefined partition table.
`modify`	Enables you to change all the slices in the partition table. This command is preferred over the individual `change` `x` `partition` commands.
`name`	Enables you to specify a name for the current partition table.
`print`	Displays the current partition table.
`label`	Writes the partition map and the label to the current disk.
`quit`	Exits the `partition` menu.

x86: `fdisk` Menu

The `fdisk` menu is available on x86 based systems only and appears similar to the following.

```
format> fdisk
Total disk size is 8924 cylinders
Cylinder size is 16065 (512 byte) blocks

Cylinders
Partition   Status    Type          Start   End   Length    %
=========   ======    ============  =====   ===   ======   ===
1                EFI               0  8924    8925    100



SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Edit/View extended partitions
6. Exit (update disk configuration and exit)
7. Cancel (exit without updating disk configuration)
```

```
Enter Selection:
```

The following table describes the `fdisk` menu items.

TABLE 10-3 x86: Descriptions for `fdisk` Menu Items

Menu Item	Description
`Create a partition`	Creates an `fdisk` partition. You must create a separate partition for each OS such as Oracle Solaris or DOS. There is a maximum of four partitions per disk. You are prompted for the size of the `fdisk` partition as a percentage of the disk.
`Specify the active partition`	Enables you to specify the partition to be used for booting. This menu item identifies where the first stage boot program looks for the second stage boot program.
`Delete a partition`	Deletes a previously created partition. This command destroys all the data in the partition.
`Change between Solaris and Solaris2 Partition IDs`	Changes partition IDs from 130 (`0x82`) to 191 (`0xbf`) and back again.
`Edit/View extended partitions`	Manages partition information that is generally used for booting.
`Exit (update disk configuration and exit)`	Writes a new version of the partition table and exits the `fdisk` menu.
`Cancel (exit without updating disk configuration)`	Exits the `fdisk` menu without modifying the partition table.

`analyze` Menu

The `analyze` menu contents are displayed as follows:

```
format> analyze

ANALYZE MENU:
read     - read only test   (doesn't harm SunOS)
refresh  - read then write  (doesn't harm data)
test     - pattern testing  (doesn't harm data)
write    - write then read      (corrupts data)
compare  - write, read, compare (corrupts data)
purge    - write, read, write   (corrupts data)
verify   - write entire disk, then verify (corrupts data)
print    - display data buffer
setup    - set analysis parameters
config   - show analysis parameters
quit
analyze>
```

The following table describes the `analyze` menu items.

TABLE 10-4 Descriptions for `analyze` Menu Items

Subcommand	Description
`read`	Reads each sector on the current disk. Repairs defective blocks as a default.
`refresh`	Reads then writes data on the current disk without harming the data. Repairs defective blocks as a default.
`test`	Writes a set of patterns to the disk without harming the data. Repairs defective blocks as a default.
`write`	Writes a set of patterns to the disk then reads back the data on the disk. Destroys existing data on the disk. Repairs defective blocks as a default.
`compare`	Writes a set of patterns to the disk, reads back the data, and then compares it to the data in the write buffer. Destroys existing data on the disk. Repairs defective blocks as a default.
`purge`	Removes all data from the disk so that the data cannot be retrieved by any means. Data is removed by writing three distinct patterns over the entire disk (or a section of the disk). If the verification passes, a hex-bit pattern is written over the entire disk (or a section of the disk). Repairs defective blocks as a default.
`verify`	In the first pass, writes unique data to each block on the entire disk. In the next pass, reads and verifies the data. Destroys existing data on the disk. Repairs defective blocks as a default.
`print`	Displays the data in the read/write buffer.

`setup`	Enables you to specify the following analysis parameters:
	
	`Analyze entire disk? yes`
	`Starting block number:` *depends on drive*
	`Ending block number:` *depends on drive*
	`Loop continuously? no`
	`Number of passes: 2`
	`Repair defective blocks? yes`
	`Stop after first error? no`
	`Use random bit patterns? no`
	`Number of blocks per transfer: 126 (0/`*n*`/`*nn*`)`
	`Verify media after formatting? yes`
	`Enable extended messages? no`
	`Restore defect list? yes`
	`Restore disk label? yes`

`config`	Displays the current analysis parameters.
`quit`	Exits the `analyze` menu.

`defect` Menu

The `defect` menu contents are displayed as follows:

```
format> defect

DEFECT MENU:
primary  - extract manufacturer's defect list
grown    - extract manufacturer's and repaired defects lists
both     - extract both primary and grown defects lists
print    - display working list
dump     - dump working list to file
quit
defect>
```

The following table describes the `defect` menu items.

TABLE 10-5 The `defect` Menu Item Descriptions

Subcommand	Description
`primary`	Reads the manufacturer's defect list from the disk drive and updates the in-memory defect list.
`grown`	Reads the grown defect list and then updates the in-memory defect list. Grown defects are defects that have been detected during analysis.
`both`	Reads both the manufacturer's defect list and the grown defect list. Then, updates the in-memory defect list.
`print`	Displays the in-memory defect list.
`dump`	Saves the in-memory defect list to a file.
`quit`	Exits the `defect` menu.

Rules for Input to `format` Commands

When you use the Format utility, you need to provide various kinds of information. This section describes the rules for this information. For information on using `format`'s help facility when you specify data, see "Getting Help on the Format Utility" on page 187.

Specifying Numbers to `format` Commands

Several places in the Format utility require number as input. You must either specify the appropriate data or select a number from a list of choices. In either case, the help facility causes `format` to display the upper and lower limits of the number expected. Simply enter the appropriate number. The number is assumed to be in decimal format unless a base is explicitly specified as part of the number (for example, 0x for hexadecimal).

The following are examples of integer input:

```
Enter number of passes [2]: 34
Enter number of passes [34] 0xf
```

Specifying format Command Names

Command names are needed as input whenever the Format utility displays a menu prompt. You can abbreviate the command names, as long as what you type is sufficient to uniquely identify the command desired.

For example, use p to access the partition menu from the Format menu. Then, type p to display the current slice table.

```
format> p
PARTITION MENU:
0      - change `0' partition
1      - change `1' partition
2      - change `2' partition
3      - change `3' partition
4      - change `4' partition
5      - change `5' partition
6      - change `6' partition
7      - change `7' partition
select - select a predefined table
modify - modify a predefined partition table
name   - name the current table
print  - display the current table
label  - write partition map and label to the disk
quit
partition> p
```

Specifying Disk Names to format Commands

At certain points in the Format utility, you must name something. In these cases, you are free to specify any string you want for the name. If the name has white space in it, the entire name must be enclosed in double quotation marks ("). Otherwise, only the first word of the name is used.

For example, if you want to identify a specific partition table for a disk, you can use the name subcommand that is available from the partition menu:

```
partition> name
Enter table name (remember quotes): "new disk3"
```

Getting Help on the Format Utility

The Format utility provides a help facility that you can use whenever the Format utility is expecting input. You can request help about what input is expected by typing a question mark (`?`). The Format utility displays a brief description of what type of input is needed.

If you type a `?` at a menu prompt, a list of available commands is displayed.

The man pages associated with the Format utility include the following:

- `format`(1M) – Describes the basic Format utility capabilities and provides descriptions of all command-line variables.
- `format.dat`(4) – Describes disk drive configuration information for the Format utility.

♦ ♦ ♦ CHAPTER 11

Managing Tape Drives Tasks

This chapter describes how to manage tape drives in the Oracle Solaris OS.

This is a list of the information in this chapter:

This is a list of the step-by-step instructions in this chapter.

Choosing Which Media to Use

Oracle Solaris systems are generally backed up by using 1/2-inch tape cartridge devices or high capacity devices. You can review Oracle's current tape product line in http://www.oracle.com/us/products/storage/overview/index.html. Select the Tape Storage tab.

See Oracle's current tape storage documentation in http://www.oracle.com/technetwork/documentation/tape-storage-curr-187744.html.

The media that you choose depends on the availability of the equipment that supports it and of the media (usually tape) that you use to store the files. Although you must do the backup from a local system, you can write the files to a remote device.

Backup Device Names

You specify a tape or device to use for backup by supplying a logical device name. This name points to the subdirectory that contains the “raw” device file and includes the logical unit number of the drive. Tape drive naming conventions use a logical, not a physical, device name, as follows: `/dev/rmt/`*N*`[`*d*`][b][n]`.

`/dev/rmt`	Raw magnetic tape subdirectory of the devices directory.
n	Drive number, with 0 as the first drive and *n* as the last drive.
d	Density which you can optionally specify by selecting one from the following choices: ■ `l` – low ■ `m` – medium ■ `h` – high ■ `u` – ultra ■ `c` – compressed
b	Optional character if you want to indicate the tape's BSD compatible behavior.
n	Optional character if you want to specify *no-rewind*. Omit this character to specify the default behavior, which is *rewind*.

If you don't specify the density, a tape drive typically writes at its *preferred* density, which is typically the highest density the tape drive supports. Most SCSI drives can automatically detect the density or format on the tape and read it accordingly. To determine the different densities that are supported for a drive, look at the `/dev/rmt` subdirectory. This subdirectory includes the set of tape device files that support different output densities for each tape.

Also, a SCSI controller can have a maximum of seven SCSI tape drives.

The following are examples of tape device names with corresponding rewind and density values.

- `/dev/rmt/0` – first drive, rewind. Preferred density is used.
- `/dev/rmt/0n` – first drive, no rewind. Preferred density is used.
- `/dev/rmt/1m` – second drive, medium density, rewind.
- `/dev/rmt/1hn` – second drive, high density, no rewind.

Displaying Tape Drive Status

You can use the `status` option with the `mt` command to get status information about tape drives. The `mt` command reports information about any tape drives that are described in the `/kernel/drv/st.conf` file.

▼ How to Display Tape Drive Status

1. **Ensure that tapes are loaded into the drives.**
2. **For every tape whose status you want to check, type the following command:**

```
# mt -f /dev/rmt/drive-number status
```

Example 11-1 Displaying Tape Drive Status

The following example shows the status for a QIC-150 tape drive (`/dev/rmt/0`):

```
$ mt -f /dev/rmt/0 status
Archive QIC-150 tape drive:
sense key(0x0)= No Additional Sense   residual= 0   retries= 0
file no= 0   block no= 0
```

The following example shows the status for an Exabyte tape drive (`/dev/rmt/1`):

```
$ mt -f /dev/rmt/1 status
Exabyte EXB-8200 8mm tape drive:
sense key(0x0)= NO Additional Sense residual= 0  retries= 0
file no= 0   block no= 0
```

The following example shows a quick way to poll a system and locate all of its tape drives:

```
$ for drive in 0 1 2 3 4 5 6 7
> do
> mt -f /dev/rmt/$drive status
> done
Archive QIC-150 tape drive:
sense key(0x0)= No Additional Sense   residual= 0   retries= 0
file no= 0   block no= 0
/dev/rmt/1: No such file or directory
/dev/rmt/2: No such file or directory
/dev/rmt/3: No such file or directory
/dev/rmt/4: No such file or directory
/dev/rmt/5: No such file or directory
/dev/rmt/6: No such file or directory
/dev/rmt/7: No such file or directory
$
```

Handling Magnetic Tape Cartridges

If errors occur when a tape is being read, you can retension the tape, clean the tape drive, and then try again.

Retensioning a Magnetic Tape Cartridge

Retension a magnetic tape cartridge with the `mt` command.

For example:

```
$ mt -f /dev/rmt/1 retension
$
```

Note - Do not retension non-QIC tape drives.

Rewinding a Magnetic Tape Cartridge

To rewind a magnetic tape cartridge, use the `mt` command.

For example:

```
$ mt -f /dev/rmt/1 rewind
$
```

Guidelines for Drive Maintenance and Media Handling

A backup tape that cannot be read is useless. So, periodically clean and check your tape drives to ensure correct operation. See your hardware manuals for instructions on procedures for cleaning a tape drive. You can check your tape hardware by copying some files to the tape, reading the files back, and then comparing the original files with the copied files.

Be aware that hardware can fail in ways that the system does not report.

Always label your tapes after a backup. This label should never change. Every time you do a backup, make another tape label that contains the following information:

- The backup date
- The name of the machine and file system that is backed up

- The backup level
- The tape number (1 of *n*, if the backup spans multiple volumes)
- Any information specific to your site

Store your tapes in a dust-free safe location, away from magnetic equipment. Some sites store archived tapes in fireproof cabinets at remote locations.

You should create and maintain a log that tracks which media (tape volume) stores each job (backup) and the location of each backed-up file.

♦ ♦ ♦ CHAPTER 12

Writing CDs and DVDs

This chapter provides step-by-step instructions for writing and copying data CDs and DVDs and audio CDs with the `cdrw` command.

This is a list of the information in this chapter:

Working With Audio CDs and Data CDs and DVDs

You can use the `cdrw` command in the `media/cdrw` package to write file systems for CDs and DVDs in ISO 9660 format with Rock Ridge or Joliet extensions on CD-R,CD-RW, DVD-RW, or DVD+RW media devices.

You can use the `cdrw` command to perform the following tasks:

- Create data CDs and DVDs.
- Create audio CDs.
- Extract audio data from an audio CD.
- Copy CDs and DVDs.
- Erase CD-RW media.

For information on recommended CD-R or CD-RW devices, go to http://www.oracle.com/us/sun/index.html.

CD/DVD Media Commonly Used Terms

This section defines commonly used terms related to CD/DVD media.

Term	Description
CD-R	CD read media that can be written once and after that, can only be read from.
CD-RW	CD rewritable media that can be written to and erased. CD-RW media can only be read by CD-RW devices.
DVD-R	Digital video disk (recordable) that can be written once and after that, can only be read from. These devices have much larger capacity than CD-R media.
DVD+R	Digital video disk (recordable) that can be written once and after that, can only be read from. DVD+R devices have more complete error management system than DVD-R, which allows for more accurate burning to media, independent of the quality of the media.
DVD-RW	Digital video disk (rewritable) with storage capacity equal to a DVD-R. This media can be re-recorded by first erasing the entire disk.
DVD+RW	Digital video disk (random-access rewritable) with storage capacity equal to a DVD+R. This medium allows overwriting of individual blocks without erasing the entire disk.
DVD-RAM	Digital video disk (random access memory, rewritable) with circular rather than spiral tracks and hard sectoring.
ISO 9660	ISO, an acronym for Industry Standards Organization, is an organization that sets standards for computer storage formats. An ISO 9660 file system is a standard CD or DVD file system that enables you to read the same CD or DVD on any major computer platform. The standard, issued in 1988, was written by an industry group named High Sierra, named after the High Sierra Hotel in Nevada. Almost all computers with CD or DVD drives can read files from an ISO 9660 file system.
Joliet extensions	Adds Windows file system information.
Rock Ridge extensions	Adds UNIX file system information. (Rock Ridge is named after the town in the movie Blazing Saddles.) **Note -** These extensions are not mutually exclusive. You can specify both `mkisofs -R` and `-j` options for compatibility with both systems. (See *mkisofs* for details.)
MMC-compliant recorder	Acronym for Multi Media Command, which means these recorders comply with a common command set. Programs that can write to one MMC-compliant recorder should be able to write to all other recorders.
Red Book CDDA	Acronym for Compact Disc Digital Audio, which is an industry standard method for storing digital audio on compact discs. Also known by the term “Red Book” format. The official industry specification calls for one or more audio files sampled in 16-bit stereo sound at a sampling rate of 44.1 kilohertz (kHz).

Commonly used terms when writing to CD media are listed in the following table.

Term	Description
blanking	The process of erasing data from the CD-RW media.
session	A complete track with lead-in and lead-out information.
track	A complete data or audio unit.

Writing CD and DVD Data and Audio CDs

The following procedures are described in this section:

The process of writing to a CD or DVD cannot be interrupted and needs a constant stream of data. Consider using the `cdrw -S` option to simulate writing to the media to verify that the system can provide data at a sufficient rate for writing to the CD or DVD.

Write errors can be caused by one of the following problems:

- The media cannot handle the drive speed. For example, some media are only certified for 2x or 4x speeds.
- The system is running too many heavy processes that are starving the writing process.
- The image is on a remote system, and network congestion is causing delays in reading the image.
- The source drive is slower than the destination drive.

If any of these problems occur, you can lower the writing speed of the device by using the `cdrw -p` option.

For example, the following command shows how to simulate writing at 4x speed:

```
$ cdrw -iS -p 4 image.iso
```

Note - CD-R, CD-RW (not MRW formatted), DVD-R, and DVD-RW media support simulation mode (`-S`), but DVD-RAM, DVD+R, DVD+RW, and any MRW-formatted media and some others do not support simulation mode. The following message is displayed if simulation mode is not supported:

```
Media does not support simulated writing
```

For more information about media types, see "CD/DVD Media Commonly Used Terms" on page 195.

For more information, see `cdrw`(1).

Restricting User Access to Removable Media

By default, all users can access removable media. However, you can restrict user access to removable media by setting up a role through administrative rights. Access to removable media is restricted by assigning the role to a limited set of users.

For a discussion of using roles, see Chapter 3, "Assigning Rights in Oracle Solaris," in "Securing Users and Processes in Oracle Solaris 11.2 ".

▼ How to Restrict User Access to Removable Media With Administrative Rights

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in "Securing Users and Processes in Oracle Solaris 11.2 ".

2. **Set up a role that includes the Device Management rights.**

   For more information, see Chapter 3, "Assigning Rights in Oracle Solaris," in "Securing Users and Processes in Oracle Solaris 11.2 ".

   ```
   # roleadd -m -d /export/home/muser -c "mediauser role" \
                  -A solaris.device.cdrw -P All muser
   ```

3. **Add users who need to use the `cdrw` command to the newly created role.**

   ```
   # usermod -R muser username
   ```

How to Identify a CD or DVD Writer

1. **Identify the CD or DVD writers on the system.**

   For example:

```
$ cdrw -l
Looking for CD devices...
Node              |    Connected Device            |  Device type
----------------------+--------------------------------+-----------------
cdrom0               | YAMAHA   CRW8824S         1.0d | CD Reader/Writer
```

2. **Identify a specific CD or DVD writer.**

   For example:

```
$ cdrw -a filename.wav -d cdrom2
```

3. **Identify whether the media is blank or whether a table of contents exists on the media.**

   For example:

```
$ cdrw -M

Device : YAMAHA   CRW8824S
Firmware : Rev. 1.00 (26/04/00)
Media is blank
%
```

How to Check the CD or DVD Media

The `cdrw` command works with or without removable media services running. For more information about disabling or enabling removable media services, see “How to Disable or Enable Removable Media Services” on page 243.

1. **Insert a CD or DVD into the drive.**

   The CD or DVD can be any CD or DVD that the drive can read.

2. **Check that the drive is connected properly by listing the drive.**

```
$ cdrw -l
Looking for CD devices...
Node                  Connected Device               Device type
----------------------+--------------------------------+-----------------
cdrom1               | YAMAHA   CRW8824S         1.0d | CD Reader/Writer
```

3. **(Optional) If you do not see the drive in the list, select one of the following so that the system recognizes the drive.**

- Add the drive without rebooting the system.

  ```
  # devfsadm
  ```

- Restart removable media services

Creating a Data CD or DVD

Prepare the data first by using the `mkisofs` command to convert the file and file information into the High Sierra format used on CDs or DVDs.

▼ How to Create an ISO 9660 File System for a Data CD or DVD

1. **Insert a blank CD or DVD into the drive.**

2. **Create the ISO 9660 file system on the new CD or DVD.**

   ```
   $ mkisofs -r /pathname > cd-file-system
   ```

`-r`	Creates Rock Ridge information and resets file ownerships to zero.
/pathname	Identifies the path name used to create the ISO 9660 file system.
> cd-file-system	Identifies the name of the file system to be put on the CD or DVD.

3. **Copy the file system onto the CD or DVD.**

   ```
   $ cdrw -i cd-file-system
   ```

   The `-i` *cd-file-system* specifies the image file for creating a data CD or DVD.

Example 12-1 Creating an ISO 9660 File System for a Data CD or DVD

The following example shows how to create an ISO 9660 file system for a data CD or DVD.

```
$ mkisofs -r /home/dubs/dir > dubs_cd
Total extents actually written = 56
Total translation table size: 0
Total rockridge attributes bytes: 329
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 8000
56 extents written (0 Mb)
```

Then, copy the file system onto the CD.

```
$ cdrw -i dubs_cd
Initializing device...done.
Writing track 1...done.
Finalizing (Can take several minutes)...done.
```

How to Create a Multi-Session Data CD

This procedure describes how to put more than one session on a CD. This procedure includes an example of copying the `infoA` and `infoB` directories onto the CD.

1. Create the file system for the first CD session.

```
$ mkisofs -o infoA -r -V my_infoA /data/infoA
Total translation table size: 0
Total rockridge attributes bytes: 24507
Total directory bytes: 34816
Path table size(bytes): 98
Max brk space used 2e000
8929 extents written (17 Mb)
```

`-o infoA`	Identifies the name of the ISO file system.
`-r`	Creates Rock Ridge information and resets file ownerships to zero.
`-V my_infoA`	Identifies a volume label to be used as the mount point by removable media services.
`/data/infoA`	Identifies the ISO image directory to create.

2. Copy the ISO file system for the first session onto the CD.

```
$ cdrw -iO infoA
Initializing device...done.
Writing track 1...done.
done.
Finalizing (Can take several minutes)...done.
```

`-i` *infoA*	Identifies the name of the image file to write to the CD.
`-O`	Keeps the CD open for writing.

3. Reinsert the CD after it is ejected.

4. Identify the path name of the CD media to include in the next write session.

```
$ eject -n
.
```

```
.
.
cdrom0 -> /vol/dev/rdsk/c2t4d0/my_infoA
```

Note the `/vol/dev/...` path name.

5. Identify the next writeable address on the CD to write the next session.

```
% cdrw -M /cdrom
Device : YAMAHA   CRW8424S
Firmware : Rev. 1.0d (06/10/99)

Track No. |Type    |Start address
----------+--------+-------------
1         |Audio   |0
2         |Audio   |33057
3         |Data    |60887
4         |Data    |68087
5         |Data    |75287
Leadout   |Data    |84218

Last session start address: 75287
Next writable address: 91118
```

Note the address in the `Next writable address` output so that you can provide this address when you write the next session.

6. Create the next ISO file system for the next CD session, and write it onto the CD.

```
$ mkisofs -o infoB -r -C 0,91118 -M /vol/dev/rdsk/c2t4d0/my_infoA
/data/infoB
Total translation table size: 0
Total rockridge attributes bytes: 16602
Total directory bytes: 22528
Path table size(bytes): 86
Max brk space used 20000
97196 extents written (189 Mb)
```

`-o` *infoB*	Identifies the name of the ISO file system.
`-r`	Creates Rock Ridge information and resets file ownerships to zero.
`-C` *0,91118*	Identifies the starting address of the first session and the next writable address.
`-M /vol/dev/rdsk/c2t4d0/my_infoA`	Specifies the path of the existing ISO image to be merged.
`/data/infoB`	Identifies the ISO image directory to create.

Creating an Audio CD

You can use the `cdrw` command to create audio CDs from individual audio tracks or from `.au` and `.wav` files.

The supported audio formats are described in the following table.

Format	Description
`sun`	Oracle `.au` file with data in Red Book CDDA format
`wav`	RIFF (`.wav`) file with data in Red Book CDDA format
`cda`	`.cda` file with raw CD audio data, which is 16-bit PCM stereo at 44.1 k Hz sample rate in little-endian byte order
`aur`	`.aur` files with raw CD data in big-endian byte order

If no audio format is specified, the `cdrw` command tries to determine the audio file format based on the file extension. The case of the characters in the extension is ignored.

▼ How to Create an Audio CD

This procedure describes how to copy audio files onto a CD.

1. **Insert a blank CD into the CD-RW drive.**

2. **Change to the directory that contains the audio files.**

   `$` **`cd`** *`/myaudiodir`*

3. **Copy the audio files onto the CD.**

   `$` **`cdrw -a`** *`track1.wav track2.wav track3.wav`*

   The `-a` option creates an audio CD.

Example 12-2 Creating an Audio CD

The following example shows how to create an audio CD.

```
$ cdrw -a bark.wav chirp.au meow.wav
Initializing device...done.
Writing track 1...done.
done.
Writing track 2...done.
```

```
Writing track 3...done.
done.
Finalizing (Can take several minutes)...done.
```

The following example shows how to create a multi-session audio CD. The CD is ejected after the first session is written. You would need to reinsert the CD before the next writing session.

```
$ cdrw -aO groucho.wav chico.au harpo.wav
Initializing device...done.
Writing track 1...done.
done.
Writing track 2...done.
Writing track 3...done.
done.
Finalizing (Can take several minutes)...done.
<Re-insert CD>
$ cdrw -a zeppo.au
Initializing device...done.
Writing track 1...done.
done.
Finalizing (Can take several minutes)...done.
```

How to Extract an Audio Track on a CD

Use the following procedure to extract an audio track from a CD and copy the audio track to a new CD.

If you don't use the `cdrw -T` option to specify the audio file type, the `cdrw` command uses the filename extension to determine the audio file type. For example, the `cdrw` command detects that this file is a `.wav` file.

```
$ cdrw -x 1 testme.wav
```

1. **Insert an audio CD into the CD-RW drive.**

2. **Extract an audio track.**

   ```
   $ cdrw -x -T audio-type 1 audio-file
   ```

`-x`	Extracts audio data from an audio CD.
`T` *audio-type*	Identifies the type of audio file to be extracted. Supported audio types are `sun`, `wav`, `cda`, or `aur`.
audio-file	Identifies the audio track to be extracted.

3. **Copy the track to a new CD.**

   ```
   $ cdrw -a audio-file
   ```

Example 12-3 Extracting and Creating Audio CDs

The following example shows how to extract the first track from an audio CD and name the file `song1.wav`.

```
$ cdrw -x -T wav 1 song1.wav
Extracting audio from track 1...done.
```

This example shows how to copy a track to an audio CD.

```
$ cdrw -a song1.wav
Initializing device...done.
Writing track 1...done.
Finalizing (Can take several minutes)...done.
```

How to Copy a CD

This procedure describes how to extract all the tracks from an audio CD into a directory and then copy all of them onto a blank CD.

Note - By default, the `cdrw` command copies the CD into the `/tmp` directory. The copying might require up to 700 MB of free space. If there is insufficient space in the `/tmp` directory for copying the CD, use the `-m` option to specify an alternate directory.

---

1. **Insert an audio CD into a CD-RW drive.**

2. **Create a directory for the audio files.**

   ```
   $ mkdir /music_dir
   ```

3. **Extract the tracks from the audio CD.**

   ```
   $ cdrw -c -m music_dir
   ```

   An `Extracting audio ...` message is display for each track.

   The CD is ejected when all the tracks are extracted.

4. **Insert a blank CD and press Return.**

   After the tracks are extracted, the audio CD is ejected. You are prompted to insert a blank CD.

Example 12-4 Copying a CD

This example shows how to copy one CD to another CD. You must have two CD-RW devices to do this task.

```
$ cdrw -c -s cdrom0 -d cdrom1
```

How to Erase CD-RW Media

You have to erase existing CD-RW data before the CD can be rewritten.

- **Erase the entire media or just the last session on the CD by selecting one of the following:**

  - Erase the last session only.

    ```
    $ cdrw -d cdrom0 -b session
    ```

    Erasing just the last session with the `-b session` option is faster than erasing the entire media with the `-b all` option. You can use the `-b session` option even if you used the `cdrw` command to create a data or audio CD in just one session.
  - Erase the entire media.

    ```
    $ cdrw -d cdrom0 -b all
    ```

♦ ♦ ♦ CHAPTER 13

Managing USB Devices

This chapter provides overview information and step-by-step instructions for using Universal Serial Bus (USB) devices in Oracle Solaris. The following topics are covered:

For general questions about USB, go to http://www.usb.org/about/faq.

For general information about dynamic reconfiguration and hot-plugging, see Chapter 2, "Dynamically Configuring Devices".

For information on configuring USB printers, see Chapter 1, "Setting Up and Administering Printers by Using CUPS (Overview)," in "Configuring and Managing Printing in Oracle Solaris 11.2 ".

About USB Support in Oracle Solaris

Oracle Solaris supports all USB 3.0 devices that use the USB 3.0 `xhci` host controller driver, with the exception of audio devices. Backward compatibility with previous USB versions enables you to use the same hardware and software components for USB 2.0 , 1.1, and 1.0 devices on USB 3.0 ports. However, for audio devices, only USB 2.0, 1.1, and 1.0 are supported.

A single XHCI host controller supports all speeds of the USB devices. If you are using USB 2.0 ports for devices, different host controller interface drivers are dynamically assigned depending on whether those devices support USB 2.0.

Note - For devices that are not supported by a USB driver, refer to the USB library documentation in `/usr/share/doc/libusb/libusb.txt`.

For better performance, always connect USB 3.0 and USB 2.0 devices to corresponding USB 3.0 and USB 2.0 ports. A USB 3.0 or USB 2.0 port can be on any of the following components:

- A USB PCI card
- A USB hub that is connected to a USB port
- A SPARC or x86 computer motherboard

Note - A USB 2.0 PCI card might be needed for older SPARC platforms.

References for USB Information in Oracle Solaris

The following man pages provide information about USB versions:

USB 3.0	`xhci`(7D)
USB 2.0	`ehci`(7D); `usba`(7D)
USB 1.1	`ohci`(7D)

The following man pages provide information about specific USB devices:

Generic USB drivers	`ugen`(7D)
Keyboard and mouse devices	`hid`(7D)
Hubs	`hubd`(7D)
Serial devices	`usbser_edge`(7D); `usbsprl`(7D); `usbsksp`(7D)
Storage devices	`scsa2usb`(7D)
Device management	`libusb`(3LIB)

For information about specifications of different USB versions, go to http://www.usb.org/home.

USB Device Features and Compatibility Issues

To identify the speed of your USB device, check the `/var/adm/messages` file for messages similar to the following:

```
Dec 13 17:05:57 mysystem usba: [ID 912658 kern.info] USB 2.0 device
(usb50d,249) operating at hi speed (USB 2.x) on USB 2.0 external hub:
storage@4, scsa2usb0 at bus address 4
```

Except where indicated, Oracle Solaris supports USB devices on both SPARC and x86 based systems. Additional storage devices might work by modifying the `scsa2usb.conf` file. For more information, see the `scsa2usb`(7D) man page.

The following sections provide additional information about specific USB devices.

Bus-Powered Devices

USB hubs are not self-powered. They supply power to connected devices by deriving it from the USB bus to which they are connected. Consequently, and also because of power management, power to these downstream devices is limited. Therefore, avoid overloading these hubs. Note specifically the following limitations:

- You cannot cascade two bus-powered hubs.
- Each bus-powered hub is limited to a maximum of 100 mA only for each port.
- You can connect only self-powered or low bus-powered devices, not high bus-powered devices, to a bus-powered hub.
- Some hubs or devices can report a false power source. With such hubs, the connection might be unpredictable.

USB Keyboards and Mouse Devices

On SPARC based systems, do not remove the keyboard and mouse during a system reboot or at the `ok` prompt. During the boot process, the OpenBoot PROM (OBP) limits keyboard and mouse devices to the motherboard root hub ports only. You can move the keyboard and mouse to another hub at any time after a system reboot. These devices become fully functional after you plug them to their ports.

On SPARC based systems, note the following issues with regards to these devices:

- The power key behaves differently between a USB keyboard and a type 5 keyboard. On a USB keyboard, the SUSPEND/SHUTDOWN key suspends or shuts down the system. However, that key cannot power up the system.
- On legacy SPARC based systems, USB keyboard and mouse devices do not work simultaneously with Type 3, 4, or 5 keyboards.

For information about multiple keyboard and mouse device support, see the `virtualkm`(7D) man page.

USB Hubs and Host Controller

The USB host controller has an embedded hub called the *root hub* whose ports are visible on the system's back panel.

When using USB hubs, avoid the following:

- Cascading hubs beyond four levels on either SPARC based systems or x86 based systems. On SPARC systems, the OpenBoot PROM cannot reliably probe beyond four levels of devices.
- Cascading bus-powered hubs. A bus-powered hub does not have its own power supply.
- Connecting to a bus-powered hub a device that requires a large amount of power, such as a USB diskette drive, which might drain the hub of power for other devices. Bus-powered hubs might deny connection to these devices.

Note - A USB 3.0 host controller supports control, bulk, and interrupt transfer types. However, unlike previous USB versions, it does not support isochronous transfer type.

SPARC: USB Power Management

Suspending and resuming USB device services are fully supported on SPARC systems. However, do not suspend a device that is busy. Likewise, never remove a device when the system is powered off under a suspend shutdown.

When power management is enabled on the system, the USB framework manages power on all devices. For example, the hub driver suspends the port to which the device is connected.

Devices that support *remote wake up* can notify the system to restore power to the device's path so that the device can be used. The host system can also restore power to the device if an application sends an I/O to the device.

Power management is implemented on all devices that support remote wake-up capability. On USB printers, power management functions only between two print jobs. On devices that use the generic USB driver (UGEN), power management runs only when the devices are closed.

Managing USB Mass Storage Devices

In Oracle Solaris, USB mass storage devices are hot-pluggable and share the same features as most removable media devices.

A hot-pluggable device is automatically mounted and immediately available in the `/media` directory. Users can readily access the device. If no automatic mount occurs, you can mount devices manually with the `mount` command. The following example mounts a disk with the FAT file system:

```
mount -F pcfs /dev/dsk/c2t0d0s0:c /mnt
```

USB storage devices are power managed, except those that support `LOG SENSE` pages. Devices with `LOG SENSE` pages are typically SCSI drives connected through a USB-to-SCSI bridge device.

To configure or manage USB storage devices, use the following commands:

- The `rmformat` command creates slices or displays all USB devices with their media inserted.
- The `fdisk` command partitions a USB device.

Caution - Do not use the `format` command or the `rmformat -F` command to physically format a USB drive.

Applications might work differently with USB mass storage devices. For example, some earlier applications might make incorrect assumptions about the size of the media because previously, only smaller devices like diskettes were removable.

To revert to a previous Oracle Solaris behavior that treated USB mass storage devices as removable media devices, update the `/etc/driver/drv/scsa2usb.conf` file. For more information, see `scsa2usb`(7D).

Hot-Plugging USB Mass Storage Devices

Hot-plugging a device means adding or removing the device without shutting down the operating system or rebooting the system. These devices are also automatically configured or unconfigured without user intervention.

All USB devices are hot-pluggable. However, non-removable USB storage devices are identified as hot-pluggable devices only at the driver level. When these devices are connected or disconnected, the changes caused by automatic configuration or unconfiguration occur at the kernel level and do not affect the use of the devices.

The removable media services manage the mounting of removable media, including hot-pluggable devices. To determine whether the service is running, type the following command:

```
# svcs hal dbus rmvolmgr
STATE          STIME    FMRI
online         May_03   svc:/system/dbus:default
```

```
online         May_03   svc:/system/hal:default
online         May_03   svc:/system/filesystem/rmvolmgr:default
```

With hot-plugged devices, the file system can be mounted from the device if the device is valid and recognized. While file system mounts occur automatically, you can mount the file system manually if necessary.

Before disconnecting a hot-pluggable device, issue an `eject` command first. Use the `eject -l` command to determine the name of the device. Failure to eject before disconnecting the device might damage the device's file system, even though the device itself is released and the port becomes available for use.

When you connect a USB device, the device is immediately added to the system's device hierarchy and is included in the output of the `prtconf` command.

If you disconnect a device, the device is removed from the system's device hierarchy only if no application is using the device. If an application is using a device that becomes disconnected, the device node remains. However, the driver controlling the device stops all activity on the device. Any new I/O activity issued to this device returns an error. The system then prompts you to connect the original device. If the device is no longer available, stop the application. After a few seconds, the port is released.

Note - Removing an active or open device might damage data integrity. Always close the device before removing it. However, you can move the keyboard and mouse even while these devices are active.

Displaying USB Information

You can access information about removable media with or without using removable media services. For information on accessing information on removable media with GNOME's File Manager, see the GNOME desktop documentation.

After being formatted, the USB device is usually mounted under the `/media/`*label* directory.

The device nodes are created under the `/dev/rdsk` directory for character devices and under the `/dev/dsk` directory for block devices. Device links are created when the devices are hot-plugged. For more information, see the `scsa2usb`(7D) man page.

For more information about mounting and unmounting USB devices, see "How to Mount or Unmount a USB Mass Storage Device" on page 223.

The following examples show how to display USB information.

- Using the `prtconf` command

  In this example, the output is truncated to display only USB device information.

```
$ prtconf
usb, instance #0
hub, instance #2
device, instance #8
interface (driver not attached)
printer (driver not attached)
mouse, instance #14
device, instance #9
keyboard, instance #15
mouse, instance #16
storage, instance #7
disk (driver not attached)
communications, instance #10
modem (driver not attached)
data (driver not attached)
storage, instance #0
disk (driver not attached)
storage, instance #1
disk (driver not attached)
```

- Using the `rmformat` command

```
$ rmformat
Looking for devices...
1. Logical Node: /dev/rdsk/c3t0d0p0
Physical Node: /pci@0,0/pci108e,534a@2,1/storage@3/disk@0,0
Connected Device: SanDisk  Cruzer Micro     0.3
Device Type: Removable
Bus: USB
Size: 245.0 MB
Label: <None>
Access permissions: Medium is not write protected.
```

Creating File Systems on USB Storage Devices

The file systems are automatically mounted on hot-pluggable devices. On certain removable devices, you might need to create the file system after the devices are connected. This section provides some examples of how to create the file systems.

Before you create the file system, ensure that the device is unmounted. To create a file system, use the following command syntax:

```
# mkfs -F FS-type -o FS-type-specific-options raw-device-file
```

FS-type Type of file system you want to create.

FS-type-specific-options	Options that are specific to the type of file system you are creating such as size for FAT file systems.
raw-device-file	Disk partition on which to write the file system.

Note - You must format USB diskettes first before creating file systems on the diskettes. All other mass storage devices require only a file system before you can use them.

The following example shows how to create a PCFS file system on a 1.4-MB diskette on a SPARC system:

```
# mkfs -F pcfs /dev/rdsk/c2t0d0p0
Construct a new FAT file system on /dev/rdsk/c2t0d0p0: (y/n)? y
```

The following example shows how to create a PCFS file system on a 1.4-MB diskette on an x86 system:

```
# mkfs -F pcfs /dev/rdsk/c2t0d0s2
Construct a new FAT file system on /dev/rdsk/c2t0d0s2: (y/n)? y
```

The following example shows how to create a PCFS file system on a 100-MB USB memory stick on a SPARC system:

```
# mkfs -F pcfs /dev/rdsk/c5t0d0s2:c
```

The following example shows how to create a PCFS file system on a 100-MB USB memory stick on an x86 system:

```
# mkfs -F pcfs /dev/rdsk/c5t0d0p0:c
```

The following example shows how to create a legacy UFS file system.

```
# newfs /dev/rdsk/c0t0d0s6
```

Consider using the `newfs -f 4096` option or `newfs -T` option for large USB hard disks.

Note - UFS file system overhead consumes a significant portion of space on a diskette, due to a diskette's limited storage capacity.

▼ How to Add a USB Mass Storage Device

1. **Connect the USB mass storage device.**
2. **Verify that the USB device has been added.**

```
$ rmformat
```

The following example shows the output of the command after you connect a storage device.

```
$ rmformat
Looking for devices...
1. Logical Node: /dev/rdsk/c3t0d0p0
Physical Node: /pci@0,0/pci108e,534a@2,1/storage@3/disk@0,0
Connected Device: SanDisk  Cruzer Micro     0.3
Device Type: Removable
Bus: USB
Size: 245.0 MB
Label: <None>
Access permissions: Medium is not write protected.
```

3. **Verify that the device is automatically mounted under the `/media` directory.**

   Use one of the following commands:

   - `ls /media/NONAME`

     For example:

     ```
     $ ls /media/NONAME
     aa  bb
     ```

   - `rmmount -l`

     For example:

     ```
     $ rmmount -l
     /dev/dsk/c3t0d0p0:1  rmdisk0,NONAME,/media/NONAME
     ```

How to Remove a USB Mass Storage Device

1. **Stop any active applications that are using the device.**

2. **Unmount the device.**

   Use one of the following commands:

   - As a regular user: `rmumount` *name*

     For example:

     ```
     $ rmumount NONAME
     ```

   - As an administrator: `umount` *name*

     For example:

     ```
     # umount /media/NONAME
     ```

3. **Remove the device.**

How to Create a File System on a USB Mass Storage Device

Devices typically already have default slices. If the default slices are not acceptable refer to the following sections to create the slices:

- "How to Modify Partitions and Create a PCFS File System on a USB Mass Storage Device" on page 217
- "How to Create a Solaris Partition and Modify the Slices on a USB Mass Storage Device" on page 221

1. **Become an administrator.**

2. **Add the USB device to your system. For information about hot-plugging USB devices, see:**

   - "Hot-Plugging USB Mass Storage Devices" on page 211
   - "Hot-Plugging USB Devices With the `cfgadm` Command" on page 226

3. **(Optional) Identify the USB device by using the `rmformat` command.**

   In the following example, the device is `c2t0d0p0`.

```
# rmformat
Looking for devices...
1. Logical Node: /dev/rdsk/c2t0d0p0
Physical Node: /pci@0,0/pci108e,534a@2,1/hub@7/floppy@1/disk@0,0
Connected Device: MITSUMI  USB FDD          1039
Device Type: Floppy drive
Bus: USB
Size: 1.4 MB
Label: <None>
Access permissions: Medium is not write protected.
```

4. **Perform this step only if you need to format a USB diskette. Otherwise, skip to the next step.**

   a. **Insert a diskette into the diskette drive.**

   b. **Format the diskette, if necessary.**

      ```
      # rmformat -F long raw-device-file
      ```

      where *raw-device-file* is the disk partition on which to write the file system.

Caution - Use the `rmformat -F` command **only** on USB diskettes.

5. **Determine the file system type to create, if necessary.**

6. **Unmount the device, if necessary.**

   For more information about unmounting a USB device, see "How to Mount or Unmount a USB Mass Storage Device" on page 223.

7. **Create the file system.**

   Select one of the following:

   - Create a ZFS pool and file system on a USB stick if you need to transfer data to another system.

     ```
     # zpool create device temp-pool
     # zfs create temp-pool/data
     ```

     The following example creates a ZFS pool for the device `c5t0d0` and then creates the file system on the pool.

     ```
     # zpool create c5t0d0 usbpool
     # zfs create usbpool/data
     ```

   - Create a PCFS file system.

     For information and examples, see "Creating File Systems on USB Storage Devices" on page 213

   - Create a legacy UFS file system.

     For information and examples, see "Creating File Systems on USB Storage Devices" on page 213

How to Modify Partitions and Create a PCFS File System on a USB Mass Storage Device

The following steps describe how to delete an existing partition, create a new partition, and then create a PCFS file system on the USB device. Make sure you backup any data before you perform this task.

1. **Become an administrator.**

2. **Modify the partitions as follows:**

a. Start the `fdisk` utility.

```
# fdisk device
```

b. On the interactive interface that is displayed, choose the appropriate options for the following actions as prompted.

- Delete the partition.
- Specify the partition number to delete.
- Create the partition.
- Select the partition type to create.
- Specify the percentage of disk space to use for the partition.
- Specify whether the new partition will be active or inactive.
- Choose to update the disk configuration and exit.

3. Unmount the device, if necessary.

For more information about unmounting a USB device, see "How to Mount or Unmount a USB Mass Storage Device" on page 223.

4. Create the PCFS file system on this partition.

```
# mkfs -F FS-type -o FS-type-specific-option raw-device-file
```

Example 13-1 Modifying the Partitions of a USB Diskette

In the following example, the disk configuration of `c3t0d0p0` is modified, starting with the creation of a new partition.

```
# fdisk /dev/rdsk/c3t0d0p0

     Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

Cylinders
Partition   Status    Type          Start   End   Length    %
=========   ======    ============  =====   ===   ======   ===
1       Active    Solaris2          1    28      28     97

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)
Enter Selection: 3      Partition will be deleted.


     Total disk size is 29 cylinders
```

```
Cylinder size is 2048 (512 byte) blocks

Cylinders
Partition   Status    Type          Start   End   Length    %
=========   ======    ============  =====   ===   ======   ===
1       Active    Solaris2          1    28      28     97

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)
Specify the partition number to delete (or enter 0 to exit): 1      Partition number to delete.

Partition deleted.      Deletion is completed.


      Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

Cylinders
Partition   Status    Type          Start   End   Length    %
=========   ======    ============  =====   ===   ======   ===

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)
Enter Selection: 1      Partition will be created.


      Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

Cylinders
Partition   Status    Type          Start   End   Length    %
=========   ======    ============  =====   ===   ======   ===

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)
Select the partition type to create:
1=SOLARIS2  2=UNIX        3=PCIXOS     4=Other
5=DOS12     6=DOS16       7=DOSEXT     8=DOSBIG
```

```
9=DOS16LBA  A=x86 Boot    B=Diagnostic C=FAT32
D=FAT32LBA  E=DOSEXTLBA   F=EFI        0=Exit? c      Partition type is selected.


       Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

Cylinders
Partition   Status    Type          Start   End   Length    %
=========   ======    ============  =====   ===   ======   ===

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)
Select the partition type to create:
Specify the percentage of disk to use for this partition (or type "c" to
specify the size in cylinders). 100      Percentage of disk space is specified.


             Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

Cylinders
Partition   Status    Type          Start   End   Length    %
=========   ======    ============  =====   ===   ======   ===

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)
Select the partition type to create:
Should this become the active partition? If yes, it  will be activated
each time the computer is reset or turned on.
Please type "y" or "n". n      Inactive partition is selected.


      Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

Cylinders
Partition   Status    Type          Start   End   Length    %
=========   ======    ============  =====   ===   ======   ===
1                 Win95 FAT32       1    28      28     97

SELECT ONE OF THE FOLLOWING:
1. Create a partition
```

```
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)
Enter Selection: 5      Disk configuration will be updated.

# mkfs -F pcfs -o fat=32 /dev/rdsk/c3t0d0p0:c
Construct a new FAT file system on /dev/rdsk/c3t0d0p0:c: (y/n)? y      File system is created.
```

How to Create a Solaris Partition and Modify the Slices on a USB Mass Storage Device

The following steps illustrate how to create a Solaris partition and modify the slices.

Make sure you back up any data before you perform this task.

1. **Become an administrator.**

2. **(Optional) Display the partition information on the system.**

   # **fdisk** *device*

3. **Display the current slices.**

   # **prtvtoc** *device*

4. **Create a text file with the slice information.**

   Make sure each slice starts on a cylinder boundary. For example, slice 1 starts at 822280000 bytes, which is the cylinder size in bytes multiplied by 1000.

   For more information, see the `-s` option description in `rmformat`(1).

5. **Create the slices by referring to the slice file created in the previous step.**

   # **rmformat -s** *slice-file device*

6. **View the new slice information.**

   # **prtvtoc** *device*

Example 13-2 Creating Slices on a USB Device

In the following example, slices are created on the device `c5t0d0s2`. The slice file `myslice` has been created with the following content:

```
slices: 0 = 0, 5GB, "wm", "home" :
```

```
1 = 8225280000, 6GB :
2 = 0, 44GB, "wm", "backup" :
6 = 16450560000, 15GB
```

The file will be used when creating the new slices.

```
# fdisk /dev/rdsk/c5t0d0s2
No fdisk table exists. The default partition for the disk is:

a 100% "SOLARIS System" partition

Type "y" to accept the default partition,  otherwise type "n" to edit the
partition table.
```

y *The current partition is accepted and will not be modified.*

```
# prtvtoc /dev/rdsk/c5t0d0s2
* /dev/rdsk/c5t0d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*     255 tracks/cylinder
*   16065 sectors/cylinder
*    5836 cylinders
*    5836 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
0      0    00          0  93755340  93755339
2      0    00          0  93755340  93755339

# rmformat -s myslice /dev/rdsk/c5t0d0s2

# prtvtoc /dev/rdsk/c5t0d0s2
* /dev/rdsk/c5t0d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*     255 tracks/cylinder
*   16065 sectors/cylinder
*    5836 cylinders
*    5836 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
* Unallocated space:
*       First     Sector    Last
*       Sector     Count    Sector
*    10485760   5579240  16064999
*    28647912   3482088  32129999
```

```
*    63587280  30168060  93755339
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
0      8    00          0  10485760  10485759
1      3    01   16065000  12582912  28647911
2      5    00          0  92274688  92274687
6      4    00   32130000  31457280  63587279
```

How to Mount or Unmount a USB Mass Storage Device

The following steps illustrate how to mount and unmount a USB mass storage device.

1. **Become an administrator.**

2. **(Optional) Identify the device.**

```
$ rmformat
```

The following example identifies a physical diskette device as `c2t0d0p0`.

```
$ rmformat
Looking for devices...
1. Logical Node: /dev/rdsk/c3t0d0p0
Physical Node: /pci@0,0/pci108e,534a@2,1/storage@3/disk@0,0
Connected Device: SanDisk  Cruzer Micro     0.3
Device Type: Removable
Bus: USB
Size: 245.0 MB
Label: <None>
Access permissions: Medium is not write protected.
```

3. **Select one of the following to mount or unmount a USB mass storage device:**

   - Mount a USB mass storage device as a console user.

     You can use the `rmmount` command with device nicknames, mount points, or device paths, similar to the following:

```
$ rmmount rmdisk0
$ rmmount NONAME
$ rmmount /dev/dsk/c3t0d0p0:1
```

     The following example mounts `c2t0d0p0` whose mount point is `NONAME`.

```
$ rmmount NONAME
NONAME /dev/dsk/c2t0d0p0 mounted
$ ls /media/NONAME
```

```
AA.TXT
```

- Unmount a USB mass storage device as a console user.

  The following example unmounts `c2t0d0p0` whose mount point is `NONAME`.

  ```
  $ rmumount NONAME
  NONAME /dev/dsk/c2t0d0p0 unmounted
  ```

- Mount a USB mass storage device as superuser.

  The following example mounts `c1t0d0s2` with a UFS file system:

  ```
  $ mount /dev/dsk/c1t0d0s2 /mnt
  ```

  The following example mounts `c3t0d0s2` with a PCFS file system on a SPARC system:

  ```
  $ mount -F pcfs /dev/dsk/c3t0d0s2:c /mnt
  ```

  The following example mounts `c3t0d0p0` with a PCFS file system on an x86 system:

  ```
  $ mount -F pcfs /dev/dsk/c3t0d0p0:c /mnt
  ```

  The following example mounts `c1t0d0s2` with a read-only HSFS file system:

  ```
  $ mount -F hsfs -o ro /dev/dsk/c1t0d0s2 /mnt
  ```

- Unmount a USB mass storage device as superuser.

  The following example unmounts a storage devise.

  ```
  $ fuser -c -u /mnt
  $ umount /mnt
  ```

4. Eject the device.

This step is optional for DVD, CD, or diskette devices.

```
$ eject /dev/rdsk/c1t0d0s2
```

The following example ejects `c1t0d0s2`.

Troubleshooting Tips for USB Mass Storage Devices

Check the `/var/adm/messages` file for failures to enumerate the device. In cases of failures that involve hubs, insert the USB hub or remove a hub and connect it directly to a root USB hub.

- If you have problems accessing a device that was connected while the system is running, issue following command:

  ```
  # devfsadm
  ```

- Do not move devices around if the system has been powered down by a suspend operation. For more information, see "USB Power Management" on page 210.
- If a device has been removed while in use by applications and is no longer available, stop the applications. Use the `prtconf` command to see whether the device node has been removed.

Disabling Specific USB Drivers

You can disable specific types of USB devices by disabling their client driver. For example, USB printers can be disabled by disabling the `usbprn` driver that directs them. Disabling `usbprn` does not affect other kinds of devices, such as USB storage devices.

The following table identifies some USB device types and their corresponding drivers.

Device Type	Driver to Disable
Audio	`usb_ac` and `usb_as`
HID (usually keyboard and mouse)	`hid`
Storage	`scsa2usb`
Printer	`usbprn`
Serial	`usbser_edge`

If you disable a driver for a USB device that is still connected to the system, a warning message similar to the following is displayed:

```
usba10: WARNING: usba:    no driver found for device name
```

▼ How to Disable Specific USB Drivers

1. **Become an administrator.**
2. **Use the `pfedit` command to edit the `/etc/system` file.**

   ```
   # pfedit /etc/system
   ```

3. **Add an `exclude` line that refers to the driver alias entry.**

   The following example excludes the USB printer driver.

   ```
   exclude: usbprn
   ```

4. **Reboot the system.**

```
# init 6
```

How to Remove Unused USB Device Links

Use this procedure if a USB device is removed while the system is powered off. Removing the USB device while the system is powered off can leave device links for devices that do not exist.

1. **Become an administrator.**
2. **Close all applications that might be accessing the device.**
3. **Remove the unused links for a specific USB class.**

   Choose one of the following commands:

   - `devfsadm -C -c` *USB-class*

     The following example removes the unused links for the audio class of devices.

     ```
     # devfsadm -C -c audio
     ```

   - `devfsadm -C`

     The command removes all dangling links.

Hot-Plugging USB Devices With the `cfgadm` Command

You can add and remove a USB device from a running system without using the `cfgadm` command. However, the `cfgadm` command enables you to perform *logical* hot-plug operations on the USB device. In logical hot-plug operations, you do not have to physically handle the device. Thus, you can remotely disable or reset a non functioning USB device without having to remove the device itself. The `cfgadm` command also enables you to display the USB device tree, including manufacturer and product information.

The `cfgadm` command works on USB devices in the same manner that it does on other hot-pluggable devices. For more conceptual and overview information about how the command works, see the relevant sections in Chapter 2, “Dynamically Configuring Devices”.

The following `cfgadm` commands apply to all devices including USB devices. See detailed information in Chapter 2, “Dynamically Configuring Devices”:

`cfgadm -l[a]` Displays information about devices.

`cfgadm -c configure` *device*; `cfgadm -c unconfigure` *device*	Configures or unconfigures a device.
`cfgadm -c connect` *device*; `cfgadm -c disconnect` *device*	Connects or disconnects a device.
`cfgadm -x` *usb-option device*	Perform logical operations on a USB device. The command is specific to USB devices only.

Note - The `prtconf` command can also display configuration information of all devices including USB devices.

The following examples show the use of the `cfgadm` command to perform hot-plug operations on USB devices. The sample output displays only information relevant to USB devices. The actual command output depends on the devices on your system.

EXAMPLE 13-3 Displaying USB Bus Information

```
$ cfgadm
Ap_Id                  Type         Receptacle   Occupant     Condition
usb0/4.5               usb-hub      connected    configured   ok
usb0/4.5.1             usb-device   connected    configured   ok
usb0/4.5.2             usb-printer  connected    configured   ok
usb0/4.5.3             usb-mouse    connected    configured   ok
usb0/4.5.4             usb-device   connected    configured   ok
usb0/4.5.5             usb-storage  connected    configured   ok
usb0/4.5.6             usb-communi  connected    configured   ok
usb0/4.5.7             unknown      empty        unconfigured ok
```

Using `usb0/4.5.1` as an example, you obtain the following information from the output:

- `usb0` refers to the first USB controller.
- The 3 dot-separated numbers after the device name indicate that in addition to the root hub, the system has 2 external hubs.
  - The first number refers to port 4 of the controller's root hub.
  - The second number refers to port 5 of the first-level external hub.
  - The third number refers to port 1 of the second-level external hub.

Note that as a useful practice, you should display information about the system devices before and after the actual hot-plugging step. The information helps you in the following ways:

- You can properly identify the bus or device to dynamically reconfigure.
- You can verify from the state of the bus or device that the reconfiguration completed successfully.

You can also customize the specific information that you want about a USB device. The following example provides full information about the devices.

```
$ cfgadm -l -s "cols=ap_id:info"
Ap_Id       Information
usb0/4.5.1  Mfg: Inside Out Networks Product: Edgeport/421 NConfigs: 1
            Config: 0  : ...
usb0/4.5.2  Mfg: <undef> Product: <undef>   NConfigs: 1 Config: 0 ...
usb0/4.5.3  Mfg: Mitsumi Product: Apple USB Mouse NConfigs: 1
            Config: 0 ...
usb0/4.5.4  Mfg: NMB  Product: NMB USB KB/PS2 M NConfigs: 1 Config: 0
usb0/4.5.5  Mfg: Hagiwara Sys-Com  Product: SmartMedia R/W  NConfigs: 1
            Config: 0 : ...
usb0/4.5.6  Mfg: 3Com Inc.  Product: U.S.Robotics 56000 Voice USB Modem
            NConfigs: 2 ...
usb0/4.5.7
```

EXAMPLE 13-4 Unconfiguring a USB Device

You can unconfigure a USB device and leave it physically connected to the system. The device continues to be included in a `prtconf` command output. However, no driver will attach to the device.

```
# cfgadm -c unconfigure usb0/4.7
Unconfigure the device: /devices/pci@8,700000/usb@5,3/hub@4:4.7
This operation will suspend activity on the USB device
Continue (yes/no)? y

# cfgadm -l
Ap_Id                  Type         Receptacle   Occupant     Condition
usb0/4.5               usb-hub      connected    configured   ok
usb0/4.5.1             usb-device   connected    configured   ok
usb0/4.5.2             usb-printer  connected    configured   ok
usb0/4.5.3             usb-mouse    connected    configured   ok
usb0/4.5.4             usb-device   connected    configured   ok
usb0/4.5.5             usb-storage  connected    configured   ok
usb0/4.5.6             usb-communi  connected    configured   ok
usb0/4.5.7             unknown      empty        unconfigured ok
usb0/4.6               usb-storage  connected    configured   ok
usb0/4.7               usb-storage  connected    unconfigured ok     Device is unconfigured.
```

EXAMPLE 13-5 Configuring a USB Device

```
# cfgadm -c configure usb0/4.7

# cfgadm usb0/4.7
Ap_Id                Type         Receptacle  Occupant    Condition
```

```
usb0/4.7              usb-storage  connected   configured  ok
```

EXAMPLE 13-6 Logically Disconnecting a USB Device

In this example, the USB device remains physically connected to the system. However, it becomes unusable after being logically disconnected. The `prtconf` command output will not include this device in the output.

```
# cfgadm -c disconnect -y usb0/4.7
# cfgadm usb0/4.7
Ap_Id                 Type      Receptacle    Occupant      Condition
usb0/4.7              unknown   disconnected  unconfigured  ok      The device is disconnected and
 becomes unavailable.
```

EXAMPLE 13-7 Logically Connecting a USB Device

You can connect a USB device remotely without physical access to the device. However, the device must be physically connected to the system for a logical connection to succeed.

```
# cfgadm -c configure usb0/4.7

# cfgadm usb0/4.7
Ap_Id                 Type         Receptacle  Occupant    Condition
usb0/4.7              usb-storage  connected   configured  ok      The device is connected and becomes
 available.
```

EXAMPLE 13-8 Logically Disconnecting a USB Device Subtree

In this example, the entire hierarchy of devices below the hub is removed.

```
# cfgadm -c disconnect -y usb0/4

# cfgadm usb0/4
Ap_Id                 Type       Receptacle   Occupant     Condition
usb0/4                unknown    disconnected unconfigured ok      All devices under port 4 of the hub are
 disconnected.
```

EXAMPLE 13-9 Resetting a USB Device

Resetting a device removes and then recreates the device. You can reset a device that behaves erratically. This example assumes that the device is not in use. Resetting fails if any application is using the device.

```
# cfgadm -x usb_reset -y usb0/4.7

# cfgadm usb0/4.7
Ap_Id                 Type         Receptacle   Occupant     Condition
usb0/4.7              usb-storage  connected    configured   ok
```

About Multi-Configurations of a USB Device

A USB device configuration defines how a device presents itself to the operating system. USB device configuration differs from system device configurations that use the `cfgadm` command and which are discussed in other sections in this book.

Some USB devices support multiple configurations. However, only one configuration can be active at a time. You can identify devices with multi-configurations by using the `cfgadm -lv` command. From the command output, two parameters provide multi-configuration information:

- `Nconfigs` indicates the number of configurations for the device.
- `Config` indicates the configuration that is currently active and is therefore the default configuration.

Changes to the default configuration persist across reboots, hot-removes, and the reconfiguration of the device, provided that the device is reconnected to the same port.

▼ How to Change the Default Configuration of a Multi-Configuration USB Device

1. **Become an administrator.**
2. **Make sure that the device is not in use.**
3. **(Optional) Display the current USB device configuration.**

   ```
   # cfgadm -lv [device]
   ```

   You can specify the device if you already know the `Ap_Id` whose default configuration you are changing.

4. **Change the default USB configuration.**

   Ensure that you also confirm the action at the prompt.

   ```
   # cfgadm -x usb_config -o config=config-number  device
   ```

   The following example changes the configuration of `usb0/4` from `1` to `2`.

   ```
   # cfgadm -x usb_config -o config=2 usb0/4
   Setting the device: /devices/pci@1f,0/usb@c,3:4
   to USB configuration 2
   This operation will suspend activity on the USB device
   Continue (yes/no)? yes
   ```

5. **Verify that the device changed.**

```
# cfgadm -lv device
```

The following example shows the changes for `usb0/4` after the previous step.

```
# cfgadm -lv usb0/4
Ap_Id  Receptacle   Occupant     Condition  Information
When  Type  Busy   Phys_Id
usb0/4 connected    unconfigured ok         Mfg: Sun
2000  Product: USB-B0B0 aka Robotech
With 6 EPPS High Clk Mode    NConfigs: 7   Config: 2 :     Default configuration is now 2.
EVAL Board Setup unavailable
usb-device   n     /devices/pci@1f,0/usb@c,3:4
```

Using USB Audio Devices

For audio devices in Oracle Solaris, only USB 2.0, 1.1, and 1.0 are supported, not USB 3.0. Additionally, the support is for those devices that are play-only, record-only, or record and play.

Oracle Solaris USB audio support is implemented by a pair of drivers:

- Audio control driver (`usb_ac`) - provides the controlling interface to user applications. For more information, see the `usb_ac`(7D) man page.
- Audio streaming driver (`usb_as`) - processes audio data messages during play and record. It sets sample frequency and precision, and encodes requests from the `usb_ac` driver. For more information, see the `usb_as`(7D) man page.

Some audio devices can set volume under software control. A STREAMS module, `usb_ah`, is pushed on top of the human interface device (HID) driver for managing this function. For more information, see the `usb_ah`(7M) man page.

The primary audio device is `/dev/audio`. To verify that `/dev/audio` is pointing to USB audio, use the `mixerctl` command. For example:

```
%$ mixerctl
Device /dev/audioctl:
Name    = USB Audio
Version = 1.0
Config  = external

Audio mixer for /dev/audioctl is enabled
```

You access connected USB audio devices with the `audioplay` and `audiorecord` command through the `/dev/sound/`*N* device links.

Note that the `/dev/audio` and `/dev/sound/`*N* devices can refer to speakers, microphones, or combination devices. If you refer to the incorrect device type, the command fails. For example, the `audioplay` command fails if you try to use it with a microphone.

You can select a specific default audio device for most Oracle Solaris audio applications, such as `audioplay` and `audiorecord`, by setting the `AUDIODEV` shell variable or by specifying the `-d` option for these commands. However, setting `AUDIODEV` does not work for third-party applications that have `/dev/audio` hardcoded as the audio file.

A USB plugged-in audio device automatically becomes the primary audio device, `/dev/audio`, provided that `/dev/audio` is not in use. It remains the primary audio device even after the system is rebooted. If additional USB audio devices are plugged in, the last one becomes the primary audio device.

For instructions on switching `/dev/audio` between on-board audio and USB, refer to "Problem With Reverting to On-Board Audio Device" on page 234, and `usb_ac`(7D).

Displaying Information About USB Audio Devices

To list the primary audio device on the system, use the `ls -l` command on the `/dev/audio` device link. As you plug in additional USB devices, the command output identifies the device to which `/dev/audio` connects. This device automatically becomes the primary audio device.

For example, by default, the system's audio is the on-board audio.

```
$ ls -l /dev/audio
lrwxrwxrwx 1 root root 7 Feb 13 08:47 /dev/audio -> sound/0
```

Suppose you connect a USB speaker to the system. The command output would then be as follows:

```
$ ls-l /dev/audio
lrwxrwxrwx 1 root root 10 Feb 13 08:46 /dev/audio -> usb/audio0
```

If you add another audio device, such as a USB microphone, the command output would also change.

```
$ ls -l /dev/audio
lrwxrwxrwx 1 root root 10 Feb 13 08:54 /dev/audio -> usb/audio1
```

To list all the USB audio devices that are connected to the system, use the same `ls` command on all the audio device links.

```
$ ls -lt /dev/audio*
lrwxrwxrwx   1 root  root      7 Jul 23 15:46 /dev/audio -> usb/audio0
lrwxrwxrwx   1 root  root     10 Jul 23 15:46 /dev/audioctl ->
usb/audioctl0/

% ls -lt /dev/sound/*
lrwxrwxrwx   1 root  root     74 Jul 23 15:46 /dev/sound/1 ->
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:...
lrwxrwxrwx   1 root  root     77 Jul 23 15:46 /dev/sound/1ctl ->
```

```
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:...
lrwxrwxrwx   1 root  other    66 Jul 23 14:21 /dev/sound/0 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx   1 root  other    69 Jul 23 14:21 /dev/sound/0ctl ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audioctl
```

Note that despite the multiple audio devices, the `/dev/usb/audio0` is the primary audio device.

You can also use the `prtconf` command to display USB device information.

```
$ prtconf
.
usb, instance #0
hub, instance #0
mouse, instance #0
keyboard, instance #1
device, instance #0
sound-control, instance #0
sound, instance #0
input, instance #0
.
```

To change the primary audio device to a USB audio device, plug the USB device to the system. The `/dev/audio` link automatically points to this device. To revert to using the on-board audio device, simply remove the USB device.

Troubleshooting USB Audio Device Problems

Sometimes, USB speakers do not produce any sound, even though the driver is attached and the volume is set to high. Hot-plugging the device might not change this behavior.

To restore sound, power cycle the USB speakers.

About Audio Device Ownership

When you plug in a USB audio device while logged in to the console, the console is the owner of the `/dev/*` entries. Therefore, you can use the audio device provided you remain logged in. If you were not logged in, then `root` is the owner of the device. However, if you log in to the console and attempt to access the USB audio device, then device ownership changes to the console. For more information, see the `logindevperm(4)` man page.

The behavior is different with remote logins. If you attempt to access the device remotely, the ownership does not change and therefore the device is secure. For example, unauthorized users cannot remotely listen to conversations over a microphone that is owned by someone else.

Problem With Reverting to On-Board Audio Device

You might observe that after removing the USB device, the `/dev/audio` does not point back to `/dev/sound/0`. This error indicates that the system did not revert to using the on-board audio device as the primary audio device. Do one of the following workarounds:

- Shut down the system, and use the `boot -r` command.
- As root, issue the `devfsadm -i` command.

♦ ♦ ♦ CHAPTER 14

Managing Removable Media

This chapter describes how to manage and access removable media in the Oracle Solaris OS.

This is a list of the information in this chapter:

About Removable Media

Removable media services enable you to access removable media but without requiring administrative rights, unlike with manual mounting.

If the media contains a file system and a label, the media label name is used to name the */media/pathname* mount point. If a label is not present, the disk model name is used to name the media, such as `/media/cdrom`. A generic *nickname* is used only for legacy symbolic links. For example, `/rmdisk/rmdisk0`.

If your system has more than one type of removable device, see the following table for their access points.

TABLE 14-1 How to Access Data on Removable Media

Media	Unlabeled Media Pathnames	Labeled Media Pathname Examples
Hard disk	`/media/`*usb-disk* or the legacy path `/rmdisk/rmdisk0`	`/media/00JB-00CRA0`
DVD	`/media/cdrom`	`/media/Oracle_Solaris-`*n*`-Live-X86`

To identify mounted media on your system, use the `rmmount -l` command. For example:

```
# rmmount -l
/dev/dsk/c5t0d0p0       rmdisk6,/media/FD-05PUB
/dev/dsk/c4t0d3p0       rmdisk5,/media/223UHS-SD-MMC
/dev/dsk/c2t0d0s2       cdrom1,cd1,sr1,Oracle_Solaris-11_1-AI-SPARC,  \
                           /media/Oracle_Solaris-11_1-AI-SPARC
/dev/dsk/c3t0d0p0       rmdisk2,/media/00JB-00CRA0
```

In the sample output, the mounted devices are as follows:

- `/dev/dsk/c5t0d0p0` – USB diskette
- `/dev/dsk/c4t0d3p0` – CF card in a USB card reader
- `/dev/dsk/c2t0d0s2` – DVD-ROM
- `/dev/dsk/c3t0d0p0` – Removable USB disk

Removable Media Considerations

File system formats in Oracle Solaris consist of the basic “bit” formatting, in addition to the structure to support an Oracle Solaris file system. The procedures required to prepare a media for each type of file system are different. Therefore, before you format a diskette, consider which procedure to follow.

About Formatting Removable Media

You can use the `rmformat` command to format and perform other management tasks on removable media. File systems are mounted automatically. So, you might have to unmount media before you can format it, if the media contains an existing file system.

The `rmformat` command has three formatting options:

- `quick` – This option formats media without certification or with limited certification of certain tracks on the media.
- `long` – This option completely formats media. For some devices, the use of this option might include the certification of the whole media by the drive.
- `force` – This option formats completely without user confirmation. On media with a password-protection mechanism, this option clears the password before formatting. On media without password protection, this option forces a long format.

Managing Removable Media

The following guidelines and procedures are described in this section:

How to Load Removable Media

1. **Insert the media.**

2. **Ensure that the media is formatted.**

   If you aren't sure, insert the media and check the status messages in the system console window, as described in Step 3. If you need to format the media, go to "How to Format a Removable Device (rmformat)" on page 238.

3. **(Optional) Notify volume management if you are using a legacy, non-USB diskette device.**

   ```
   $ volcheck -v
   ```

   Two status messages are possible:

`media was found`	Volume management detected the media and will attempt to mount it in the directory described in "Using Removable Media Names" on page 242. If the media is not formatted properly, additional messages are displayed. You must then format the media before volume management can mount it. For more information, see "How to Format a Removable Device (rmformat)" on page 238.
`no media was found`	Volume management did not detect the media. Ensure that the media is inserted properly, and run `volcheck` again. If unsuccessful, check the media, which could be damaged. You can also try to mount the media manually.

4. **Verify that the media was mounted by listing its contents.**

   ```
   $ ls /media/media-name
   ```

How to Format a Removable Device (rmformat)

You can use the rmformat command to format a diskette. By default, this command creates two partitions on the media: partition 0 and partition 2 (the whole media).

1. **Verify that removable media service is running. If so, you can use the shorter nickname for the device name.**

```
# svcs hal dbus rmvolmgr
STATE          STIME    FMRI
online         Apr_09   svc:/system/dbus:default
online         Apr_09   svc:/system/hal:default
online         Apr_09   svc:/system/filesystem/rmvolmgr:default
```

For information on restarting removable media services, see "How to Disable or Enable Removable Media Services" on page 243. For information on identifying media device names, see "Using Removable Media Names" on page 242.

2. **Format the device.**

```
$ rmformat -F [ quick | long | force ] device-name
```

where *device-name* includes the full path to the device, such as /dev/*device*.

See "About Formatting Removable Media" on page 236 for more information on rmformat formatting options.

If the rmformat output indicates bad blocks, see "How to Repair Bad Blocks on Removable Media" on page 240.

3. **(Optional) Label the device.**

```
$ rmformat -b label device-name
```

For information on creating a DOS label, see mkfs_pcfs(1M).

How to Create a File System on Removable Media

1. **(Optional) If necessary, format the media.**

2. **(Optional) Create an alternate Solaris partition table.**

```
$ rmformat -s slice-file device-name
```

A sample slice file appears similar to the following:

```
slices: 0 = 0, 30MB, "wm", "home" :
```

```
1 = 30MB, 51MB :
2 = 0, 94MB, "wm", "backup" :
6 = 81MB, 13MB
```

3. **Become an administrator.**

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

4. **Determine the appropriate file system type and select one of the following:**

   ```
   # mkfs -F FS-type -o FS-type-specific-options device
   ```

   where *FS-type-specific-options* refers to options that are specific to the file system type you are creating.

   The following example creates a PCFS file system.

   ```
   # mkfs -F pcfs -o nofdisk,size=9800 /dev/rdsk/c11t0d0p0
   ```

   The following example create a UDFS file system.

   ```
   # mkfs -F udfs /dev/rdsk/c0t1d0p0
   ```

How to Create a File System on a DVD-RAM

Use this procedure to create a file system on a DVD-RAM.

1. **Become an administrator.**

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

2. **Create a file system on the DVD-RAM device.**

   For example, create a UDFS file system, as follows:

   ```
   # mkfs -F udfs /dev/rdsk/c0t0d0s2
   ```

3. **Mount the file system.**

   For example, mount a UDFS file system, as follows:

   ```
   # mount -F udfs /dev/dsk/c0t0d0s2 /mnt
   ```

4. **Verify that you can read or write to the file system.**

5. **When finished, eject the DVD-RAM.**

How to Check a File System on Removable Media

1. **Become an administrator.**

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

2. **Identify the file system type and select one of the following:**

   - **Check a UDFS file system, as follows:**

     ```
     # fsck -F udfs device-name
     ```

   - **Check a PCFS file system, as follows:**

     ```
     # fsck -F pcfs device-name
     ```

Example 14-1 Checking a PCFS File System on Removable Media

The following example shows how check the consistency of a PCFS file system on media.

```
# fsck -F pcfs /dev/rdsk/c0t4d0s2
** /dev/rdsk/c0t4d0s2
** Scanning file system meta-data
** Correcting any meta-data discrepancies
1457664 bytes.
0 bytes in bad sectors.
0 bytes in 0 directories.
0 bytes in 0 files.
1457664 bytes free.
512 bytes per allocation unit.
2847 total allocation units.
2847 available allocation units.
```

How to Repair Bad Blocks on Removable Media

You can only use the `rmformat` command to verify, analyze, and repair bad sectors that are found during verification if the drive supports bad block management. Most USB memory sticks do not support bad block management.

If the drive supports bad block management, a best effort is made to rectify the bad block. If the bad block cannot be rectified despite the best effort mechanism, a message indicates the failure to repair the media.

1. **Repair bad blocks on removable media.**

   ```
   $ rmformat -c block-numbers device-name
   ```

Supply the block number in decimal, octal, or hexadecimal format from a previous `rmformat` session.

2. **Verify the media.**

```
$ rmformat -V read device-name
```

Applying Read or Write Protection and Password Protection to Removable Media

You can apply read protection or write protection, and set a password, on removable media that support this feature.

▼ How to Enable or Disable Read and Write Protections

1. **Determine whether you want to enable or disable write protection and select the appropriate commands from the following:**

   - **Enable write protection.**

     ```
     $ rmformat -w enable device-name
     ```

   - **Enable read protection.**

     ```
     $ rmformat -r enable device-name
     ```

   - **Disable write protection.**

     ```
     $ rmformat -w disable device-name
     ```

   - **Disable read protection.**

     ```
     $ rmformat -r disable device-name
     ```

   Note - Each of these commands includes a prompt to type a password. Specify a valid password.

2. **Verify whether the media's write protection is enabled or disabled.**

```
$ rmformat -p device-name
```

Accessing Removable Media

You can access information on removable media with or without using volume management. For information on accessing information on removable media with GNOME's File Manager, see the GNOME desktop documentation.

This following procedures are described in this section:

Using Removable Media Names

Removable media is now mounted automatically in the `/media` directory. However, symbolic links to `/media` are provided from previous media mount points, `/cdrom` and `/rmdisk`, for compatibility purposes.

For example, a compact flash memory card (`/dev/dsk/c4d0p0:1`) is mounted as follows:

```
$ ls /media/memory-card-name
```

For example, a USB memory stick (`/dev/dsk/c3t0d0s0`) is mounted, as follows:

```
$ ls /media/U3
```

Guidelines for Accessing Removable Media Data

Most DVDs are formatted to the ISO 9660 standard, which is portable. So, DVDs can be mounted by volume management.

To accommodate possible different formats, a DVD is split into slices. Slices are similar in effect to partitions on hard disks. The 9660 portion is portable. If you are having trouble mounting a DVD, particularly if it is an installation DVD, make sure that its file system is appropriate for your system's architecture. For example, you can check the label on DVD.

How to Add a New Removable Media Drive

Generally, most modern bus types support hot-plugging. This means you can insert a disk in an empty slot and the system recognizes it.

For more information about hot-plugging devices, see Chapter 2, “Dynamically Configuring Devices”.

1. **Become an administrator.**
2. **Connect the new media drive.**

   See your hardware handbook for specific instructions.
3. **Confirm that the system sees the new media drive.**

```
# rmformat
Looking for devices...
```

How to Disable or Enable Removable Media Services

Occasionally, you might want to manage media without using removable media services. This section describes how to disable and enable removable media services.

Disabling these services means that you would have to mount all media manually by using the `mount` command.

1. **Ensure that the media is not being used.**

   If you are not sure whether you have found all users of the media, use the `fuser` command, see “How to Determine If Removable Media Is Still in Use” on page 245.
2. **Become an administrator.**
3. **Select one of the following:**

   - **You can disable some or all removable media features in this release:**

     - **To prevent volumes from mounting outside of user sessions, disable the `rmvolmgr` service. For example:**

```
# svcadm disable rmvolmgr
```

- **To prevent any volume management, disable the `dbus`, `hal`, and `rmvolmgr` services.**

  ```
  # svcadm disable rmvolmgr
  # svcadm disable dbus
  # svcadm disable hal
  ```

  Disabling these services means that you would have to mount all media manually by using the `mount` command.

- **Enable removable media services.**

  ```
  # svcadm enable rmvolmgr
  # svcadm enable dbus
  # svcadm enable hal
  ```

▼ How to Access Information on Removable Media

1. **Insert the media.**

   The media is mounted after a few seconds.

2. **List the contents of the media. For example:**

   ```
   # ls /media/Oracle_Solaris-11_1-AI-SPARC
   auto_install      export           proc             solarismisc.zlib
   bin               home             reconfigure      system
   boot              jack             root             tmp
   dev               mnt              sbin
   devices           platform         solaris.zlib
   ```

3. **(Optional) Copy the file that you identified in the previous step.**

Example 14-2 Accessing Information on Removable Media

This example shows how to access information on a USB memory stick.

```
$ ls /media/usb-name
```

This example shows how to access information on a DVD.

```
$ ls /media
Oracle_Solaris-11_1-AI-SPARC cdrom
```

How to Determine If Removable Media Is Still in Use

1. **Become an administrator.**
2. **Identify the processes that are accessing the media.**

   ```
   # fuser -u /media
   ```

   The -u displays the user of the media.

   For more information, see `fuser`(1M).

3. **(Optional) Kill the process accessing the media.**

   ```
   # fuser -u -k /media
   ```

   The -k option kills the processes accessing the media.

Caution - Killing the processes that are accessing the media should only be used in emergency situations.

4. **Verify that the process is gone.**

   ```
   # pgrep process-ID
   ```

How to Eject Removable Media

1. **Ensure that the media is not being used.**

   Remember, media is “being used” if a shell or an application is accessing any of its files or directories. If you are not sure whether you have found all users of a DVD (for example, a shell hidden behind a desktop tool might be accessing it), use the `fuser` command. See “How to Determine If Removable Media Is Still in Use” on page 245.

2. **Eject the media.**

   ```
   # eject media
   ```

   For example, for a DVD, you would do the following:

   ```
   # eject cdrom
   ```

   For example, for a USB memory stick, you would do the following:

```
# eject rmdisk0
```

Tip - You can view the removable device name with the `eject -l` command.

Accessing Removable Media on a Remote System

Removable media can be shared with remote systems by using NFS. For additional information about using NFS, see “Managing Network File Systems in Oracle Solaris 11.2”.

The following procedures are described in this section:

- “How to Make Local Media Available to Other Systems” on page 246
- “How to Access Removable Media on Remote Systems” on page 247

▼ How to Make Local Media Available to Other Systems

You can configure your system to share its media drives to make any media in those drives available to other systems. One exception is musical CDs. Once your media drives are shared, other systems can access the media they contain simply by mounting them. For instructions, see “How to Access Removable Media on Remote Systems” on page 247.

1. **Confirm that the media is loaded.**

2. **Add the following entry to the `/etc/dfs/dfstab` file.**

   For example:

   ```
   share -F nfs -o ro /media/sol_10_811_sparc
   ```

3. **Determine whether the NFS server service is running.**

   ```
   # svcs *nfs*
   ```

   The following output is returned from the `svcs` command if NFS server service is running:

   ```
   online         14:28:43 svc:/network/nfs/server:default
   ```

4. **Identify the NFS server status, and select one of the following:**

   - **If the NFS server service is running, go to Step 7.**

- **If the NFS server service is *not* running, go to the next step.**

5. **Start the NFS server service.**

```
# svcadm enable network/nfs/server
```

6. **Verify that the NFS daemons are running.**

For example:

```
# svcs -p svc:/network/nfs/server:default
STATE          STIME    FMRI
online         Aug_30   svc:/network/nfs/server:default
Aug_30        319 mountd
Aug_30        323 nfsd
```

7. **Verify that the media is available to other systems.**

If the media is available, its share configuration is displayed.

```
# share
-               /media/Oracle_Solaris-11_1-AI-SPARC   sec=sys,ro   ""
```

Example 14-3 Making Local DVDs Available to Other Systems

The following example shows how to make any local DVD available to other systems on the network.

```
# share -F nfs -o ro /media
# svcs *nfs*
# svcadm enable network/nfs/server
# svcs -p svc:/network/nfs/server:default
# share
-               /media/Oracle_Solaris-11_1-AI-SPARC   ro   ""
```

How to Access Removable Media on Remote Systems

You can access media on a remote system by manually mounting the media into your file system. Also, the remote system must have shared its media according to the instructions in "How to Make Local Media Available to Other Systems" on page 246.

1. **Select an existing directory to serve as the mount point. Or create a mount point.**

```
$ mkdir /directory
```

where /*directory* is the name of the directory that you create to serve as a mount point for the remote system's DVD.

2. **Find the name of the media that you want to mount.**

   $ **showmount -e** *system-name*

3. **As an administrator, mount the media.**

   # **mount -F nfs -o ro** *system-name*:*/media/media-name local-mount-point*

   system-name: Is the name of the system whose media you will mount.

   media-name Is the name of the media you want to mount.

   local-mount-point Is the local directory onto which you will mount the remote media.

4. **Log out as an administrator.**

5. **Verify that the media has been mounted.**

   $ **ls** */mnt*

Example 14-4 Accessing DVDs or CDs on Remote Systems

The following example shows how to automatically access the remote DVD from the remote system `starbug` using `autofs`.

```
$ showmount -e starbug
export list for starbug:
/media/Oracle_Solaris-11_1-AI-SPARC (everyone)
$ ls /net/starbug/media/
Oracle_Solaris-11_1-AI-SPARC
```

Index

D

E

F

H

I

K

L

M

N

O

P

R

S

T

U

V

W

Z